CETIA

CETIA POWERENGINE

RELEASE NOTES

PWL4-ESP

for LynxOS 3.0.1 on VMPC Family

ID 00091

SD.DT.C00-3e NOVEMBER 2000

CETIA

Corporate Headquarters
150, rue Marcellin Berthelot
ZI Toulon-Est - BP 244
83078 TOULON Cedex 9
France
Tel: +33 - (0) 4 94 16 34 00
Fax: +33 - (0) 4 94 16 34 01

USA Headquarters
1203 New Hope Road
Raleigh - NC 27610
USA
Tel: +1 - (800) 848 2330
Tel: +1 - (919) 231 8000
Fax: +1 - (919) 231 8001

International Sales Department
67, rue Charles De-Gaulle
78350 JOUY-EN-JOSAS - France
France
Tel: +33 - (0) 1 39 56 72 72
Fax: +33 - (0) 1 39 56 72 70

United Kingdom
Cornwell Business Park
31 Salthouse Road - Brackmills
Northampton - NN4 7EX
United Kingdom
Tel: +44 - (0) 1604 700 221
Fax: +44 - (0) 1604 700 112

www.cetia.com

CETIA reserves the right to make changes without further notice to any products herein. CETIA makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does CETIA assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidential damages. "Typical" parameters can and do vary in different applications. All operating parameters including "Typicals" must be validated for each customer application by customer’s technical experts. CETIA does not convey any license under its patent rights nor the rights of others. CETIA products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the CETIA product could create a situation where personal injury or death may occur. Should Buyer purchase or use CETIA products for any such unintended or unauthorized application, Buyer shall indemnify and hold CETIA and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that CETIA was negligent regarding the design or manufacture of the part.

Your comments on this manual will be welcome.
A sheet is appended for you to submit your remarks.

TABLE OF CONTENTS

1. Overview

This PowerLine Extented Standard Package for LynxOS 3.0.1 is intended to be installed after a basic LynxOS 3.0.1 release.

CAUTION **The release LynxOS 3.0.1 supported with this Extended Standard Package release is:**
– On VMPC6 platforms, the "LynxOS 3.0.1 for VMPC6 or VxG4a (ID 00088)" distributed by CETIA.
– On the other VMPC platforms, the "LynxOS 3.0.1 CT3" distributed by LynxRTS.

This Extended Standard Package mainly contains:

∗ the Dual support,

∗ the advanced VME service,

∗ the VMEOb support,

∗ the SCSIMGR driver (Ultra fast and wide support),

∗ the SCMP service,

∗ the User Flash support,

∗ the CPMC-ETH100T support,

∗ the CPMC-ETH-xx support,

∗ the CPMC-xASYNC support,

∗ the X11 multi screen server on PCI,

∗ the Device Resource Manager support,

∗ the diskless service,

∗ HPT: High Precision Timer,

∗ AltiVec support for VMPC6c boards.

This product includes software developed by the NetBSD Foundation Inc. and its contributors. See the complete copyright notice in Appendix C.

2. Release Content

The distribution of this release includes:

- These release notes.

- One CD-ROM containing the Extended Standard Package to be installed on the workstation.

CETIA DISTRIBUTION
PWL4-ESP for LynxOS 3.0.1 on VMPC Family
ID 00091

3. Associated Documentation

- SD.DT.B96: CETIA PowerEngine - LynxOS Release 3.0.1 Documentation Kit,
- SD.DT.C08: CETIA PowerEngine - Readme First for LynxOS 3.0.1,
- For a VMPC6 platform:
  SD.DT.C04: CETIA PowerEngine - Release Notes LynxOS 3.0.1 for VMPC6 or VxG4a Family,

4. Required Configuration

WARNING

If you have a board with 256 MB or more of memory, as LynxOS kernel supports only 256 MB of memory and cannot boot with a memory of 256 MB or more, you must:

1. Reserve the additional memory (with the **`R`** option or with **`mscd`** parameters). Then, this part of memory can be used with **`map_iomem, smem_create`** or **`BAT`**.
2. Install the CRP2131 fix or reserve just one page (by using the **`@R1`** boot option).

4.1 Hardware Configuration

- ■ Minimal hardware configuration needed for installation:
  - ⇒ A CETIA Single Board Computer, either:
    - ♦ VMPC4a with E.C. Level 1010 or later,
    - ♦ VMPC4b with E.C. Level 2040 or later,
    - ♦ VMPC5a with E.C. Level 2040 or later,
    - ♦ VMPC5a-Dual with E.C. Level 2040 or later,
    - ♦ VMPC6a, VMPC6a/RA and VMPC6a/RC with E.C. Level 1000 or later,
    - ♦ VMPC6a-Dual, VMPC6a-Dual/RA and VMPC6a-Dual/RC with E.C. Level 1000 or later,
    - ♦ VMPC6c, VMPC6c/RA and VMPC6c/RC with E.C. Level 3450 or later,
    - ♦ VMPC6c-Dual, VMPC6c-Dual/RA and VMPC6c-Dual/RC with E.C. Level 3450 or later.
  - ⇒ a SCSI hard disk,
  - ⇒ a CD-ROM drive.
- ■ The hardware configuration may also include:
  - ○ On VMPC6 platforms:
    - ⇒ One serial board with 16 asynchronous serial I/O RS-232-C/RS-422 lines (VMEOb).
    - ⇒ An ICPMC-6 Carrier Board.
    - ⇒ And either a graphics console configuration including:
      - ♦ a PMC graphics board (CPCIGx or CPMC-GTX-8) plugged into either:
        - • the VMPC board,
        - • the PMC carrier board (right PMC slot on the ICPMC-6 - the ICPMC-6 front panel facing you).
      - ♦ an USB CHERRY keyboard,
      - ♦ a multi-standard graphics monitor.
    - ⇒ Or a VT100-compatible terminal connected to the SERIAL port (set up for 8-1-N-9600 data-stop-parity-baud).
  - ○ On VMPC4 and VMPC5 platforms:
    - ⇒ One or two Memory Mezzanine Boards (IPMD).
    - ⇒ On a VMPC4a or VMPC4b platform, one L2-Cache Mezzanine Board (IPML2 or IPML2a).
    - ⇒ One serial board with 16 asynchronous serial I/O RS-232-C/RS-422 lines (VMEOb).
    - ⇒ A PMC Carrier Board (ICPMC or ICPMC-6 for VMPC4b and VMPC5 platforms PCB C).
    - ⇒ And either a graphics console configuration including:
      - ♦ a PMC graphics board (CPCIGx or CPMC-GTX-8) plugged into eithre:
        - • the VMPC board,
        - • the PMC Carrier Board (left PMC slot on the ICPMC - the ICPMC front panel facing you or the right PMC slot on the ICPMC-6 - the ICPMC-6 front panel facing you).
      - ♦ an IBM 101 key qwerty PS/2 keyboard,
      - ♦ a multi-standard graphics monitor.

⇒ Or a VT100-compatible terminal connected to the SERIAL port (set up for 8-1-N-9600 data-stop-parity-baud).

NOTE

To determine if you have got an ICPMC or ICPMC-6 carrier board look at it. If a DEC/Intel PCI-PCI bridge is present in the middle of the PCB and if the following line appears with the `CONF` firmware command:

Bus 1 Slot 1 : Digital Equipment Corporation

This is an ICPMC carrier board.
Otherwise, you have got an ICPMC-6 carrier board.

WARNING

Not all PMCs are supported by the VMPC boards under LynxOS 3.0.1.

Not all PMCs can be plugged onto the Carrier Board (ICPMC, ICPMC-6).

If you want to add a PMC to your VMPC configuration, PLEASE CONTACT CETIA.

WARNING

If you must install a CPMC-4ASYNC to an ICPMC carrier board, it is recommended to install the PMC to the middle or right slot of the ICPMC (ICPMC facing you) because the CPMC-4ASYNC could not work on the left slot (CRP2042).

4.2 Firmware Configuration

The version of the VMPCBug firmware must be at least:

- ♦ 1.4 for VMPC4a platforms,
- ♦ 1.7 for VMPC4b platforms,
- ♦ 2.2 for VMPC4b platforms E.C. Level ≥ 2140 (PCB C),
- ♦ 1.8 for VMPC5a or VMPC5a-Dual platforms,
- ♦ 2.2 for VMPC5a or VMPC5a-Dual platforms E.C. Level ≥ 2140 (PCB C),
- ♦ 2.2 for VMPC6a, VMPC6a/RC, VMPC6a/RA, VMPC6a-Dual, VMPC6a-Dual/RA or VMPC6a-Dual/RC platforms.
- ♦ 2.4.1 for VMPC6c, VMPC6c/RC, VMPC6c/RA, VMPC6c-Dual, VMPC6c-Dual/RA or VMPC6c-Dual/RC platforms.

CAUTION

VMPCBug firmware version must be at least:

- – 2.3 if there are several VMPC boards into the rack.
- – 2.5 if the configuration contains a CPMC-GTX-8 Graphics PMC. Otherwise, you must install a VT100 console on the serial port of the VMPC to boot the software.

WARNING

A mapping example of the CETIA VMPC boards is given in the Programmer’s Reference Guide associated with each board.

The mapping of the CETIA VMPC boards is ENTIRELY RECONFIGURABLE BY LynxOS and depends on the hardware configuration.

For more information about the mapping of your VMPC board, PLEASE CONTACT CETIA.

5. Extended Standard Package Installation

The release can be directly installed on a CETIA machine running LynxOS 3.0.1. This is called a "native installation" or "self hosted".
It may also be installed as a "cross-development environment" or on a development or host workstation. This method allows cross-compilation on the host to create binaries and kernels that can be transferred to the native system and executed there.

Before installing this release you have to read the "Readme First for LynxOS 3.0.1" (SD.DT.C08). It gives information about your LynxOS 3.0.1 packages installation order.

5.1 Native Installation

It is assumed that you are using the Bash shell.

1. Power up the machine and boot LynxOS 3.0.1.
2. Insert the **PWL4-ESP for LynxOS 3.0.1** CD-ROM into the drive.
3. Create the `/cdrom` directory, if it does not already exist:
   ```
   # mkdir /cdrom
   ```
4. Mount the CD-ROM drive under `/cdrom` :
   ```
   # mount -o ro /dev/sd6 /cdrom
   ```
5. Run the `Install.esp` command:
   ```
   # cd /
   # /cdrom/Install.esp /cdrom
   ```
6. Remove the CD-ROM from the drive.
   ```
   # umount /cdrom
   ```
7. Rebuild the nodes and reboot the system:
   ```
   # cd /dev
   # rm *
   # mknod -a
   # reboot
   ```

5.2 Cross Installation

First, you must have installed the basic software associated with your VMPC platform on the host machine.

It is assumed that you are using a Korn shell compatible shell. Check your SHELL environment variable.

Check that **`/bin`** is before **`.`** into the path. Check that the following links have been created by the BSP installation:
link from **`sys/lynx.os`** to **`sys/bsp.vmpc`**
link from **`sys/galaxy`** to **`sys/galaxy.vmpc`**

1. Mount the **PWL4-ESP for LynxOS 3.0.1** CD-ROM under a directory (**`/cdrom`** for example).
2. Check that **`ENV_PREFIX`** environment variable is set to the directory name where the cross environment package has been installed on the host workstation and that the **`SETUP.bash`** has been run.
3. Run the installation with the **`Install.esp`** command:

```
# cd $ENV_PREFIX
# $SHELL /cdrom/Install.esp /cdrom
```

6. Boot Options

The boot option syntax is:

```
        bop [<device>[[,]<controller>]][@boot_options][;<firmware_options>]
```

For example:

```
        COMMAND> bop 0 c @R4096
```

The available boot options are:

R*nnn*: ***nnn*** pages of memory (4KB) are reserved at the end of the memory and are not used by the kernel (***nnn*** is a decimal value and not hexadecimal).

P*n*: The size of the PowerPC page table is set to: ($2^{n} *$ 256 KB) ($1 \leq n \leq 9$). The default size is 1/64th of the memory size. This option is only used to extend the page table. It is used for physical memory high mappings (> 100 MB). If the following message appears "`panic: mappage`", it is necessary to increase the page table size. (***n*** is a decimal value and not hexadecimal).

I: The cache mode is "inhibited" (default is Copyback).

If the kernel is into an image generated with **`mkimage`**, the **`cflag`** line of the **`.spec`** file describes the boot options. But **`cflag`** is only available on AIX or LynxOS. On other platforms, you can patch the image generated with **`mkimage`** at the offset 0xF00 and replace character by your option starting with `@`.

7. VME Advanced Service

To install the CETIA VME advanced service, uncomment the corresponding line in the file **`/sys/lynx.os/CONFIG.TBL`** (or **`/sys/galaxy/CONFIG.TBL`** for a microkernel):

```
I:almavme.cfg
```

The VME advanced service is installed by default.

CAUTION Check that the following line is disabled:

```
I:avme.cfg
```

A new kernel must be created and installed. Type the following commands:

```
LynxOS# cd /sys/lynx.os
LynxOS# make install
```

Then, reboot the system:

```
LynxOS# reboot -N
```

Once the VME service is installed, the **`almavmechan`** CETIA command on VMPC is enabled.

See the online **vmekit** and **vmekit_vmpc** man pages for more information on the VME driver development.

8. VMEOb Board Support

8.1 Overview

This release contains the following to support VMEOb boards:

- a **`vmeo`** driver which tests the presence of the VMEOb boards and allows the management of these boards (see utilities),
- utilities: **`vload, vlsdev, vdump`** (see Appendix A).
- a **`vmeotty`** driver.

The boards which are taken into account by the system are deduced from the configuration of the **`vmeo`** driver. For each board which is configured, a "logical number" is given. This number allows the association of the node names to the VMEOb boards and their lines (see Software configuration below).

8.2 Hardware configuration

The required hardware configuration is:

- one VMPC CPU board,
- one or two VMEOb boards.

WARNING

Other hardware configurations exist:

- several VMEOb in synchronous mode,
- working in VME clusters,
- more than two VMEOb boards, several CPUs, ...

CONTACT CETIA for these configurations.

8.3 Software configuration

The VMEOb boards can be at one of the following addresses:

Coder Wheel	VME Address
1	0xE8000000
2	0xE8400000
3	0xE8800000
4	0xE8C00000
5	0xE9000000
6	0xE9400000
7	0xE9800000
8	0xE9C00000
9	0xEA000000
A	0xEA400000
B	0xEA800000
C	0xEAC00000

The following configuration is the default: two VMEOb.

Logical number	0	1
Coder wheel	4	6
VME address	0xE8C00000	0xE9400000
VME IT used	4	4
IT vector	0xc5	0xc7

CAUTION Check that there is no conflict with other boards on the VME bus or with the VME IT levels and vectors used.

To change this configuration, in **`/sys/dheaders/vmeoinfo.h`** modify:

```
#define WHEEL_b0       4     /* coding wheel for board 0. default: 4 */
#define IT_LVL_b0      4     /* IT level for board 0: default 4 */
#define WHEEL_b1       6     /* coding wheel for board 1. default: 6 */
#define IT_LVL_b1      4     /* IT level for board 1: default 4 */
                             /* it vector or 0 if protocol not available */
                             /* tty = protocol 0 */
#define VECT_b0_p0     0xc5  /* it vector for board 0 proto 0 def:0xc5*/
#define VECT_b1_p0     0xc7  /* it vector for board 1 proto 0 def:0xc7*/
```

WARNING

— The **`xx_p1`** or **`xx_p2`** constants have no significance if the complementary options are installed.

— The configuration does not depend on the presence of the boards. Thus, for the default configuration, if a VMEOb with the coder wheel on 4 is not set, only the VMEOb with the coder wheel on 6 is taken into account. If there is no VMEOb with coder wheel on 4 and 6 but if there is a VMEOb with the coder wheel setting on another value, this board is not taken into account.

8.4 VMEOb Driver Implementation

The **`/usr/bin/Install.vmeob`** script automates all the VMEOb driver installation steps and generates a new kernel with this driver. This script also adds the **`vload`** command to load the firmware to the **`/bin/rc`** file.

The steps run automatically by the **`Install.vmeob`** script are detailed below:

CAUTION The VMEOb boards run only if the **almavme** VME driver is installed. This driver is exclusive to the **avme** driver.

Once the configuration is built:

— If **`vmeoinfo.h`** is modified, recompile **`/sys/devices/vmeoinfo.c`**:

```
        # cd /sys/devices
        # touch vmeoinfo.c
        # make all
```

— In **`/sys/lynx.os/CONFIG.TBL`**, enable the following lines:

```
        vmeo.cfg
        vmeotty.cfg
```

WARNING The line **`vmeo.cfg`** must be before **`vmeotty.cfg`** in the file.

— Rebuild the kernel and nodes.

```
        # cd /sys/lynx.os
        # make install
        # cd /dev
        # rm *
        # mknod -a /etc/nodetab
```

The board with the logical number ***`i`*** will be associated with the node **`vmeo`*`i`***. The ports of each VMEOb board will be associated with a node **`vtty`*`n`*** as:

- **vtty0** to **vtty15** corresponding to ports 0 to 15 of the VMEOb 0,
- **vtty16** to **vtty31** corresponding to ports 0 to 15 of the VMEOb 1.

— Reboot the system.

— Download the firmware for each VMEOb which has the logical number ***`i`***:

```
        /usr/local/bin/vload -bi -f /usr/local/bin/vmeob.24
```

(see **`vload`** command).

WARNING This firmware must be downloaded at each reboot. These commands can be placed in the **`/bin/rc`** file.

To uninstall the VMEOb driver, use the **`/usr/bin/Uninstall.vmeob`** script.

8.5 Activation of the VMEOb lines in tty mode

To determine the available VMEOb boards, see the **`vlsdev`** command. See above for the association between the **`vtty`*`n`*** nodes and the VMEOb ports.

Next, the activation of these lines is the same as for any other TTY line (see manual: tty).

9. Implementing the VMPCx-Dual platform

NOTE

The master processor is the first processor, LynxOS is first booted on this processor.
It drives the serial line 1.

The slave processor is the second processor.
It drives the serial line 2.

9.1 Master Processor Configuration

The master processor kernel is built in **`/sys/lynx.os`** as a standard LynxOS kernel.
Drivers are configured as usual by editing the **`/sys/lynx.os/CONFIG.TBL`** file or by running the usual install scripts.

But, in order to use the VMPCx-Dual board in bi-processor mode, you must configure the kernel as a master kernel. This is automatically done by using the script:

```
        LynxOS# /usr/bin/Install.master
```

This script installs a new kernel. Then, the system must be rebooted by the command:

```
        LynxOS# reboot -aN
```

9.2 Slave Processor Configuration

The slave processor kernel is built in the **`/sys/lynx.os.s`** directory.

There is an error in the distribution, the file **`/sys/lynx.os.s/nodetab`** is linked to **`sys/lynx.os/nodetab`**.
This is wrong and you must delete this link.

A romable slave image is created by running the script **`mkslave`** into **`/sys/lynx.os.s`**. This script rebuilds the slave kernel and generates the image file **`rom_dualslave`** with a ram file system. This ram file system is defined by the specification file **`dualslave.spec`** used by the **`mkimage`** utility. Therefore, if you want to add a file in the ram file system, you must add this file in **`dualslave.spec`**.

The hardware resources (SCSI, VME, PCI, ...) are not shared and must be allocated to one of the two processors by selecting the drivers in the corresponding **`CONFIG.TBL`** file. However, for some resources, communication mechanisms exist.
For more information about these mechanisms, contact CETIA.

9.3 Configuring PMCs on a Dual Board

It is possible with PCI Kit to drive each PMC independently by ONE of the 2 CPUs.

The following steps must be done to configure PMCs:

- On the Master processor:
  - In the directory **`/sys/lynx.os`**, edit the **`CONFIG.TBL`** file:

```
cd /sys/lynx.os
vi CONFIG.TBL
```

  - Uncomment the line:

```
I:pcibus.cfg
```

- On the Slave processor:
  - In the directory **`/sys/lynx.os.s`** edit the **`CONFIG.TBL`** file:

```
cd /sys/lynx.os.s
vi CONFIG.TBL
```

  - Uncomment the line:

```
 I:pcibus_client.cfg
```

- PMC installation:
  Include each PMC driver in the CONFIG.TBL file of the concerned CPU and NOT in the other one.

- Then generate the new kernel for the master:

```
cd /sys/lynx.os
make install
reboot -aN
```

- Generate the new romable slave image:

```
cd /sys/lynx.os.s
./mkslave
```

NOTE

This feature is only available for PMC using CETIA PCI-Kit. It is not available for DRM. If you have PMC drivers using DRM you could only include its on the master CPU.

WARNING

There are some restrictions when the **pcibus_client** driver is included into the slave processor kernel. This driver copies the PCI device table list from the master CPU and then it takes every unused PMC and PMC interrupt to him. So the master PMC must be installed and the interrupt must be connected before the slave CPU **pcibus_client** driver installation. Also, the **pcibus_client** driver supports the ICPMC-6 carrier board but not the ICPMC carrier board.

9.3.1 Ethernet PMC

- PMC on the Master CPU:
  Please refer to the CPMC-ETH100T or CPMC-ETH-xx Chapter.

- PMC on the Slave CPU:
  - ♦ Run the **`Uninstall`** script associated with you ethernet PMC, to uninstall the PMC ethernet driver from the master CPU if it was installed.
    - For CPMC-ETH100T PMC:
      `Uninstall.eth100t`
    - For CPMC-ETH-xx PMC:
      `Uninstall.cpmc-eth`
  - ♦ Edit the **`/sys/lynx.os.s/CONFIG.TBL`** file and uncomment the driver associated with the Ethernet PMC:
    - For CPMC-ETH100T PMC:
      `I:cpmc-eth100t_slave.cfg`
    - For CPMC-ETH-xx PMC:
      `I:cpmc-eth_slave.cfg`
  - ♦ Edit the **`rc.network`** file of the slave processor and add the Ethernet interface installation:
    - For CPMC-ETH100T:

```
# This starts the CPMC-ETH100T interface
start_it /bin/ifconfig cpmc-etha0 ip_address
```

    - For CPMC-ETH-xx:

```
This starts the CPMC-ETH-xx interface
start_it /bin/ifconfig cpmc-ethxa0 ip_address
```

- 1 PMC on the master CPU and the other one on the slave CPU (Only if these Ethernet PMCs are identicals). In this case, just follow these steps:
  - ♦ Run the **`Install`** script associated with your ethernet PMC, to install the PMC ethernet driver from the master CPU.
  - ♦ Edit the configuration file of you ethernet device in the **`/sys/cfg`** directory (**`cpmc-eth100t.cfg`** for CPMC-ETH100T and **`cpmc-eth.cfg`** for CPMC-ETH-xx) then comment the interfaces you don’t want to be configured on the master CPU.

    In the following case only the first CPMC-ETH100T PMC will be configured on the master:

```
#
# CPMC-ETH100T PMC board
#
# Up to 4 CPMC-ETH100T can be used simultaneously.
# 1rst CPMC-ETH100T interface
C:CPMC-ETH100Ta: \
       ::::: \
       :::eth100tinstall:eth100tuninstall
D:cpmc-etha:eth100ta_info::
N:eth100ta:0:

# 2nd CPMC-ETH100T interface
#C:CPMC-ETH100Tb: \
#        ::::: \
#        :::eth100tinstall:eth100tuninstall
```

```
#D:cpmc-ethb:eth100tb_info::
#N:eth100tb:0:

# 3rd CPMC-ETH100T interface
#C:CPMC-ETH100Tc: \
#       ::::: \
#       :::eth100tinstall:eth100tuninstall
#D:cpmc-ethc:eth100tc_info::
#N:eth100tc:0:

# 4th CPMC-ETH100T interface
#C:CPMC-ETH100Td: \
#        ::::: \
#        :::eth100tinstall:eth100tuninstall
#D:cpmc-ethd:eth100td_info::
#N:eth100td:0:
```

- Edit the **/sys/lynx.os.s/CONFIG.TBL** file and uncomment the driver associated with the ethernet PMC.

- Edit the **rc.network** file of the slave processor and add the ethernet interface installation:
  - For CPMC-ETH100T:

```
# This starts the CPMC-ETH100T interface
start_it /bin/ifconfig cpmc-etha0 ip_address
```

  - For CPMC-ETH-xx:

```
# This starts the CPMC-ETH-xx interface
start_it /bin/ifconfig cpmc-ethxa0 ip_address
```

9.3.2 CPCIGx graphics PMC

If you want your CPCIGx graphics PMC to be drived by the slave processor, then follow these steps for uninstallation of the graphics driver on the master, and installation of the graphics driver on the slave CPU.

- Driver uninstallation on the master:
  - Edit the master **CONFIG.TBL** file:

```
cd /sys/lynx.os
vi CONFIG.TBL
```

  - Comment the following lines:

```
I:usb.cfg
I:atc.cfg
n:vt00:@atc0::
n:vt01:@atc1::
n:vt02:@atc2::
n:vt03:@atc3::
```

- Driver installation on the slave:
  - Edit the master **CONFIG.TBL** file:

```
cd /sys/lynx.os.s
vi CONFIG.TBL
```

  - Add the following lines:

```
I:usb.fcg
I:atc.cfg
I:cetiamouse.cfg
n:vt00:@atc0::
n:vt01:@atc1::
n:vt02:@atc2::
n:vt03:@atc3::
```

- Then add in the ram file system of the slave processor the directory used during the graphics installation, for example **/X11**. Its content will be mounted either via NFS or on a partition reserved to the slave processor.
  - ♦ Add in the **/bin** directory of the ram file system the script file **/bin/runx**.
  - ♦ Copy **/etc/ttys** to **/etc/ttys.slave**. The content of these files must be set as follows:
    - **/etc/ttys**:

```
/dev/con:1:default96:dumb:/bin/login
...
/dev/atc0:1:default:vt100at:/bin/login
/dev/atc1:1:default:vt100at:/bin/login
...
```

    - **/etc/ttys.slave**:

```
/dev/con:0:default96:dumb:/bin/login
...
/dev/atc0:1:default:vt100at:/bin/login
/dev/atc1:1:default:vt100at:/bin/login
...
```

- Finally, add at least the following links in the **dualslave.spec** file, as follows:

```
directory = /usr/lib
          symlink /XM/usr/lib/X11 /usr/lib/X11
directory = /usr/bin
          symlink /XM/usr/bin/X11 /usr/bin/X11
```

9.3.3 CPMC-4ASYNC, CPMC-8ASYNC, CPMC-16ASYNC PMCs

Four Scripts are provided to install/uninstall these devices:

- **Install.Xasync, Uninstall.Xasync** (for the master CPU),
- **Install.Xasync.s, Uninstall.Xasync.s** (for the slave CPU).

So, you can use these scripts to configure your CPMC-xASYNC. Take care not to configure the same PMC either on the master and the slave CPU.

9.4 Starting the System

9.4.1 Booting LynxOS on the Master

Memory must be reserved for the slave processor. For example, if you have a 32MB memory and the slave processor uses 16MB you have to reserve 16MB = 4096 x 4KB pages. To do this, use the boot option @R4096.

The L2-cache must be in "write through" mode if the processor is a PowerPC750 revision 2. To configure or check this mode, use the VMPCBug command:

```
        conf ;m.
```

NOTE The host machine name should be changed in `/net/rc.network` and perhaps also in `/etc/hosts`.

9.4.2 Booting Slave Processor

The `artload()` utility boots the slave processor from the master:

```
        artload -b <image_file> <memory_address> <size_memory_of_the_board_in_MB>
```

For example, with a memory size of 128MB and if the slave is loaded at 64MB with the `/sys/lynx.os.s/rom_dualslave` image file, the command will be:

```
        artload -b /sys/lynx.os.s/rom_dualslave 0x04000000 128
```

The example script `load_slave` executes this command:

```
bash# load_slave 128
load_slave [cpu_size]
example for a 64Mb VMPC-Dual board : load_slave 64
default is 32Mb board

Load slave at 67108864; (master must be booted with option @R16384)

artload ...
ZMAGIC (LynxOS 3.x.x mkimage) Rom File Exec Format
Downloading "/sys/lynx.os.s/rom_dualslave" on slave at 0x4004000
Downloading Header (size=0x20)...
Downloading text/data at 0x4004020
Read done 7798784 bytes
Boot options are "" at 0x4004f00
Start address of "/sys/lynx.os.s/rom_dualslave" on CPU board 0x4000000 is : 0x4020
CPU Dram size is : 128Mb
Slave number is  : 1
Starting Slave(1) 0x4000000 at 0x4020
bash#
```

NOTE You can modify the `load_slave` example script, if your parameters are different.

When the slave processor leaves the firmware, it prints `GO at 0x04004020` on the serial line 2.
Then, the LynxOS banner is printed to the serial line 2 and you can log in as usual.

When the slave processor is rebooted, the master processor must be also rebooted if you want to boot again the slave processor.

9.5 Communication between Processors

VMPCx-Dual resource sharing:

Serial line 1:	**cpu-0**
Serial line 2:	**cpu-1**
VME driver:	Managed by **cpu-0** (**`almavme.cfg`**).
PCI driver:	Managed by one of the two CPUs. Is allocated only for the managing CPU.
On-board SCSI:	Managed by **cpu-0**.
On-board Ethernet:	**cpu-0**
COM driver:	**cpu-0** : **`com_master`**. **cpu-1** : **`com_slave`**.
Reboot:	**cpu-0** reboot will involve the VMPCx-Dual and **cpu-1** reset. You must first reboot **cpu-1** to obtain a clean system halt.
Real time clock:	If one of the two processors modifies the date, the other one is not told about it and its date is not changed. To take into account this new date, the other processor must be rebooted.

NOTE The **`artcom`** driver provides a complementary sharing mechanism. Contact CETIA for more information about its use.

The SCMP service permits TCP/IP exchanges between the two processors which communicate via the shared DRAM. Therefore, NFS mounting, rsh commands, etc. can be done between both processors.

NOTE The slave CPU can only be downloaded from the master processor of its own board with the **`artload()`** command. It cannot be downloaded with the **`slaveboot()`** command.

9.6 SCMP with Dual Board

To configure SCMP on a single dual board, execute as follows:

- Configure the master as described in the Chapter "SCMP Support".
- Edit the `/sys/lynx.os.s/CONFIG.TBL` file, uncomment the lines:

  ```
  I:almavme_client.cfg
  I:bplane.cfg
  ```

  and rebuild the slave kernel.
- Edit the `/net/rc.network.dualslave` file, initialize:

  ```
  bplane_name=
  ```

  uncomment the lines:

  ```
  start_it /bin/ifconfig bplane0 "$bplane_name"
  start_it /net/bpconfig >/dev/null
  ```

  and then build the image (with `mkslave`) and download the slave processor (with `artload`).

WARNING If the slave CPU must be rebooted, you have to reboot the master CPU otherwise the slave CPU should not be reload (CRP2126).

Firmware Configuration:
The firmware **VME to DRAM A32 Gap** should be the sum of the master CPU VME gap and the slave CPU VME gap. So, it is twice of the SCMP VME address range. When the VME address range and the `bplane.conf` are chosen, each board should be programmed under firmware with the same VME address and size as LynxOS for the board.
The VME firmware address is:

(VME Board id) * (VME to DRAM A32 Gap) + (VME to DRAM A32 Base Offset).

Only **VME to DRAM A32 Base Offset** should be calculated because other parameters are known.

For example:

- The contents of the `/etc/bplane.conf` file for a configuration with 3 VMPCx-Dual boards (Board ids set to 0, 2, 4) with a VME_address_range of 64MB is:

  ```
  # type boot  addr       io      hostname
     5     0  0x00000000 0x0       cpu0
     5     0  0x04000000 0x1       cpu1
     5     0  0x08000000 0x200     cpu2
     5     0  0x0C000000 0x201     cpu3
     5     0  0x10000000 0x400     cpu4
     5     0  0x14000000 0x401     cpu5
  ```

- The board 0 firmware programmation is:

  ```
  VME Board id : 0
  VME to DRAM A32 Base Offset : 0x0
  VME to DRAM A32 Gap : 0x8000000
  VME to DRAM A32 Base Address is : 0x0
  ```

- The board 2 firmware programmation is:

```
VME Board id : 2
VME to DRAM A32 Base Offset : 0xF8000000
VME to DRAM A32 Gap : 0x8000000
VME to DRAM A32 Base Address is : 0x08000000
```

- The board 4 firmware programmation is:

```
VME Board id : 4
VME to DRAM A32 Base Offset : 0xF0000000
VME to DRAM A32 Gap : 0x8000000
VME to DRAM A32 Base Address is : 0x10000000
```

9.7 Additional Information

■ The new SCSI driver `simncr` cannot be used to share the SCSI resource by both CPUs yet.
In other words, only the lines

```
I:sysscsi.cfg
```

and

```
I:sysscsi_client.cfg
```

may be both validated in the `CONFIG.TBL` files for the master CPU and slave CPU.
There is no equivalent `simncr_client.cfg` configuration file, yet.

WARNING Raw devices like tape SCSI devices cannot be used by the slave processor.

■ **If the processor version is less than 3.0 on the VMPC5a-Dual or VMPC6a-Dual:**
The L2-cache must be in "Write Through" mode on each processor. Use the VMPCBug command `conf ;m` to modify or check this mode. The LynxOS kernel does not modify the L2-cache configuration.

■ **VMEOb driver:**
By default, the VMEOb board is only managed by one CPU (refer to `lynx.os/CONFIG.TBL`).
However, in the case of the VMPCx-Dual, you can share the VMEOb resources between both CPUs. For this:

- ♦ For CPU0: generate the kernel with VMEOb as usual.
- ♦ For CPU1: before generating the kernel (`/sys/lynx.os.s/a.out`), modify the interrupt allocation in `/sys/dheaders/vmeoinfo.h`. The allocation rule is: 1 interrupt level per CPU and 1 interrupt vector per protocol.

  Example: Set `IT_LVL_b0` to 5 (the default value is 4 which is used by CPU0).

  1. Set `IT_LVL_b0` to 5 in `/sys/dheaders/vmeoinfo.h`.
  2. Recompile `/sys/devices/vmeoinfo.c` and archive:

```
# cd /sys/devices
# touch vmeoinfo.c
# make all
```

3. Generate the CPU1 kernel:

   ```
   # cd /sys/lynx.os.s
   ```

   Add **I:vmeo.cfg** in **CONFIG.TBL** (uncomment the line)

   ```
   # mkslave
   ```

4. Remember to restore **IT_LVL_b0** with its initial value if you must generate a new CPU0 kernel. In this case:
   a. Set **IT_LVL_b0** to 4 in **/sys/dheaders/vmeoinfo.h**.
   b. Recompile **/sys/devices/vmeoinfo.c** and archive:

      ```
      # cd /sys/devices
      # touch vmeoinfo.c
      # make all
      ```

   c. Generate the new CPU0 kernel.

10. SCMP Support

CAUTION SCMP is limited to 8 processors (8 single boards or 4 dual boards).

CAUTION In a dual configuration, first you have to proceed Chapter "Implementing the VMPCx-Dual platform".
The firmware version must be at least 2.3.

WARNING The **slaveboot** man page is wrong. The syntax is described is § 10.3 "Diskless Configuration". The **`slaveboot`** command does not manage the slave processor of dual boards.

WARNING The driver must be compilated if the **`slaveboot`** command is used:

```
cd /sys/drivers.vmpc/vme-bplane
make all
```

and then rebuild the kernel.

The SCMP driver distributed in this release is the standard LynxOS SCMP driver. The LynxOS documentation **User Guide** is valid for VMPC boards.

There are many configurations to use SCMP:

- the board boots on the SCSI disk (see § 10.2),
- the board boots from the Ethernet network or from the on-board flash on a local ramfilesystem,
- the board boots from the Ethernet network or from the on-board flash and mounts the filesystem from the Ethernet network,
- the board boots from the VME bus and mounts the filesystem from the SCMP network (see § 10.3 and § 10.4).

10.1 SCMP Configuration Files

Three files define the SCMP configuration:

- **`/etc/bplane.conf`**,
- **`/sys/devices/if_bplaneinfo.c`**,
- **`/sys/lynx.os/conf.c`**.

The **`almavme.cfg`** driver should be into the kernel before the SCMP driver.

10.1.1 /etc/bplane.conf file

The **`/etc/bplane.conf`** file defines the number of VMPC boards and their addresses on the VME bus. Each line describes one board in the SCMP configuration. The parameters are:

- type of the board: always 5 for a VMPC board,
- boot: always 0,
- VME address of the board: it depends on your configuration,
- VME IO address of the board: it depends on your configuration,
- hostname: name of the board, it must be also defined in **`/etc/hosts`**.

10.1.1.1 Standard case

First, you have to define the VME address range used by each board.

The maximum VME address range for all VMPC boards is 512MB. Therefore, if you have 5 boards with 128MB of DRAM, you can not reserve 128MB of VME space per board, you must reduce to 64MB.
On the other hand, the slave VME space must be enough to access to the SCMP internal buffers. These buffers are reserved when LynxOS is booting. After boot, use the **`ps`** command to know how much memory is used by LynxOS and SCMP buffers.
For instance, for a memory size of 128MB, the **`ps`** command will say:

```
pid ppid pgrp pri text   stk data      time dev   user   S name
 29    1   29  17  300    28  116      0.02 con   root   W /bin/bash
 30   29   30  17   52    12   28      0.03 con   root   C /bin/ps
116508K/0K free physical/virtual, 564K used (in this display)
```

116MB are free, so 12MB are used by LynxOS and SCMP buffer. The VME address range can be 16MB or more.
This must be done for each board and the maximum for all boards must be used for each board.
The VME address range must be a power of 2 (8, 16, 32, 64, ...).
The VME address range must be set in the file **`sys/lynx.os/conf.c`** with the global variable **`pwb_vme_gap`**.
The VME address range must be programmed under the firmware environment on each board with the **`conf`** command:

```
COMMAND> conf;m
 ...
 VME to DRAM A32 Gap             = 8000000 ?   -> set the VME address range.
 ...
```

The board identifier of each board must be programmed under firmware also with the **`conf`** command (each board has its own board id):

```
COMMAND> conf;m
 ...
 VME Board Id (0x0-0xFF)         = 0 ?       -> set the board id (0 to n)
 ...
```

CAUTION

Increment the VME ID with each board, i.e. if the first board VME ID is 0 the second board VME ID is 1, etc.
The VME board IDs must form a single continuous sequence with no gap.

Then, you can program the file **`/etc/bplane.conf`**:

- VME address of the board: it is (board_id * VME_address_range)
- VME IO address of the board: it is (board_id * 0x100)

Here is the contents of the **`/etc/bplane.conf`** file for a 3 VMPC board configuration with a VME_address_range of 64MB and board ids set to 0, 1 and 2.

```
# type boot  addr      io      hostname
   5     0  0x00000000 0x0      cpu0
   5     0  0x04000000 0x100    cpu1
   5     0  0x08000000 0x200    cpu2
```

10.1.1.2 Dual CPU board case

For a VMPC-Dual board, the configuration will be the same except:

- One board has two entries in the file.
- The file **`/sys/lynx.os/conf.c`** must be modifed to set the kernel load address for the slave CPU. This is done with the variable **`pwb_slv_load`**.
- VME IO address for the slave CPU is equivalent to the value for the master CPU plus 1.
- Here is an example for 3 VMPC-Dual boards (board id 0, 2 and 4) and 1 VMPC board (board id 6) with a 64MB VME address range.

  ```
  # type boot  addr    io      hostname
    5     0 0x00000000 0x0      cpu0
    5     0 0x04000000 0x1      cpu1
    5     0 0x08000000 0x200    cpu2
    5     0 0x0C000000 0x201    cpu3
    5     0 0x10000000 0x400    cpu4
    5     0 0x14000000 0x401    cpu5
    5     0 0x18000000 0x600    cpu6
  ```

  The file **`/etc/bplane.conf`** must be the same on each board.
- On slave processor **`almavme_client`** should be included into the kernel instead of the **`almavme`** driver.
- Refer to section "SCMP with Dual Board" for more information.

10.1.2 /sys/devices/if_bplaneinfo.c file

`MAX_NO_PROCESSORS` defined in this file, indicates the maximum number of processors in the SCMP environment. Therefore, the number of CPUs of the VMPC boards must be smaller than this number.
If you have 5 VMPC5a, **`MAX_NO_PROCESSORS`** must be at least 5.
If you have 3 VMPC6a-Dual, **`MAX_NO_PROCESSORS`** must be at least 6.

10.1.3 /sys/lynx.os/conf.c file

The **`pwb_vme_gap`** variable defines the VME address range for each board.

The **`pwb_slv_load`** variable is only used for VMPC-Dual boards. It defines the kernel load address for the slave CPU.

The files **`/sys/devices/if_bplaneinfo.c`** and **`/sys/lynx.os/conf.c`** must be the same for each board.

Also **`VMPCOPT_MCP`** should not be set into **`vmpc_option`** data (set 0 if you don’t want options).

10.2 Enabling SCMP

The following settings must be done to install SCMP:

- the board must be configured under firmware,
- the files **`/etc/bplane.conf`** and **`/sys/lynx.os/conf.c`** must be written,
- the file **`/sys/devices/if_bplaneinfo.c`** must be written, compiled and archived into the device library,

  ```
  # cd /sys/devices
  # make all
  ```

- The file **`/etc/hosts`** must include the IP name and address of the sub-network.
- the SCMP driver must be installed. Use the commands:

  ```
  # cd /sys/lynx.os
  # make install.tcpip
  ```

  Answer the questions, when the following menu appears:

```
.
.
.
Which rc.network file should be modified?

  1: Work directly on /net/rc.network
  2: Make local version of rc.network file

please enter response: [1] 1

  /net/rc.network : /bin/ifconfig lo0
  /net/rc.network : echo Network interface
  /net/rc.network : start_it /net/inetd

Network Interface Configuration
===============================
  1. [ethernet] Ethernet Interface
  2. [scmp] SCMP VMEbus Backplane Network
  3. [quit] Network interface configuration done

Please select an option by number or [keyword]: 2

Configuring SCMP VMEbus Backplane Network...
```

```
--------------------------------------------

  CONFIG.TBL : I:bplane.cfg

Please Select the Board Type [mvme1600, mvme2600,PowerCore,vmpc]: vmpc
  CONFIG.TBL : I:almavme.cfg
  CONFIG.TBL : I:almavme_client.cfg
  CONFIG.TBL : #I:avme.cfg
Do you want to configure this system as a diskless client? n

Which hosts data base file should be modified?

  1: Work directly on /etc/hosts
  2: Make local version of hosts database file

please enter response: [1] 1


Enter name for the SCMP interface: cpu0

Searching hosts for host cpu0.
```

If the host name already exists in the **`/etc/hosts`** file, the following line is displayed:

```
y.y.y.y   cpu0

Modify the entry for host cpu0? [y,n] (y): n

  /net/rc.network : bplane_name="cpu0"
  /net/rc.network : start_it /bin/ifconfig bplane0
  /net/rc.network : start_it /net/bpconfig

Network Interface Configuration
===============================
  1. [ethernet] Ethernet Interface
  2. [scmp] SCMP VMEbus Backplane Network
  3. [quit] Network interface configuration done

Please select an option by number or [keyword]: 3

Additional Network Configuration
================================
Please enter your system's hostname (eg. lynxdemo): lynxdemo


Searching hosts for host lynxdemo.
x.x.x.x   lynxdemo

Modify the entry for host lynxdemo? [y,n] (y): n
.
.
.
```

This command adds the **almavme, bplane** and **hbtcpip** drivers in the file **/sys/lynx.os/CONFIG.TBL** and also modifies the file **/net/rc.network**. It adds the SCMP hostname **bplane_name** and uncomments the two lines:

```
start_it /bin/ifconfig bplane0 "$bplane_name"
start_it /net/bpconfig >/dev/null
```

Finally, the kernel is rebuilt and installed.

Now, you can rebuild the nodes.

```
# reboot -aN
```

10.3 Diskless Configuration

This configuration is as follows:

- a master board with disk,
- many other boards without disk, downloaded by the master.

Execute the following steps:

1. Configure the master as described in section "Enabling SCMP" except for the **/etc/bplane.conf** file which must contain the IP address. For example:

```
# type boot  addr      io      hostname
   5     0  0x00000000 0x0      192.93.162.1
   5     0  0x02000000 0x100    192.93.162.2
   5     0  0x04000000 0x200    192.93.162.3
   5     0  0x06000000 0x300    192.93.162.4
```

   Also, **.** must be added in **$PATH**.

2. Edit the **/etc/inetd.conf** file and uncomment the line:

```
tftp       dgram   udp     wait     root      /net/tftpd      tftpd
```

   Then, find the inetd pid:

```
ps -ax | grep inetd
23    1    23   17   80      8    28     0.00     root     W /net/inetd
```

   And kill this pid:

```
kill -1 23
```

3. Create a ***/clients*** directory:

```
mkdir /clients
```

   And create a link per diskless board:

```
ln -s / /clients/cpu1
ln -s / /clients/cpu2
ln -s / /clients/cpu3
```

4. Edit the **`/etc/exports`** file and add a line per diskless board:

   ```
   /clients/cpu1 root=cpu1
   /clients/cpu2 root=cpu2
   /clients/cpu3 root=cpu3
   ```

   Then, export the new directories:

   ```
   exportfs -va
   ```

5. Generate an image for the first diskless board and download it:
   - Configure the diskless:

     ```
     # cd /sys/romkit/scmp_boot
     # make config CPUID=1
     Creating '/sys/romkit/scmp_boot/sys-1'...
     Creating '/sys/romkit/scmp_boot/sys-1/dist'...

     SCMP Netboot Configuration
     **************************


     TFTP-Server Configuration
     =========================

     Please specify the location of the TFTP-directory [/tftpboot]:

     Enter the IP-Address the TFTP-Server (aaa.bbb.ccc.ddd) [192.1.1.1]: 192.93.162.1

        Checking IP-Address 192.93.162.1...ok.


     NFS Server Configuration
     ========================

     Specify the name of the NFS-Server for the root filesystem [scmp-cpu1]: cpu0

     Specify the directory containing the root filesystem on the NFS-Server [/clients/cpu2]: /clients/cpu1


     Diskless Client Configuration
     =============================

     Enter the IP-Address of the diskless client (aaa.bbb.ccc.ddd) [192.1.1.2]: 192.93.162.2

        Checking IP-Address 192.93.162.2...ok.

     Hosts database to use on this client [/etc/hosts]:

     SCMP configuration database to use on this client [/etc/bplane.conf]:

        Writing file netboot.config-1...ok.
     ```

   - Generate the diskless image:

     ```
     # make all CPUID=1

     Creating '/sys/romkit/scmp_boot/sys-1/lynx.os'...
     Using template '/sys/romkit/scmp_boot/sys-1/rc.network'...

     Building '/sys/romkit/scmp_boot/sys-1/lynx.os/a.out'...

        You will be asked several questions about your target
     ```

```
   configuration during the configuration process.

/bin/make  PARSE=Done COMPONENTS="tcpip nfs" all.all

Configuring TCP/IP...
*********************

Scanning networks: ethernet scmp

  CONFIG.TBL : I:hbtcpip.cfg
  CONFIG.TBL : #I:nullnux.cfg
  /sys/romkit/scmp_boot/sys-1/rc.network : /bin/ifconfig lo0
  /sys/romkit/scmp_boot/sys-1/rc.network : echo Network interface
  /sys/romkit/scmp_boot/sys-1/rc.network : start_it /net/inetd

Network Interface Configuration
===============================

   1. [ethernet] Ethernet Interface
   2. [scmp] SCMP VMEbus Backplane Network
   3. [quit] Network interface configuration done

Please select an option by number or [keyword]: 2

Configuring SCMP VMEbus Backplane Network...
--------------------------------------------

  CONFIG.TBL : I:bplane.cfg

Please Select the Board Type [mvme1600, mvme2600,PowerCore,vmpc]: vmpc
  CONFIG.TBL : I:almavme.cfg
  CONFIG.TBL : I:almavme_client.cfg
  CONFIG.TBL : #I:avme.cfg

Network Interface Configuration
===============================

   1. [ethernet] Ethernet Interface
   2. [scmp] SCMP VMEbus Backplane Network
   3. [quit] Network interface configuration done

Please select an option by number or [keyword]: 3

Configuring NFS ...
*******************

Install NFS client support [y,n] (y): y
Install NFS server support [y,n] (y): y
Install NFS client caching [y,n] (y): y
Install NFS file locking [y,n] (y): y

  CONFIG.TBL : I:nfs.cfg
  CONFIG.TBL : #I:nullnfs.cfg
  /sys/romkit/scmp_boot/sys-1/rc.network : mount -a -t nfs
  /sys/romkit/scmp_boot/sys-1/rc.network : start_it /net/unfsio
  CONFIG.TBL : I:nfssvc.cfg
  /sys/romkit/scmp_boot/sys-1/rc.network : start_it /net/portmap
  /sys/romkit/scmp_boot/sys-1/rc.network : start_it /net/mountd
  /sys/romkit/scmp_boot/sys-1/rc.network : start_it /net/nfsd
  /sys/romkit/scmp_boot/sys-1/rc.network : [ -s /etc/exports
  /sys/romkit/scmp_boot/sys-1/rc.network : echo exporting
  /sys/romkit/scmp_boot/sys-1/rc.network : start_it /bin/exportfs
  CONFIG.TBL : I:lock.cfg
  /sys/romkit/scmp_boot/sys-1/rc.network : start_it /net/portmap
  /sys/romkit/scmp_boot/sys-1/rc.network : start_it /net/rpc.statd
  /sys/romkit/scmp_boot/sys-1/rc.network : start_it /net/rpc.lockd.svc
  /sys/romkit/scmp_boot/sys-1/rc.network : start_it /net/rpc.lockd.clnt
  CONFIG.TBL : I:nfscache.cfg

/bin/make COMPONENTS=tcpip nfs PARSE=Done common_all

Creating bsp.vmpc.o ...

/usr/bin/ld -r -o bsp.vmpc.o rkboot.o mmuinit.o  etrap.o htime.o ioint.o main.o
ftrap.o time.o procinit.o tlbmiss.o version.o start.o trap.o powerpc_asm.o power
pc_uk.o kputc_ppc.o initmem.o sysctl_list.o hw_init.o hw_intr.o hw_delay.o hw_li
b.o hw_time.o hw_eeprom.o hw_stub.o hw_pci.o

Processing CONFIG.TBL ...

config CONFIG.TBL CONFIG.h nodetab sysdevices.h /sys/cfg.ppc

Rebuilding conf.o ...

/usr/bin/gcc -I. -c -I.  -DLynx -D__Lynx__  -fwritable-strings -fcommon -D__powe
rpc__  -DLYNXOS -D__LYNXOS -fwritable-strings -fcommon -fvolatile-global -fvolat
ile -finhibit-size-directive -I. -D__LYNXOS_RELEASE="3.0.1" -D__LYNXOS_VERSION
="110298-G" -D__vmpc__ -I. -I/usr/include -I/sys/dheaders -I. -I/../common/sys
romkitscmp_bootsys-1lynx.os -I/usr/include -I/../common/usr/include conf.c

Creating timestamp.o ...

Current date: Mon May 17 11:24:02 PDT 1999
/usr/bin/gcc -c timestamp.c

Linking kernel ...

/usr/bin/ld -bnogc -Tppclynx.x -Ttext b0004000 -N -e .start  -o a.out bsp.vmpc.o
 conf.o info.o timestamp.o -L/sys/lib -lsyscalls -ldevices -ldrivers_vmpc -ldriv
ers -ltcpip -lnfs_server -lnfs_client -lstreams   -lkernel -lsyscalls -lmisc

Using template ‘/sys/romkit/scmp_boot/ppc.netboot.spec’...
Creating ‘/sys/romkit/scmp_boot/sys-1/rc’...
   Reading netboot.config-1...ok.
   Writing rc...ok.
Using /etc/bplane.conf as SCMP configuration...
Creating Boot-Image (KDI) ‘/sys/romkit/scmp_boot/sys-1/lynxos.netboot’...
mkimage /sys/romkit/scmp_boot/ppc.netboot.spec /sys/romkit/scmp_boot/sys-1/lynxo
s.netboot
adding root device rd0 0 0 (0)

File system used:  421 blocks &  356 inodes
File system free:  506 blocks &   11 inodes

load map (static):
OS text -> 0x00004000
OS data -> 0x000fd000
OS stab -> 0x00113140
Root fs -> 0x0013b000
Resident-> 0x001af000

load map (runtime):
OS text -> 0x00004000 (0x000f9000) OS Entry: 0x00004020
OS data -> 0x000fd000 (0x00016140)
OS bss  -> 0x00113140 (0x00027528)
OS stab -> 0x0013a668 (0x000271cb)
Root fs -> 0x00162000 (0x00074000)
Resident-> 0x001d6000 (0x000de000)
OS end  -> 0x002b4000
Creating '/sys/romkit/scmp_boot/sys-1/fstab'...
   Reading netboot.
   Writing fstab...ok.
```

- Install the image in **`/tftpboot`**:

```
# make install CPUID=1
```

- Download the image:

```
# cd /tftpboot
# slaveboot 0 0x4000 1 0x4020 C05DA202.lynxos.netboot
```

6. Redo the preceding step for the other diskless boards with their parameters.

The **`slaveboot`** syntax is:

```
slaveboot address 0x4000 boardID 0x4020 romfile
```

- ***`address`***: the CPU physical address of the PCI to VME channel opened by SCMP for the board minus 0xD0000000.
  Type the **`almavmechan -printall`** command to obtain the physical address for each board then subtract 0xD0000000 from this address.
- ***`boardID`***: board ID of the board (0, 1, 2, ...). Refer to the firmware command **`CONF`** to obtain the board ID.
- ***`romfile`***: boot image generated by **`mkimage`**.

In the case of a dual CPU board, the slave CPU cannot be downloaded with the **`slaveboot()`** command.

10.4 Common Diskless Configuration

It is possible to generate a common diskless image for every board. But, in this case the following IP address should be into the **`/etc/hosts`** file:

```
192.93.162.100  cpu0
192.93.162.1    cpu1
192.93.162.2    cpu2
192.93.162.3    cpu3
192.93.162.4    cpu4
etc
```

NOTE

If you want to use other IP numbers, you have to change some files into the **`sys-VX`** directory. Then, in this directory, execute:

```
touch rc.network
```

and in **`/sys/romkit/scmp_boot`**:

```
make install -f MakefileVX
```

Because the IP address is **`192.93.162.`*`X`***, ***`X`*** is the board ID. Of course **`/etc/bplane.conf`** must be created, **`tftpserver`** must be enabled and **`/clients/cpuVX`** must be exported for each board. You have to add **`.`** into the PATH because of the scripts. Also the **VME Gap** must be set up into the **`conf.c`** file and the board must configurated under firmware.

The common image generation for a single board is done by the following commands:

```
# cd /sys/romkit/scmp_boot
# make install -f MakefileVX
Creating '/sys/romkit/scmp_boot/sys-VX/dist'...
Creating '/sys/romkit/scmp_boot/sys-VX/lynx.os'...
Using template '/sys/romkit/scmp_boot/sys-VX/rc.network'...

Building '/sys/romkit/scmp_boot/sys-VX/lynx.os/a.out'...
```

You will be asked several questions about your target configuration during the configuration process.

```
/bin/make  PARSE=Done COMPONENTS="tcpip nfs" all.all

Configuring TCP/IP...
*********************

Scanning networks: ethernet scmp

  CONFIG.TBL : I:hbtcpip.cfg
  CONFIG.TBL : #I:nullnux.cfg
  /sys/romkit/scmp_boot/sys-VX/rc.network : /bin/ifconfig lo0
  /sys/romkit/scmp_boot/sys-VX/rc.network : echo Network interface
  /sys/romkit/scmp_boot/sys-VX/rc.network : start_it /net/inetd

Network Interface Configuration
===============================

   1. [ethernet] Ethernet Interface
   2. [scmp] SCMP VMEbus Backplane Network
   3. [quit] Network interface configuration done

Please select an option by number or [keyword]: 2

Configuring SCMP VMEbus Backplane Network...
--------------------------------------------

  CONFIG.TBL : I:bplane.cfg

Please Select the Board Type [mvme1600, mvme2600,PowerCore,vmpc]: vmpc
  CONFIG.TBL : I:almavme.cfg
  CONFIG.TBL : I:almavme_client.cfg
  CONFIG.TBL : #I:avme.cfg

Network Interface Configuration
===============================

   1. [ethernet] Ethernet Interface
   2. [scmp] SCMP VMEbus Backplane Network
   3. [quit] Network interface configuration done

Please select an option by number or [keyword]: 3


Install NFS client support [y,n] (y): y
Install NFS server support [y,n] (y): y
Install NFS client caching [y,n] (y): y
Install NFS file locking [y,n] (y): y

...

Linking kernel ...
```

```
/usr/bin/ld -bnogc -Tppclynx.x -Ttext b0004000 -N -e .start  -o a.out bsp.vmpc.o
 conf.o info.o timestamp.o -L/sys/lib -lsyscalls -ldevices -ldrivers_vmpc -ldriv
ers -ltcpip -lnfs_server -lnfs_client -lstreams   -lkernel -lsyscalls -lmisc

Using /etc/bplane.conf as SCMP configuration...
Creating Boot-Image (KDI) ‘/sys/romkit/scmp_boot/sys-VX/lynxos.netboot’...
mkimage /sys/romkit/scmp_boot/ppc.netboot.specVX /sys/romkit/scmp_boot/sys-VX/ly
nxos.netboot
adding root device rd0 0 0 (0)

File system used:  447 blocks &  358 inodes
File system free:  504 blocks &   13 inodes

load map (static):
OS text -> 0x00004000
OS data -> 0x0010b000
OS stab -> 0x00124078
Root fs -> 0x00151000
Resident-> 0x001c8000

load map (runtime):
OS text -> 0x00004000 (0x00107000) OS Entry: 0x00004020
OS data -> 0x0010b000 (0x00019078)
OS bss  -> 0x00124078 (0x00029188)
OS stab -> 0x0014d200 (0x0002c391)
Root fs -> 0x0017a000 (0x00077000)
Resident-> 0x001f1000 (0x000e9000)
OS end  -> 0x002da000
Creating ‘/sys/romkit/scmp_boot/sys-VX/fstab’...
   Reading netboot.config-VX...ok.
   Writing fstab...ok.

SCMP Netboot Installation
*************************

   Reading netboot.config-VX...ok.

   Directory /tftpboot exists.
   Copying /sys/romkit/scmp_boot/sys-VX/lynxos.netboot to /tftpboot/VX.lynxos.ne
tboot...ok.
   Copying /sys/romkit/scmp_boot/sys-VX/fstab to /tftpboot/VX.fstab...ok.
   Copying /etc/hosts to /tftpboot/VX.hosts...ok.
```

Then you have into **`/tftpboot`** directory:

```
# ls -l
-rw-r--r--   1 root          230 Nov 25 18:28 VX.fstab
-rw-r--r--   1 root        30420 Nov 25 18:28 VX.hosts
-rwxr-xr-x   1 root      2805760 Nov 25 18:28 VX.lynxos.netboot*
```

VX.lynxos.netboot is the common image that is downloaded by **slaveboot** on every board. Then, create some one link on **VX.fstab** and **VX.hosts** per board:

For CPU1:

```
# cd /tftpboot
# ln -s VX.fstab C05DA201.fstab
# ln -s VX.hosts C05DA201.hosts
```

NOTE

`C05DA202` corresponds to the hexadecimal value of the IP address.

It is possible to modified your kernel into **./sys/romkit/scmp_boot/sys-VX/lynx.os**.

Dual Board in Diskless Configuration

For VMPC-Dual board, you can also generate on common image for the master CPUs and one common image for the slave CPU.

For the master CPU you have first to configure master mode with **/usr/bin/Install.master script** and set **pwb_slv_load** into **/sys/lynx.os/conf.c**. Then into **/sys/romkit/scmp_boot** edit **ppc.netboot.specMX** file and change if necessary the line:

```
cflags=@R16384
```

depending on the slave CPU address.

Then, it is the same as single CPU but used **MX** instead of **VX**:

```
make install -f MakefileMX
cd /tftpboot
ln -s MX.fstab C05DA202.fstab
ln -s MX.hosts C05DA202.hosts
```

For the slave CPU the suffix **SX** must be used instead of **VX** or **MX** but the generation must be do into two steps:

```
make all -f MakefileSX
Network Interface Configuration
==============================

   1. [ethernet] Ethernet Interface
   2. [scmp] SCMP VMEbus Backplane Network
   3. [quit] Network interface configuration done

Please select an option by number or [keyword]: 3

Configuring NFS ...
*******************

Install NFS client support [y,n] (y): y
Install NFS server support [y,n] (y): y
Install NFS client caching [y,n] (y): y
Install NFS file locking [y,n] (y): y
...
```

Then copy the slave **CONFIG.TBL** file:

```
# cd /sys/romkit/scmp_boot/sys-SX/lynx.os
# cp /sys/lynx.os.s/CONFIG.TBL .
```

Add the two lines of **CONFIG.TBL** files:

```
        I:almavme_client.cfg
        I:bplane.cfg
```

Generate again the kernel:

```
        # make all
```

And then generate the image:

```
        # cd ../..
        # make install -f MakefileSX
        # cd /tftpboot
        # ln -s SX.fstab C05DA201.fstab
        # ln -s SX.hosts C05DA201.hosts
```

the image **SX.lynxos.netboot** can be loaded by **artload**.

11. SCSIMGR Support

The new SCSI driver **`simncr`** supports Ultra fast and wide SCSI modes.
The **`simncr`** driver is not the default SCSI driver. The default one is the **`ll-scsi`** driver.

To install the new SCSI driver:

- Edit the file **`/sys/devices/simncr_info.c`** to define the transfer type for each SCSI device (normal, fast, wide or both fast and wide). For each, the choice may be:

  ```
  ALLOW_DISCONNECT
  ALLOW_DISCONNECT | SYNCHRONOUS_NEGOTIATE,
  ALLOW_DISCONNECT | WIDE_NEGOTIATE,
  ALLOW_DISCONNECT | SYNCHRONOUS_NEGOTIATE | WIDE_NEGOTIATE,
  ```

  Then, recompile the file and archive the data:

  ```
  # cd sys/devices
  # make all
  ```

- Configure the **`simncr`** driver in the file **`/sys/lynx.os/CONFIG.TBL`**. The standard **`ll-scsi`** driver must be disabled. You must have:

  ```
  I:simncr.cfg
  #I:sysscsi.cfg
  ```

  Rebuild the kernel:

  ```
  # cd sys/lynx.os
  # make install
  ```

  Then reboot the system by the command:

  ```
  # reboot -N
  ```

NOTE

The device names for the **`simncr`** driver are not the same as for the **`ll-scsi`** driver. They are:

`/dev/sdncr.`*`x`*

or

`/dev/rstncr.`*`x`*

where ***`x`*** is the peripheral SCSI ID.

WARNING

Problems may occur with some SCSI devices when disconnection is selected. Do not set this option in this case.

CAUTION

Be careful to respect Ultra SCSI specification for cable length and number of devices. This information is given in the "Connection Guide" for VMPC6 and VMPC6/RA boards and in the "User's Guide" for VMPC6/RC boards. It is possible to come back to Fast mode by setting **`dma_fifo_size`** field to 1 in the **`/sys/devices/simncr_info.c`** file.

12. High Precision Timer

12.1 Overview

The CETIA High Precision Timer is based on the decrementer register of the PowerPC. It allows the user to manage delays much smaller than the system clock tick (10 ms).

This service can be implemented for each CPU of a VMPCx-dual board simultaneously.

12.2 Configuring

The maximum number of timer entries managed by the timer driver (each entry defines one timer) is defined in the file **`$ENV_PREFIX/sys/lynx.os/conf.c`** with the global variable **`nbrtimer`**, so it can be changed easily. By default, **`nbrtimer`** is initialized to 8.

To configure the High Precision Timer, uncomment the following line in the file **`$ENV_PREFIX/sys/lynx.os/CONFIG.TBL`**:

```
        I:timer.cfg
```

A new kernel must be rebuilt and installed by the following commands:

```
        LynxOS# cd $ENV_PREFIX/sys/lynx.os
        LynxOS# make install
```

Then reboot the system if you are in a native development environment:

```
        LynxOS# reboot -Na
```

See the online man page **timerdrvr** for a detailed description of the High Precision Timer.

To implement the High Precision Timer in a microkernel, configure the files mentioned above which are present in the directory **`$ENV_PREFIX/sys/galaxy`**.

To implement the High Precision Timer on the slave CPU side of a VMPCx-dual board, configure the files mentioned above which are present in the directory **`$ENV_PREFIX/sys/lynx.os.s`**.
The High Precision Timer can be implemented on both CPUs of a VMPCx-Dual, simultaneously.

An example of HPT code is given on the CD-ROM (file **`./exampl/hpt.c`**).

13. Promable Kernel and Diskless Service

13.1 Building a promable kernel

To generate a LynxOS promable kernel, execute the following steps:

1. Generate a **`lynx.os`** kernel using the **`ramdisk`** driver: the **`ramdisk.cfg`** line must be enabled in the **`/sys/lynx.os/CONFIG.TBL`** file (default case).

2. Generate a "ROMable" **`romfile`** file by using **`mkimage`**. For this, see the **mkimage** and **mkimage.spec.cetia** manual pages.
   The **`romfile`** file can be saved in a PROM. Copy the PROM contents to the local DRAM at address 0x4000 and execute:
   ```
           go 4020
   ```
   Copying at 0x4000 is the first step that the promable kernel automatically does if the **`romfile`** file is not at this address.

   It is possible to boot the **`romfile`** file directly from disk. Enter the firmware command:
   ```
           bop 0,c
   ```
   to boot the kernel loader **`preboot`**, then
   ```
           b s0b romfile
   ```
   to boot the **`romfile`**

13.2 Building a downloadable file

1. Generate a **`romfile`** promable kernel with the **`mkimage`** utility.

2. Run the command:
   ```
           bash# mkbootfile ebo_file romfile
   ```
   The output file ***`ebo_file`*** must be downloaded onto the target by the **`ebo`** command.

   You may also run the command:
   ```
           bash# mkbootprep /lynxos.boot romfile ebo_file
   ```
   to generate a PReP output file.
   The output file ***`ebo_file`*** must be downloaded onto the target by the **`ebop`** command.

Note that:
`ebo_file` is also a bootable tape file. It can be written onto a tape by executing the command:
```
        bash# cp ebo_file /dev/rtape
```
and booted by the firmware command:
```
        COMMAND> bo 20,c
```
or, if the file is a PReP boot file:
```
        COMMAND> bop 20,c
```

13.3 Configuring a diskless server

1. Run **`tftp`**:
   a. Enable the corresponding line in the **`/etc/inetd.conf`** file.
   b. Reboot the system, to take into account the **`/etc/inetd.conf`** modification.

2. Run **`bootd`**:

   ```
   bash#  /usr/local/bin/bootd &
   ```

   To insert the start-up in the **`/net/rc.network`** file, add the following command:

   ```
   nohup /usr/local/bin/bootd > /dev/null
   ```

3. Edit **`/etc/bootdb`** using ***`ebo_file`*** as the file to be transferred:

   ```
   DEFAULT
       internet = usehosts
       server = server_name
       rootfs = /tmp
       default = ebo_file

   target_name
       ether = 00.00.de.30.5a.a5
       rootfs =  /tmp
       default =  ebo_file
   ```

   Replace:
   - ***`server_name`*** by the machine name where **`bootd`** runs,
   - ***`target_name`*** by the name of the target machine,
   - ***`00.00.de.30.5a.a5`*** by the Ethernet number of the target CPU,
   - ***`/tmp`*** by the ***`ebo_file`*** access path,
   - ***`ebo_file`*** by the name of the output file given with the **`mkbootfile`** command (or **`mkbootprep`** command for a PReP file).

4. On the target, execute the firmware command:

   ```
   COMMAND> ebo
   ```

   if you used the **`mkbootfile`** utility or

   ```
   COMMAND> ebop
   ```

   if you used the **`mkbootprep`** utility.

   If your server is an AIX system, depending on the AIX version, to boot by Ethernet instead of the **`ebo`** command you have to enter:

   ```
   ebo ,<target_internet_address>
   ```

14. User Flash Support

14.1 Overview

User Flash is present on VMPC4b, VMPC5 or VMPC6 boards.

User Flash is used to save user data and to keep them when power is down. Basically, the User Flash is used in standalone system configurations and usually contains a kernel and a Ram file system.

Complete information about how to use the Flash is given below in the paragraph **"Configuring the Flash"**. See also the online man page by typing the command:

```
LynxOS# man pwbflash
```

14.2 Installation

To install the User Flash driver, you must validate the following line in the file **/sys/lynx.os/CONFIG.TBL**:

```
I:pwbflash.cfg
```

The User Flash can be cut into several parts or RAM disks. The layout of the Flash has to be described in the file **/sys/devices/pwbflashinfo.c**. By default, two 2Mbyte RAM disks are defined, but more RAM disks with different sizes can be defined. The table **pwbflashinfo[]** defines the size (in 512-byte blocks) of each RAM disk. To validate a new configuration, change and compile the file **/sys/devices/pwbflashinfo.c**:

```
LynxOS# cd /sys/devices
LynxOS# make all
```

Each RAM disk is represented by a raw device **/dev/rpwbflashx** and a block device **/dev/pwbflashx** where x is the minor. By default, two nodes (two devices) are defined. Update the file **/sys/cfg/pwbflash.cfg** to add more nodes to be compatible with the layout defined in the configuration file **/sys/devices/pwbflashinfo.c**.

A global variable **romable_isinflash** is defined in the configuration file **/sys/devices/pwbflashinfo.c**. This variable must be set to 1 for a romable kernel which will be downloaded into the User Flash but must set to 0 for a romable kernel not resident in User Flash (network boot).

Then, rebuild a new kernel by typing the following commands:

```
LynxOS# cd /sys/lynx.os
LynxOS# touch CONFIG.TBL
LynxOS# make install
```

Reboot the system by the command:

```
LynxOS# reboot -N
```

To get more information about the Flash driver, see the online man page.

CAUTION The User Flash is write enabled by a hardware link (LK3) (refer to the hardware documentation associated with your VMPC board for more information about the LK3 link setting). To manage the User Flash under LynxOS so each time the Flash driver is installed by the kernel, the LK3 must:

- not be set for VMPC6 PCB E or PCB F,
- be set for VMPC6 PCB G and upper.

This link is also required for booting from User Flash if the VMPCBug firmware is version 1.8 or lower.

14.3 Utility programs

Three utility programs are provided in the directory **`/usr/local/bin`** to manage the User Flash:

- The **`erase`** utility erases by default the first two Flash devices. Type the command:

  ```
  LynxOS# erase -help
  ```

  to get the command usage.

- The **`to_flash`** utility copies a file into a Flash device. Type the command:

  ```
  LynxOS# to_flash -help
  ```

  to get the command usage.

- The **`ram_to_flash`** utility updates the file system contained in the first Flash device **`/dev/rpwbflash0`**. This command must be used only with the Flash functional mode 1 (see below).

14.4 Configuring the Flash

Two functional modes are provided:

- The first mode (called mode 1) corresponds to the user who needs to constantly adapt the system to the environment. So, the file system must be writable. In this case, the User Flash will contain a kernel with a RAM file system which is copied into the system memory at run time. A ROM file system can be downloaded into another Flash device and be mounted after booting.
- In the second mode (called mode 2), the environment is well known and constant. In this case, the User Flash will contain a kernel and a ROM file system which is not copied into the system memory at run time. A larger space is available in the system memory for the applications. The kernel and the ROM file system are on two different Flash devices.

Flash Block locking/unlocking is not yet supported by the flash driver for VMPC4b, VMPC5 or VMPC6 platforms.

14.4.1 Mode 1

as described above and define the wanted Flash layout in the **/sys/devices/pwbflashinfo.c** file. Then, generate a new kernel and reboot the system.

To create the kernel romable image, create a **mkimage** specification file as described in the **mkimage.spec.cetia** man page and run the **mkimage** utility.
Then, run the **mkbootfile** CETIA utility to generate the bootable image.
Then, copy the bootable image into the first User Flash logical device by using the **to_flash** utility. To do it quicker, erase the User Flash first.

```
# mkimage -v ppc.spec romfile
# mkbootfile lynx.img romfile
# erase /dev/rpwbflash0
# to_flash lynx.img /dev/rpwbflash0
```

Then, reboot the firmware.

To boot LynxOS from the User Flash, type the following command under firmware:

```
COMMAND> bo 0 5
```

Assuming the command has been added to the RAM file system, if you want to update the file system in User Flash after changing or adding a file at run time, type the following command under LynxOS:

```
# ram_to_flash
```

The **ram_to_flash** command only manages **/dev/rpwbflash0**.

To install a ROM file system into another User Flash device, use the following. This is an example.

```
# mkramdisk -n /dev/rd1 -r /dev/rrd1 4096
to create a 2M byte ramdisk

# mkfs -v /dev/rd1
to create the file system on /dev/rd1

# mount /dev/rd1 /mnt

# cp /init /mnt
copy useful files into the file system

# umount /mnt

# cp /dev/rd1 /tmp/romdisk
to create a ROM disk file

# erase /dev/rpwbflash1
to erase the Flash device 1

# to_flash /tmp/romdisk /dev/rpwbflash1
to copy the ROM disk file into the Flash device 1.
```

This file system can be mounted at run time but cannot be changed.

14.4.2 Mode 2

To use the User Flash in mode 2, install the User Flash driver as described above and define the wanted Flash layout in the **/sys/devices/pwbflashinfo.c** file. Then, generate a new kernel and reboot the system.

At least two RAM disks must be defined: the first one will contain the romable kernel, the second one will contain the ROM file system (used as **/**).

Add **VMPCOPT_UFLASH** flag in the file **/sys/lynx.os/conf** (**vmpc_option** data).

For this specification file, the file system is empty to gain space in the system memory. Run the **mkimage** utility and then the **mkbootfile** CETIA utility to create the bootable image. Then, copy the image into the User Flash by using the **to_flash** utility. To do it quicker, erase the User Flash first, both **/dev/rpwbflash0** and **/dev/rpwbflash1** devices.

```
# mkimage -v ppc.spec romfile
# mkbootfile lynx.img romfile
# erase /dev/rpwbflash0
# to_flash lynx.img /dev/rpwbflash0
```

The Flash device **/dev/rpwbflash1** must contain a file system. To do this, use the **mkramdisk** utility. The **mkfs** utility cannot be used directly on **/dev/pwbflash1** because Flash block devices are not writable. The size of the RAM disk must be lower or equal to the Flash device. Then, copy the RAM disk into the Flash by using the CETIA **to_flash** utility.

```
# mkramdisk -n /dev/rd1 -r /dev/rrd1 4096
to create a 2M byte ramdisk

# mkfs -v /dev/rd1
to create the file system on /dev/rd1

# mount /dev/rd1 /mnt

# cp /init /mnt
copy files into the file system

# mkdir -p /mnt/dev
# cd /mnt/dev
# mknod -a /etc/nodetab
to create nodes (/etc/nodetab is associated with the romable kernel)

# cd /
# umount /mnt

# cp /dev/rd1 /tmp/romdisk
to create a ROM disk file

# to_flash /tmp/romdisk /dev/rpwbflash1
to copy the ROM disk file into the Flash device
```

Then, reboot the machine.

To boot LynxOS from the User Flash, type the following command under firmware:

```
        COMMAND> bo 0 5
```

The **`ram_to_flash`** utility must not be used in mode 2 because in this case the root file system is not in RAM.

15. Graphics Support

CAUTION The graphics support for LynxOS 3.0.1 on CETIA VMPC platforms with CPMC-GTX-8 graphics PMC is described in the **Release Notes DR-GTX for LynxOS 3.0.1 on VMPC Family" (SD.DT.C09).**

This chapter describes the graphics support for LynxOS 3.0.1 on CETIA VMPC platforms with CPCIGx graphics PMC.

15.1 Overview

The graphics package included in this release mainly contains:

- the CETIA mouse driver,
- the CETIA X server for CPCIGx graphics board which supports:
  - ♦ multiple screens,
  - ♦ depths of 15, 16 and 24 planes in TrueColor,
  - ♦ screen sizes of 640x480, 1024x768 and 1280x1024.

  The CETIA X server and its configuration files are distributed in the tar file **`/Xcetia.tar`**.

New install scripts **`/usr/bin/Install.XM`** and **`/usr/bin/Uninstall.XM`** are also installed when the graphics package is downloaded.
These can be used in a cross development environment as well as in a native environment.

The complete X11 and Motif releases are not included in this distribution and must be extracted from the **LynxOS 3.0.1 for VMPC6 or VxG4a Family** CD-ROM on VMPC6 platforms or **LynxOS 3.0.1 - CT3** CD-ROM on the other VMPC platforms.

15.2 Before Installing XM

In order to be able to start X immediately after installation is finished, before installing XM configure the **cetiamouse** driver in the **`/sys/lynx.os/CONFIG.TBL`** file. To do that, uncomment the line:

```
        I:cetiamouse.cfg
```

The AT console must be also selected before TTY. In the **`CONFIG.TBL`** file, the line **`I:atc.cfg`** must be before **`I:com.cfg`**.

The minimum disk size required to install the entire X11, Motif and PosixWorks Desk distributions is about 260 MB.

15.3 Installing XM on a LynxOS PowerPC Target

It is assumed that you are using the bash shell. Check your SHELL environment variable.

1. Power up the machine and boot the LynxOS 3.0.1.
2. Insert the **LynxOS 3.0.1 for VMPC6 or VxG4a Family** CD-ROM on VMPC6 platforms or **LynxOS 3.0.1 - CT3** CD-ROM on the other VMPC platforms, corresponding to your installed LynxOS release, into the CD-ROM drive (SCSI ID =6).
3. Create a directory such as **`/cdrom`** on which to mount the CD-ROM if it does not already exist:

```
# mkdir /cdrom
```

4. Mount the CD-ROM on **`/cdrom`**:

```
# mount -o ro /dev/sd6 /cdrom
```

5. Create a directory such as **`/X11`** where the X11 and Motif distributions will be installed:

```
# mkdir /X11
```

6. Launch the install script:

```
# $SHELL /usr/bin/Install.XM /cdrom
```

   The following steps show you the messages displayed when the **`/usr/bin/Install.XM`** script is running.

7. Select option 1 **Modify Default Values** in the main menu.

   Configure to install X and Motif from the CD-ROM mounted on the directory **`/cdrom`** to the directory **`/X11`**:

```
          *** Hardware Configuration Menu ***

          1) Install Directory       : /

          Installation from          : /cdrom

          Q)uit Installation
          R)eturn to Previous Menu

     Enter Option: [R] 1
```

NOTE X and Motif are installed in the same directory.

```
     Product installation directory (absolute path) : /X11

          *** Hardware Configuration Menu ***

          1) Install Directory       : /X11

          Installation from          : /cdrom

          Q)uit Installation
          R)eturn to Previous Menu

     Enter Option: [R]
```

8. Return to the main menu.

9. Select option 2 **Choose Products to Install**.

   Configure to install X and Motif if needed. The default is:

   ```
                *** Hardware Configuration Menu ***


                1) Install X (y/n)         : y
                2) Install Motif (y/n)     : y
                3) PosixWorks Desk         : y
                        Selecting (y) will cause the PosixWorks Desk
                        environment to be run by default.  Selecting (n)
                        will result in a basic mwm or twm environment
                        to be run instead.

                4) Advanced Installation options

                Q)uit Installation
                R)eturn to Previous Menu

           Enter Option: [R]
   ```

   Just enter the corresponding number to toggle the option.

10. By default, the kernel will be automatically rebuilt at the end of the installation. This is needed to configure the CETIA mouse driver, if it is not already done. Also, some environment files such as **`/sys/lynx.os/uparam.h`** will be updated during installation.

    If you really don't want to rebuild the kernel at the end of the installation, select option 4 **Advanced Installation options** in the Software Configuration Menu and toggle the option.

11. Return to the main menu.

12. Select option B **Begin Installation** to start the installation.

13. At the end of the installation, remake the nodes:

    ```
           # reboot -aN
    ```

15.4 Installing XM in a Cross Development Environment

Installation is basically the same as in the native environment. So please, refer to the Section above for more details about the installation steps.

The **`$ENV_PREFIX`** environment variable must be set to the location of a cross development environment.

The directory ***`<install-X11>`*** where the software is to be installed can be created anywhere but is usually created under **`$ENV_PREFIX`** to be easily used on a native LynxOS system over NFS (diskless system).

If it is not under **`$ENV_PREFIX`**, the links made in **`$ENV_PREFIX`** for use during cross developments (such as from **`$ENV_PREFIX/usr/bin/X11`** to ***`<install-X11>`*****`/usr/bin/X11`**) will not exist on it.

One solution is to create a directory path ***`<install-X11>`*** on the native system and to NFS mount the remote X11 install directory on this one.

It is assumed that you are using a Korn Shell compatible shell. Check your SHELL environment variable.

1. Mount the **LynxOS 3.0.1 for VMPC6 or VxG4a Family** CD-ROM on VMPC6 platforms or **LynxOS 3 .0.1 - CT3** CD-ROM on the other VMPC platforms by using the `mount` command appropriate for the host system (on `/cdrom` for example).
2. Launch the install script:

```
        # $ENV_PREFIX/usr/bin/Install /cdrom
```

The rest of this document assumes a native LynxOS 3.0.1 PowerPC installation.

15.5 Uninstallation

To uninstall X and Motif, type the following command:

- In a native environment (assuming you are using the bash shell):

```
        # $SHELL /usr/bin/Uninstall.XM
```

- In a cross development environment (assuming you are using a Korn shell compatible shell):

```
        # $SHELL $ENV_PREFIX/usr/bin/Uninstall.XM
```

15.6 Starting X11

15.6.1 General Information

You must have plugged in:

- one (or more) CPCIGx graphics PMC,
- an USB Logitech mouse (on the VMPC6 platforms) or a PS/2 IBM mouse (on the other VMPC platforms),
- an USB CHERRY keyboard (on the VMPC6 platforms) or a QWERTY PS/2 IBM keyboard (on the other VMPC platforms),
- a multi-standard monitor.

If PosixWorks Desk is installed, then **xdm** is automatically launched when the system is running in multi-user mode. This was configured in the `/etc/ttys` during the desktop installation by selecting the command file `/bin/runx` for **/dev/atc0** instead of **/bin/login**.

If the desktop is not installed, start X from the console by giving the command:

```
        # xinit >errors 2>&1
```

Looking inside the file later, you should find something like the following, depending on your system:

```
XFree86 Version 3.1.2 / X Window System
protocol Version 11, revision 0, vendor release 6001)
Operating System: LYNXOS PPC CETIA
Configured drivers:
(using VT number 3)

XF86Config: /usr/lib/X11/XF86Config
(**) stands for supplied, (--) stands for probed/default values
(**) Mouse: type: PS/2, device: /dev/mouse, baudrate: 1200
(**) Xcetia: Graphics device ID: "CPCIGX"
```

```
(**) Xcetia: Monitor ID: "Philips"
(**) Xcetia: Graphics device ID: "CPCIGX"
(**) Xcetia: Monitor ID: "Philips"
(**) Xcetia: Graphics device ID: "CPCIGX"
(**) Xcetia: Monitor ID: "Philips"
(**) Xcetia: Graphics device ID: "CPCIGX"
(**) Xcetia: Monitor ID: "IBM"
(**) FontPath set to "/usr/lib/X11/fonts/misc,/usr/lib/X11/fonts/100dpi,
     /usr/lib/X11/fonts/Speedo,/usr/lib/X11/fonts/Type1"

(--) Xcetia: MemBase 0xc1000000 PCI Device 1 Busnumber 1
(--) Xcetia: Vendor/chipset is CETIA (CPCIGX)
(**) Xcetia: (mem: 2000k numclocks: 0 vendor: CETIA membase: 0xc1000000)
(--) Xcetia: Maximum allowed dot-clock: 135.000 MHz
(**) Xcetia: Mode "1280x1024": mode clock = 107.300
(--) Xcetia: Virtual resolution set to 1280x1024
(--) Xcetia: Using hardware cursor
(--) Xcetia: Using 8 bits per RGB value

(--) Xcetia: screen 1 ignored

(--) Xcetia: screen 2 ignored

(--) Xcetia: screen 3 ignored
(**) Xcetia: MemSize: 0x200000 MemConfig: 0xc810003d srtctl: 0x1e3
```

If the server does not start with similar looking messages, try looking up the message you get in the section below "Exceptional Situations".

You can kill the server by typing the abort sequence on the keyboard. The default sequence is **control-alt-backspace**. Press the keys in the right order and keep them held down.

15.6.2 Multi Screens

To start X on more than one screen you must use the `-rawscreen` option described in the **Xcetia** man page. For example, to use a second screen, type the command:

```
# xinit -- -rawscreen 1
```

Use the option `-edges` described in the **Xcetia** man page to connect multiple screens together. If you run X as above, you will get an horizontal layout of the 2 screens (default is `-edges ’0E1’`). But if you run X with:

```
# xinit -- -edges ’0S1’ -rawscreen 1
```

you will get a vertical layout for screens 0 and 1.

15.6.3 Depth and Screen Size

This information is given in the file `/usr/lib/X11/XF86Config` in the section **Screen**.
There is one section per screen.
The default values are:

```
        Depth   8
        Modes   "1280x1024"
```

Where:

- **`Depth`** is the pixel size,
- **`Modes`** is the screen size.

Other values are commented out in the configuration file and can be used.
The X server will refuse to run with size combinations which surpass the video ram dimensions.

15.7 Useful Tips

- You can get information on the CETIA X server and graphics boards by starting with

  ```
  $ man Xcetia
  ```

  The `xman` command can also be used to browse the man pages.
- By setting the resource

  ```
  *customization:-color
  ```

  in your `.Xdefaults`, many applications will draw in colour instead of black and white.
- The keyboard and mouse must be plugged in before starting the server for them to be recognised. If they are unplugged and replugged when the server is running they may no longer work correctly.
- The mouse and keyboard must not be manipulated while the server is starting.

15.8 Exceptional Situations

Following is a list of possible interpretations for various messages marking exceptions.

- **When starting the server:**
  - **`Binding TCP socket: Address already in use`**
    `Cannot establish tcp listening socket`
    The server is trying to use the same network connection as another server. Give a `:`***`number`*** argument when running a second server or make sure that the first is killed.
  - **`TRANS(SocketCreateListener): failed to bind listener`**
    `TRANS(SocketINETCreateListener): TRANS(SocketCreateListener) () failed`
    `TRANS(MakeAllCOTSServerListeners) failed to create listener for tcp`
    TCP/IP is not configured. This is not mandatory for X11 on the CVME or VMPC platforms, and the R6 server will continue to run.

15.8.1 Fixed Bugs

Fixed bugs are relative to the CETIA X11R6 release **PWL-OP-X11 for LynxOS 2.5.0 on CVME and VMPC ID 97287**.

- CRP1911:
  Plane mask not respected when filling background colour.

15.8.2 Known Problems

- The `sync_on_green` option is not supported by the CPCIGx hardware.
- Some of the extensions present in the server are not supported though may work. They are: **SYNC, XC-MISC, MIT-SHM, XIE, BIG-REQUESTS, XTEST, XTestExtension1**.
- The **MIT-SCREEN-SAVER** extension can only be used once per run of the server. If the client using it dies, it cannot be used again.
- `xmkmf` and `imake` cannot be used on cross development platforms. If necessary, Makefiles should be built on native LynxOS platforms but compiling can then be done on the cross platform.
- The X11 bell cannot be run on VMPC platforms (no audio on these platforms).

16. PMC Support

16.1 CPMC-ETH100T Support

16.1.1 Overview

This chapter describes the software installation of the device driver and network interface of the CPMC-ETH100T PMC.

The documentation associated with the PMC is **CPMC-ETH100T Board - Fast Ethernet Adapter, reference: CA.DT.123**. Please, refer to this document to install the CPMC-ETH100T correctly.

A category 5 cable must be always used to connect the CPMC-ETH100T for 100 Mbit/s use.

The **cpmc-eth100t** device driver supports the IEEE802.3 data link.

It can be configured for 10Mbit/s or 100Mbit/s data rates.

It operates only through sockets in the `AF_INET` or `AF_RAWETH` domain.

Details about the driver itself are available in the associated **cpmc-eth100t(4)** on-line manual page.

16.1.2 Network Configuration

TCP/IP service must be installed.

1. Configure the data rate of each Ethernet interface by editing the file **`/sys/devices/cpmc-eth100tinfo.c`**:

```
struct eth100tinfo eth100t0_info = {
                    2,     /* 1 = 10Mb/s ; 2 = 100 Mb/s */
                    0,     /* automatic ethernet address search */
                    { 0x00, 0x00, 0x00, 0x00, 0x00 },
                    0,     /* must be 0 */
                    1      /* 1: enable ICPMC support, 0 : disable */
};
```

   The following values should be chosen:
   1 for a **10Mbit/s** data rate,
   2 for a **100Mbit/s** data rate (default value).

2. Recompile the file:

```
LynxOS# cd /sys/devices
LynxOS# make all
```

3. Configure the device driver and the network interface:
   run the script file:

   ```
   /usr/bin/Install.eth100t
   ```

   Then, reboot the system by typing the command:

   ```
   LynxOS# reboot -aN
   ```

   The nodes will be automatically redone.

   To remove the network interface, run the script file:

   ```
   /usr/bin/Uninstall.eth100t
   ```

   Then, reboot the system by typing the command:

   ```
   LynxOS# reboot -aN
   ```

The two next chapters explain how to configure manually the cpmc-eth100t network interface but it is automatically done when you run the script

```
/usr/bin/Install.eth100t
```

16.1.2.1 Host Database

An IP number/hostname couple must be defined in the **`/etc/hosts`** file to define a new network address family.

Example:

```
192.10.11.100     my_name            /* for default amd or dec interface */

192.20.21.1       my_other_name      /* for the cpmc-eth100t interface */
```

16.1.2.2 Network Interface

The cpmc-eth100t network interface must be configured by typing the command:

```
LynxOS# ifconfig cpmc-etha0 my_other_name
```

where ***`my_other_name`*** is the new network address family (see section "Host Database").

This command can be directly included in the **`/net/rc.network`** configuration file.

NOTE

Up to four CPMC-ETH100Ts can be simultaneously used. The network interface names associated with the four possible CPMC-ETH100Ts are **cpmc-etha0, cpmc-ethb0, cpme-ethc0** and **cpmc-ethd0**.

16.1.3 Additional information

- A warning message is printed by **`ifconfig`** to the system console if there is no carrier:

  ```
  cpcm-eth100t: No carrier - transceiver cable problem ?
  ```

16.2 CPMC-ETH-xx Support

16.2.1 Overview

This chapter describes the software installation of the device driver and network interface of the CPMC-ETH-xx PMC.

The documentation associated with the PMC is **CPMC-ETH Board - Fast Ethernet Adapter, reference: CA.DT.131**. Please, refer to this document to install the CPMC-ETH-xx correctly.

The **cpmc-ethx** device driver supports the IEEE802.3 data link.

It can be configured for 10Mbit/s or 100Mbit/s data rates.

It operates only through sockets in the `AF_INET` or `AF_RAWETH` domain.

Details about the driver itself are available in the associated **cpmc-eth(4)** on-line manual page.

16.2.2 Network Configuration

TCP/IP service must be installed.

CPMC-ETH-xx default configuration mode is **AUTOSENSE**.

1. If you want to remove the **AUTOSENSE** mode, edit the file **`/sys/drivers.vmpc/cpmc-eth/cpmc-eth-dc21143.c`** and comment the line:

   ```
   #define AUTOSENSE.
   ```

   Then in the **`/sys/drivers.vmpc/cpmc-eth`** directory, run:

   ```
   make all.
   ```

2. Manually, configure the data rate of each Ethernet interface by editing the file **`/sys/devices/cpmc-ethinfo.c`**:

   ```
   struct dcinfo dc1_infoa = {
                       1,     /* 1 = 10Mb/s ; 2 = 100 Mb/s */
                       0,     /* automatic ethernet address search */
                       { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
                       1      /* pci bus */
   };
   ```

   The following values should be chosen:
   1 for a **10Mbit/s** data rate,
   2 for a **100Mbit/s** data rate (default value).

3. Recompile the file:

   ```
   LynxOS# cd /sys/devices
   LynxOS# make all
   ```

4. Configure the device driver and the network interface:

WARNING There is a bug into the **`Install.cpmc-eth`** script.
The **`rc.network`** file is not updated because the **`Install.cpmc-eth`** script checks for the line about the amd0 interface which does not exist anymore. So the line to start the cpmc-ethxa0 interface is not added.
Change the string **amd0** to **dec0** in the file ***`$ENV_PREFIX/`*****`usr/bin/Install.cpmc-eth`** before running it.

Run the script file:

```
/usr/bin/Install.cpmc-eth
```

Then, reboot the system by typing the command:

```
LynxOS# reboot -aN
```

The nodes will be automatically redone.

To remove the network interface, run the script file:

```
/usr/bin/Uninstall.cpmc-eth
```

Then, reboot the system by typing the command:

```
LynxOS# reboot -aN
```

The two next chapters explain how to configure manually the cpmc-eth network interface but it is automatically done when you run the script

```
/usr/bin/Install.cpmc-eth
```

16.2.2.1 Host Database

An IP number/hostname couple must be defined in the **`/etc/hosts`** file to define a new network address family.

Example:

```
192.10.11.100     my_name          /* for default amd or dec interface */

192.20.21.1       my_other_name    /* for the cpmc-eth interface */
```

16.2.2.2 Network Interface

The cpmc-eth network interface must be configured by typing the command:

```
LynxOS# ifconfig cpmc-ethxa0 my_other_name
```

where ***`my_other_name`*** is the new network address family (see section "Host Database").

This command can be directly included in the **`/net/rc.network`** configuration file.

NOTE Up to four CPMC-ETH-xxs can be simultaneously used. The network interface names associated with the four possible CPMC-ETH-xxs are **cpmc-ethxa0, cpmc-ethxb0, cpme-ethxc0** and **cpmc-ethxd0**.

16.3 CPMC-xASYNC Support

16.3.1 Overview

This Chapter describes the software installation of the device driver for the CPMC-xASYNC PMC.

The CPMC-xASYNC PMCs provide 4, 8 or 16 16550 UART based serial ports. This product is a low cost option extending the number of serial ports available with CETIA CPU boards.

Details about the driver itself are available in the associated **dr-Xasync(4)** on-line manual page.

The following PMCs are supported:

- CPMC-4ASYNC: 4 RS232 Asynchronous lines,
- CPMC-4ASYNC422: 4 RS422 Asynchronous lines,
- CPMC-8ASYNC232: 8 RS232 Asynchronous lines,
- CPMC-8ASYNC422: 8 RS232 Asynchronous lines,
- CPMC-16ASYNC232: 16 RS232 Asynchronous lines,
- CPMC-16ASYNC422: 16 RS422 Asynchronous lines,
- CPMC-4+4ASYNC: 4 RS232 Asynchronous lines and 4 RS422 Asynchronous lines.

The PMC board technical documents are:

- Technobox Inc. - 4-Port RS232/RS422/RS485 Asynchronous - Communication Adapter - PMC Product Manual
- Technobox Inc. - 8-Port RS232/RS422/RS485 Asynchronous - Communication Adapter - PMC Product Manual
- Technobox Inc. - 16-Port RS232/RS422/RS485 Asynchronous - Communication Adapter - PMC Product Manual

16.3.2 Configuration

■ To configure the device driver, run the script file:

```
LynxOS# /usr/bin/Install.Xasync
```

Then, reboot the system by typing the command:

```
LynxOS# reboot -aN
```

■ To remove the device driver, run the script file:

```
LynxOS# /usr/bin/Uninstall.Xasync
```

Then, reboot the system by typing the command:

```
LynxOS# reboot -aN
```

The device driver is associated with the configuration file **`/sys/devices/cpmc-Xasyncinfo.c`**. This file may be modified to configure the serial ports as wanted.

16.3.3 Additional information

- Terminal login: Terminal logins may be obtained on CPMC-xASYNC serial ports by adding in the file **`/etc/ttys`** one line for each serial port:

  ```
  DEVICE:1:default96:wyse50:/bin/login
  ```

  Where ***`DEVICE`*** is the device name of the serial port **`/dev/p`*`X`*`a`*`Y`*`com`*`Z`*** with:
  X: 4, 8 or 16
  Y: 0 or 1 (depending on the setting of the PMC on the PCI bus).
  Z: port number (from 1 to 4 for CPMC-4ASYNC, from 1 to 8 for CPMC-8ASYNC, from 1 to 16 for CPMC-16ASYNC).

- Outputs on front panel or P2 connector: Refer to the Technobox documentation associated with your CPMC-xASYNC.

  CAUTION For VMPC platforms, you have to order the PMC I/Os manufacturing option.

- RS232/RS422/RS485 port configuration: Refer to the Technobox documentation associated with your CPMC-xASYNC.

- When the system is booted, type the command:

  ```
  LynxOS# devices
  ```

  to check if device nodes (serial lines) are correctly installed (**`/dev/p4a?com*, /dev/p8a?com*, /dev/p16a?com*`**).

17. AltiVec Support

This release includes the following altivec support:

OS Kernel support: The kernel allow multi thread user application to run ALTIVEC code. The ALTIVEC default mode is set to NON JAVA Mode when a new thread is started. Also a **SIG_FPE** signal is sent to the thread that executes a denormalizated code in JAVA MODE.

Assembler The assembler is modified to support ALTIVEC instruction. Its name is **`/bin/as.vec`**. If you want "**as.vec**" to become the assembler used by your compiler, replace the "**as**" version of you compiler. (usually in **`/usr/ppc-xcoff-lynxos/bin`** directory).

Debugger The GDB debugger is modified to support ALTIVEC instruction. It disassembles ALTIVEC code and print ALTIVEC register with the "**`info vec-registers`**" command. Its name is **`/bin/gdb.vec`**.

WARNING Alignment Issue: the load and store ALTIVEC instructions align the address to 16-byte boundary. Your code MUST used 16-byte boundary aligned addresses.

WARNING VSIPL Library: the **PWL4-CROSS-G4** or **PWL4-COMP-G4** package must be used to compile and link a program with the VSIPL library.

Kernel implementation remarks:

- AltiVec unit must not be used into a signal handler because the AltiVec context is not saved and restored.
- **`longjmp()`** and **`setjmp()`** do not save and restore the AltiVec context, so do not use AltiVec between these two routines.
- The AltiVec registers are initialized when a new thread or process is created. So **`fork()`** initializes the AltiVec registers but does not copy it and **`exec()`** does not initialize it.

A documentation of the PowerPC G4 and the VSIPL library is on the CD-ROM (**`/doc`**).

18. Troubleshooting

To send a bug to CETIA, complete the software problem report which is at the end of this document or in **`/usr/cetia_bugreport`**.

A general **Known Problems** chapter for the LynxOS 3.0.1 release is available in the Lynx documentation **Release Notes**.

- Micro kernel generation:
  A micro kernel application must contain the following line to be linked:

  ```
  int os_load_address =0;
  ```

- On the VMPC platforms, the User Flash is write enabled by a hardware link. This LK3 link (on board between the User Flash device and the front panel) must be set in to manage the User Flash under LynxOS each time the Flash driver is installed by the kernel (refer to the CPU board User’s Guide for more information about the LK3 link setting).

  This link is also required for booting from User Flash if the VMPCBug firmware is version 1.7 or lower.

- Flash Block locking/unlocking is not supported by the flash driver for VMPC platforms.

- CRP2042:
  The CPMC-4ASYNC does not work on the left side slot on an ICPMC carrier board (ICPMC facing you).

19. Fixed Bugs

- CRP1954:
  Problem when using galaxy (choice of kernel modules).

- CRP1976:
  Problem to configure a cluster with dual boards under LynxOS 3.0.1.

- CRP1988:
  Address conflict between User Flash and some PMCs.

- CRP2011:
  Activating the VME driver in **`CONFIG.TBL`** stops the system boot.

- CRP2013:
  High resolution timers loose their accuracy on busy system.

- CRP2025:
  `BLCK_CAPACITY` (rpwbflash2) returns the rpwbflash0's one.

- CRP2028:
  AltiVec instructions are not supported under LynxOS 3.0.1

- CRP2029:
  New Ethernet not supported.

- CRP2030:
  The USB mouse events are not correctly managed.

- CRP2063:
  `ping -f`, under SCMP cluster, has some "duplicate packets".

- CRP2073:
  `drm_register_isr()` cannot attach an IRQ for PMC Dolphin.

- CRP2074:
  Bad management of external interrupts stacking.

- CRP2083:
  The slave CPU of a dual board must be booted BEFORE the master made any SCMP activity.

20. Additional Information

- **Board without RTC battery:**
  The VMPC Boards without RTC battery have the following restrictions:
  - the `date` command returns randow value,
  - `gettimeofday()` precision is only 10 ms,
  - the High Precision Timer precision is reduced to 10 ms.

- The on-line man pages are available for most CETIA additions as follows:
  - `pwbflash` is a man page that describes the user flash driver for VMPC4b, VMPC5a and VMPC5a-Dual.
  - `pcidevs` is a man page that shows the PCI devices.
  - `pciinfo` is a man page that manages the PCI devices.
  - `pcikit` is a man page that describes the PCI device driver kit and kernel service calls for PCI device drivers.
  - `atc_cpcigx` is a man page that describes the ATC (VT100 emulator) driver as implemented on the CETIA CPCIGx graphics board.
  - `ckprintf` is a kernel routine which sends messages to the circular buffer of messages.
  - `kvtop` is a kernel routine that can be called by device drivers to convert a given virtual address into a physical address, for example for use in DMA.
  - `map_iomen, unmap_iomen` are kernel routines which are called by the drivers to map or unmap the physical memory.
  - `prdebug` is a command which displays the content of the circular buffer of messages sent by the **ckprintf** routine.
  - `vmekit` is a kit of VME service calls which must be used during VME driver development.
  - `vmekit_vmpc` is a kit of VME service calls which must be used during VME driver development.
  - `almavmechan` is an utility which controls the PCI-VME bridge.
  - `mkbootfile` is used for the diskless generation.
  - `vdump,vlsdev,vload,vmeotty` are man pages relative to VMEOb boards.
  - `cpmc-eth100t` is a man page which describes the driver for the CPMC-ETH100T PMC.
  - `cpmc-eth` is a man page which describes the driver for the CPMC-ETH-xx PMC.
  - `dr-Xasync` is a man page which describes the driver for the CPMC-xASYNC PMC.
  - `timerdrvr` is a man page which describes the High Precision Timer.
  - `fastintr` is a man page which describes the SMI system.

- **sysppc()** extensions:
  - **sysppc(MAP_IOMEM, phys_addr, size, flags)** calls `map_iomem`.
  - **sysppc(UNMAP_IOMEM, log_addr)** calls `unmap_iomem`.
  - **sysppc(GET_VMEID)** returns the VME identification.
  - **sysppc(GET_CPU)** returns the CPU type (PPC603, PPC604, PPC750).
  - **sysppc(GET_BOARD)** returns the board identification.
  - **sysppc(GET_FREQ)** returns the CPU bus frequency.
  - **sysppc(GET_DRAM)** returns the system memory size.
  - **sysppc(GET_DRAMR)** returns the reserved memory size (boot option R).
  - **sysppc(GET_VMPC)** returns the type of VMPC board (VMPC_4A, VMPC_4B, VMPC_5A, ...).

- **ECC support:**
  ECC can be enabled by adding the `VMPCOPT_ECC` flag in the `/sys/lynx.os/conf.c` file (`vmpc_option` data). In addition, Lynx must be booted with the `;P` option after the power up:

```
# bop 0 c ;P
```

- **PCI Device Resource Manager support:**
  The PCI bus can be managed by the **pcibus** driver or by the new Device Resource Manager (DRM).

  The PCI bus is handled by:
  - the **pcibus** driver, if `pcibus.cfg` is included in the `CONFIG.TBL` file,
  - or by the Device Resource Manager, if it is not included in the `CONFIG.TBL` file.

- **BAT:**
  The PowerPC BAT Register can be used to improve performance when many MBytes are accessed. Contact CETIA to used it.

- **Fast Interrupt Response:**
  Refer to the **fastintr** man page.

Appendix A: Manual Pages

The CETIA manual pages are now only available online. The following section describes these manual pages.

almavmechan(1)	VME bus utility for VMPC board
mkbootfile(1)	make an IPL boot image for VMPC board

atc_cpcigx(4)	atc on CPCIGx
cpmc-eth(4)	CPMC-ETH-xx PMC
cpmc-eth100t(4)	CPMC-ETH100T PMC
dr-Xasync(4)	CPMC-xASYNC PMC
prdebug(4)	display circular buffer of kernel messages
pwbflash, erase, to_flash, ram_to_flash(4)	User Flash driver and utilities for VMPC4b
timerdrvr(4)	High Precision Timer
vdump(4)	display VMEOb memory
vload(4)	Load firmware on VMEOb board (with reset eventually)
vlsdev(4)	show VMEOb boards configuration
vmeotty(4)	tty driver for the VMEO board

mkimage.spec.cetia(5)	specification file for the mkimage utility

almavmekit, almavmeinfo.c, almapci_channel_alloc, almapci_channel_free, almavme_channel_alloc, almavme_channel_free alma_dma_start, alma_dma_wait, alma_ctrl, alma_dram_to_vme, alma_vme_to_dram(9)	VME service calls on the VMPC platform
ckprintf(9)	CETIA kernel printf
fastintr(9)	SMI management
kvtop(9)	convert virtual address to physical address
map_iomem, unmap_iomem(9)	map physical memory addresses
vmekit, vmeintcontrol, vmeinterrupt, wprobe, bprobe, lprobe(9)	VME service calls for VME device drivers on CETIA boards

Appendix B: Writing PMC Device Driver with ICPMC on VMPC

NOTE This appendix must not be used on ICPMC-6 Carrier Board.

A device driver for a PCI master on a PMC card plugged onto an ICPMC on a VMPC requires some specific initializations if the PMC is able to be PCI master. In order to prevent deadlock between the VMPC hostbridge and the ICPMC PCI-PCI bridge, the DRAM memory accessed by the PMC must not be in the PowerPC data cache.

There are many ways of preventing data from being in the PowerPC data cache:

a. The data are mapped with data cache inhibited.
   This solution is not good because performance may be severely degraded.

b. The data may be mapped with the data cache set in write-through mode.
   This solution is the best because the source code modification is minimal. Nevertheless it is not always possible to map data in write-through mode if the buffer read or written by the PMC is not allocated by the device driver (for instance a user buffer).

c. The data must be flushed from cache before the PMC reads or writes the data.
   Flushing the data from cache can be done by the device driver in each specific case when it is required. Once flushed, the data must not be accessed by the CPU until the driver completes its transfer.

For LynxOS , the two last workarounds are implemented as follows:

1. **Data cache in write through mode:**

   LynxOS does not support data cache write through mode. Therefore the device driver must create this new mode by using a PowerPC **BAT** (the PowerPC has 4 data block-address translation registers). The fourth **BAT** is free, so it can be used to map the whole memory in write through cache mode along the same model as **PHYSBASE** space that is in copy-back mode. This new space can be called **PHYSWBASE**, its value is 0x50000000. So, add the following lines in your device driver source code:

   ```
   #ifndef PHYSWBASE
   #define PHYSWBASE 0x50000000
   #endif
   ```

   and in the driver installation routine set up the **BAT** by calling this **setupbatphysw()** routine:

   ```
   setupbatphysw()
   {
   asm("sync");
   asm("mtspr 542, %0" : : "r"(0x50001ffe));
   asm("mtspr 543, %0" : : "r"(0x52));
   asm("sync");
   asm("isync");
   }
   ```

   The driver memory allocation need only be modified for memory that is read and written by the PMC:

   ♦ **sysbrk()** must be replaced by **alloc_cmem()** and **sysfree()** by **free_cmem()**,

- After **alloc_cmem()** or **get1page()**, the memory space must be changed from **PHYSBASE** to **PHYSWBASE**. For example:

```
char *p;

p = alloc_cmem(0x2000);
p = (char*) ((int)p + PHYSWBASE - PHYSBASE);    /* <- add this line */
```

  In addition, before calling **free_cmem() and free1page() the memory space must return to PHYSBASE**. For example:

```
p = (char*) ((int)p + PHYSBASE - PHYSWBASE);    /* <- add this line */
free_cmem(p,0x2000);
```

- Physical address conversion must be changed if the address is in the **PHYSWBASE** space:

```
        Physical_address = logical_address - PHYSWBASE
```

  For example:

```
char *p,*physical_addr;

p = alloc_cmem(0x2000);
p = (char*) ((int)p + PHYSWBASE - PHYSBASE);
physical_addr = (char*) ((int)p - PHYSWBASE); /*physical address conversion*/
```

  If your PMC only accesses **PHYSWBASE** space, the workaround is done. If not the second workaround (Data cache flush) must also be implemented.

2. **Data cache flush:**

When the PMC driver has to read or write directly into a buffer that it has not allocated itself, it is necessary to flush the buffer before the PMC accesses it. For example, for read and write driver entry, it is necessary to flush data cache. The following new routine does the data cache flush:

```
        flush_dcache(char *addr, int size)
```

The first argument is the logical address aligned on 32 bytes, the second argument is the size in bytes. The assembler source code is the following (in a C source file):

```
asm("
        .align 4
        .globl flush_dcache
        .globl .flush_dcache
        .csect flush_dcache[DS]
flush_dcache:
        .long .flush_dcache, TOC[tc0], 0
        .csect .text[PR]
        addis   10,0,0
        addi    0,0,32
fdc_loop:
        dcbf    10,3
        add     10,10,0
        cmp     0,10,4
        blt     fdc_loop
        sync
        blr
");
```

Appendix C: NetBSD Foundation Copyright Notice

CETIA

Vos remarques sur ce document/Technical publications remark form

Titre/Title : **CETIA PowerEngine - Release Notes PWL4-ESP for LynxOS 3.0.1 on VMPC Family**

No de référence/Reference No : **SD.DT.C00-3e**
ID production/Product ID: **00091**

ERREURS DETECTEES / ERRORS IN PUBLICATION

AMELIORATIONS SUGGEREES / SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

- Vos remarques et suggestions seront attentivement examinées.
  Si vous désirez une réponse écrite, veuillez indiquer votre adresse postale complète.
- Your comments will be promptly investigated by qualified technical personnel and action will be taken as required.
  If you require a written reply, furnish your complete mailing address below.

DATE :

NOM/NAME :

SOCIETE/COMPANY :

ADRESSE/ADDRESS :

Envoyer cet imprimé avec vos remarques à :
Please mail this form with remarks to:

CETIA
Service Documentation Technique CETIA
150, rue Marcellin Berthelot
ZI Toulon Est - BP 244
83078 TOULON Cedex 9
FRANCE
Tél : (+33) 04 94 16 34 00