ADLINK TECHNOLOGY INC.

aTCA-N700

Dual Cavium CN6880 Packet Processor
GbE ATCA Processor Blade

Software User's Manual



Manual Revision:	1.00
Revision Date:	August 28, 2013
Part Number:	50- 1G027-1000

Advance Technologies; Automate the World.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Revision History

Rev	Date	Description
1.00	2013/08/28	Initial release

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Preface

Copyright 2013 ADLINK Technology, Inc.

This document contains proprietary information protected by copyright. All rights are reserved. No part of this manual may be reproduced by any mechanical, electronic, or other means in any form without prior written permission of the manufacturer.

Disclaimer
The information in this document is subject to change without prior notice in order to improve reliability, design, and function and does not represent a commitment on the part of the manufacturer. In no event will the manufacturer be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or inability to use the product or documentation, even if advised of the possibility of such damages.

Environmental Responsibility
ADLINK is committed to fulfill its social responsibility to global environmental preservation through compliance with the European Union's Restriction of Hazardous Substances (RoHS) directive and Waste Electrical and Electronic Equipment (WEEE) directive. Environmental protection is a top priority for ADLINK. We have enforced measures to ensure that our products, manufacturing processes, components, and raw materials have as little impact on the environment as possible. When products are at their end of life, our customers are encouraged to dispose of them in accordance with the product disposal and/or recovery programs prescribed by their nation or company.

Trademarks
Product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table of Contents

REVISION HISTORY II
PREFACE III
LIST OF TABLES VI
LIST OF FIGURES VI
1. INTRODUCTION 1
2. SOFTWARE OVERVIEW 3
2.1. SDK 4
2.1.1. LMP-SDK 4
2.1.2. BCM-SDK 5
2.1.3. NP-SDK 5
2.2. BSP 5
2.2.1. LMP-BSP 5
2.2.2. BCM-BSP 5
2.2.3. NP-BSP 6
3. S/W DELIVERABLES 7
3.1. Contents of S/W Package 7
3.2. Installing S/W Package (for developer) 9
3.2.1. Prerequisites 9
3.2.1.1. LMP-SDK 9
3.2.1.2. NP-SDK 10
3.2.1.3. ipmitool / BCM SDK 12
3.2.2. Compiling aTCA-N700 S/W package - LMP 12
3.2.2.1. Patching 12
3.2.2.2. Build 13
3.2.3. Compiling aTCA-N700 S/W package - NP 14
3.2.3.1. Patching 14
3.2.3.2. Build 14
4. DEVICE DRIVERS 16
5. BOOTING 18
5.1. Booting Image Redundancy 19
5.1.1. LMP U-Boot Redundancy 19
5.1.2. LMP OS Redundancy 21
5.1.3. OCTEON U-Boot Redundancy 23
5.2. Booting via tftp 24
5.3. Booting via NFS 24
5.4. Booting via USB storage 25
5.5. Install / Upgrade 26
5.5.1. Installing LMP 26
5.5.2. Installing NP 27
5.5.3. Installing IPMC 28
5.5.4. Upgrade LMP U-Boot only 29
6. HARDWARE DIAGNOSTICS 30
6.1. Logging 31
6.2. POST 32
6.3. Base Switch 33
6.4. IPMI commands 34
6.5. Default Environment Variables 35
7. OCTEON HOST UTILITIES 38

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

8. SWITCH MANAGEMENT 40
8.1. Networking 41
8.1.1. Port Mapping 42
8.2. Initialization 44
8.3. bcm.user 44
8.3.1. Port configuration 45
8.3.2. Vlan configuration 46
8.3.3. Pvlan configuration 46
8.3.4. Spanning-tree group configuration 47
8.3.5. Mac configuration 48
8.3.6. Port interframe gap configuration 49
8.3.7. Port statistics 49
8.3.8. Port mirroring configuration 50
8.3.9. rcload 50
8.3.10. Switch configuration 51
8.3.11. LED 52
8.3.12. Base Switch 53
9. BOARD MANAGEMENT PACKAGE 56
9.1. nbm-uenv 56
9.2. nbm-robo 57
9.3. nbm-cpld 57
9.4. nbm-relay 57
9.5. nbm-mdio 57
9.6. nbm-util 57
9.7. nbm-diag 58
10. IPMI 61
10.1. ipmievd 61
10.2. ipmitool 62
11. IPMC ACCESS VIA ETHERNET 63
11.1. IPMI-over-LAN 63
11.1.1. IOL Overview 63
11.1.2. LAN, Channel and Session Configuration Commands 63
11.2. Serial-over-LAN 66
11.2.1. SOL Overview 66
11.2.2. SOL Commands 66
GETTING SERVICE 69

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

List of Tables

TABLE 1-1 ABBREVIATIONS 2
TABLE 3-1 DELIVERABLES 8
TABLE 4-1 SUMMARY OF DEVICES AND ACCESS 16
TABLE 5-1 NVM MAP 18
TABLE 5-2 TFTP AND PORT SETTINGS 24
TABLE 6-1 DEVICES AND TEST 30
TABLE 6-2 DEVICES COVERED BY THE POST 32
TABLE 6-3 IPMI COMMAND LIST 34
TABLE 6-4 NETWORKING ENVIRONMENT VARIABLES 36
TABLE 6-5 SDBOOT PARAMETERS 37
TABLE 7-1 OCTEON HOST UTILITIES 38
TABLE 8-1 BCM.USER PORT ASSIGNMENT WITH RTM1 42
TABLE 8-2 BCM.USER PORT ASSIGNMENT WITH AN RTM2 43
TABLE 8-3 SFP+ PORT LED OPERATIONS 52
TABLE 9-1 DIRECTIVES IN THE CONFIGURATION FILE 60
TABLE 10-1 IPMITOOL OPTIONS 62

List of Figures

FIGURE 2-1 ATCA-N700 SOFTWARE COMPONENTS 4
FIGURE 8-1 BROADCOM SDK COMPONENTS 40
FIGURE 8-2 BROADCOM DEVICE NETWORKING INTERFACES (RTM1) 41
FIGURE 8-3 BROADCOM DEVICE NETWORKING INTERFACES (RTM2) 42
FIGURE 8-4 ACCESS TO BCM.USER 44
FIGURE 10-1 IPMI BLOCK DIAGRAM 61
FIGURE 11-1 IPMI-OVER-LAN DIAGRAM 63
FIGURE 11-2 GET CHANNEL INFO COMMAND 64
FIGURE 11-3 SET CHANNEL ACCESS COMMAND 64
FIGURE 11-4 SET USER NAME COMMAND 64
FIGURE 11-5 GET CHANNEL ACCESS COMMAND 64
FIGURE 11-6 SET USER PASSWORD COMMAND 64
FIGURE 11-7 GET LAN CONFIGURATION PARAMETERS COMMAND 65
FIGURE 11-8 SET LAN CONFIGURATION PARAMETERS COMMAND 65
FIGURE 11-9 SERIAL-OVER-LAN DIAGRAM 66
FIGURE 11-10 SERIAL-OVER-LAN DIAGRAM 66
FIGURE 11-11 GET SOL CONFIGURATION PARAMETERS COMMAND 67
FIGURE 11-12 TESTING SERIAL-OVER-LAN 67
FIGURE 11-13 TERMINATING SERIAL-OVER-LAN SESSION 68

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1. Introduction

The aTCA-N700 is a single slot AdvancedTCA blade powered by dual OCTEON II multi-core processors for high performance packet processing applications such as LTE wireless, security and other networking applications. The blade provides up to 40Gbps full duplex packet processing performance with flexible network interfaces.

The aTCA-N700 is designed to provide high performance security enabled intelligent deep packet inspection and processing functions that are required by high-end network appliances, security appliances, switches, and servers. The major building block is the Cavium CN6880 packet processor subsystem.

The Cavium CN6880 has 2MB of L2 cache which is shared by 32 cnMIPS cores. Each cnMIPS core also has dedicated 37K Icache and 32K Dcache. DDR3 memory has total 400Gbps bandwidth. Hardware acceleration blocks are also available for the most advanced networking and security applications. RegEx engine enables high throughput pattern matching with optional co-processors.

The blade also has a high performance, non-blocking, local switch fabric to interconnect CN6880, backplane, and I/O ports altogether. This feature allows the data path to be configured in many different ways to fully utilize the multi-core CPUs and provide the flexibility for application programming. There are basically 3 modes of data path operation: serial, parallel, and by pass, which can be configured depending on the application requirements.

The application areas of the blade are summarized but not limited to:

- Wireless backhaul aggregation Switch/Router, which requires Deep Packet Inspection capability
- Security Appliances such as Firewalls, VPN and IDS/IPS
- Content based fine grain load balancing switches
- Front end I/O card with traffic management for high capacity, value added appliances for multi-service offering
- Network Monitoring and Analysis Services
- Ethernet aggregation nodes

This document describes a software overview and provides basic operation instructions for the users.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table 1-1 Abbreviations

Terms	Description
ATCA	Advanced Telecom Computing Architecture
Base Interface	ATCA Base Interface on the Zone-2 connectors to connect to the dual HUBs, 4 differential pairs per Base Channel.
Blade	A printed circuit board assembly that plugs into a chassis
BSP	Board Support Package
Boot-flash	Flash memory where the bootloader program is stored
Bootloader (bootrom)	Code used to boot the System before loading OS
DDR	Double Data Rate DRAM memory
DIMM	Dual Inline Memory Module
ECC	Error Correction Control
E-keying	Standard defined by PICMG useful to verify whether the ATCA blade plugged in is compliant to the fabric link capabilities
Fabric Interface	ATCA Fabric Interface on the Zone-2 connectors to connect to the dual Switch Fabric Blades
Fabric Slot	A slot supporting a link connection to/from each node slot
FRU	Field Replaceable Unit
GPIO	General Purpose Input/Output
Hot-swap	Functions of replacing system components without shutting down the system
IPMB	Intelligent Platform Management Bus
IPMB-L	IPMB Local interface
IPMC	Intelligent Platform Management Controller
IPMI	Intelligent Platform Management Interface
JTAG	Joint Test Action Group
LMP	Local Management CPU
MAC	Media Access Controller
MMC	Module Management Controller (on RTM or AMC)
NP or NPU	Network Processor Unit
PCI	Peripheral Component Interconnect
PCIe	Peripheral Component Interconnect-Express
PHY	Physical Layer or Physical Layer Device
PICMG	PCI Industrial Computer Manufacturers Group
RGMII	Reduced Gigabit Medium Independent Interface
RXAUI	Reduced XAUI, having 2 lanes@6.25Gbps
RTM	Rear Transition Module
SDR	Sensor Data Record
SDRAM	Synchronous Dynamic Random Access Memory
SFP	Small Form Pluggable (1G interface)
SFP+	Small Form Pluggable Plus (10G interface)
SGMII	Serial Gigabit Medium Independent Interface
SOL	Serial Over LAN
SPI	Serial Peripheral Interface Bus
QSFP+	Quad Small Form Pluggable Plus (40G interface)
UART	Universal Asynchronous Receiver-Transmitter
VLP	Very Low Profile, DRAM Module Form Factor Spec
XAUI	10 Gigabit (X) Attachment Unit Interface, having 4 lanes@3.125Gbps
XLAUI	40 Gigabit (XL) Attachment Unit Interface

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2. Software Overview

The aTCA-N700 has one Local Management Processor (LMP), dual Network Processor Units and another intelligent device, the IPMC. The Local Management Processor (LMP) P2041, OCTEON #1 (Top, right), OCTEON #2 (Bottom, left) and the SmartFusion IPMC. Since two NPs (OCTEONs) are in an identical configuration, the software for the NPs is identical. Therefore, on the blade, we have three software suites: the LMP software suite, the NP software suite, and the IPMC software suite. Each software suite comes in the form of a bootloader image and an OS image in one or more of the flash memories on the board, respectively. The LMP software suite, a dedicated board management and control plane module, includes the U-Boot Bootloader, the Open Linux OS, the Broadcom Switch software, the NP interface software, the IPMI software and the LMP BOMA (Board Management). All the functional modules other than U-Boot Bootloader become a part of the LMP OS image.

The NP software suite, a data plane and its supporting modules, includes the U-Boot Bootloader, the Open Linux OS, the OCTEON II software, and the OCTEON BOMA (Board Management) package.

The IPMC software suite, a chassis management module, includes vendor's bootloader, OS and IPMI package.

ADLINK's patches applied to the many of the standard modules include the following functions.

- **Booting Image Redundancy**
- **Device drivers**
- **H/W diagnostics**
- **S/W image upgrade**
- **LNI (LMP-NP Interface)**
- **RTM management**

Shown in Figure 2-1 are the software building blocks for the blade, in terms of their device (CPU) associations. Each of the components will be described in detail in the following sections.

The details on the vendor's original release software can be found in the documents listed below and will not be covered in this document

- *Mentor Embedded Linux Essentials for QorIQ Processors with Data Path Acceleration*, Rev. 1, 6/2011, Freescale
- *OCTEON Programmer's Guide(The Fundamentals), 2009, Cavium*
- Network Switching Software Platform Guide, 2012, Broadcom

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

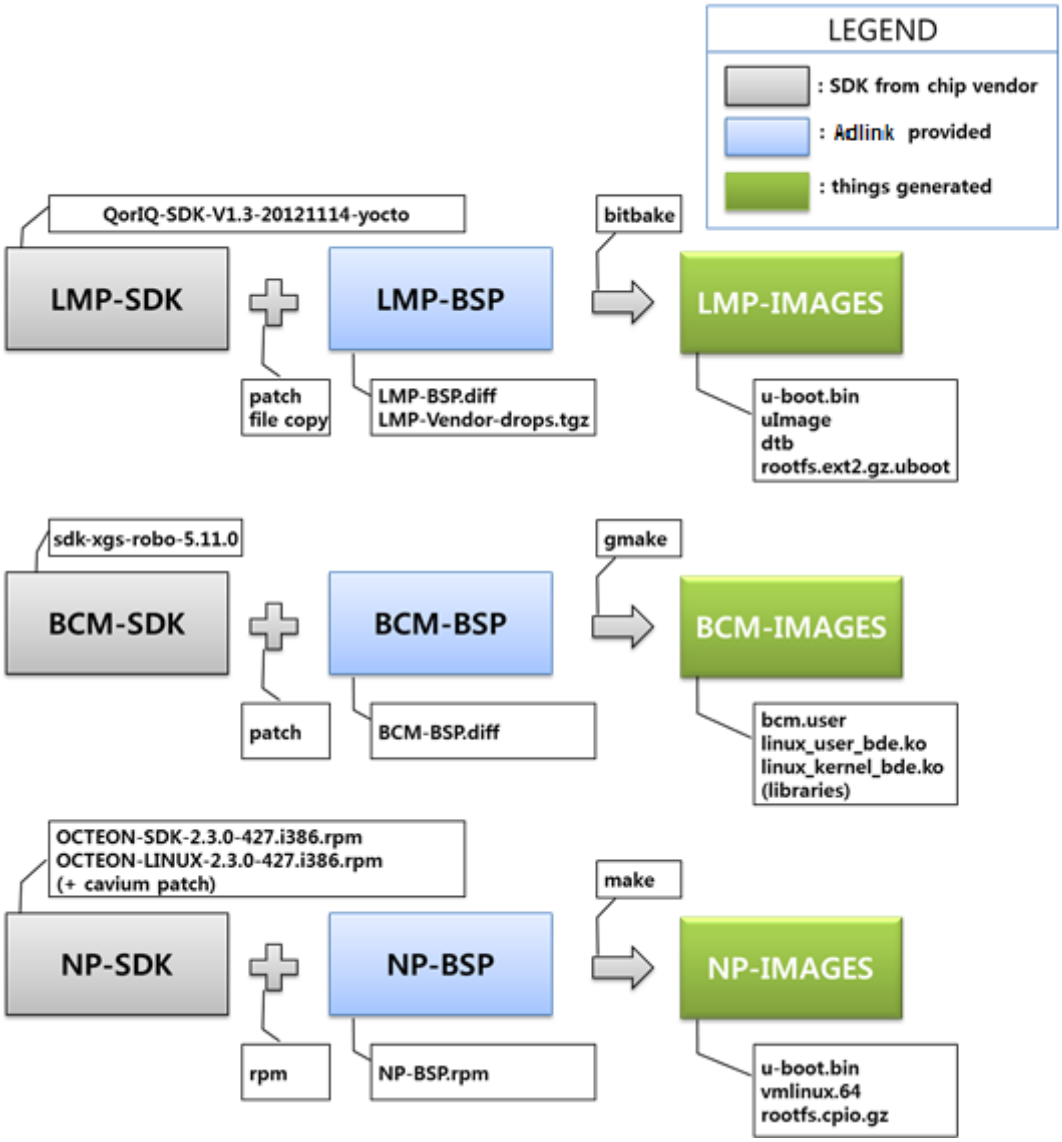


Figure 2-1 aTCA-N700 Software Components

2.1. SDK

2.1.1. LMP-SDK

The LMP-SDK is LMP (Local Management Processor) P2041 SDK is provided by Freescale.

As of December, 2012, most up-to-date version is ***1.3***, distributed in an iso file (QorIQ-SDK-V1.3-SOURCE-20121114-yocto.iso). After installation, the default directory generated is *QorIQ-SDK-V1.3-20121114-yocto*.

The directory structure follows LTIB (Linux Target Image Builder, See http://ltib.org), which enables *bitbake* tool to generate target images.

※ The details for the Freescale SDK installation and operation are found in ~/documents after the *iso* installation.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.1.2. BCM-SDK

The BCM-SDK is switch (BCM56842, BCM53115) SDK from Broadcom. Each application developer of the SDK requires a S/W license from Broadcom.

As of December, 2012, most up-to-date version for BCM5684x is ***5.11.0.***

(Note: 6,x version is available, but it's not for BCM5684x)

2.1.3. NP-SDK

The NP-SDK is NP (OCTEON CN6880) SDK from Cavium. Each application developer of the SDK requires a S/W license from Cavium.

As of December, 2012, the most up-to-date version is ***2.3.0 (patch level 4)***.

2.2. BSP

2.2.1. LMP-BSP

To generate the LMP images using the LMP-SDK for aTCA-N700, there should be patches to the existing standard package. Such patches include *LMP-Vendor-drops.tgz* composed of the rootrfs files and *LMP-BSP.diff* composed of Linux kernel and U-Boot changes.

LMP-Vendor-drops.tgz contains the following.

- **BCM executing files**
  bcm.user : BCM configuration SHELL program
  Linux_user_bde.ko, Linux_kernel_bde.ko : Linux kernel modules to access Broadcom switch physically.
  config.bcm, rc.soc : configuration files for bcm.user
- **ADLINK utilities**
  nbm-uenv : aTCA-N700 S/W upgrade program
  nbm-robo : base switch control program
  nbm-cpld : CPLD control program
  nbm-relay : fabric switch control daemon
  nbm-util : RTM Management program
  nbm-diag : aTCA-N700 H/W diagnostics program

The source codes for LMP-Vendor-drops.tgz may be released upon request.

2.2.2. BCM-BSP

The board specific switch configuration is reflected on to BCM-BSP, which uses the BCM-SDK APIs. The BCM-BSP is a collection of patch files to generate the aTCA-N700 Broadcom switch S/W.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2.3. NP-BSP

The board specific OCTEON II NPU configuration is reflected on to NP-BSP, which uses the NP-SDK APIs. The BCM-BSP is a collection of patch files to generate the aTCA-N700 NPU S/W.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

3. S/W deliverables

3.1. Contents of S/W Package

ADLINK's aTCA-N700 S/W package, which is to be released after the development is completed, is composed of the following components.

- **aTCA-N700_ sw-package_$(PACKAGE_VERSION).bsp.tgz :** LMP-BSP + BCM-BSP + NP-BSP
- **aTCA-N700_ sw-package_$(PACKAGE_VERSION).bin.tgz :** LMP-IMAGES + NP-IMAGES

  - *$(PACKAGE_VERSION):* $(SDK_VERSION).$(MAJOR_VERSION).$(MINOR_VERSION)
  - *$(SDK_VERSION):* base SDKs (LMP-SDK, BCM-SDK, NP-SDK) version change enforces increment of the version. (Starts from 1)
  - *$(MAJOR_VERSION):* Increments at a structure change or a feature change. (Starts from 0)
  - *$(MINOR_VERSION):* Increments at a S/W bug fix. (Starts from 0)

- **aTCA-N700_lmp-bsp.$(LMP_BSP_VERSION).diff:** Patches for LMP Linux and U-Boot sources. Included in *aTCA-N700_ sw-package_$(PACKAGE_VERSION).bsp.tgz*
- **aTCA-N700_lmp-vendor-drop.$(LMP_VD_VERSION).tgz:** Collection of rootfs files to create the image. Included in *aTCA-N700_ sw-package_$(PACKAGE_VERSION).bsp.tgz*

  - *$(LMP_BSP_VERSION)*: v.$(LMP_BSP_REVISION_NUMBER)
  - *$(LMP_VD_VERSION)*: v.$(LMP_APP_REVISION_NUMBER)
  - *$(LMP_BSP_REVISION_NUMBER):* Increments at a U-Boot or Linux kernel source change. (Starts from 1)
  - *$(LMP_APP_REVISION_NUMBER):* Increments at an LMP application change. (Starts from 1)

- **aTCA-N700_ bcm-bsp_$(BCM_BSP_VERSION).diff:** Broadcom S/W source patches and configurations. Included in *aTCA-N700_ sw-package_$(PACKAGE_VERSION).bsp.tgz*

  - *$(BCM_BSP_VERSION):* v.$(BCM_BSP_REVISION_NUMBER)
  - *$(BCM_BSP_REVISION_NUMBER):* Increments at a SBCM S/W change. (Starts from 1)

- **aTCA-N700_ np-bsp_$(NP_BSP_VERSION).rpm:** Patches to the NP Linux and the U-Boot sources. Included in *aTCA-N700_ sw-package_$(PACKAGE_VERSION).bsp.tgz*

  - *$(NP_BSP_VERSION):* v.$(NP_BSP_REVISION_NUMBER)
  - *$(NP_BSP_REVISION_NUMBER):* Increments at an NP (Octeon) S/W change. (Starts from 1)

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table 3-1 Deliverables

Item	Type	Contents	Description
lmp.patch	patch	QorIQ-DPAA-SDK generates (compiles) LMP U-Boot, linux kernel, and rootfs images with the patch.	**Usage example:** 1. Make sure that QorIQ-DPAA-SDK is installed. **2. patch –p1 < lmp.patch** **3. tar –xvzf lmp-vendor-drops.tgz** 4. Follow the compilation process by QorIQ-DPAA-SDK.
lmp-vendor-drops.tgz	binary	LMP linux application layer includes the following: - Broadcom switch S/W (bcm.user), - H/W diagnostics program - ipmi utility - CPLD access utility - upgrade utility	lmp.patch assumes that lmp-vendor-drops.tgz file is properly copied to the QorIQ-DPAA-SDK source tree.
bcm.patch	patch	BCM-SDK generates (compiles) Broadcom switch program with the patch.	**Usage example:** 1. Make sure that the BCM-SDK is installed. **2. patch –p1 <bcm.patch** 3. Follow the compilation process by the BCM-SDK.
np.rpm	rpm	OCTEON-SDK generates (compiles) NP U-Boot and linux images with U-Boot, linux, and executive patches.	**Usage example:** 1. Make sure that the OCTEON-SDK (rpm file) is installed. **2. rpm –ivh np.rpm** 3. Follow the compilation process by OCTEON-SDK.
uboot.lmp	binary	bootloader image for LMP	*upgrade* utility is used to load the images.
dtb.lmp	binary	device table binary image for LMP	
linux.lmp	binary	linux kenel image for LMP	
rootfs.lmp	binary	root file system image for LMP	
uboot.np	binary	bootloader image for LMP	
linux.np	binary	linux image for NP	
booter.ipmc	binary	booter image for IPMC	
firmware.ipmc	binary	firmware image for IPMC	
fruinfo.ipmc	binary	FRU info image for IPMC	
release-note	pdf	Package version change list and file names are provided.	
user-guide	pdf	S/W manual	

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

3.2. Installing S/W Package (for developer)

This section describes the procedures to install the S/W package from ADLINK for the aTCA-N700.

3.2.1. Prerequisites

3.2.1.1. LMP-SDK

1. The host machine should have the Freescale SDK installed.
   Version supported: QorIQ-SDK-V1.3-SOURCE-20121114-yocto.iso
2. With the Freescale SDK installed, fsl-image-full generation should be verified working.
   (After this step, sysroot directory is generated, which is used for aTCA-N700 S/W package compilation.)
3. aTCA-N700 S/W package users should have the access to the Freescale SDK toolchains.

For the details for Freescale SDK installation, refer to the following:

QorIQ-SDK-V1.3-SOURCE-20121114-yocto.iso documents directory

The following example is for QorIQ-SDK-V1.3-SOURCE-20121114-yocto installation on CentOS 6.3 host. (Run as a root account.)

```
[root@localhost ~]# vi /etc/sudoers
…
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
yglim  ALL=(ALL)       NOPASSWD: ALL <= add this line

[root@localhost ~]# mkdir /mnt/QorIQ-DPAA-SDK-20111026
[root@localhost ~]# mount -o loop QorIQ-SDK-V1.3-SOURCE-20121114-yocto.iso /mnt/QorIQ-SDK-V1.3
```

An installation example is shown below: the SDK will be installed in /home/yglim/work/QorIQ-SDK-V1.3-20121114-yocto with this sequence.

```
[yglim@localhost ~]$ /mnt/QorIQ-DPAA-SDK-20111026/install
Checking ISO integrity ...
You are about to install the SDK built by Yocto

Before installing , you must read and accept the EULA
(End User License Agreement) which will be presented next.

Do you want to continue ? Y|n
Y

…

I have read and accept the EULA (yes|no):
yes

Where do you want to install SDK? (/home/yglim)
/home/yglim/work
Making target directory /home/yglim/work/QorIQ-SDK-V1.3-20121114-yocto
Installing to /home/yglim/work/QorIQ-SDK-V1.3-20121114-yocto

Installation complete.

your SDK installation has been placed in
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
/home/yglim/work/QorIQ-SDK-V1.3-20121114-yocto, please execute following steps to
continue:

    $ cd /home/yglim/work/QorIQ-SDK-V1.3-20121114-yocto
    $ ./scripts/host-prepare.sh
    $ source ./fsl-setup-poky <-m machine>
    Optional parameters: [-j jobs] [-t tasks] [-s path] [-p] [-l] [-h]
    Supported ppc machines: mpc8536ds mpc8544ds mpc8548cds mpc8572ds p1010rdb p1020mbg
p1020rdb p1020utm p1021rdb p1022ds p1023rds p1024rdb p1025rdb p2020ds p2020rdb p2041rdb
p3041ds p4080ds p5020ds-64b p5020ds p5040ds-64b p5040ds
    * [-j jobs]:  number of jobs for make to spawn during the compilation stage.
    * [-t tasks]: number of BitBake tasks that can be issued in parallel.
    * [-s path]:  append an extra path to build_machine_release folder
    * [-l]:       lite mode. To help conserve disk space, deletes the building
                 directory once the package is built.
    * [-p]:       append cache and source mirrors (For FSL Internal Use Only)
    * [-h]:       help

    $ bitbake <task-name>

[yglim@localhost ~]$ cd /home/yglim/work/QorIQ-SDK-V1.3-20121114-yocto

[yglim@centos63 QorIQ-SDK-V1.3-20121114-yocto]$ ./scripts/host-prepare.sh
Verifying sudo permission to execute yum command.
Install packages needed to build Yocto, please wait, it may take a while
Loaded plugins: fastestmirror, refresh-packagekit, security
Loading mirror speeds from cached hostfile
 * base: centos.tt.co.kr
 * extras: centos.tt.co.kr
 * updates: centos.tt.co.kr
Setting up Group Process
Checking for new repos for mirrors
Package flex-2.5.35-8.el6.x86_64 already installed and latest version
…
Done. You're ready to go with Yocto build now

[yglim@centos63 QorIQ-SDK-V1.3-20121114-yocto]$ source ./fsl-setup-poky -m p2041rdb -j 4
-t 2
Configuring for p2041rdb board type
Run the following commands to start a build:
    bitbake fsl-image-lsb-sdk
    bitbake fsl-image-minimal
    bitbake fsl-image-full
    bitbake fsl-image-kvm
    bitbake fsl-image-flash
    bitbake fsl-image-core
To return to this build environment later please run:
    source /home/yglim/work/QorIQ-SDK-V1.3-20121114-
yocto/build_p2041rdb_release/SOURCE_THIS

[yglim@centos63 build_p2041rdb_release]$ bitbake fsl-image-full
Pseudo is not present but is required, building this first before the main build
…
(this takes many hours)
…

[yglim@centos63 build_p2041rdb_release]$ cd
[yglim@centos63 ~]$ sudo umount /mnt/QorIQ-DPAA-SDK-20111026
```

Now, you have installed the LMP SDK on the /home/yglim/work/QorIQ-SDK-V1.3-20121114-yocto directory.

3.2.1.2. NP-SDK

NP-SDK RPM packages from OCTEON should be installed in the build host. And also, patch level 4 for this RPM package should be adopted in the package. RPM package patch file list is shown below.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

- OCTEON-COMPONENTS-COMMON-2.3.0-62.i386.rpm
- OCTEON-PCI-NIC-2.3.0-85.i386.rpm
- OCTEON-LINUX-2.3.0-427.i386.rpm
- OCTEON-SDK-2.3.0-427.i386.rpm
- OCTEON-PCI-BASE-2.3.0-84.i386.rpm
- OCTEON-PCI-CNTQ-2.3.0-85.i386.rpm
- sdk_2.3.0_update_p4.tgz

Default installation location of these RPM package is /usr/local/Cavium_Networks/OCTEON-SDK. Copying the directory to your working directory is recommended, as shown below.

```
[yglim@centos63 octeon_packages]$ pwd
/home/yglim/work/octeon_packages

[yglim@centos63 octeon_packages]$ ls
OCTEON-COMPONENTS-COMMON-2.3.0-62.i386.rpm    OCTEON-PCI-CNTQ-2.3.0-
85.i386.rpm    sdk_2.3.0_update_p4.tgz
OCTEON-LINUX-2.3.0-427.i386.rpm                    OCTEON-PCI-NIC-2.3.0-
85.i386.rpm
OCTEON-PCI-BASE-2.3.0-84.i386.rpm                  OCTEON-SDK-2.3.0-427.i386.rpm

[yglim@centos63 octeon_packages]$ sudo rpm -i *.rpm

[yglim@centos63 octeon_packages]$ tar xzf sdk_2.3.0_update_p4.tgz

[yglim@centos63 octeon_packages]$ cd /usr/local/Cavium_Networks/OCTEON-SDK

[yglim@centos63 OCTEON-SDK]$ sudo patch -s -p1 <
/home/yglim/work/octeon_packages/sdk_2.3_patch_release4/sdk.patch

[yglim@centos63 OCTEON-SDK]$ cd linux

[yglim@centos63 linux]$ sudo patch -s -p2 <
/home/yglim/work/octeon_packages/sdk_2.3_patch_release4/linux.patch

[yglim@centos63 linux]$ cd ..

[yglim@centos63 OCTEON-SDK]$ sudo
/home/yglim/work/octeon_packages/sdk_2.3_patch_release4/octeon_copy_2_3_0_p4_f
iles.sh /home/yglim/work/octeon_packages/sdk_2.3_patch_release4
Copying files from /home/yglim/work/octeon_packages/sdk_2.3_patch_release4
/usr/local/Cavium_Networks/OCTEON-SDK
/usr/local/Cavium_Networks/OCTEON-SDK
/usr/local/Cavium_Networks/OCTEON-SDK
/usr/local/Cavium_Networks/OCTEON-SDK
/usr/local/Cavium_Networks/OCTEON-SDK
/usr/local/Cavium_Networks/OCTEON-SDK
/usr/local/Cavium_Networks/OCTEON-SDK

[yglim@centos63 OCTEON-SDK]$ cd ..

[yglim@centos63 Cavium_Networks]$ ls
OCTEON-SDK

[yglim@centos63 Cavium_Networks]$ cp -dpR OCTEON-SDK/ ~/work/OCTEON-SDK-
2.3.0
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Now, you have installed NP SDK on the /home/yglim/work/OCTEON-SDK-2.3.0 directory.

3.2.1.3. ipmitool / BCM SDK

The board management features are heavily related with ipmitool. The ipmitool patch set for aTCA-N700 board is also provided in the released source packages. The vanilla source code of ipmitool-1.8.11 should be prepared to generate proper root filesystem for aTCA-N700.

Furthermore, BCM SDK is also needed to build rootfilesystem for aTCA-N700.

3.2.2. Compiling aTCA-N700 S/W package - LMP

You should already have vanilla codes of ipmitool, linux kernel source and bcm-sdk on the host machine.

Linux kernel source can be extracted from the LMP-SDK.

```
# ls
ipmitool-1.8.11  linux-qoriq-sdk-3.0.48-r5  sdk-xgs-robo-5.11.0
```

3.2.2.1. Patching

Copy all source packages that were released by ADLINK to your build directory (in this case: /home/yglim/work/QorIQ-SDK-V1.3.aTCA-N700).

```
# ls
ipmitool-1.8.11  linux-qoriq-sdk-3.0.48-r5  sdk-xgs-robo-5.11.0
# tar -xvzf aTCA-N700_sw-package_v0.0.1.lmp-bsp.tgz
# tar -xvzf aTCA-N700_sw-package_v0.0.2.lmp-bsp.tgz
# ls
contrib                                          files
scriptsVERSION
ipmitool-1.8.11                                  sdk-xgs-robo-5.11.0
linux-qoriq-sdk-3.0.48-r5                        target
Makefile                                         U-Boot-2011.09
naboma README

# make patch
```

In order to make roofs image, copying basic RFS image file from QorIQ-SDK to your build directory is needed.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Basic RFS file name should be **fsl-image-core-aTCA-N700-20121213061548.rootfs.tar.gz**

```
# mv fsl-rfs-core.gz fsl-image-core-aTCA-N700-20121213061548.rootfs.tar.gz
ls
contrib                                          files
scriptsVERSION
ipmitool-1.8.11                                  sdk-xgs-robo-5.11.0
linux-qoriq-sdk-3.0.48-r5                        target
Makefile                                         U-Boot-2011.09
naboma README
fsl-image-core-aTCA-N700-20121213061548.rootfs.tar.gz
```

3.2.2.2. Build

Environment variables described below should be configured. ${QORIQ_SDK} and ${QORIQ_BSP} should be configured properly as your build environment.

```
export QORIQ_SDK=/home/yglim/work/QorIQ-SDK-V1.3-20121114-yocto
export QORIQ_BSP=/home/yglim/work/QorIQ-SDK-V1.3.aTCA-N700

export
QORIQ_SYSROOT=${QORIQ_SDK}/build_p2041rdb_release/tmp/sysroots/p2041rdb
export
QORIQ_TOOLCHAIN_INCLUDE=${QORIQ_SDK}/build_p2041rdb_release/tmp/sysroots/x
86_64-linux/usr/lib/ppce500mc-fsl-linux/gcc/powerpc-fsl-
linux/4.6.2/include
export QORIQ_BSP_WORK_BASE=${QORIQ_BSP}

export PATH=${QORIQ_SDK}/build_p2041rdb_release/tmp/sysroots/x86_64-
linux/usr/bin/ppce500mc-fsl-
linux:${QORIQ_SDK}/scripts:${QORIQ_SDK}/bitbake/bin/:${PATH}

export CROSS_COMPILE=powerpc-fsl-linux-
export ARCH=powerpc
export CPP="${CROSS_COMPILE}gcc -E --sysroot=$QORIQ_SYSROOT"
export LD="${CROSS_COMPILE}ld --sysroot=$QORIQ_SYSROOT"
export AR="${CROSS_COMPILE}ar"
export RANLIB="${CROSS_COMPILE}ranlib"
export CCLD="${CROSS_COMPILE}gcc -mcpu=e500mc -maix-struct-return --
sysroot=$QORIQ_SYSROOT"
export F77="${CROSS_COMPILE}g77 -mcpu=e500mc -maix-struct-return --
sysroot=$QORIQ_SYSROOT"
export CC="${CROSS_COMPILE}gcc -mcpu=e500mc -maix-struct-return --
sysroot=$QORIQ_SYSROOT"
export TOOLCHAIN_OPTIONS="--sysroot=$QORIQ_SYSROOT"
export CXX="${CROSS_COMPILE}g++ -mcpu=e500mc -maix-struct-return --
sysroot=$QORIQ_SYSROOT"
```


LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Run as described below to build LMP images. After build process, three target images (LMP U-Boot, dtb, linux kernel uImage) will be located in the **${QORIQ_BSP}/target** directory. Ramdisk rootfilesystem file will be located in the **${QORIQ_BSP}**.

```
# make binpack
# make rfs
# ls ${QORIQ_BSP}/target/boot
aTCA-N700.dtb    U-Boot.bin    uImage
# ls ${QORIQ_BSP}/rootfs.ext2.gz.uboot
```

3.2.3. Compiling aTCA-N700 S/W package - NP

3.2.3.1. Patching

Copy all source packages released by ADLINK to your NP build directory. (In this case: /home/yglim/work/OCTEON-SDK-2.3.0), and untar them.

```
# tar xzf aTCA-N700_sw-package_v0.0.1.np-bsp.tgz
# tar xzf aTCA-N700_sw-package_v0.0.2.np-bsp.tgz
# for i in `ls octeon.*.patch`; do patch -p0 < $i; mv $i $i.done; done
```

3.2.3.2. Build

Environment variables described below should be configured. **${PATH}** and **${OCTEON_ROOT}** should be configured properly as your build environment.

```
source env-setup OCTEON_CN68XX_PASS2_1

export OCTEON_CPPFLAGS_GLOBAL_ADD="-
DUSE_RUNTIME_MODEL_CHECKS=1 -
DCVMX_ENABLE_PARAMETER_CHECKING=0 -
DCVMX_ENABLE_CSR_ADDRESS_CHECKING=0 -
DCVMX_ENABLE_POW_CHECKS=0"

export PATH=/home/yglim/work/OCTEON-SDK-
2.3.0/tools/bin:/home/yglim/work/OCTEON-SDK-2.3.0/host/bin:${PATH}

export OCTEON_MODEL=OCTEON_CN68XX_PASS2_1
export OCTEON_ROOT=/home/yglim/work/OCTEON-SDK-2.3.0
```

Build NP images as below. After build process, three target images (NP U-Boot, linux, rootfilesystem image) will be located in the directories, as described below

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
# cd ${OCTEON_ROOT}/linux
# make link-kernel-se-files; make kernel; make kernel-deb;

# cd ${OCTEON_ROOT}/bootloader/U-Boot/
# make aTCA-N700_config; make

# ls ${OCTEON_ROOT}/linux/kernel_2.6/linux/vmlinux.64

# ls ${OCTEON_ROOT}/bootloader/U-Boot/U-Boot-octeon_aTCA-N700.bin

# ls ${OCTEON_ROOT}/linux/embedded_rootfs/rootfs.cpio.gz
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4. Device Drivers

The Device Drivers provide device accessibility for the devices on the aTCA-N700 specific devices. The vendor's drivers are ported to the blade environments with necessary changes.

Table 4-1 Summary of devices and access

Device	ID	Description	Host	Program	Remark
P2041	LMP	Local Management Processor	LMP	(All LMP Linux programs)	LMP-SDK + ADLINK patch
Mini VLP DIMM	LMP_MEM	DRAM for LMP	LMP	LMP Linux kernel	LMP-SDK + ADLINK patch
BCM56842	FABRIC_SWITCH	multilayer switch	LMP	bcm.user	BCM-SDK + ADLINK patch
BCM53115	BASE_SWITCH	base switch	LMP	bcm.user	BCM-SDK + ADLINK patch
CPLD	CPLD	CPLD for system control	LMP	nbm-cpld	
BCM54612E	MGMT_PHY	PHY for mgmt Ethernet port of LMP	LMP	nbm-mdio	
BCM54612E	BASE_PHY	PHY for base Ethernet port of LMP (btw. LMP & BASE_SWITCH)	LMP	nbm-mdio	
M24256	LMP_EEPROM	mac addresses for LMP	LMP	nbm-eeprom	
DS1388	RTC	Real Time Clock	LMP	date, hwclock...	LMP-SDK (Linux standard utilities)
MAX5820	PWR_CTRL	power control for FABRIC_SWITCH	LMP	i2cget, i2cset...	LMP-SDK (Linux standard utilities)
Front SFP(+)	FP1, FP2, FP3, FP3	Front SFP(+) module	LMP	i2cget, i2cset...	LMP-SDK (Linux standard utilities)
I2CIO	F_I2CIO	I2C to IO buffer	LMP	i2cget, i2cset...	LMP-SDK (Linux standard utilities)
PCA9546	F_I2CHUB	I2C HUB device	LMP	(kernel)	vitual I2C bus
NTPM-RTM1/2	RTM_CTRL_DEVS	retimer, sfp(+) modules, i2c switch devices on the RTM	LMP	nbpm-sfp	optional
SPI_FLASH	SPI_FLASH1 SPI_FLASH2	boot flash for U-Boot	LMP	mtd-utilities	LMP-SDK + ADLINK patch Linux MTD system based
CHIP_SSD	CHIP_SSD1 CHIP_SSD2	OS, data flash for Linux	LMP	mount, ...	LMP-SDK (Linux standard utilities)
A2F500	IPMC	IPMC	LMP	*ipmitool*	open source + ADLINK patch
A2F500	IPMC	IPMC	SHMC	-	PICMG standard
CN5880	OCTEON_R OCTEON_L	right octeon left octeon	LMP	oct-remote-*	OCTEON host utilities (oct-remote-load, oct-remote-*bootcmd*...)

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Device	ID	Description	Host	Program	Remark
CN5880	OCTEON_R OCTEON_L	right octeon left octeon	NP	(All NP Linux programs)	NP-SDK + ADLINK patch
DIMM	DIMM_OCTEON_R DIMM_OCTEON_L	DIMM set of OCTEON_R DIMM set of OCTEON_L	NP	NP Linux kernel	NP-SDK + ADLINK patch
NOR_FLASH	NOR_FLASH_R_1 NOR_FLASH_R_2 NOR_FLASH_L_1 NOR_FLASH_L_1	Nor Flash1 of OCTEON_R Nor Flash2 of OCTEON_R Nor Flash1 of OCTEON_L Nor Flash2 of OCTEON_L	NP	mtd-utilities	NP-SDK + ADLINK patch Linux MTD system based
NAND_FLASH	NAND_FLASH_R_1 NAND_FLASH_R_2 NAND_FLASH_L_1 NAND_FLASH_L_2	Nand Flash1 of OCTEON_R Nand Flash2 of OCTEON_R Nand Flash1 of OCTEON_L Nand Flash2 of OCTEON_L	NP	mtd-utilities	NP-SDK + ADLINK patch Linux MTD system based

.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

5. Booting

The aTCA-N700 provides various NVMs (Non Volatile memory) to store software images and log system events for different purposes, especially used for a booting procedure.

The memory map for the devices is illustrated in Table 5-1.

Table 5-1 NVM Map

Host	Device	Size	Contents	format	U-Boot	linux	Remark
LMP	SPI_FLASH1	0xA0000	U-Boot image	binary	read/write	/dev/mtd0	Same S/W access point to SPI_FLASH1, SPI_FLASH2. Selected by a CPLD register
		0x20000	U-Boot Env.	binary	read/write	/dev/mtd1	
		0x40000	Reserved	binary	read/write	/dev/mtd2	
	SPI_FLASH2	0xA0000	U-Boot image	binary	read/write	/dev/mtd0	
		0x20000	U-Boot Env.	binary	read/write	/dev/mtd1	
		0x40000	Reserved	binary	read/write	/dev/mtd2	
	CHIP_SSD1	256 MiB	OS images	VFAT	read only	/dev/sda1 (/bootA)	
		256 MiB	System config	EXT3	no access	/dev/sda2 (/var/NVMA)	
		(remains)	user defined	user defined	no access	/dev/sda3	
	CHIP_SSD2	256 MiB	OS images	VFAT	read only	/dev/sda4 (/var/bootB)	
		256 MiB	System config	EXT3	no access	/dev/sda5 (/var/NVMB)	
		(remains)	user defined	user defined	no access	/dev/sda6	
OCTEON_R	NOR_FLASH1	1MiB	U-Boot image	binary	read/write	/dev/mtd0	Same S/W access point to NOR_FLASH1, NOR_FLASH2. Selected by a CPLD register
		6MiB	Reserved	binary	read/write	/dev/mtd1	
		1MiB	U-Boot Env.	binary	read/write	/dev/mtd2	
	NOR_FLASH2	1MiB	U-Boot image	binary	read/write	/dev/mtd0	
		6MiB	Reserved	binary	read/write	/dev/mtd1	
		1MiB	U-Boot Env.	binary	read/write	/dev/mtd2	
	NAND_FLASH1	256 MiB	OS images	UBIFS	read only	/dev/ubi0 (/bootA)	
		256 MiB	System config	UBIFS	no access	/dev/ubi1 (/var/NVMA)	
		(remains)	user defined	user defined	no access	-	
	NAND_FLASH2	256 MiB	OS images	UBIFS	read only	/dev/ubi2 (/bootB)	
		256 MiB	System config	UBIFS	no access	/dev/ubi3 (/var/NVMB)	
		(remains)	user defined	user defined	no access	-	

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Host	Device	Size	Contents	format	U-Boot	linux	Remark
OCTEON_L	NOR_FLASH1	1MiB	U-Boot image	binary	read/write	/dev/mtd0	Same S/W access point to NOR_FLASH1, NOR_FLASH2. Selected by a CPLD register
		6MiB	Reserved	binary	read/write	/dev/mtd1	
		1MiB	U-Boot Env.	binary	read/write	/dev/mtd2	
	NOR_FLASH2	1MiB	U-Boot image	binary	read/write	/dev/mtd0	
		6MiB	Reserved	binary	read/write	/dev/mtd1	
		1MiB	U-Boot Env.	binary	read/write	/dev/mtd2	
	NAND_FLASH1	256 MiB	OS images	UBIFS	read only	/dev/ubi0 (/bootA)	
		256 MiB	System config	UBIFS	no access	/dev/ubi1 (/var/NVMA)	
		(remains)	user defined	user defined	no access	-	
	NAND_FLASH2	256 MiB	OS images	UBIFS	read only	/dev/ubi2 (/bootB)	
		256 MiB	System config	UBIFS	no access	/dev/ubi3 (/var/NVMB)	
		(remains)	user defined	user defined	no access	-	

5.1. Booting Image Redundancy

The aTCA-N700 S/W package provides booting image redundancy to survive a single point of failure during a booting process. When a device with a CPU core boots up from an image stored in a storage device, the system guarantees to boot up fine by having redundant bootable images stored separately in a primary and a redundant storage devices. The bootloader U-Boot image and the Linux operating system image are stored in different storage devices.
The booting image redundancy applies to both the LMP and the dual Octeons as summarized below.

- LMP U-Boot image redundancy
- LMP OS image redundancy
- NP U-Boot image redundancy
- NP OS image redundancy

5.1.1. LMP U-Boot Redundancy

The U-Boot image is stored in a SPI flash memory. On the aTCA-N700, the LMP has two of such U-Boot flash memories in a primary and a redundant fashion.

To select one of the redundant images, use the following example.

```
=> ipmi active-flash              -> shows current active flash (result = 0 or 1)
active-flash: 0                   -> means active-flash is the first one.
=> ipmi switch-active              -> change active flash to the second one (toggle
fashion).
                                  -> A reset occurs automatically.
U-Boot 2011.09-svn118 (Jul 08 2013 - 19:25:54)

CPU0:    P2041E, Version: 1.1, (0x82180111)
Core:    E500MC, Version: 2.2, (0x80230022)
Clock Configuration:
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
          CPU0:1200 MHz, CPU1:1200 MHz, CPU2:1200 MHz, CPU3:1200 MHz,
          CCB:600    MHz,
          DDR:600    MHz (1200 MT/s data rate) (Asynchronous), LBC:37.500 MHz
          FMAN1: 500 MHz
          PME:     300 MHz
L1:       D-cache 32 kB enabled
          I-cache 32 kB enabled
Board: aTCA-N700, 36-bit Addressing, bank#1                -> this shows active flash
number
I2C:     ready
SPI:     ready
DRAM:    Detected UDIMM i-DIMM
          DDR: 2 GiB (DDR3, 64-bit, CL=8, ECC on)
Date:    2013-07-16 (Tuesday) Time: 20:33:54
IPMC:    OK
          Manuf.:    ADLINK Technology
          Name:       aTCA-N700
          Part#:     aTCA-N700
          Version: A2
          Serial#:
          Slot:       4
          Reset:     FRU control command(opt.: cold reset)
L2:       128 KB enabled
L3:       1024 KB enabled(Corenet Platform Cache)
MMC:      FSL_ESDHC : Card is not detected
SF:       Detected SST25VF080B with page size 4 KiB, total 1 MiB
EEPROM:OK
PCIe1: Root Complex, x1, regs @ 0xfe200000
          01:00.0        - 177d:0091 - Processor
PCIe1: Bus 00 - 01
PCIe2: Root Complex, x2, regs @ 0xfe201000
          03:00.0        - 14e4:b842 - Network controller
PCIe2: Bus 02 - 03
PCIe3: Root Complex, x1, regs @ 0xfe202000
          05:00.0        - 177d:0091 - Processor
PCIe3: Bus 04 - 05
In:       serial
Out:      serial
Err:      serial
XGS:      0xa0000868 : 0xd13f0201
OK(0.9V)
ROBO:    OK
SF:       Detected SST25VF080B with page size 4 KiB, total 1 MiB
Net:      Initializing Fman
SF:       Detected SST25VF080B with page size 4 KiB, total 1 MiB
Fman1: Uploading microcode version 106.1.4
FM1@DTSEC4, FM1@DTSEC5
Hit any key to stop autoboot:    0
=> ipmi active-flash               -> shows current active flash (result = 0 or 1)
active-flash: 1                      -> means active-flash is the second one.
=> ipmi switch-active              -> change active flash to the first one (toggle fashion).
                                       -> A reset occurs automatically.
U-Boot 2011.09-svn118 (Jul 08 2013 - 19:25:54)

CPU0:    P2041E, Version: 1.1, (0x82180111)
Core:    E500MC, Version: 2.2, (0x80230022)
Clock Configuration:
          CPU0:1200 MHz, CPU1:1200 MHz, CPU2:1200 MHz, CPU3:1200 MHz,
          CCB:600    MHz,
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
          DDR:600    MHz (1200 MT/s data rate) (Asynchronous), LBC:37.500 MHz
          FMAN1: 500 MHz
          PME:      300 MHz
L1:       D-cache 32 kB enabled
          I-cache 32 kB enabled
Board: aTCA-N700, 36-bit Addressing, bank#0                -> this shows active flash
number
I2C:      ready
SPI:      ready
DRAM:    Detected UDIMM i-DIMM
          DDR: 2 GiB (DDR3, 64-bit, CL=8, ECC on)
Date:    2013-07-16 (Tuesday) Time: 20:39:20
IPMC:    OK
          Manuf.:    ADLINK Technology
          Name:       aTCA-N700
          Part#:     aTCA-N700
          Version: A2
          Serial#:
          Slot:       4
          Reset:     FRU control command(opt.: cold reset)
L2:       128 KB enabled
L3:       1024 KB enabled(Corenet Platform Cache)
MMC:      FSL_ESDHC : Card is not detected
SF:       Detected SST25VF080B with page size 4 KiB, total 1 MiB
EEPROM:OK
PCIe1: Root Complex, x1, regs @ 0xfe200000
          01:00.0        - 177d:0091 - Processor
PCIe1: Bus 00 - 01
PCIe2: Root Complex, x2, regs @ 0xfe201000
          03:00.0        - 14e4:b842 - Network controller
PCIe2: Bus 02 - 03
PCIe3: Root Complex, x1, regs @ 0xfe202000
          05:00.0        - 177d:0091 - Processor
PCIe3: Bus 04 - 05
In:       serial
Out:      serial
Err:      serial
XGS:      0xa0000868 : 0xd13f0201
OK(0.9V)
ROBO:    OK
SF:       Detected SST25VF080B with page size 4 KiB, total 1 MiB
Net:      Initializing Fman
SF:       Detected SST25VF080B with page size 4 KiB, total 1 MiB
Fman1: Uploading microcode version 106.1.4
FM1@DTSEC4, FM1@DTSEC5
Hit any key to stop autoboot:    0
=>
```

5.1.2. LMP OS Redundancy

The aTCA-N700 provides dual Chip SSD devices for a primary storage and a redundant storage for the OS image.

To select one of the redundant images, use the following example.

```
=> run satabootA                          -> boot from the primary SSD storage.
SATA0 (1.5 Gbps)
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
            SATA1 (1.5 Gbps)
reading uImage

3815369 bytes read
reading aTCA-N700.dtb

28054 bytes read
WARNING: adjusting available memory to 30000000
## Booting kernel from Legacy Image at 01000000 ...
    Image Name:     Linux-3.0.48-rt70-svn33
    Created:          2012-12-18     7:36:48 UTC
    Image Type:     PowerPC Linux Kernel Image (gzip compressed)
    Data Size:       3815305 Bytes = 3.6 MiB
    Load Address: 00000000
    Entry Point:    00000000
    Verifying Checksum ... OK
## Flattened Device Tree blob at 00c00000
    Booting using the fdt blob at 0xc00000
    Uncompressing Kernel Image ... OK
    Loading Device Tree to 03fe6000, end 03fffd95 ... OK
Using aTCA-N700 machine description
Memory CAM mapping: 256/256/256 Mb, residual: 1280Mb
…
Yocto (Built by Poky 7.0) 1.2 aTCA-N700 ttyS0

aTCA-N700 login: root
root@aTCA-N700:~# cat /proc/cmdline
root=/dev/sdb2 rw console=ttyS0,115200    -> this shows bootup is completed using
the secondary SSD.

root@aTCA-N700:~# reboot
…
=> run satabootB                                  -> boot from the secondary SSD storage.

SATA0 (1.5 Gbps)
            SATA1 (1.5 Gbps)
reading uImage

3815369 bytes read
reading aTCA-N700.dtb

28054 bytes read
WARNING: adjusting available memory to 30000000
## Booting kernel from Legacy Image at 01000000 ...
    Image Name:     Linux-3.0.48-rt70-svn33
    Created:          2012-12-18     7:36:48 UTC
    Image Type:     PowerPC Linux Kernel Image (gzip compressed)
    Data Size:       3815305 Bytes = 3.6 MiB
    Load Address: 00000000
    Entry Point:    00000000
    Verifying Checksum ... OK
## Flattened Device Tree blob at 00c00000
    Booting using the fdt blob at 0xc00000
    Uncompressing Kernel Image ... OK
    Loading Device Tree to 03fe6000, end 03fffd95 ... OK
Using aTCA-N700 machine description
Memory CAM mapping: 256/256/256 Mb, residual: 1280Mb
…
Yocto (Built by Poky 7.0) 1.2 aTCA-N700 ttyS0
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
aTCA-N700 login: root
root@aTCA-N700:~# cat /proc/cmdline
root=/dev/sda2 rw console=ttyS0,115200
```

5.1.3. OCTEON U-Boot Redundancy

The OCTEON U-Boot image is stored in a NOR flash memory. On the aTCA-N700, each NP has two of such U-Boot flash memories in a primary and a redundant fashion. The flash device selection is done by the CPLD and configurable by a CPLD register on. Shown below is a command example of such setting.

```
[ NOR_FLASH1 for NP-A (top) ]
root@aTCA-N700:~# nbm-cpld flashctl @npa 0          <- select
NOR_FLASH1(bank#0) for NP-A
root@aTCA-N700:~# nbm-cpld flashctl @npa            <- show Active Flash for NP-A
NPA BANK#: 0
root@aTCA-N700:~# nbm-cpld reset np 0               <- execute reset

[ NOR_FLASH2 NP-A (top) ]
root@aTCA-N700:~# nbm-cpld flashctl @npa 1          <- select
NOR_FLASH2(bank#1) for NP-A
root@aTCA-N700:~# nbm-cpld flashctl @npa            <- show Active Flash for NP-A
NPA BANK#: 1
root@aTCA-N700:~# nbm-cpld reset np 0               <- execute reset

[ NOR_FLASH1 for NP-B (bottom) ]
root@aTCA-N700:~# nbm-cpld flashctl @npb 0          <- select
NOR_FLASH1(bank#0) for NP-B
root@aTCA-N700:~# nbm-cpld flashctl @npb            <- show Active Flash for NP-B
NPA BANK#: 0
root@aTCA-N700:~# nbm-cpld reset np 0               <- execute reset

[ NOR_FLASH2 NP-B (bottom) ]
root@aTCA-N700:~# nbm-cpld flashctl @npb 1          <- select
NOR_FLASH2(bank#1) for NP-B
root@aTCA-N700:~# nbm-cpld flashctl @npb            <- show Active Flash for NP-B
NPA BANK#: 1
root@aTCA-N700:~# nbm-cpld reset np 0               <- execute reset
```

Shown below is a command example of such setting.

```
aTCA-N700# run ubibootA                             -> boot from primary NAND
Flash
Creating 1 MTD partitions on "nand0":
0x000000000000-0x000020000000 : "mtd=0"
…
BusyBox v1.18.4 (2013-02-27 02:59:52 KST) built-in shell (ash)
Enter 'help' for a list of built-in commands.
~ # cat /proc/cmdline
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
mtdparts=octeon_nand0:512m(osA),1536m(rfsA);octeon_nand1:512m(osB),1536m(rfs
B) ubi.mtd=2 root=ubi0:rfs rw rootfstype=ubifs primary console=ttyS0,115200
~ # reboot
…

aTCA-N700# run ubibootB                          -> boot from secondary NAND
Flash
Creating 1 MTD partitions on "nand1":
0x000000000000-0x000020000000 : "mtd=0"
…
BusyBox v1.18.4 (2013-02-27 02:59:52 KST) built-in shell (ash)
Enter 'help' for a list of built-in commands.


~ # cat /proc/cmdline

mtdparts=octeon_nand0:512m(osA),1536m(rfsA);octeon_nand1:512m(osB),1536m(rfs
B) ubi.mtd=4 root=ubi0:rfs rw rootfstype=ubifs secondary console=ttyS0,115200
```

5.2. Booting via tftp

To use tftp booting, either the management port or the Base Interface port are available to connect to the tftp server. **Table 5-2** describes the details.

Table 5-2 tftp and port settings

Interface	cmd example	Server
MGMT phy	=> setenv serverip 192.168.0.142 => setenv ipaddr 192.168.78.199 => setenv ethact FM1@DTSEC4 => run tftpboot	tftp server on the same LAN through the faceplate port
Base port	=> setenv serverip 10.0.0.142 => setenv ipaddr 10.0.0.2 => setenv ethact FM1@DTSEC5 => robo init => robo enable 0 => robo enable 6 => run tftpboot	tftp server connected through the backplane

5.3. Booting via NFS

To boot to linux in U-Boot using Network File System (NFS), some NFS Server configuration needs to be in place as shown below.

/etc/exportfs file should contain the directory information for the file system.

```
# cat /etc/exportfs
/tftpboot/aTCA-N700T/RFS/rootfs *(rw,no_root_squash)

# exportfs                              -> exportfs reconfiguration
# mknod rootfs/dev/console c 5 1        -> Generate a special file in rootfs/dev/
```

On U-Boot, configuration setup is needed for NFS boot-up. Pre-defined U-Boot environment variables can be used.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
=> setenv rootpath /tftpboot/aTCA-N700T/RFS/rootfs    -> Set rootfs path information
=> setenv netdev eth0                                 -> For REV1 set netdev as eth0(mgmt port)

=> setenv serverip 192.168.0.142                      -> set NFS server IP address
=> setenv ipaddr 192.168.34.56                        -> set my IP address
=> setenv gatewayip 192.168.0.1                       -> set gateway IP address
=> setenv netmask 255.255.0.0                         -> set netmask
=> run nfsboot
```

5.4. Booting via USB storage

To boot from a USB storage attached to the blade face plate connector with a pre-programmed images on the storage, see the following example.

```
=> usb start
=> fatload usb0 0 $loadaddr kal
=> fatload usb0 0 $fdtaddr kad
=> fatload usb0 0 $ramdiskaddr kar
=>   setenv   bootargs   root=/dev/ram   rw   console=$consoledev,$baudrate
ramdisk_size=460000
=> bootm $loadaddr $ramdiskaddr $fdtaddr
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

5.5. Install / Upgrade

The software image install scripts provides a way to install or upgrade images on the devices. The software delieverables shown below should be located on the tftpserver to use it.

- uImage : LMP linux kernel image file (to be used on tftpboot)
- aTCA-N700.dtb: LMP linux dtb image file (to be used on tftpboot)
- rootfs.ext2.gz.uboot: LMP linux rootfs image file (to be used on tftpboot)
- fsl-rfs-full.tgz: LMP linux rootfs image file (to be installed on SATA)
- vendor-drops.tgz: LMP linux application archives file (to be installed on SATA)
- vmlinux.64: NP linux kernel image(to be installed on NAND)
- rootfs.cpio.gz: NP linux RFS image(to be installed on NAND)
- ipmc.tgz: IPMC firmware image(to be installed on IPMC)

From chapter 5.5.1 to 5.5.3, ramdisk boot-up sequence is commonly used.

5.5.1. Installing LMP

LMP boot-up using the ramdisk image and lmp-installer script on the ramdisk image should be used to install all LMP deliverables (U-Boot, kernel, dtb, and rootfs). The script considers redundant parts of the each device. All primary and secondary part can be installed or upgraded using this script.

Run tftpboot on lmp U-Boot, like below.

```
=> setenv tftp_bootfile uImage
=> setenv tftp_fdtfile aTCA-N700.dtb
=> setenv tftp_ramdiskfile rootfs.ext2.gz.uboot
=> setenv ipaddr 192.168.35.2
=> setenv serverip 192.168.1.213
=> setenv tftpboot 'tftp $loadaddr $tftp_bootfile; tftp $fdtaddr
$tftp_fdtfile; tftp $ramdiskaddr $tftp_ramdiskfile; run ramdiskargs; run
bootrfs;'
=> run tftpboot
```

After LMP Linux boot-up using ramdisk image is completed, edit and run lmp-installer on linux, as shown below. Ipaddr / serverip need to be changed according to your environment. lmp-installer downloads all image files from tftpserver, and install those to proper locations.

```
root@aTCA-N700:~# vi /sbin/lmp-installer
#!/bin/sh

#usage:                                                    lmp-installer
<LMP_IP><TFTPSERVER_IP><BASEDIR><VENDOR_DROPS><RFS>
#ex)  /lmp-installer  192.168.12.34  192.168.1.213  /media/ram  vendor-
drops.tgz rfs.

ipaddr=192.168.37.2
serverip=192.168.1.213
filebase=/media/ram
vendor=vendor-drops.tgz
rfs=fsl-rfs-full.tgz
...
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
root@aTCA-N700:~# lmp-installer
…

root@aTCA-N700:~# reboot
```

Boot-up with new U-Boot image, U-Boot environment variables should be updated, as shown below.

```
cpld flashctl lmp 0
setenv bootdelay 3
setenv bootcmd "ipmi bmc 100 1; run satabootA"
saveenv

cpld flashctl lmp 1
setenv bootcmd "ipmi bmc 100 1; run satabootB"
saveenv
reset
```

5.5.2. Installing NP

LMP boot-up using ramdisk image and np-installer script on the ramdisk image should be used to install all NP deliverables (U-Boot, kernel, and rootfs). The script installs NP deliverables using a PCIe channel.

Run tftpboot on lmp U-Boot, like below. This is same as the installing LMP procedure (previous chapter).

```
=> setenv tftp_bootfile uImage
=> setenv tftp_fdtfile aTCA-N700.dtb
=> setenv tftp_ramdiskfile rootfs.ext2.gz.uboot
=> setenv ipaddr 192.168.35.2
=> setenv serverip 192.168.1.213
=> setenv tftpboot 'tftp $loadaddr $tftp_bootfile; tftp $fdtaddr
$tftp_fdtfile; tftp $ramdiskaddr $tftp_ramdiskfile; run ramdiskargs; run
bootrfs;'
=> run tftpboot
```

After LMP Linux boot-up using ramdisk image is completed, edit and run np-installer on linux, as shown below. The MAC address for each management port of NP process should be provided.

```
root@aTCA-N700:~# vi /sbin/np-installer

#!/bin/sh

#usage:                                                    np-installer
<LMP_IP><TFTPSERVER_IP><BASEDIR><UBOOT><LINUX><RFS><NPA_MACBASE><NPB_MAC
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
BASE>
#ex) /np-installer 192.168.12.34 192.168.1.213 /media/ram oct-U-Boot.bin
vmlinux.64 rootfs.cpio.gz

ipaddr=192.168.37.2
serverip=192.168.1.213
filebase=/tftpboot
uboot=oct-U-Boot.bin
linux=vmlinux.64
rfs=rootfs.cpio.gz
#npamac=00:30:64:0F:5B:19 # board#5
#npbmac=00:30:64:0F:5B:1E # board#5
npamac=00:30:64:0F:5B:33
npbmac=00:30:64:0F:5B:38
...
```

```
root@aTCA-N700:~# np-installer
```

5.5.3. Installing IPMC

LMP boot-up using ramdisk image and ipmc-installer script on the ramdisk image should be used to install IPMC image.

Run tftpboot on lmp U-Boot, like below.

```
=> setenv tftp_bootfile ulmage
=> setenv tftp_fdtfile aTCA-N700.dtb
=> setenv tftp_ramdiskfile rootfs.ext2.gz.uboot
=> setenv ipaddr 192.168.35.2
=> setenv serverip 192.168.1.213
=> setenv tftpboot 'tftp $loadaddr $tftp_bootfile; tftp $fdtaddr
$tftp_fdtfile; tftp $ramdiskaddr $tftp_ramdiskfile; run ramdiskargs; run
bootrfs;'
=> run tftpboot
```

After LMP Linux boot-up using ramdisk image is completed, edit and run lmp-installer on linux, as shown below.

Run tftpboot on lmp U-Boot, like below.

```
root@aTCA-N700:~# vi /sbin/ipmc-installer
```

```
#!/bin/sh

#usage: ipmc-installer <LMP_IP><TFTPSERVER_IP><BASEDIR><IPMCPKG>
#ex) /ipmc-installer 192.168.12.34 192.168.1.213 /media/ram ipmc.tgz

ipaddr=192.168.32.2
serverip=192.168.1.213
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
filebase=/media/ram
ipmcpkg=ipmc.tgz
...


root@aTCA-N700:~# ipmc-installer
... (board reboot automatically)
```

5.5.4. Upgrade LMP U-Boot only

To replace former U-Boot image on the first and the secondary SPI-flash, U-Boot commands and environmental variables can be used as shown below. This can be used upgrading U-Boot only.

```
=> setenv ipaddr 192.168.95.67                    -> set my ip address
=> setenv serverip 192.168.0.142                    -> set tftp server ip
address
=> setenv netmask 255.255.0.0                     -> set netmask
=> setenv ethact FM1@DTSEC5                        -> set ethernet interface
to be used

=> tftp 0x500000 new-U-Boot                         -> download U-Boot image
from tftpserver
=> mw.b 0xe8000003 0x0                              -> select the first SPI-
flash
=> sf probe 0
=> sf erase 0 0xa0000
=> sf write 0x500000 0 0xa0000                      -> write U-Boot image to
the first SPI-flash

=> mw.b 0xe8000003 0x80                             -> select the secondary
SPI-flash
=> sf probe 0
=> sf erase 0 0xa0000
=> sf write 0x500000 0 0xa0000
=> reset

=> cpld flashctl lmp 0
=> setenv bootdelay 3
=> setenv bootcmd "ipmi bmc 100 1; run satabootA"     -> change U-Boot
env.variables for the first U-Boot
=> saveenv

=> cpld flashctl lmp 1
=> setenv bootcmd "ipmi bmc 100 1; run satabootB"  -> change U-Boot
env.variables for the secondary U-Boot
=> saveenv
```


LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

6. Hardware Diagnostics

The aTCA-N700 Software Package provides the basic functionalities to test the devices on the board. Table 6-1 illustrates the overview of the device testing.

Table 6-1 Devices and Test

ID	Description	Host	Test Description	Tools
LMP_MEM	DRAM for LMP	LMP	write/read/compare	Memtester (open source)
FABRIC_SWITCH	multilayer switch	LMP	BCM re-initialization	restarting bcm.user and check log message
BASE_SWITCH	base switch	LMP	BCM re-initialization	restarting bcm.user and check log message
CPLD	CPLD for system control	LMP	write/read/compare CPLD test register	nbm-cpld
MGMT_PHY	PHY for mgmt Ethernet port of LMP	LMP	backup/write/read/compare/restore test register	nbm-mdio
BASE_PHY	PHY for base Ethernet port of LMP (btw. LMP & BASE_SWITCH)	LMP	backup/write/read/compare/restore test register	nbm-mdio
LMP_EEPROM	mac addresses for LMP	LMP	write/read/compare	i2cset/i2cget
RTC	Real Time Cloc	LMP	time elpased	hwclock
PWR_CTRL	power control for FABRIC_SWITCH	LMP	write/read/compare	i2cset/i2cget
FRONT_SFPxx	SFP(+) modules (xx: 1~4)	LMP	read test register	i2cset/i2cget
FRONT_IOBUF	Front io BUF	LMP	detection on i2c bus	i2cprobe
FRONT_I2C_HUB	Front i2c switch	LMP	detection on i2c bus	i2cprobe
RTM_SFPxx	SFP(+) modules (xx: 1~12)	LMP	read test register	i2cset/i2cget
RTM_RETIMERx	RTM retimers (xx: 1~6)	LMP	backup/write/read/compare/restore test register	i2cset/i2cget
RTM_IOBUFxx	RTM io BUF (xx: 1~4)	LMP	detection on i2c bus	i2cprobe
RTM_I2C_HUBxx	RTM i2c switch (xx: 1~4)	LMP	detection on i2c bus	i2cprobe
SPI_FLASH1 SPI_FLASH2	boot flash for U-Boot	LMP	backup/write/read/compare/restore data on flash(reserved area)	flashcp, flash_erase, cp, diff
CHIP_SSD1 CHIP_SSD2	OS, data flash for linux	LMP	file write/read/compare/remove	cp, rm, diff
IPMC	IPMC	LMP	*ipmitool*	*ipmitool*
OCTEON_R OCTEON_L	right octeon left octeon	LMP	backup/write/read/compare/restore test register	oct-remote-csr

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

ID	Description	Host	Test Description	Tools
DIMM_OCTEON_R **DIMM_OCTEON_L**	DIMM set of OCTEON_R DIMM set of OCTEON_L	NP	write/read/compare	Memtester (open source)
NOR_FLASH_R_1 **NOR_FLASH_R_2** **NOR_FLASH_L_1** **NOR_FLASH_L_2**	NorFlash1 of OCTEON_R NorFlash2 of OCTEON_R NorFlash1 of OCTEON_L NorFlash2 of OCTEON_L	NP	backup/write/read/compare/restore data on flash(reserved area)	flashcp, flash_erase, cp, diff
NAND_FLASH_R_1 **NAND_FLASH_R_2** **NAND_FLASH_L_1** **NAND_FLASH_L_2**	NandFlash1 of OCTEON_R NandFlash2 of OCTEON_R NandFlash1 of OCTEON_L NandFlash2 of OCTEON_L	NP	backup/write/read/compare/restore data on flash(reserved area)	flashcp, flash_erase, cp, diff

6.1. Logging

Linux kernel and system S/W like ipmievd and nlnid use syslog mechanism to save their log data to a file.
The detail syslog information is as below:

- syslog version: busybox v1.17.1 (from LMP SDK)
- syslog config (/etc/syslog.confg)

```
DESTINATION="file"              # log destinations (buffer file remote)
MARKINT=20                      # interval between --mark-- entries
[min]
REDUCE=no                       # reduced-size logging
BUFFERSIZE=64                   # buffer: size of circular buffer [kByte]
LOGFILE=/var/NVMA/messages      # file: where to log
   ⇨ This file is located on CHIP_SSD. Refer to Table 6 1 VM Map.
REMOTE=loghost:514              # remote: where to log
FOREGROUND=no                   # run in foreground (don't use!)
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

6.2. POST

The Power-On-Self-Test (POST) is performed during bootloader booting up. The POST includes the major devices on the board. The POST result is printed out on the monitoring port at the end of the bootloader booting. Table 6-2 summarizes the devices and testing.
(The blade without the NAND flash skips the test; CONFIG_SYS_POST_NAND_FPGA test and CONFIG_SYS_POST_NAND_FLASH test.)

Table 6-2 Devices covered by the POST

No.	POST device ID	Description
1	CONFIG_SYS_POST_MEMORY	LMP DDR DRAM memory testing. The specified memory region is tested with Read/Write.
2	CONFIG_SYS_POST_IPMC	Sample IPM commands are executed to see if it is operational.
3	CONFIG_SYS_POST_SPI_EEPROM	spi flash probe is executed to check if SPI eeprom is operational.
4	CONFIG_SYS_POST_MGMT_PHY	The PHY access through MII is performed to check if it is operational.
5	CONFIG_SYS_POST_RTC	RTC device is verified for time of the day operation.
6	CONFIG_SYS_POST_EEPROM	The EEPROM is checked by I2C probe to see if it is operational.
7	CONFIG_SYS_POST_NAND_FPGA	The NAND fpga is checked by read/write operation to check if it is operational.
8	CONFIG_SYS_POST_NAND_FLASH	The post of NAND flash will pass if init process passed.
9	CONFIG_SYS_POST_CPLD	The CPLD is checked by read/write operation to check if it is operational.
10	CONFIG_SYS_POST_SWITCH_BASE	The SWITCH (base) access through SPI is performed to check if it is operational.
11	CONFIG_SYS_POST_SWITCH_FAB	The SWITCH (fabric) access through PCI is performed to check if it is operational.
12	CONFIG_SYS_POST_POWER_CTRL	The Power controller is checked by I2C probe to see if it is operational.
13	CONFIG_SYS_POST_SYNC_IN	The SYNC (in) is checked by I2C probe to see if it is operational.
14	CONFIG_SYS_POST_NP_TOP	The NP (TOP) access through PCI is performed to check if it is operational.
15	CONFIG_SYS_POST_NP_BOT	The NP (BOT) access through PCI is performed to check if it is operational.
16	CONFIG_SYS_POST_I2C_MUX_0	The I2C Muxes are checked by I2C probe to see if it is operational.
17	CONFIG_SYS_POST_I2C_MUX_1	The I2C Muxes are checked by I2C probe to see if it is operational.
18	CONFIG_SYS_POST_I2C_MUX_2	The I2C Muxes are checked by I2C probe to see if it is operational.
19	CONFIG_SYS_POST_IO2I2C_INS	IO buffer device is verified by probing.
20	CONFIG_SYS_POST_IO2I2C_TXF	IO buffer device is verified by probing.
21	CONFIG_SYS_POST_IO2I2C_LOS	IO buffer device is verified by probing.
22	CONFIG_SYS_POST_IO2I2C_TXDIS	IO buffer device is verified by probing.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

6.3. Base Switch

The base port switch is managed by “robo” command. The usage is as the following.

```
=> robo
robo - Base Switch specific commands

Usage:
robo [command] ...
robo w show                      - show port stats of base switch
robo init                      - init forwarding function of base switch
robo close                     - close forwarding function of base switch
robo enable|disable <port>      - enable/disable base switch port(IMP port: >5)
robo sread <page> <reg> <len>   - read base switch register
robo swrite <page> <reg> <data> - write base switch register
```

The base switch port status is monitored with a command in the following example.

```
=> robo show
Switch State: Managed (Forwarding)

Port Name  STP    Link       Speed    Tx(U)    TX(B)    Tx(M)    Rx(U)    RX(B)    Rx(M)
=============================================================================================
0    IPMC   DIS    Up       100(H)        0        0        0        0        0        0
1    LMP    NOSTP  Up      1000(F)        0       29       55        0        0        0
2    Front  NOSTP  Up      1000(F)        0        0        0        0    15366     4639
3    BaseB  DIS    Up      1000(F)        0       14       32        0        0        0
4    BaseA  DIS    Up      1000(F)        0        0        0        0        0        0
5    OctB   DIS    Down    1000(H)        0        0        0        0        0        0
-    OctA   DIS    Up      1000(F)        0        0        0        0        0        0
```

The “robo” command in LMP U-Boot is equivalent to the command “nbm-robo” on the LMP linux.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

6.4. IPMI commands

The Intelligent Platform Management Interface (IPMI) is provided as a tool to monitor and management the blade. The commands are in available in the bootloader environment, which are summarized in Table 6-3.

Table 6-3 IPMI command list

Parameter	Description
reset	reset board by IPMC cmd
sol-mac <MAC_ADDR>	set mac address for sol
sol-mac	get mac address for sol
get-last-rst	get last reset reason
bmc <time> <act>	enable BMC watchdog with timer
active-flash	display active-flash number
switch-active	change active-flash
get-addr-info	get address info
get-msg-add-info	get msg address info
get-msg-shelf-addr	get msg shelf addr
get-device-id	get device information
auto-boot-switch	get auto boot status
auto-boot-switch (0|1)	set auto boot status(0 or 1)

The following is an example for an IPMI command which performs a BMC watchdog reset.

```
=> ipmi get-addr-info
PICMGid:         0x00
HWAddr:          0x47
IPMB_0Addr:      0x8e
Reserved:        0xff
FRU_dev_info:    0x00
Site No:         0x00
Sitetype:        0x00

=> ipmi get-msg-add-info
chNo:          0x00
netFn:         0xb6
chksum_0:      0x4a
rsSA:          0x20
rqSeq:         0x00
cmdCode:       0x01
compCode:      0x00
PICMGid:       0x00
hwAddr:        0x47
rqSA:          0x8e
fixedData_0:   0xff
FRUid:         0x00
siteNo:        0x04
siteType:      0x00
chksum_1:      0x07
=> ipmi get-msg-shelf-addr
chNo:           0x00
netFn:          0xb6
chksum_0:       0x4a
rsSA:           0x20
rqSeq:          0x00
cmdCode:        0x02
compCode:       0x00
PICMGid:        0x00
ShelfAddr type: 0x03
ShelfAddrLen:   0x14
ShelfAddr[0]:  0x30
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
ShelfAddr[1]:  0x30
ShelfAddr[2]:  0x30
ShelfAddr[3]:  0x30
ShelfAddr[4]:  0x30
ShelfAddr[5]:  0x31
ShelfAddr[6]:  0x20
ShelfAddr[7]:  0x41
ShelfAddr[8]:  0x30
ShelfAddr[9]:  0x31
ShelfAddr[10]:  0x30
ShelfAddr[11]:  0x31
ShelfAddr[12]:  0x46
ShelfAddr[13]:  0x46
ShelfAddr[14]:  0x46
ShelfAddr[15]:  0x46
ShelfAddr[16]:  0x46
ShelfAddr[17]:  0x46
ShelfAddr[18]:  0x2e
ShelfAddr[19]:  0x2e

chksum_1:       0xc6

=> ipmi get-device-id
Device ID:              0x12
Device Revision:        0x80
Major Firmware Revision: 0x01
Minor Firmware Revision: 0x00
IMPI Version:           0x02
Device Support Value:   0x2d
Manufacturer ID:        0x00 5f 13
Product ID:             0x07 00

=> ipmi get-addr-info
PICMGid:        0x00
HWAddr:         0x47
IPMB_0Addr:     0x8e
Reserved:       0xff
FRU_dev_info:   0x00
Site No:        0x00
Sitetype:       0x00
```

6.5. Default Environment Variables

Various environment variables are used in U-Boot, which are also available for users to configure. Shown below are the examples.

```
baudrate=115200
bootcmd=run nandboot
bootdelay=10
bootfile=packages/aTCA-N700T/linux.latest.bin
bootrfs=bootm $loadaddr $ramdiskaddr $fdtaddr
c=ffe
consoledev=ttyS0
ethact=FM1@DTSEC4
ethprime=FM1@DTSEC4
fdtaddr=c00000
fdtfile=packages/aTCA-N700T/narinet_ntpm8000.latest.dtb
fman_ucode=0x7fe3fe88
hwconfig=fsl_ddr:ctlr_intlv=cacheline
loadaddr=1000000
mtdids=nand0=A,nand1=B
mtdparts=mtdparts=A:4m(dtbA),8m(kernelA),100m(rfsA);B:4m(dtbB),8m(kernelB),100m(rfsB)
nandboot=ipmi en_bmc 120 1; run nibA; run normalargs; run bootrfs; ipmi di_bmc;ipmi
switch-active
nandboot_nobmc=run nibA; run normalargs; run bootrfs
netdev=eth0
nfsargs=setenv bootargs root=/dev/nfs rw nfsroot=$serverip:$rootpath
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
ip=$ipaddr:$serverip:$gatewayip:$netmask:$hostname:$netdev:off
console=$consoledev,$baudrate $othbootargs
nfsboot=run nfsargs; tftp $loadaddr $bootfile; tftp $fdtaddr $fdtfile; bootm $loadaddr -
$fdtaddr
nibA=nboot $loadaddr kernelA; nboot $ramdiskaddr rfsA; nboot $fdtaddr dtbA
nibB=nboot $loadaddr kernelB; nboot $ramdiskaddr rfsB; nboot $fdtaddr dtbB
normalargs=setenv bootargs root=/dev/ram rw console=$consoledev,$baudrate root=/dev/ram
rw ramdisk_size=400000 post_res=$post_result
nukeenv=sf probe 0;sf erase 0xa0000 0x10000
partition=nand0,0
post_result=fff00000
ramdiskaddr=2000000
ramdiskfile=packages/aTCA-N700T/rootfs.latest.bin
rootpath=/tftpboot/nfsroot
sdboot=mmc dev;fatload mmc0 0:1 $loadaddr uimage;fatload mmc0 0:1 $ramdiskaddr
rootfs.ext2.gz.uboot; fatload mmc0 0:1 $fdtaddr aTCA-N700.dtb;run normalargs; run
bootrfs
stderr=serial
stdin=serial
stdout=serial
syncenv=ntpm8000 sf-device 1; saveenv;ntpm8000 sf-device 0
tftpboot=run tib_1; run normalargs; run bootrfs
tib_1=tftp $loadaddr $bootfile;tftp $fdtaddr $fdtfile; tftp $ramdiskaddr $ramdiskfile
```

The environment variables for networking interfaces are as shown in Table 6-4.

Table 6-4 Networking environment variables

Env	Description
Serverip	tftp server ip address for tftpboot command
Ipaddr	ip address of the blade
netmask	Network mask of the blade
ethact	Ethernet device used for current ipaddr
ethprime	Highest priority Ethernet device when multiple devices are present. At the blade rebooting, ethact is configured based on ethprime.

The blade without the NAND flash uses sdboot bootcmd.

sdboot=mmc dev;fatload mmc0 0:1 $loadaddr uimage;fatload mmc0 0:1 $ramdiskaddr rootfs.ext2.gz.uboot; fatload mmc0 0:1 $fdtaddr aTCA-N700.dtb;run normalargs; run bootrfs

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table 6-5 sdboot parameters

No	command	Description
1	mmc dev	Activate the mmc deivce.
2	fatload mmc0 0:1 $loadaddr uimage	Load kernel image at mmc0 interface 0 device 1 partition.
3	fatload mmc0 0:1 $ramdiskaddr rootfs.ext2.gz.uboot	Load rootfile system at mmc0 interface 0 device 1 partition.
4	fatload mmc0 0:1 $fdtaddr aTCA-N700.dtb	Load dtb image at mmc0 interface 0 device 1 partition.
5	run normalargs	Set Environment. setenv bootargs root=/dev/ram rw console=$consoledev,$baudrate root=/dev/ram rw ramdisk_size=400000 post_res=$post_result
6	run bootrfs	Execute Linux booting. bootm $loadaddr $ramdiskaddr $fdtaddr

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

7. OCTEON Host Utilities

The OCTEON-SDK provides OCTEON host utilities which apply only to OCTEON or Intel HOST architecture. For the LMP (Freescale) porting, however, the OCTEON-SDK sources (OCTEON-SDK/host/remote-utils) can be accompanied into the LMP source tree for cross compilation. There may be limitations in using assembler based modules on the LMP.

SW7 configuration selects PCIe operation mode for each OCTEON processor. “F_OCTEON Remote Control select Mode” or “S_OCTEON Remote Control select Mode” should be selected to use this utility properly. Regarding SW7, please refer to the aTCA-N700 Hardware User Manual, Section 4.1.2.

These utilities support the remote protocol using two PCI channels which are connected to each OCTEON Processor. To select an OCTEON processor to be controlled by this utility, an environmental variable needs to be setup before using the utilities, as shown below

```
[ to select OCTEON TOP ]
# export OCTEON_REMOTE_PROTOCOL=PCI:0

[ to select OCTEON BOTTOM ]
# export OCTEON_REMOTE_PROTOCOL=PCI:0
```

Table 7-1 OCTEON Host Utilities

Name	Description
oct-remote-boot	Boot OCTEON and load bootloader over remote connection (PCI) Usage: oct-remote-boot [--memonly] [--board=board_name] [bootloader filename]
oct-remote-bootcmd	Give command to bootloader running on OCTEON in PCI slot. Command must be double quoted if it contains whitespace. Usage: oct-remote-bootcmd "command"
oct-remote-console	Connect to a remote console. Usage: oct-remote-console [--noraw] <CONSOLE_NUMBER>
oct-remote-core	Display the state of an OCTEON core. Usage: oct-remote-core <core>
oct-remote-csr	Read or write a CSR by name. Usage: oct-remote-csr CSR [value] [CSR [value] ...]
oct-remote-ddr	OCTEON DDR2 memory test over a remote interface Note: Requires use of oct-*-boot to configure memory controller Usage: oct-remote-ddr <Mbytes to test>
oct-remote-ddr-tune	Perform DDR tuning tests on remote Octeon.
oct-remote-load	Load the file into OCTEON memory at supplied address Usage: oct-remote-load <address> <file>

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

oct-remote-memory	Read or write bytes at a physical memory address. Usage: oct-remote-memory [-w width] [-c count] address [value]
oct-remote-pow	The state of the OCTEON POW can be captured and displayed through the remote interface
oct-remote-profile	Profiling of any code running on OCTEON can be performed
oct-remote-qlm	Read and display all QLM jtag settings.
oct-remote-reset	Usage: oct-remote-reset [options] Options: --wait=n: Wait n seconds before polling for bootloader completion. If 0, then don't wait at all. (Default is 4 seconds) --h or -h: print this usage message
oct-remote-save	Read OCTEON DRAM and write contents to a file. Usage: oct-remote-save <address> <length> <file>
oct-remote-tra	L2 transactions and CSR accesses can be traced

Please find more information about these utilities in the OCTEON SDK documentation.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

8. Switch Management

The Broadcom SDK is a part of the LMP software suite. Shown in Figure 8-1 are the Broadcom switch S/W package components which are associated with the two on-board switches, the BCM56842 fabric switch and the BCM53115 base switch. The configuration interfaces are PCI-e for BCM56842 and SPI for the BCM53115. Both interfaces are connected to the LMP.

The bcm.user is a user interface shell program which can be accessed to configure the switches. The kernel modules *linux_user_bde.ko, and linux_kernel_bde.ko* simply provide switch hardware access points and most of the functionalities are accompanied to the *bcm.user*. The aTCA-N700 S/W package provides the *user mode* BCM-SDK only. The *kernel mode* BCM-SDK, where most of the functionalities are included in the kernel with *bcm.user* for a simple user interface, is not supported in the aTCA-N700.

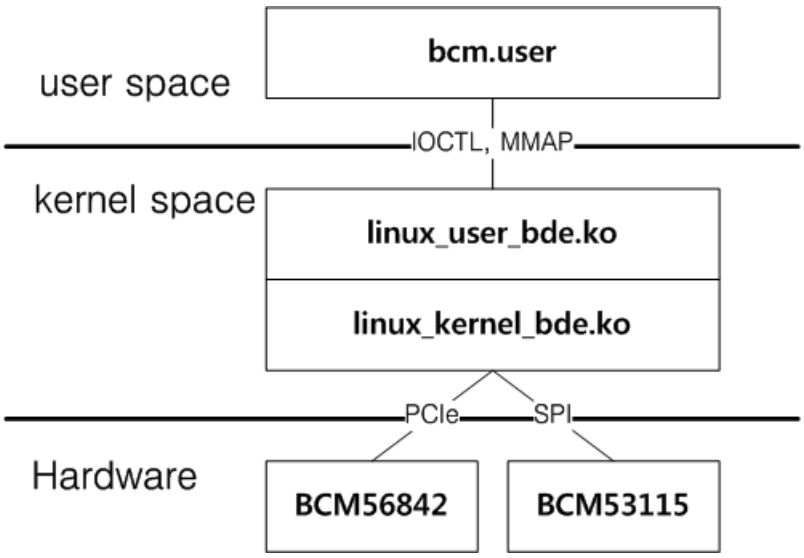


Figure 8-1 Broadcom SDK components

The Fabric Switch (BCM56842) can also be paired with the ADLINK Broadcom Switch Manager software for management. The ADLINK Broadcom Switch Manager includes functions of port configuration, customized port mirror, VLAN setting, trunk group, ACL, static route, and management API with a more user-friendly CLI to enable more straightforward API development.

The details on the ADLINK Broadcom Switch Manager release software can be found in the ADLINK Broadcom Switch Manager Configuration Guide and will not be covered in this document.

Under default settings, the system will boot to the BCM SDK shell. Please contact your ADLINK representative to order the ADLINK Broadcom Switch Manager software.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

8.1. Networking

The network interfaces are detailed in Figure 8-2 and Figure 8-3. The RTM1 is an optional installment which has twelve of SFP+ (10G) ports, whereas the RTM2 has two QSFP+ ports (40G) and four SFP+ ports. The front SFP+ ports are present, but are omitted in the figures.

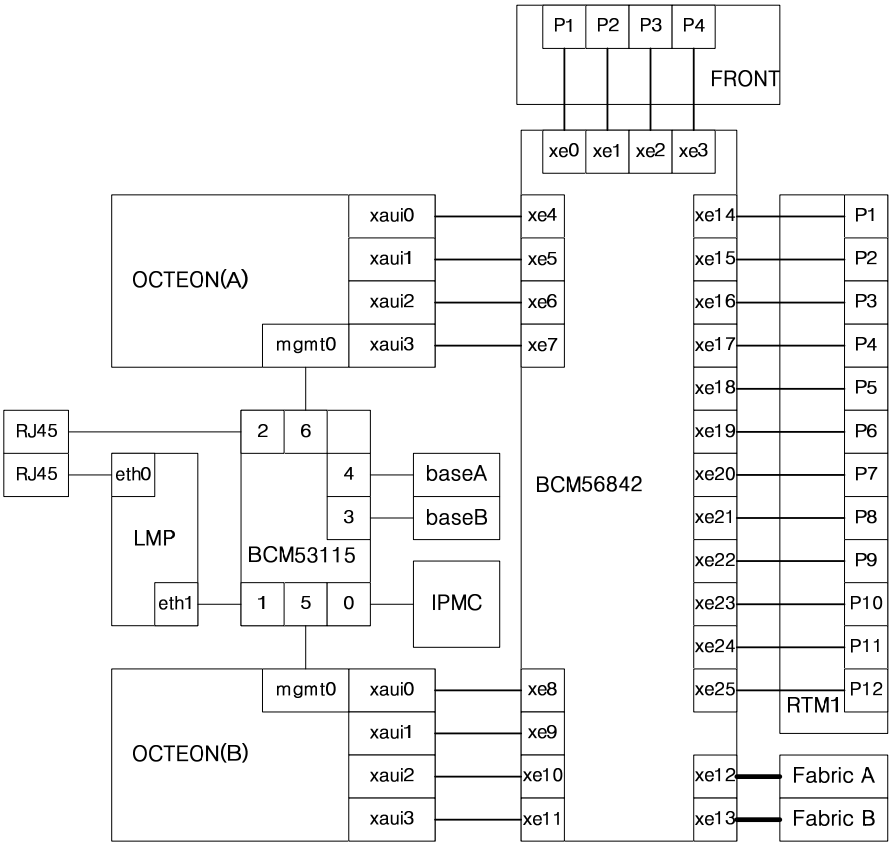


Figure 8-2 Broadcom device networking interfaces (RTM1)

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

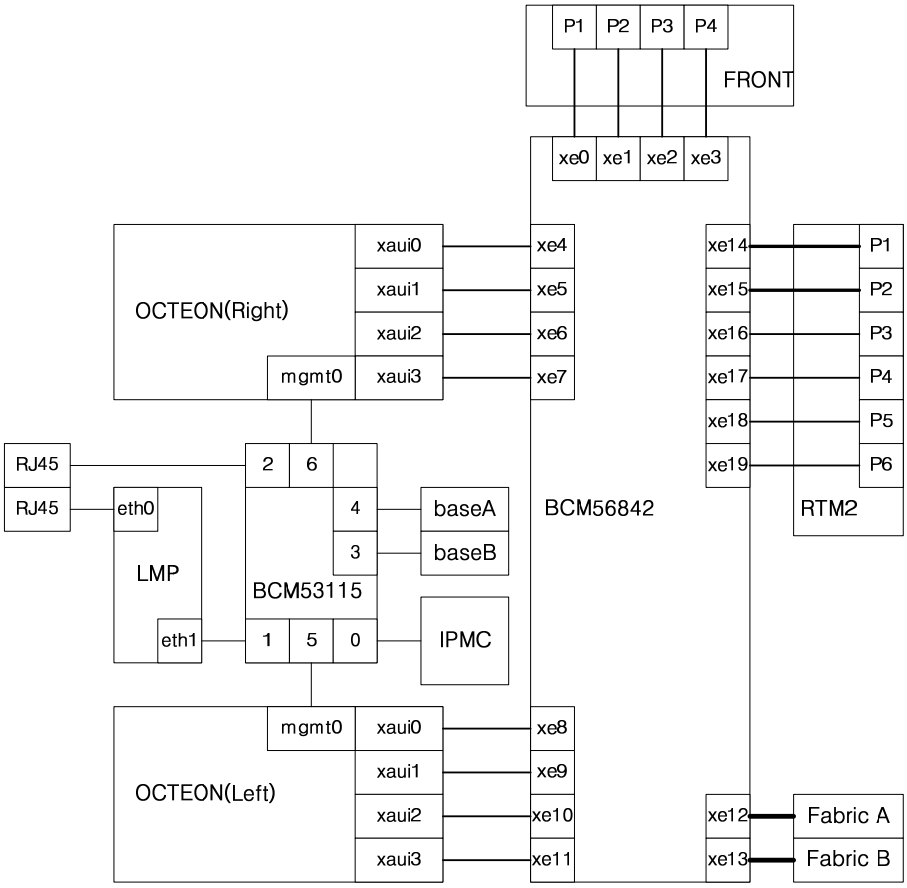


Figure 8-3 Broadcom device networking interfaces (RTM2)

8.1.1. Port Mapping

Bcm.user uses port names as described in Table 8-1 without an RTM, and Table 8-2 with an RTM. The front blade only configuration is according to */usr/local/bcm/config.bcm.config*. The configuration with an RTM is according to */usr/local/bcm/config.bcm.rtm config*.

Table 8-1 bcm.user port assignment with RTM1

Name	speed (Mbps)	Interface Type	Auto nego	Description
xe0	1000 10000	GMII SFI	off	Front port 1
xe1	1000 10000	GMII SFI	off	Front port 2
xe2	1000 10000	GMII SFI	off	Front port 3
xe3	1000 10000	GMII SFI	off	Front port 4
xe4	10000	XAUI	off	NP(TOP) Port 0
xe5	10000	XAUI	off	NP(TOP) Port 1
xe6	10000	RXAUI	off	NP(TOP) Port 2
xe7	10000	RXAUI	off	NP(TOP) Port 3
xe8	10000	XAUI	off	NP(BOT) Port 0
xe9	10000	XAUI	off	NP(BOT) Port 1
xe10	10000	RXAUI	off	NP(BOT) Port 2
xe11	10000	RXAUI	off	NP(BOT) Port 3
xe12	1000	GMII	on	FabricA

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

	10000 40000	XGMII(XAUI) KR4		
xe13	1000 10000 40000	GMII XGMII(XAUI) KR4	on	FabricB
xe14	1000 10000	GMII SFI	off	Rear Port 1
xe15	1000 10000	GMII SFI	off	Rear Port 2
xe16	1000 10000	GMII SFI	off	Rear Port 3
xe17	1000 10000	GMII SFI	off	Rear Port 4
xe18	1000 10000	GMII SFI	off	Rear Port 5
xe19	1000 10000	GMII SFI	off	Rear Port 6
xe20	1000 10000	GMII SFI	off	Rear Port 7
xe21	1000 10000	GMII SFI	off	Rear Port 8
xe22	1000 10000	GMII SFI	off	Rear Port 9
xe23	1000 10000	GMII SFI	off	Rear Port 10
xe24	1000 10000	GMII SFI	off	Rear Port 11
xe25	1000 10000	GMII SFI	off	Rear Port 12

Table 8-2 bcm.user port assignment with an RTM2

Name	speed (Mbps)	Interface Type	Auto nego	Description
xe0	1000 10000	GMII SFI	off	Front port 1
xe1	1000 10000	GMII SFI	off	Front port 2
xe2	1000 10000	GMII SFI	off	Front port 3
xe3	1000 10000	GMII SFI	off	Front port 4
xe4	10000	XAUI	off	NP(TOP) Port 0
xe5	10000	XAUI	off	NP(TOP) Port 1
xe6	10000	RXAUI	off	NP(TOP) Port 2
xe7	10000	RXAUI	off	NP(TOP) Port 3
xe8	10000	XAUI	off	NP(BOT) Port 0
xe9	10000	XAUI	off	NP(BOT) Port 1
xe10	10000	RXAUI	off	NP(BOT) Port 2
xe11	10000	RXAUI	off	NP(BOT) Port 3
xe12	1000 10000 40000	GMII XGMII(XAUI) KR4	on	FabricA
xe13	1000 10000 40000	GMII XGMII(XAUI) KR4	on	FabricB

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

xe14	40000	XLAUI	off	Rear Port 1
xe15	40000	XLAUI	off	Rear Port 2
xe16	1000 10000	GMII SFI	off	Rear Port 3
xe17	1000 10000	GMII SFI	off	Rear Port 4
xe18	1000 10000	GMII SFI	off	Rear Port 5
xe19	1000 10000	GMII SFI	off	Rear Port 6

8.2. Initialization

During booting up, /etc/init.d/bcm.sh script is executed and the switch software is launched.

Here are lists of actions performed by /etc/init.d/bcm.sh.

1. Loading kernel modules (linux_kernel_bde, linux_user_bde)
2. Making device node file (/dev/linux_user_bde)
3. It makes CWD (current work directory) to be /usr/local/bcm.
4. Running bcm.user using nbm-netserve.
5. Checking and waiting done using log file of nbm-netserve.

Once activated, *bcm.user* can be accessed in the following way.

```
[root@N700]$ telnet localhost 9895
```

This access is performed via *nbm-netserve* as illustrated in Figure 5-1.

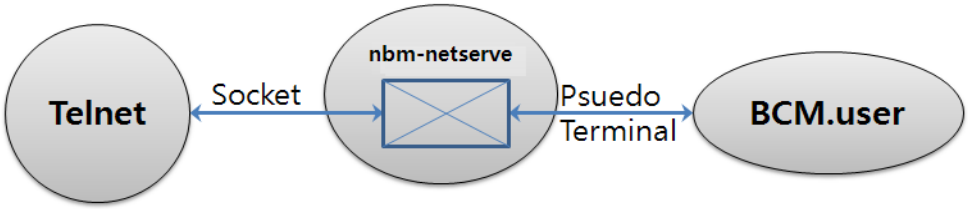


Figure 8-4 Access to bcm.user

Here's an example for bcm.user.

```
[root@N700]$ telnet localhost 9895
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

To exit to Shell mode only to terminate telnet session, use "quit".

```
BCM.0> quit
```

8.3. bcm.user

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

This is a shell environment provided by BCM SDK to configure the fabric Switch from Broadcom for various functions. Refer to the Broadcom documents for the details.

In the following sections, we describe aTCA-N700 dependant configurations and their commands.

8.3.1. Port configuration

Users can set the attributes of a port such as speed, auto-negotiation and enable by 'port' command.
'port' command usage is as shown below.

```
Usage (PORT): Parameters: <pbmp> [[ENCap=IEEE|HIGIG|B5632|HIGIG2]
         [AutoNeg=on|off] [ADVert=<portmode>]
         [LinkScan=on|off|hw|sw] [SPeed=10|100|1000] [FullDuplex=true|false]
         [TxPAUse=on|off] [RxPAUse=on|off] [STationADdr=<macaddr>]
         [LeaRN=<learnmode>] [DIScard=none|untag|all] [VlanFilter=<value>]
         [PRIOrity=<0-7>] [PortFilterMode=<value>]
         [PHymaster=<Master|Slave|Auto|None>] [Enable=<true|false>]
         [FrameMax=<value>]
         [MDIX=Auto|ForcedAuto|ForcedNormal|ForcedXover]
         [LoopBack=NONE|MAC|PHY]]
         | [[EEE] [ENable=<enable|disable>|TxIDleTime=<0-2560>|TxWakeTime=<0-
2560>|STats=<get>]]
         If only <ports> is specified, characteristics for that port are
         displayed. <ports> is a standard port bitmap (see "help pbmp").
         If AutoNeg is on, SPeed and DUPlex are the ADVERTISED MAX values.
         If AutoNeg is off, SPeed and DUPlex are the FORCED values.
         SPeed of zero indicates maximum speed.
         LinkScan enables automatic scanning for link changes with updating
         of MAC registers, and EPC_LINK (or equivalent)
         PAUse enables send/receive of pause frames in full duplex mode.
         <learnmode> is a numeric value controlling source lookup failure
         packets; it may include bit 0 to enable hardware L2 learn, bit 1
         to copy SLF packets to CPU, bit 2 to forward SLF packets.
         VlanFilter drops input packets that not tagged with a valid VLAN
         that contains the port. For XGS3, VlanFilter takes a value 0/1/2/3 where
         bit 0 turns on/off ingress filter and bit 1 turns on/off egress filter.
         PRIOrity sets the priority for untagged packets coming on this port.
         PortFilterMode takes a value 0/1/2 for mode A/B/C (see register manual).
         EEE: Energy Efficient Ethernet.
```

Some examples of port command are described in the following.

** Set port xe0 ~ xe3 speed to be 10G and turn off auto-negotiation.

```
BCM.0> port xe0-xe3 sp=10000 an=off
```

** Display the status of port xe0 ~ xe25.

```
BCM.0> ps xe0-xe3
          ena/  speed/ link auto    STP                lrn  inter   max  loop
     port link  duplex scan neg?   state   pause  discrd ops   face frame  back
      xe0  up    10G FD   SW  No   Forward  TX RX   None   FA    SFI 16356
      xe1  up    10G FD   SW  No   Forward  TX RX   None   FA    SFI 16356
      xe2  up    10G FD   SW  No   Forward  TX RX   None   FA    SFI 16356
      xe3  up    10G FD   SW  No   Forward  TX RX   None   FA    SFI 16356
```

** Disable port xe0.

```
BCM.0> port xe0 en=false

BCM.0> ps xe0
          ena/  speed/ link auto    STP                lrn  inter   max  loop
     port link  duplex scan neg?   state   pause  discrd ops   face frame  back
      xe0  !ena  10G FD   SW  No   Forward  TX RX   None   FA    SFI 16356
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

8.3.2. Vlan configuration

To configure vlan for ports, use the ‘vlan’ command.

```
Usage (VLAN): Usages:
          vlan create <id> [PortBitMap=<pbmp> UntagBitMap=<pbmp>]
                                          - Create a VLAN
          vlan destroy <id>                - Destroy a VLAN
          vlan clear                      - Destroy all VLANs
          vlan add <id> [PortBitMap=<pbmp> UntagBitMap=<pbmp>
                                          - Add port(s) to a VLAN
          vlan remove <id> [PortBitMap=<pbmp>] - Remove ports from a VLAN
          vlan MulticastFlood <id>  [Mode]     - Multicast flood setting
          vlan show                       - Display all VLANs
          vlan default [<id>]              - Show or set the default VLAN
```

Some examples of the *vlan* command are described in the following.

** Create Vlan 2and add port xe0, xe2 with untaggedoption and display the resultingvlan table.

```
BCM.0> vlan create 2 pbm=xe0,xe2 ubm=xe0,xe2

BCM.0> vlan show
vlan 1   ports cpu,xe (0x00000000000000000000000000000000ffffffff), untagged xe
(0x00000000000000000000000000000000fffffffe) MCAST_FLOOD_UNKNOWN
vlan 2   ports xe0,xe2 (0x0000000000000000000000000000000000000000a), untagged
xe0,xe2 (0x00000000000000000000000000000000000000a) MCAST_FLOOD_UNKNOWN
```

** Add port xe3 with tagged option in vlan 2.

```
BCM.0> vlan add 2 pbm=xe3
BCM.0> vlan show 2
vlan 2   ports xe0,xe2-xe3 (0x0000000000000000000000000000000000001a), untagged
xe0,xe2 (0x00000000000000000000000000000000000a) MCAST_FLOOD_UNKNOWN
```

** Remove port xe3 in vlan 2.

```
BCM.0> vlan remove 2 pbm=xe3
BCM.0> vlan show 2
vlan 2   ports xe0,xe2 (0x0000000000000000000000000000000000000a), untagged
xe0,xe2 (0x00000000000000000000000000000000000a) MCAST_FLOOD_UNKNOWN
```

** Remove vlan 2.

```
BCM.0> vlan destroy 2
BCM.0> vlan show
vlan 1   ports cpu,xe (0x00000000000000000000000000000000ffffffff), untagged xe
(0x00000000000000000000000000000000fffffffe) MCAST_FLOOD_UNKNOWN
```

8.3.3. Pvlan configuration

To set a default vlan for a port, use the ‘pvlan’ command.

```
pvlan show <pbmp>
      - Show PVLAN info for these ports.
pvlan set <pbmp> <vid>
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
      - Set default VLAN tag for port(s)
        Port bitmaps are read from the VTABLE entry for the VID.
        <vid> must have been created and all ports in <pbmp> must
        belong to that VLAN.
```

Some examples of the *pvlan* command are described in the following.

** Set port xe0, xe2 default vlan= 2.

```
BCM.0> pvlan xe0,xe2 2
```

** Display port xe0, xe2 default vlan information.

```
BCM.0> pvlan show xe0,xe2
Port xe0 default VLAN is 2
Port xe2 default VLAN is 2
```

8.3.4. Spanning-tree group configuration

For STG (Spanning-Tree Group) STG 1 contains all vlans and forwards all ports. STG usage is as shown below.

```
Usage (STG): Usages:
           stg create [<id>]           - Create a STG; optionally specify ID
           stg destroy <id>            - Destroy a STG
           stg show [<id>]             - List STG(s)
           stg add <id> <vlan_id> [...]     - Add VLAN(s) to a STG
           stg remove <id> <vlan_id> [...]  - Remove VLAN(s) from a STG
           stg stp                     - Get span tree state, all ports/STGs
           stg stp <id>                - Get span tree state of ports in STG
           stg stp <id> <pbmp> <state>  - Set span tree state of ports in STG
                                        (disable/block/listen/learn/forward)
           stg default [<id>]          - Show or set the default STG
```

Some examples of the *STG* command are described in the following.

** Create STG 2.

```
BCM.0> stg create 2
BCM.0> stg show 2
STG 2: contains 0 VLANs
```

** Add vlan 3 in STG 2.

```
BCM.0> stg add 2 3
BCM.0> stg show 2
STG 2: contains 1 VLAN (3)
```

** Configure span tree state to be *forward* for STG 2 port xe0, xe2.

```
BCM.0> stg stp 2 xe0,xe2 forward
BCM.0> stg stp 2
STG 2:
  Disable: xe1,xe3-xe34
  Forward: xe0,xe2
```

** Remove vlan 3 from STG 2.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
BCM.0> stg remove 2 3
BCM.0> stg show 2
STG 2: contains 0 VLANs
```

** Delete STG2.

```
BCM.0> stg destroy 2
BCM.0> stg show
STG 1: contains 3 VLANs (1,3-4)
Forward: xe
BCM.0>
```

8.3.5. Mac configuration

Port static mac information is configured by 'l2' command.

```
Usage (L2): Usages:
          l2 add [PortBitMap=<pbmp> or Port=<port>] [MACaddress=<mac>]
[Vlanid=<id>]
                [Trunk=true|false] [TrunkGroupId=<id>] [STatic=true|false]
                [SourceCosPriority=true|false] [DiscardSource=true|false]
                [DiscardDest]=true|false] [L3=true|false] [HIT=true|false]
                [Replace=true|false] [CPUmirror=true|false] [MIRror=true|false]
                [ReplacePriority=true|false] [RemoteLookup=true|false]
                [MacBlockPortBitMap=<pbmp>] [Group=<group>]
               - Add incrementing L2 addresses associated with port(s)
          l2 del [MACaddress=<mac>] [Count=<value>] [Vlanid=<id>]
               - Delete incrementing L2 address(s)
          l2 show [PortBitMap=<pbmp]
               - Show L2 addresses associated with port(s)
          l2 check [hit, valid, static]
               - Check full L2 table for valid entries
          l2 clear [Module=<n>] [Port=<port>] [MACaddress=<mac>]
                  [Vlanid=<id>] [TrunkGroupID=<id>] [Static=true|false]
               - Remove all L2 entries on the given module, module/port,
                  VLAN, or trunk group ID
```

Some examples of mac configuration are shown below.

** Set mac = '00:00:00:00:00:21' for port xe0 in L2 table vlan 2.

```
BCM.0> l2 add pbm=xe0 mac=0x21 v=2 static=true
```

** Display L2 table.

```
BCM.0> l2 show
mac=00:00:00:00:00:21 vlan=1 modid=0 port=1/xe0 Static ReplacePriority
```

** Remove mac '00:00:00:00:00:21' in L2 table.

```
BCM.0> l2 del mac=0x21 v=2
```

** Remove all mac information associated with vlan 1 in l2 table.

```
BCM.0> l2 clear v=1
L2: Deleting static and non-static addresses by VLAN
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

8.3.6. Port interframe gap configuration

'ipg' command is used to configure interfame gap for each port speed.

```
Usage (IPG): Parameters: [PortBitMap=<pbmp>] [SPeed=10|100|1000|2500|10000]
         [FullDuplex=true|false] [Gap=<int>]
Set the IPG register values to the specified values when the
         MAC is programmed after autonegotiation OR when the port is
         into the specific mode.  If no args are given, displays settings.
```

ipg examples are as the following.

** Set port xe0-xe21 10G (Full Duplex) interframe gap= 80.

```
BCM.0> ipg pbm=xe0-xe21 sp=10000 fd=true
           10HD    10FD   100HD   100FD  1000HD  1000FD  2500HD  2500FD 10000FD
STRETCH
xe0                                                                      80       0
xe1                                                                      80       0
xe2                                                                      80       0
xe3                                                                      80       0
xe4                                                                      80       0
xe5                                                                      80       0
xe6                                                                      80       0
xe7                                                                      80       0
xe8                                                                      80       0
xe9                                                                      80       0
xe10                                                                     80       0
xe11                                                                     80       0
xe12                                                                     80       0
xe13                                                                     80       0
xe14                                                                     80       0
xe15                                                                     80       0
xe16                                                                     80       0
xe17                                                                     80       0
xe18                                                                     80       0
xe19                                                                     80       0
xe20                                                                     80       0
xe21                                                                     80       0
```

8.3.7. Port statistics

Port statistics are available by 'show count'.

```
show Counters [Changed] [Same] [Z] [NZ] [Hex] [Raw] [<reg>][.][<pbmp>]
         Display all counters, or only specified regs and/or ports
           Changed - include counters that changed
           Same    - include counters that did not change
           Z       - include counters that are zero
           Nz      - include counters that are not zero
           All     - same as: Changed Same Z Nz
           Hex     - display counters as 64-bit hex values
           Raw     - display raw 64-bit hex values, no register name(s)
           ErDisc  - Only show those counters marked with Error/Discard
         NOTES: If neither Changed or Same is specified, Change is used.
                If neither Z or Nz is specified, Nz is used.
                Use All to display counters regardless of value.
```

** Display statistics of port xe0.

```
BCM.0> show c xe0
RUC.xe0            :            24,999,987             +0
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
RDBGC0.xe0        :              24,999,987             +0
R64.xe0           :                  97,654             +0
R127.xe0          :               6,152,568             +0
R255.xe0          :              18,652,091             +0
R511.xe0          :                  97,674             +0
RPKT.xe0          :              24,999,987             +0
RUCA.xe0          :              24,999,987             +0
RPRM.xe0          :              24,999,987             +0
RPOK.xe0          :              24,999,987             +0
RBYT.xe0          :           3,999,973,180             +0
T64.xe0           :                       1             +0
T127.xe0          :                       4             +0
T255.xe0          :                       5             +0
TPOK.xe0          :                      10             +0
TPKT.xe0          :                      10             +0
TUCA.xe0          :                      10             +0
TBYT.xe0          :                   1,299             +0
PERQ_PKT(10).xe0  :                      10             +0
PERQ_BYTE(10).xe0 :                   1,299             +0
```

8.3.8. Port mirroring configuration

'dmirror' sets mirroring option for a selected portto a target port.

```
Usage (DMIRror): Parameters: <ports> [Mode=<Off|On|Ingress|Egress|All>]
         [DestPort=<port>] [DestModule=<modid>] [DestTrunk=<tgid>]
         [MirrorTagProtocolID=<tpid>] [MirrorVlanID=<vid>
         DestTrunk overrides DestModule/DestPort.
         TPID and VLAN are set only if DestPort is a local port.
```

Port mirroring configuration examples are shown below.

** Mirror port xe0 ingress packet to xe3.

```
BCM.0> dmirror xe0 m=ingress dp=xe3
```

** Display mirrong status.

```
BCM.0> dmirror show
```

** Remove xe0 mirroring.

```
BCM.0> dmirror xe0 m=off
```

8.3.9. rcload

'rcload' executes switch command scripts.

```
Usage (RCLoad): Parameters: <file> [parameters]
         Load commands from a file until the file is complete or an error
         occurs. If optional parameters are listed after <file> they are
         pushed as local variables for the file processing. For example:
                   rcload fred a bc will result in the variables:
                            $1 = a
                            $2 = bc
                  during processing of the file.
         For platforms where FTP is used, the user name, password and host
         may be specified as: "user%password@host:" as a prefix to the
         file name
```


LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
```

** Run rcload.

```
BCM.0> rcload /tmp/traffic_10G_NP_RTM1_FABRIC
```

8.3.10. Switch configuration

Switch initialization uses /usr/local/bcm/config.bcm.atca3700 file. Switch can be restarted after this file is updated(/etc/init.d/bcm.sh restart). Also it can be updated in BCM shell.

```
Usage (CONFig): Parameters: show | refresh | save | [add] [<var>=<value>] | delete
<var>
         If no parameters are given, displays the current config vars.
           show               display current config vars
           refresh            reload config vars from non-volatile storage
           save               save config vars to non-volatile storage
           <var>=<value>      change the value of a config var
           add <var>=<value>  create a new config var
           delete <var>       deletes a config var
           clear              deletes all config vars
           =                  prompt for new value for each config var
         NOTE: changes are not retained permanently unless saved
```

Some configuration examples are described below.

** Display configuration values

```
BCM.0> config show
         os=unix
         bcm_num_cos=4
         tdma_intr_enable=1
         tslam_intr_enable=1
         tslam_dma_enable=1
         dport_map_enable=1
         dport_map_indexed=1
         dport_map_direct=1
         bcm_xlate_api_port_enable=1
         bcm_xlate_port_enable=1
         xgxs_lcpll_xtal_refclk=1
         load_firmware=1
         pbmp_xport_xe.0=0x0000000000ffffffe
         portmap_1.0=1:10
         portmap_2.0=2:10
         portmap_3.0=3:10
         portmap_4.0=4:10
         portmap_5.0=61:10
         portmap_6.0=63:10
         portmap_7.0=65:10
         portmap_8.0=69:10
         portmap_9.0=5:10
         portmap_10.0=7:10
         portmap_11.0=8:10
         portmap_12.0=13:10
         portmap_13.0=33:10
         portmap_14.0=29:10
         portmap_15.0=57:10
         portmap_16.0=58:10
         portmap_17.0=59:10
         portmap_18.0=60:10
         portmap_19.0=53:10
         portmap_20.0=54:10
         portmap_21.0=55:10
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
portmap_22.0=56:10
portmap_23.0=49:10
portmap_24.0=50:10
portmap_25.0=51:10
portmap_26.0=52:10
serdes_2wire_xaui_5=1
serdes_2wire_xaui_6=1
serdes_2wire_xaui_9=1
serdes_2wire_xaui_10=1
serdes_preemphasis_23=0xa370
serdes_preemphasis_24=0xa370
```

** Set port xe3 preemphasis =0x9.

```
BCM.0> config serdes_preemphasis_4=0x9
```

** Remove port xe3 preemphasis value.

```
BCM.0> config delete serdes_preemphasis_4
```

** Add port xe4 preemphasis= 1.

```
BCM.0> config add serdes_preemphasis_5=1
```

** Save configuration into /usr/local/bcm/config.bcm.atca3700.

```
BCM.0> config save
```

** Remove configuration.

```
BCM.0> config clear
```

8.3.11. LED

The faceplate LEDs controlled by the fabric switch are two for each SFP+ port, totaling 4 SFP+ ports, and two for Fabric A/B ports. The RTM faceplate has SFP+ 12 port LEDs. The operation of the SFP+ port LEDs is summarized in Table 8-3.

Table 8-3 SFP+ port LED operations

Picture	LED Name	LED status	Meaning	Remark
	E	On	Port Enable	
		Off	Port Disable	
	L/A	On	Link On/No Act	Port disable makes Link off.
		Off	Link off	
		Blink	Link On/Act	

LED ‘E’ is independent of link up/down states and configured as on/off by an operator configured port enable/disable status.
To control Enable LED, use both SFP TX_Disable and port enable/disable.

** LED ‘E’ On

```
[root@N700]$ nbm-sfp tx_disable y FP1
[root@N700]$ nbm-relay port xe0 en=true
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
[root@N700]$ nbm-relay ps xe0
     ena/   speed/   link   auto   STP                  lrn   inter   max  loop
port  link   duplex   scan  neg?  state    pause  discrd   ops   face  frame  back
xe0    up  1G FD   SW  Yes  Forward  TX RX   None   FA   GMII 16356
```

** LED ‘E’ Off

```
[root@N700]$ nbm-sfp tx_disable n FP1
[root@N700]$ nbm-relay port xe0 en=false
[root@N700]$ nbm-relay ps xe0
ena/   speed/   link   auto   STP                  lrn   inter   max  loop
port  link   duplex   scan  neg?  state    pause  discrd   ops   face  frame  back
xe0  !ena   1G FD   SW  Yes  Forward  TX RX   None   FA   GMII 16356
```

8.3.12. Base Switch

The proprietary command *nbm-robo* is used to configure the switch: port enable/disable and status checking. Here are the examples.

```
[root@N700]$ nbm-robo
root@aTCA-N700:~# nbm-robo
ROBO switch register tool
Usage: nbm-robo (init|close|show|clear)
               - init: initailize Base switch
               - close: Set state to 'Not forwarding'
               - show : Disply Base switch port infomation
               - clear: Clear port statistics infomation
       nbm-robo (enable|disable) <port>
               -  Enable or Disable port
       nbm-robo read  <page> <reg> <length>
               - Read Register value
       nbm-robo write <page> <reg> <length> <value>
               - Write values to Register
       nbm-robo vlan add <vid> <port> <tag/untag>
               - Add port to a VLAN
       nbm-robo vlan del <vid> <port>
               - Remove port from VLAN
       nbm-robo vlan destroy <vid>
               - Destroy a VLAN
       nbm-robo vlan show <vid>
               - Display a VLAN
       nbm-robo pvid <port> <vid>
               - Set a PVID for a port
       nbm-robo pvid show
               - Display PVID for all ports
       nbm-robo l2 add <MacAddr> <vid> <port>
               - Add L2 address associated with a port
       nbm-robo l2 del <MacAddr> <vid>
               - Delete L2 address associated with a port
       nbm-robo l2 clear
               - Delete all L2 table
       nbm-robo l2 show
               - Display all L2 table
       nbm-robo trunk (enable|disable)
               - Enable/Disable trunking
       nbm-robo trunk add <trunk_id> <port>
               - Add ports to a trunk(trunk_id: 0 or 1)
       nbm-robo trunk remove <trunk_id> <port>
               - Remove ports from a trunk
       nbm-robo trunk hash set <index>
               - Set hash index value to select egress port of a trunk
                 <index> - 0: [DMAC,SMAC],1: [DMAC], 2: [SMAC]
       nbm-robo trunk hash get
               - Get hash index value
       nbm-robo trunk show
               - Display all trunk infomation
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
nbm-robo trunk init
        - initialize trunking
```

Following example shows how to enable a port and perform connection verification such as ping operation.

```
root@aTCA-N700:~# nbm-robo show
Switch State: Managed (Forwarding)

Port Name  STP    Link       Speed    Tx(U)    TX(B)    Tx(M)    Rx(U)    RX(B)    Rx(M)
0    IPMC  DIS    Up        100(H)       0        0       12        0        0        0
1    LMP   DIS    Up       1000(F)       0        0        0        0       41        6
2    FRONT DIS    Up       1000(F)       0        0        0        0        0        0
3    BaseB DIS    Down     1000(F)       0        0        0        0        0        0
4    BaseA DIS    Up       1000(F)       0        0        0       21     1266      738
5    NP(B) DIS    Up       1000(F)       0        0        6        0        0        6
6    NP(T) DIS    Up       1000(F)       0        0        0        0        0        6
[root@N700]$ nbm-robo enable 2
[root@N700]$ nbm-robo enable 4
[root@N700]$ nbm-robo show
Switch State: Managed (Forwarding)

Port Name  STP    Link       Speed    Tx(U)    TX(B)    Tx(M)    Rx(U)    RX(B)    Rx(M)
0    IPMC   DIS    Up        100(H)       0        0       12        0        0        0
1    LMP    NOSTP  Up       1000(F)       0        0        0        0       41        6
2    FRONT  DIS    Down     1000(F)       0        0        0        0        0        0
3    BaseB  DIS    Down     1000(F)       0        0        0        0        0        0
4    BaseA  NOSTP  Up       1000(F)       0        0        0       21     1266      738
5    NP(B)  DIS    Up       1000(F)       0        0        6        0        0        6
6    NP(T)  DIS    Up       1000(F)       0        0        0        0        0        6
[root@N700]$ ifconfig eth1 90.0.0.2<= eth1: BASE switch port 1 connected with a LMP
interface
 [root@N700]$ ping 90.0.0.1
PING 90.0.0.1 (90.0.0.1): 56 data bytes
64 bytes from 90.0.0.1: icmp_seq=0 ttl=64 time=3.097 ms
64 bytes from 90.0.0.1: icmp_seq=1 ttl=64 time=0.248 ms
64 bytes from 90.0.0.1: icmp_seq=2 ttl=64 time=0.229 ms
...
```

As a Base Switchdefault configuration, put all ports in vlan 1, and set port 1(SOL) speed= 100M (Full).

```
[root@N700]$ nbm-robo init
[root@N700]$ nbm-robo show
Switch State: Managed (Forwarding)

Port Name  STP    Link       Speed    Tx(U)    TX(B)    Tx(M)    Rx(U)    RX(B)    Rx(M)
0    IPMC   DIS    Up         100(F)       0        0       12        0        0        0
1    LMP    NOSTP  Up       1000(F)       0        0        0        0       41        6
2    FRONT  DIS    Down     1000(F)       0        0        0        0        0        0
3    BaseB  DIS    Down     1000(F)       0        0        0        0        0        0
4    BaseA  NOSTP  Up       1000(F)       0        0        0       21     1266      738
5    NP(B)  DIS    Up       1000(F)       0        0        6        0        0        6
6    NP(T)  DIS    Up       1000(F)       0        0        0        0        0        6

[root@N700]$ nbm-robo vlan show 1
VID  Untagged Ports   Tagged Ports
================================
1    0 1 2 3 4 5 IMP
```

Add or delete vlan.

```
[root@N700]$ nbm-robo vlan add 2 1 untag
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
[root@N700]$ nbm-robo vlan show 2
VID  Untagged Ports   Tagged Ports
================================
2    1

[root@N700]$ nbm-robo pvid 1 2
[root@N700]$ nbm-robo pvid show
Port PVID
=========
0    1
1    2
2    1
3    1
4    1
5    1
IMP  1

[root@N700]$ nbm-robo vlan del 1 1
[root@N700]$ nbm-robo vlan show 1
VID  Untagged Ports   Tagged Ports
================================
0 2 3 4 5 IMP
```

Add or remove static mac for a port.

```
[root@N700]$ nbm-robo l2 add 00:00:00:00:00:01 1 3
[root@N700]$ nbm-robo l2 show
MacAddress        Mode    VID  Port
=================================
00:00:00:00:00:01 Static  1    3

[root@N700]$ nbm-robo l2 del 00:00:00:00:00:01 1
[root@N700]$ nbm-robo l2 show
MacAddress        Mode    VID  Port
=================================
```

port statisticscan be cleared as the following.

```
[root@N700]$ nbm-robo clear
[root@N700]$ nbm-robo show
Switch State: Managed (Forwarding)

Port Name  STP    Link       Speed    Tx(U)    TX(B)    Tx(M)    Rx(U)    RX(B)    Rx(M)
0    IPMC   DIS    Up        100(F)       0        0        0        0        0        0
1    LMP    NOSTP  Up       1000(F)       0        0        0        0        0        0
2    FRONT  DIS    Down     1000(F)       0        0        0        0        0        0
3    BaseB  DIS    Down     1000(F)       0        0        0        0        0        0
4    BaseA  NOSTP  Up       1000(F)       0        0        0        0        0        0
5    NP(B)  DIS    Up       1000(F)       0        0        0        0        0        0
6    NP(T)  DIS    Up       1000(F)       0        0        0        0        0        0
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

9. Board Management Package

The BOMA package includes functional modules for CPLD handling, and U-Boot environment variable configurations, base switch control, etc.

9.1. nbm-uenv

U-Boot environment variables are accessible in Linux by *nbm-uenv* utilities.
Usage:

nbm-uenv printenv [<VARNAME>]
nbm-uenv setenv <VARNAME> [<VALUE>]

*** Check U-Boot environment variables.

```
root@N700:~# nbm-uenv printenv
baudrate=115200
bootargs=root=/dev/ram rw console=ttyS0,115200 root=/dev/ram rw
ramdisk_size=400000
bootdelay=5
bootfile=packages/aTCA-N700T/linux.latest.bin
bootrfs=bootm $loadaddr $ramdiskaddr $fdtaddr
c=ffe
consoledev=ttyS0
eth1addr=00:0C:01:75:00:00
eth2addr=00:0C:01:75:00:01
eth3addr=00:0C:01:75:00:02
eth4addr=00:0C:01:75:00:03
ethact=FM1@DTSEC4
ethaddr=00:0C:01:75:00:00
fdtaddr=c00000
fdtfile=packages/aTCA-N700T/Adlink_N700.latest.dtb
fman_ucode=0x3fe44e50
hwconfig=fsl_ddr:ctlr_intlv=cacheline,bank_intlv=cs0_cs1
ipaddr=192.168.87.188
k=setenv bootfile kl;setenv fdtfile kd;setenv ramdiskfile kr; run tftpboot
loadaddr=1000000
mtdids=nand0=A,nand1=B
mtdparts=mtdparts=A:4m(dtbA),8m(kernelA),100m(rfsA);B:4m(dtbB),8m(kernelB),100m
(rfsB)
nandboot=ipmi en_bmc 60 1; run nibA; run normalargs; run bootrfs; ipmi
di_bmc;ipmi switch-active
nandboot_nobmc=run nibA; run normalargs; run bootfs
netdev=eth0
netmask=255.255.0.0
nfsargs=setenv bootargs root=/dev/nfs rw nfsroot=$serverip:$rootpath
ip=$ipaddr:$serverip:$gatewayip:$netmask:$hostname:$netdev:off
console=$consoledev,$baudrate $othbootargs
nfsboot=run nfsargs; tftp $loadaddr $bootfile; tftp $fdtaddr $fdtfile; bootm
$loadaddr - $fdtaddr
nibA=nboot $loadaddr kernelA; nboot $ramdiskaddr rfsA; nboot $fdtaddr dtbA
nibB=nboot $loadaddr kernelB; nboot $ramdiskaddr rfsB; nboot $fdtaddr dtbB
normalargs=setenv bootargs root=/dev/ram rw console=$consoledev,$baudrate
root=/dev/ram rw ramdisk_size=400000
partition=nand0,0
ramdiskaddr=2000000
ramdiskfile=packages/aTCA-N700T/rootfs.latest.bin
rootpath=/tftpboot/nfsroot
serverip=192.168.0.142
stderr=serial
stdin=serial
stdout=serial
tftpboot=run tib_1; run normalargs; run bootrfs
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
tib_1=tftp $loadaddr $bootfile;tftp $fdtaddr $fdtfile; tftp $ramdiskaddr
$ramdiskfile
```

9.2. nbm-robo

The base switch is connected via SPI interface to the LMP. nbm-robo provides utilities to configure the base switch. Refer to section 8.3.12 for the details.

9.3. nbm-cpld

```
This is a tool to read/write CPLD registers.
CPLD read/write tool
Usage: nbm-cpld <addr>        : read CPLD register
       nbm-cpld <addr> <value> : write CPLD register
         * addr  : 0, 1, 2, 3 ...
         * value : 0x0000 ~ 0xFF00
```

9.4. nbm-relay

nbm-relay transfers a command line to bcm.user.

```
Usage : nbm-relay [OPTION...] Command [ARGS...]
    -f, --out-file set a file name for output
    -s, --set-sep  set seperator string(ex: "BCM.0> ")
    -p, --listen   set listen address(default: 9895)
    -n, --npcmd    set NP telnet type
    -h, --help     show usage
```

** Check the port information.

```
root@aTCA-N700:~# nbm-relay ps xe0-xe3
          ena/  speed/ link auto    STP                lrn  inter   max  loop
     port link  duplex scan neg?   state   pause  discrd ops   face frame  back
      xe0  up    10G FD   SW  No   Forward  TX RX   None   FA    SFI 16356
      xe1  up    10G FD   SW  No   Forward  TX RX   None   FA    SFI 16356
      xe2  down  10G FD   SW  No   Forward  TX RX   None   FA    SFI 16356
      xe3  down  10G FD   SW  No   Forward  TX RX   None   FA    SFI 16356
```

9.5. nbm-mdio

nbm-mdio provides read/write utilities for MDC/MDIO devices.

Usage: nbm-mdio <busno> <devid> <regno>
nbm-mdio <busno> <devid> <regno> <value>

**Check register 0 of the management Ethernet port PHY device.

```
root@p3041ds:~# nbm-mdio 0 0x18 0
MDIO[0] DEVICE[0x18] REG[0x0] => 0x1140
```

9.6. nbm-util

nbm-util sets I2C devices for RTM and convert fabric switch configuration.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Usage:
nbm-util (rtm1|rtm2) (add|del) : add or delete RTM1 or RTM2
nbm-util swconfig FILENAME : convert simple config file to bcm.user command set

** Set i2c devices for RTM1

```
[root@N700]$ nbm-util rtm1 add
```

** Convert fabric switch configuration

```
[root@N700]$ cat /tmp/test
[LINK]
xe0 XFI
[VLAN]
100 xe1,xe2
[FP]
self-redirect xe2
[TX]
port xe3
time 10
[CHECK]
xe0-xe3
[root@N700]$ nbm-util swconfig /tmp/test

##### LINK CONFIG #####
port xe0 sp=10000 an=off en=true

##### VLAN CONFIG #####
vlan remove 1 pbm=xe
vlan create 100 pbm=xe1,xe2 ubm=xe1,xe2
pvlan set xe1,xe2 100

##### FP CONFIG #####
fp init
fp qset clear
fp qset add inports
fp group create 0 1
fp entry create 1 1
fp qual 1 inports xe2
fp action add 1 redirectport 0 xe2
fp entry install 1

##### START, WAIT and STOP #####
tx 100 u=yes pbm=xe3 ubm=xe3 l=68
sleep 10
port xe3 lb=none

##### STATISTICS #####
show c all r ruc.xe0-xe3
```

9.7. nbm-diag

nbm_diag performs diagnostics of aTCA-N700 board hardware resources. Depending on the purpose of the test, the test may be done by internal testing process or by external programs.
The configuration file can control the testing by directives.

The directives in the configuration file are shown in

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table 9-1.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table 9-1 Directives in the configuration file

Directive	Function	Remark
#	Start character of comment line	
<$STRING>	Defines a Test item.	The directives after this one until next “<>” or “[]” become the attribute of the Test Item.
[$STRING]	Defines a Test module.	The directives after this one until next “<>” or “[]” become the attribute of the Test module.
device: $STRING	Defines a device to test.	
extern: (yes:no)	Defines the Test program: either *nbm_diag* or an external program.	Default: no
intrusiveness: (yes|no)	Defines intrusiveness.	Default: no
test_item: $STRING	Defines nbm_diag built-in test item.	Usable only with “tester: self”. nbm_diag built-in test item list is checked by “nbm_diag –l”.
test_exec: $LINE	Defines an execution file and options for the test.	Usable only with “tester: other”
test_prepare: $LINE	Defines an execution file and options for test preparation before the test.	Usable only with “tester: other”
test_finish: $LINE	Defines an execution file and options to recover the status after the test.	Usable only with “tester: other”
test_log: $STRING	Defines a log file when the Test program generates one.	Log file may be used for the test result data.
test_result: $STRING	Defines a result file of external program including an err log when the Test program generates one.	Usable only with “tester: other” See the syntax for test result Without this, return value shall be the result of the test. default test_result file is /tmp/nbm_diag_result.txt

Each test module contains one or more test items, which are configurable. The tested results are stored in the test_result file.

The commands used with an external command have restrictions on stdin/stdout use because the test execution is converted as the following.

test_exec: ls –la
鱸 converted to “ls –la > /dev/null 2>&1”
test_exec: ls –la; echo 1234
鱸 converted to “ls –la > /dev/null 2>&1; echo 1234 > /dev/null 2>&1”

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

10. IPMI

The Intelligent Platform Management Interface, or IPMI, is a part of the ATCA standard for controlling intelligent devices that monitor a system. It enables dynamic discovery of sensors in the system and the ability to monitor the sensors and to be informed when the sensor's values change or go beyond the predefined boundaries. It also has a standardized database for field-replaceable units (FRUs) and a watchdog timer.
The IPMI interface is implemented between the LMP and the IPMC. On the LMP side, *ipmitool* on the Linux is used. The IPMC side is a Pigeon Point based implementation.

Shown in Figure 10-1 are IPMI internal blocks for the aTCA-N700.

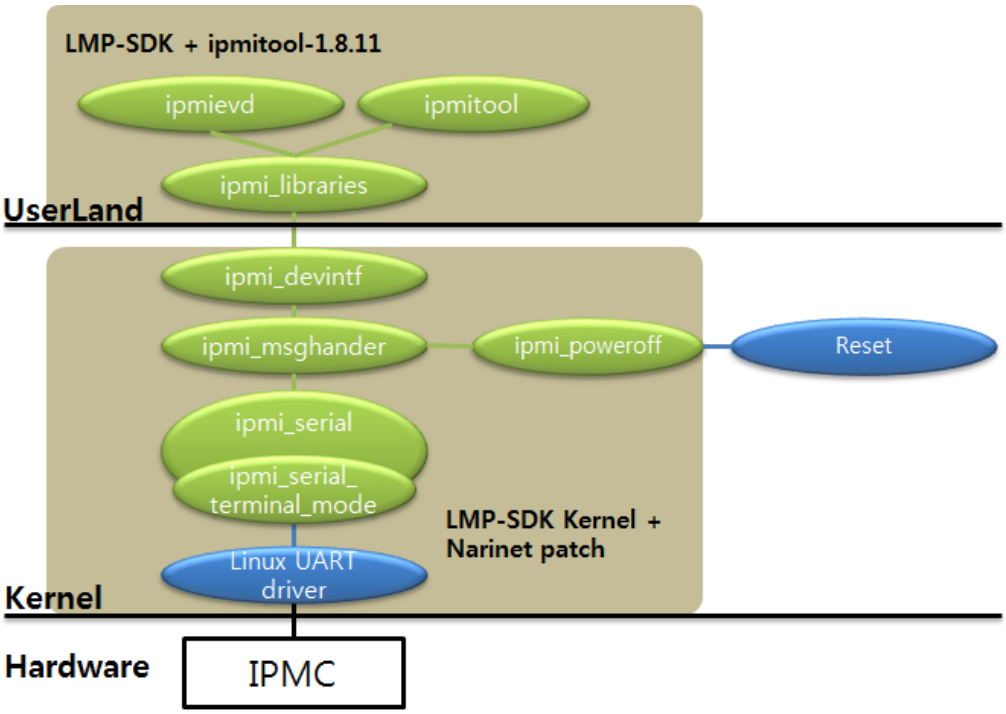


Figure 10-1 IPMI block diagram

10.1. ipmievd

impieevd is a program which performs surveillance operation for the IPMC SEL (System Event Log) and triggers user configured actions upon detecting pre-specified events.
Ipmi_event_rule.conf is a configuration file to be read by impievd to configure SEL actions.
The rule format is;

sensor_type:sensor_num:event_dir:event_type:event_data[0]:event_data[1]:event_data[2] => ${ACTION_CMD}

Following is a configuration example to perform power-off upon detecting a power state change by actions such as latch off.

```
0xF0 :0 :*    : 0x6F :0x06&0xF :*    :*          => poweroff
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

10.2. ipmitool

The *ipmitool* program provides a simple command-line interface to the LMP. It features the ability to read the sensor data repository (SDR) and print sensor values, display the contents of the System Event Log (SEL), print Field Replaceable Unit (FRU) inventory information, read and set LAN configuration parameters, and perform remote chassis power control.
The command is;

[CMD] ipmitool <option> <…>

Table 10-1 ipmitool options

Option	Description
sel list	Shows the SEL log stored in a flash device on the monitor.
sdr	Shows the sensor data on the monitor.
fru	Shows the FRU information on the monitor.

Following is an example for *ipmitool* with *sel list* option.

```
root@aTCA-N700:~# ipmitool sel list
 …
 3dc | Pre-Init Time-stamp     | FRU Hot Swap | Transition to M5 | Asserted
 3dd | Pre-Init Time-stamp     | FRU Hot Swap | Transition to M6 | Asserted
 3de | Pre-Init Time-stamp     | FRU Hot Swap | Transition to M1 | Asserted
 3df | Pre-Init Time-stamp    | OEM #0x11 |
 3e0 | Pre-Init Time-stamp     | OEM #0x13 |
 3e1 | Pre-Init Time-stamp     | FRU Hot Swap | Transition to M1 | Asserted
 3e2 | Pre-Init Time-stamp     | OEM #0x0f | State Asserted
 3e3 | Pre-Init Time-stamp     | OEM #0x10 | State Deasserted
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

11. IPMC Access via Ethernet

If the IPMC firmware on the IPMC is configured with IPMI over LAN or SOL, it is possible to use a direct Ethernet connection to the IPMC running the firmware.

To access the IPMC using an Ethernet connection, the following parameters should be specified in the command line of the *ipmitool* utility on a remote management system.

- *–I lan*
  This command line parameter instructs the *ipmitool* utility to use the RMCP protocol for communicating with the IPMC.

- *–I lanplus*
  Alternatively, this command line parameter can be specified to use the RMCP+ protocol for communicating with the IPMC, which provides enhanced authentication and encryption capabilities.

- *–H <IP address>*
  This command line parameter specifies the IP address of the IPMC.

11.1. IPMI-over-LAN

11.1.1. IOL Overview

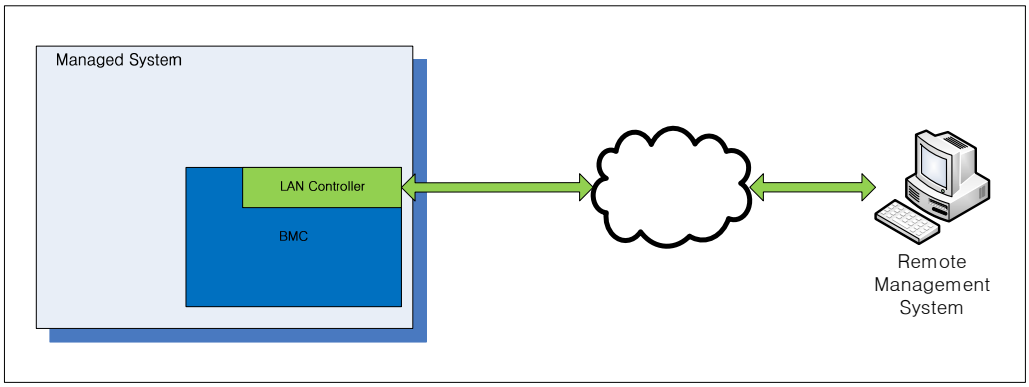


Figure 11-1 IPMI-over-LAN Diagram

IPMI messages can be transferred over LAN using UDP. The UDP datagrams are formatted to contain IPMI request and response messages, plus additional messages for discovery and authentication.

The RMCP protocol is used to pass IPMI message over LAN. In addition, the RMCP+ protocol extension can be used to pass other types of payload data over LAN (such as Serial-over-LAN), and provide enhanced authentication and encryption capabilities.

11.1.2. LAN, Channel and Session Configuration Commands

The *ipmitool* utility can be used (among other things) to read/write the LAN, channel and Session Configuration Parameters.
To be able to send configuration commands to the BMC, proper connection options must be specified on the *ipmitool* command line. The *ipmitool* commands that can be used to accomplish various tasks are listed with a task summary below.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

- Obtain general information about an ipmi channel. This command can be used to find out whether a given channel supports session establishment.

```
root@atca-3700:~# ipmitool channel info 5
Channel 0x5 info:
  Channel Medium Type   : 802.3 LAN
  Channel Protocol Type : IPMB-1.0
  Session Support       : multi-session
  Active Session Count  : 0
  Protocol Vendor ID    : 7154
  Volatile(active) Settings
    Alerting            : enabled
    Per-message Auth    : enabled
    User Level Auth     : enabled
    Access Mode         : always available
  Non-Volatile Settings
    Alerting            : enabled
    Per-message Auth    : enabled
    User Level Auth     : enabled
    Access Mode         : always available
root@atca-3700:~#
```

Figure 11-2 Get Channel Info command

- Set a privilege level for a spefied user.

ipmitool channel setaccess <channel num> <user id> privilege=<priv level>

```
root@atca-3700:~# ipmitool channel setaccess 5 2 privilege=2
root@atca-3700:~#
```

Figure 11-3 Set Channel Access command

- Change a user name.

ipmitool user set name <user id> <user name>

```
root@atca-3700:~# ipmitool user set name 2 soluser
root@atca-3700:~#
```

Figure 11-4 Set User Name command

- Obtain the access attributes associated with user.

ipmitool channel getaccess <channel number> <user id>

```
root@atca-3700:~# ipmitool channel getaccess 5 2
Maximum User IDs     : 3
Enabled User IDs     : 3

User ID              : 2
User Name            : soluser
Fixed Name           : No
Access Available     : call-in / callback
Link Authentication  : enabled
IPMI Messaging       : enabled
Privilege Level      : USER
root@atca-3700:~#
```

Figure 11-5 Get Channel Access Command

- Change a user password.

ipmitool user set password <user id> <new password>

```
root@atca-3700:~# ipmitool user set password 2 solpasswd
root@atca-3700:~#
```

Figure 11-6 Set User Password Command

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

■ Obtain the LAN configuration parameters for a given channel.

***ipmitool* lan print** *<channel num>*

```
root@atca-3700:~# ipmitool lan print 5
Set in Progress         : Set Complete
Auth Type Support       :
Auth Type Enable        : Callback :
                        : User     :
                        : Operator :
                        : Admin    :
                        : OEM      :
IP Address Source       : Unspecified
IP Address              : 172.17.172.134
Subnet Mask             : 255.255.0.0
MAC Address             : 00:30:64:0f:5b:16
Default Gateway IP      : 172.17.0.1
Default Gateway MAC     : 14:14:14:14:14:14
RMCP+ Cipher Suites     : 1,2,3,3
Cipher Suite Priv Max   : Not Available
root@atca-3700:~#
```

Figure 11-7 Get LAN Configuration Parameters Command

■ Set the management IP address for a given channel.

ipmitool lan set <channel num> <ip address> <value>

```
root@atca-3700:~# ipmitool lan set 5 ipaddr 192.168.20.100
Setting LAN IP Address to 192.168.20.100
```

Figure 11-8 Set LAN Configuration Parameters Command

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

11.2. Serial-over-LAN

11.2.1. SOL Overview

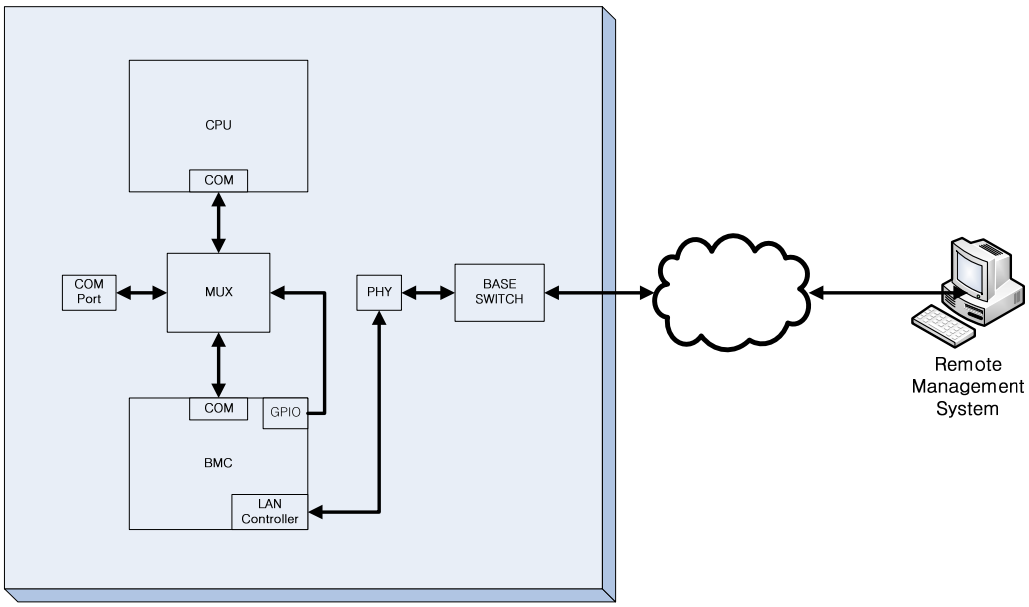


Figure 11-9 Serial-over-LAN Diagram

Serial-over-LAN (SOL) is an IPMI v2.0-defined capability that allows establishing a virtual serial console with the payload over LAN. It uses the RMCP+ protocol to encapsulate serial data in network packets and pass them between the BMC and remote console.

Check base switch configuration if both IPMC port and Management port you designed are up. An example in this chapter uses front port as the management port as shown below. For base switch configuration, please refer to Section 8.3.12.

```
root@atca-3700:~# nbm-robo show
Switch State: Managed (Forwarding)

Port Name  STP    Link       Speed     Tx(U)    TX(B)    Tx(M)    Rx(U)    RX(B)    Rx(M)
=============================================================================================
0    IPMC  NOSTP  Up         100(H)        0       96       34        0        0        0
1    LMP   DIS    Up         1000(F)       0        0        0        0        0        0
2    FRONT NOSTP  Up         1000(F)       0        0        0        0       96       47
3    BaseB DIS    Down       1000(F)       0        0        0        0        0        0
4    BaseA DIS    Up         1000(F)       0        0        0      394       12       25
5    NP(B) DIS    Down       1000(H)       0        0        0        0        0        0
6    NP(T) DIS    Up         1000(F)       0        0        0        0        0        0
```

Figure 11-10 Serial-over-LAN Diagram

11.2.2. SOL Commands

The *ipmitool* utility can be used (among other things) to read/write the SOL Configuration Parameters, and also to establish an SOL session with the payload.

To be able to send SOL commands to the IPMC, proper connection options must be specified on the *ipmitool* command line.

The *ipmitool* commands that can be used to manage the SOL Configuration Parameters and establish an SOL session are listed below.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

- Obtain the SOL configuration parameters for a given channel.

ipmitool sol info <channel num>

```
root@atca-3700:~# ipmitool sol info 5
Info: SOL parameter 'Payload Channel (7)' not supported - defaulting to 0x05
Set in progress                 : set-complete
Enabled                         : true
Force Encryption                : false
Force Authentication            : false
Privilege Level                 : ADMINISTRATOR
Character Accumulate Level (ms) : 5
Character Send Threshold        : 1
Retry Count                     : 1
Retry Interval (ms)             : 50
Volatile Bit Rate (kbps)        : 115.2
Non-Volatile Bit Rate (kbps)    : 115.2
Payload Channel                 : 5 (0x05)
Payload Port                    : 623
```

Figure 11-11 Get SOL Configuration Parameters Command

- Change one of the following SOL Configuration parameters.
  Session privilege level (<param> = privilege-level)
  Character accumulate interval (<param> = character-accumulate-level)
  Character send threshold (<param> = character-send-threshold)
  Retry count (<param> = retry-count)
  SOL Enabled status (<param> = enabled)
  Retry interval (<param> = retry-interval)
  Non-volatile bit rate (<param> = non-volatile-bit-rate)
  Volatile bit rate (<param> = volatile-bit-rate)

ipmitool sol set <param> <value>

- Establish an SOL session with the payload and enter the terminal mode. The <IP addr>, <user name> and <password> parameters are can be changed at run-time using the commands described above.

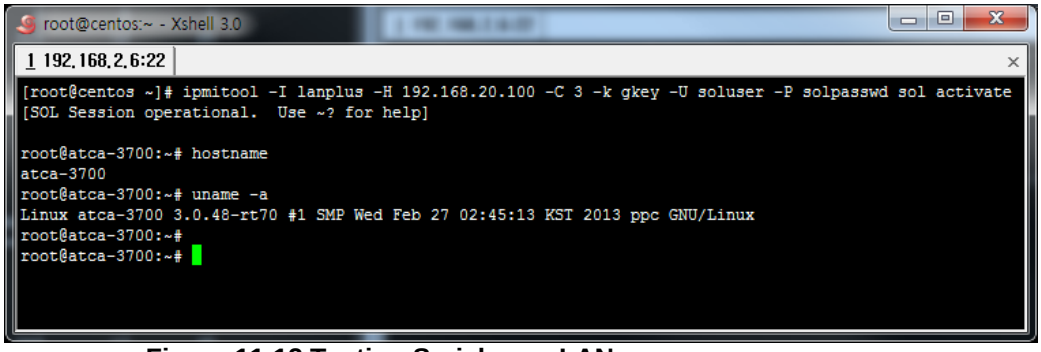


Figure 11-12 Testing Serial-over-LAN

- You can terminate a SOL session entering “~.” from the terminal.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

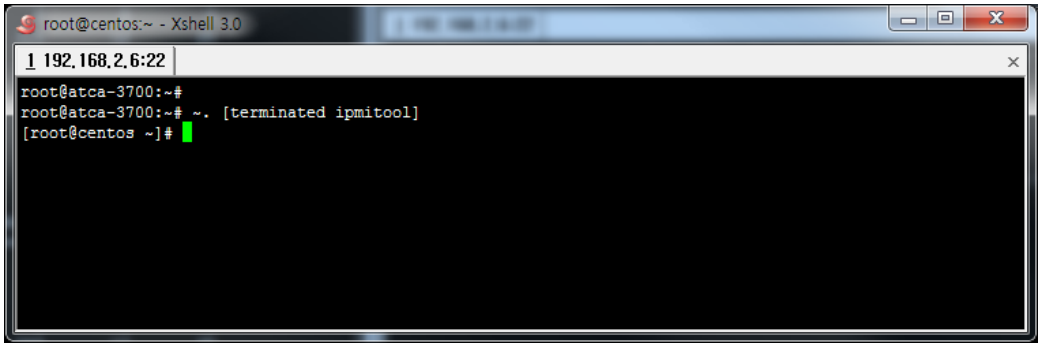


Figure 11-13 Terminating Serial-over-LAN Session

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Getting Service

ADLINK Technology, Inc.
Address: 9F, No.166 Jian Yi Road, Zhonghe District
New Taipei City 235, Taiwan
Tel: +886-2-8226-5877
Fax: +886-2-8226-5717
Email: service@adlinktech.com

Ampro ADLINK Technology, Inc.
Address: 5215 Hellyer Avenue, #110, San Jose, CA 95138, USA
Tel: +1-408-360-0200
Toll Free: +1-800-966-5200 (USA only)
Fax: +1-408-360-0222
Email: info@adlinktech.com

ADLINK Technology (China) Co., Ltd.
Address: 300 Fang Chun Rd., Zhangjiang Hi-Tech Park,
Pudong New Area, Shanghai, 201203 China
Tel: +86-21-5132-8988
Fax: +86-21-5132-3588
Email: market@adlinktech.com

ADLINK Technology Beijing
Address: Rm. 801, Power Creative E, No. 1, B/D
Shang Di East Rd., Beijing, 100085 China
Tel: +86-10-5885-8666
Fax: +86-10-5885-8625
Email: market@adlinktech.com

ADLINK Technology Shenzhen
Address: 2F, C Block, Bldg. A1, Cyber-Tech Zone, Gao Xin Ave. Sec. 7,
High-Tech Industrial Park S., Shenzhen, 518054 China
Tel: +86-755-2643-4858
Fax: +86-755-2664-6353
Email: market@adlinktech.com

LiPPERT ADLINK Technology GmbH
Address: Hans-Thoma-Strasse 11, D-68163, Mannheim, Germany
Tel: +49-621-43214-0
Fax: +49-621 43214-30
Email: emea@adlinktech.com

ADLINK Technology, Inc. (French Liaison Office)
Address: 15 rue Emile Baudot, 91300 Massy CEDEX, France
Tel: +33 (0) 1 60 12 35 66
Fax: +33 (0) 1 60 12 35 66
Email: france@adlinktech.com

ADLINK Technology Japan Corporation
Address: KANDA374 Bldg. 4F, 3-7-4 Kanda Kajicho,
Chiyoda-ku, Tokyo 101-0045, Japan
Tel: +81-3-4455-3722
Fax: +81-3-5209-6013
Email: japan@adlinktech.com

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

ADLINK Technology, Inc. (Korean Liaison Office)
Address: 8F Mointer B/D,1675-12, Seocho-Dong, Seocho-Gu,
Seoul 137-070, Korea
Tel: +82-2-2057-0565
Fax: +82-2-2057-0563
Email: korea@adlinktech.com

ADLINK Technology Singapore Pte. Ltd.
Address: 84 Genting Lane #07-02A, Cityneon Design Centre,
Singapore 349584
Tel: +65-6844-2261
Fax: +65-6844-2263
Email: singapore@adlinktech.com

ADLINK Technology Singapore Pte. Ltd. (Indian Liaison Office)
Address: 1st Floor, #50-56 (Between 16th/17th Cross) Margosa Plaza,
Margosa Main Road, Malleswaram, Bangalore-560055, India
Tel: +91-80-65605817, +91-80-42246107
Fax: +91-80- 23464606
Email: india@adlinktech.com

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000