

AAL1GATOR-32/TEMUX DEVELOPMENT KIT PLATFORM

PRELIMINARY

ISSUE 2: DECEMBER 2000

CONTENTS

1	INTRODUCTION.....	1
1.1	PURPOSE.....	1
1.2	SCOPE.....	1
1.3	APPLICATION.....	1
1.4	DOCUMENTATION.....	2
2	FEATURES.....	3
3	GENERAL DESCRIPTION.....	4
3.1	AAL1GATOR-32/TEMUX DEVELOPMENT KIT PLATFORM ARCHITECTURE ...	4
3.1.1	FULL DEVELOPMENT KIT PLATFORM CONFIGURATION.....	4
3.1.2	STANDALONE DEVELOPMENT KIT PLATFORM.....	5
4	TARGET MACHINE HARDWARE CONFIGURATION.....	6
5	REMOTE COMMAND MACHINE HARDWARE CONFIGURATION.....	7
6	SOFTWARE.....	8
6.1	PMC-SIERRA DEVELOPMENT KIT DISTRIBUTION CD-ROM.....	8
6.2	TARGET MACHINE SOFTWARE CONFIGURATION.....	11
6.2.1	HARD DISK FORMATTING.....	11
6.2.2	TARGET PLATFORM DIRECTORY STRUCTURE.....	11
6.3	SOFTWARE INSTALLATION ON THE TARGET PLATFORM.....	12
6.3.1	MAKING THE VXWORKS BOOT FLOPPY.....	13
6.3.2	SOFTWARE BINARIES INSTALLATION.....	13
6.3.3	SOFTWARE SOURCE CODE INSTALLATION.....	15
6.3.4	SOFTWARE UPDATES ON THE TARGET SYSTEM.....	16
6.4	SOFTWARE ON THE REMOTE COMMAND MACHINE.....	16

6.4.1	REMOTE COMMAND MACHINE TERMINAL CONFIGURATION	16
6.4.2	OTHER SOFTWARE ON THE REMOTE COMMAND MACHINE.....	21
7	TARGET SYSTEM STARTUP	22
8	SYSTEM COMMANDS EXECUTION OVERVIEW	23
8.1	ABOUT VXWORKS SHELL.....	23
8.2	COMMANDS INVOKED AT THE HOST (REMOTE) SYSTEM.....	24
8.3	COMMANDS INVOKED FROM THE TARGET SYSTEM	25
8.4	SYNCHRONOUS MODE OF OPERATION	25
9	GLOSSARY	26
10	APPENDIX A: VXWORKS SHELL SCRIPTS.....	27
11	APPENDIX B: CONTENT OF THE STARTUP.VXS.....	29
12	APPENDIX C: MS-DOS FILES NEEDED FOR THE AUTOMATIC INSTALLATION.....	30
13	APPENDIX D: DRIVER MAKEFILE.....	31
14	APPENDIX E: HOW TO RECREATE TORNADO 2 BUILD ENVIRONMENT	32
15	APPENDIX F: TARGET SYSTEM SELF-TEST	33
16	APPENDIX G: TARGET CONSOLE APPLICATION LEVEL DEBUG MESSAGES.....	34
17	APPENDIX H: AUTOEXEC.BAT EXAMPLE	35
18	APPENDIX I: CONFIG.SYS EXAMPLE.....	36
19	APPENDIX J: DRIVER SOURCE CODE AND BINARIES FILES LIST	37

LIST OF FIGURES

FIGURE 1 - FULL AAL1GATOR-32/TEMUX DEVELOPMENT KIT PLATFORM CONFIGURATION	4
FIGURE 2 - STANDALONE DEVELOPMENT KIT PLATFORM.....	5
FIGURE 3 - AAL1GATOR-32/TEMUX DISTRIBUTION CD-ROM DIRECTORY STRUCTURE.....	10
FIGURE 4 - DIRECTORY STRUCTURE ON THE TARGET MACHINE HARD DRIVE.....	12
FIGURE 5 - SETTING UP HYPERTERMINAL PROPERTIES.....	17
FIGURE 6 - HYPERTERMINAL PROPERTIES BOX.....	17
FIGURE 7 - COM PORT SETTINGS.....	18
FIGURE 8 - HYPERTERMINAL SETTINGS DIALOG.....	19
FIGURE 9 - ASCII SETUP DIALOG	20
FIGURE 10 - SAVE THE PROPERTIES.....	21
FIGURE 11 - USER COMMAND INVOCATION	24
FIGURE 12 - EXAMPLE OF THE VXWORKS SHELL 'LKUP' COMMAND	27
FIGURE 13 - EXAMPLE STARTUP.VXS SCRIPT	29

LIST OF TABLES

TABLE 1 - CONTENTS OF THE DEVELOPMENT KIT DISTRIBUTION CD-ROM	9
TABLE 2 – TARGET MACHINE HARD DRIVE DIRECTORY STRUCTURE COMPONENTS	12
TABLE 3 - AAL1GATOR-32 SOURCE FILES LIST	38
TABLE 4 - TEMUX SOURCE FILES LIST	39
TABLE 5 - DUPLEX SOURCE FILES LIST	39
TABLE 6 - DUPLEX SOURCE FILES LIST	40
TABLE 7 - SHARED SOURCE FILES LIST	40
TABLE 8 - TORNADO 2 BSP ADD-ON FILES	40
TABLE 9 - EXECUTABLES LIST	41

1 INTRODUCTION

The AAL1GATOR-32/TEMUX Development Kit Platform is the hardware platform on which the PMC-Sierra AAL1GATOR-32/TEMUX Development Kit board and application software can be tested and verified. This development platform enables PMC-Sierra's customers to dramatically reduce their application software development cycle.

The AAL1GATOR-32/TEMUX Development Kit consists of:

- AAL1GATOR-32/TEMUX Development Kit Board
- AAL1GATOR-32/TEMUX Development Kit software and documentation on the accompanying CD-ROM

Please, note that:

- The Pentium-based PC is not supplied by PMC-Sierra. The PC platform and its hardware and software configuration are the user's responsibility.
- The WindRiver Tornado 2 integrated development environment is not supplied by PMC-Sierra. If the user wants to continue development on the VxWorks OS it is the user's responsibility to purchase and configure the Tornado 2 development environment.
- The source code on the AAL1GATOR-32/TEMUX Development Kit Platform CD-ROM is distributed "as is". There is no backward compatibility with the latest software releases and the distributed source code shall be used for informational purposes only.

1.1 Purpose

This document provides a system specification for the AAL1GATOR-32/TEMUX Development Kit Platform.

1.2 Scope

This document describes the hardware and software requirements of the AAL1GATOR-32/TEMUX Development Kit Platform without going into the specifics of the PMC-Sierra AAL1GATOR-32/TEMUX Development Kit board. This document describes the AAL1GATOR-32/TEMUX Development Kit Platform in the level of detail required to configure and run the system. This document contains a detailed software installation procedure.

1.3 Application

This document is designed for embedded systems development based on the functionality of the PMC-Sierra devices included in this design, namely the PM73122 AAL1gator-32, PM8315 TEMUX, PM7350 S/UNI-DUPLEX and the PM4354 COMET-QUAD.

1.4 Documentation

The AAL1gator-32/TEMUX Development Kit CD includes a documentation directory containing relevant PMC-Sierra documentation. This documentation is current as of the publication date of this document. For up-to-date PMC-Sierra documentation refer to the PMC-Sierra website at www.pmc-sierra.com. The documents included on the CD are:

Document	Issue
PMC-1981419 AAL1gator-32 Datasheet	5
PMC-1991271 PM73122 (AAL1gator-32) 32-Link CES/DBCES AAL1 SAR Short Form Data Sheet	1
PMC-2000421 AAL1gator-32 Revision A Device Errata	3
PMC-1980581 S/UNI-DUPLEX Dual Serial Link PHY Multiplexer Data Sheet	5
PMC-1990147 PM7350 S/UNI-DUPLEX Dual Serial Link PHY Multiplexer Short Form Data Sheet	2
PMC-1971264 PM8315 TEMUX High Density T1/E1 Framer with Integrated VT/TU Mapper and M13 Mux Short Form Data Sheet	3
PMC-1981141 Technical Overview of the High Density T1/E1 Framer with Integrated VT/TU Mapper and M13 Multiplexer Telecom Standard Product	1
PMC-1981125 High Density T1/E1 Framer with Integrated VT/TU Mapper and M13 Multiplexer Telecom Standard Product Data Sheet	5
PMC-1990495 High Density T1/E1 Framer with Integrated VT/TU Mapper and M13 Multiplexer Telecom Standard Product Register Descriptions	3
PMC-1990677 PM8315 TEMUX Datasheet/Register Description Errata	7
PMC-1991444 AAL1gator-32/-8/-4 Driver User's Manual	1
PMC-1990799 Dual Serial Link, Phy Multiplexer (S/UNI-DUPLEX) Driver Manual	1
PMC-1990865 S/UNI-DUPLEX Driver Release Notes	3
PMC-1991611 PM8315 TEMUX (T1/E1 Multiplexer) Driver Manual	1
PMC-2000200 TEMUX Device Driver Release Notes	1
PMC-1991820 AAL1gator-32/8/4 Programmer's Guide	1
PMC-1991268 Temux Programmer's guide	1
PMC-1990887 AAL1gator-32 CES Reference Design	2
PMC-2000024 AAL1gator Product Family Technical Overview	1
PMC-2000088 Network Convergence of Voice, Data, and Video White Paper	1
PMC-1981025 S/UNI-VORTEX and S/UNI-DUPLEX Technical Overview	2
PMC-2000091 COMET-QUAD Short Form Data Sheet	1
PMC-1990315 COMET-QUAD Long Form Data Sheet	5
PMC-2000151 COMET-QUAD Programming Guide	1

2 FEATURES

- Jump-start development. Right out of the box the user can test the features of the AAL1GATOR-32/TEMUX Development Kit board and the application software supplied by PMC-Sierra. With the supplied software drivers and the evaluation board the customers can immediately start developing their own applications.

3 GENERAL DESCRIPTION

3.1 AAL1GATOR-32/TEMUX Development Kit Platform Architecture

There are two types of configurations where the AAL1GATOR-32/TEMUX Development Kit Platform can be used:

- Standalone (reduced configuration)
- Full

These two configurations are discussed below.

3.1.1 Full Development Kit Platform Configuration

The full development platform consists of two basic components:

- Target Machine
- Remote Command Machine

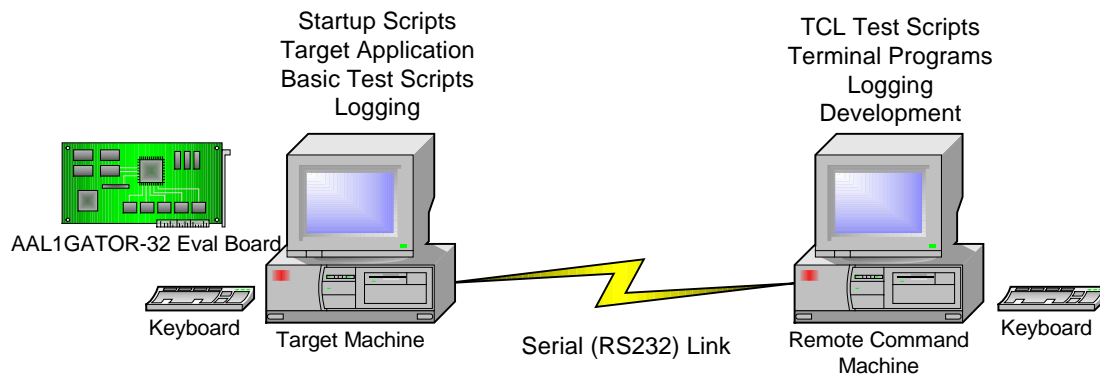


Figure 1 - Full AAL1GATOR-32/TEMUX Development Kit Platform Configuration

The target machine acts as the hardware support for the PMC-Sierra AAL1GATOR-32/TEMUX Development Kit Board. It is possible for the target machine alone to be used to test the basic functionality and features of the chipset on the board.

To extend the testing and development capabilities of the basic system, the Remote Command Machine is required. The communication link is synchronous, serial RS232. The Remote Command Machine can then be used with more powerful tools (i.e. TCL scripting) for testing and development over and above what's available on the Target Machine.

In this configuration it is not necessary to have a monitor and keyboard attached to the Target Machine since all the commands can be entered on the Remote Command Machine.

In order to establish communication, the user must connect the remote and target machine via a null modem cable.

Using the serial link a user can send the predefined set of commands (supported by the application software supplied by PMC-Sierra or by the user) and the commands will be executed on the target system. The application software will send the responses back to the Remote Command Machine.

3.1.2 Standalone Development Kit Platform

The Standalone Development Kit Platform consists of one component:

- Target Machine

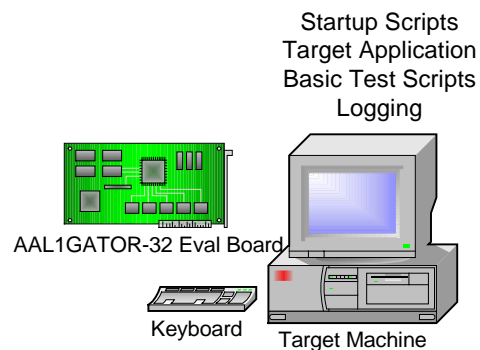


Figure 2 - Standalone Development Kit Platform

This is a minimal configuration of the Development Kit Platform. This standalone system has reduced capabilities compared to the full version. The full functionality of the AAL1GATOR-32/TEMUX Development Kit Board and the supplied software can still be tested with the minimal configuration.

In this system configuration a monitor and keyboard are necessary. The commands can be entered on the target machine keyboard and the results are visible to the user on the target machine display.

4 TARGET MACHINE HARDWARE CONFIGURATION

The target machine must have the following minimal hardware configuration:

- Intel Pentium based motherboard (minimum 233MHz) with one free PCI slot (for the development kit board).
- One available serial communication port (required for communication with the Remote Command Machine) capable of sending and receiving data at a minimum of 9600 bps. The second serial communication port is optional.
- Standard 1.44 MB Floppy Disk Drive
- CD-ROM Drive
- IDE Hard Disk Drive. Recommended size is 2.1GB.
- Keyboard
- VGA Video Adapter
- VGA Compatible Monitor

Network card is optional. There is no software network support on the target platform supplied by PMC-Sierra. If required, the user can install the network support software.

Video Graphic Capabilities. There is no requirement for any graphical capabilities of the video card on the target machine since the system will run in text mode.

5 REMOTE COMMAND MACHINE HARDWARE CONFIGURATION

In order to establish the communication with the target system the Remote Command Machine must have:

- One free standard serial communication port capable of sending and receiving data at minimum of 9600 bps. The second serial communication port is optional.

6 SOFTWARE

The current version of all PMC-Sierra software device drivers has been developed for only one operating system – VxWorks. Therefore, this document will restrict the examples and software installation procedures to cases that involves only the VxWorks operating system.

The target platform software installation and configuration has to be done locally. In other words, the full target platform including keyboard and monitor is required during this process.

6.1 PMC-Sierra Development Kit Distribution CD-ROM

The Development Kit distribution CD-ROM contains all the necessary documentation and software required for the target platform. Once installed the target system is fully functional and ready to run.

Directory Item	Comment
(D:)	Root directory of the distribution CD-ROM. Contains INSTALL.BAT. Running this file will start automatic installation of the required software binaries onto the target system hard drive.
/DOC	Location of all the required documentation.
/SOFTWARE	Root directory of the distributed software.
/SOFTWARE/ TARGET	Root directory of the software for the target system.
/SOFTWARE/ TARGET/BIN	Root directory of the software binaries required for the target system.
/SOFTWARE/ TARGET/BIN/OS	Root directory of the target system operating system.
/SOFTWARE/ TARGET/BIN/OS/ VxWorks	Directory where the VxWorks operating system and its supporting binaries is located.
/SOFTWARE/ TARGET/BIN/ DRV	Directory where the PMC-Sierra software chip drivers are located.
/SOFTWARE/ TARGET/BIN/ SCRIPTS	Directory where all of the VxWorks shell scripts are located.
/SOFTWARE/ TARGET/ SOURCE	Root directory of the source code.
/SOFTWARE/ TARGET/ SOURCE/AL32	AAL1GATOR-32 chip driver source code root directory. It is in the form of a ZIP file. The ZIP file contains directory structure of the driver source code.
/SOFTWARE/ TARGET/ SOURCE/TEMUX	TEMUX chip driver source code root directory. It is in the form of a ZIP file. The ZIP file contains directory structure of the driver source code.
/SOFTWARE/ TARGET/ SOURCE/ DUPLEX	DUPLEX chip driver source code root directory. It is in the form of a ZIP file. The ZIP file contains directory structure of the driver source code.
/SOFTWARE/	COMET-QUAD chip driver source code root directory. It is in the form of a ZIP file. The ZIP

Directory Item	Comment
TARGET/ SOURCE/ COMETQ	file contains directory structure of the driver source code.
/SOFTWARE/ TARGET/ SOURCE/ SHARED	Shared components source code root directory. It is in the form of a ZIP file. The ZIP file contains directory structure of the driver source code.
/SOFTWARE/ TARGET/ SOURCE/OS	Root directory of the operating system source distribution. Currently, only VxWorks is supported.
/SOFTWARE/ TARGET/ SOURCE/OS/ VxWorks	Root directory of the VxWorks operating system source distribution.
/SOFTWARE/ TARGET/ SOURCE/OS/ VxWorks/BSP	Location of the add-on source code for the Tornado 2 development system Board Support Package (BSP).
/SOFTWARE/ TARGET/ UTIL/ MKVXBOOT	Location of the utility software for making VxWorks boot floppy disk.

Table 1 - Contents of the Development Kit Distribution CD-ROM

ZIP/UNZIP utilities are downloadable for free from a number of Internet sites. ZIP files on the distribution CD-ROM (like the driver source code) contains a directory structure. The user should unzip the files into the root directory and the necessary directory structure will be created.

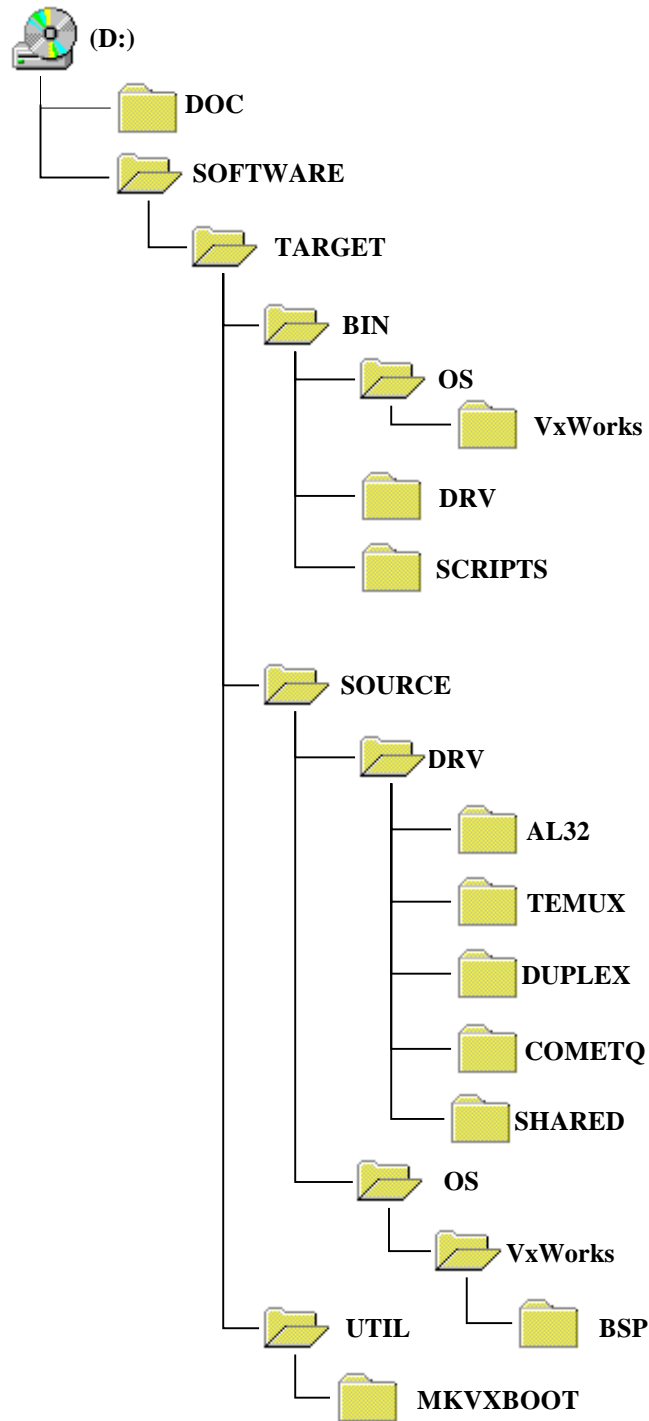


Figure 3 - AAL1GATOR-32/TEMUX Distribution CD-ROM Directory Structure

6.2 Target Machine Software Configuration

The software that is required for the system startup and operation must be located on the target machine hard drive (with the exception of the boot disk).

Since VxWorks is a real-time operating system there are not too many hard disk formats that this OS can support. One that users might find easy to install and use is FAT-16. This is also a format that operating systems like MS-DOS and MS Windows can recognize.

It is assumed that the user is using a FAT-16 disk format.

6.2.1 Hard Disk Formatting

In order to use the hard drive and install the new software the hard disk must be formatted.

The fastest way to do this is to use MS-DOS version 6 or higher and follow the MS-DOS installation procedure from the Microsoft Corporation DOS User's Manual.

The MS-DOS operating system has been chosen because of its fast boot time, fast installation time and small size. MS-DOS does not have to be installed, but the user may find some MS-DOS external commands useful during the development process (such as the DOSKEY command that repeats the commands on the command prompt). MS-DOS is required for automatic software installation on the target system.

The file system on the hard drive must be FAT-16.

VxWorks only recognizes the primary FAT-16 partition. The other partitions (if installed) may be used for other purposes.

6.2.2 Target Platform Directory Structure

A minimal fixed directory structure has to be maintained on the target machine as shown in Figure 4.

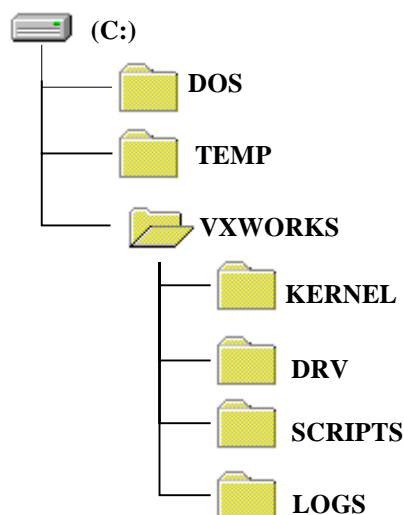


Figure 4 - Directory structure on the target machine hard drive

Directory Item	Comment
(C:)	Root directory of the FAT-16 file system.
/DOS	Directory where the Disk Operating System (DOS) might be installed.
/TEMP	Temporary directory.
/VXWORKS	Root directory of the VxWorks operating system.
/VXWORKS/ KERNEL	Directory where VxWorks kernel and VxWorks supporting binaries should be installed.
/VXWORKS/DRV	Directory where PMC-Sierra software chip drivers and supporting binaries should be installed.
/VXWORKS/ SCRIPTS	Directory where VxWorks shell script files should be installed. The most important script is <code>startup.vxs</code> which executes automatically during the system startup. This script must be located in this directory.
/VXWORKS/LOGS	Directory where application log dumps might be stored.

Table 2 – Target machine hard drive directory structure components

Normally the VxWorks directory structure will be created during automatic software installation.

6.3 Software Installation On the Target Platform

Once the target platform has been prepared properly the binaries supplied by PMC-Sierra have to be installed onto the target platform hard drive and the source code supplied by PMC-Sierra has to be installed onto the software development platform. Note that a MS-DOS CD-ROM driver must be installed properly on the target machine prior to automatic software installation.

6.3.1 Making the VxWorks Boot Floppy

The target system boots up off the VxWorks boot floppy. All the necessary files and utilities necessary to make a VxWorks boot floppy are located on the AAL1GATOR-32/TEMUX distribution CD-ROM.

The following steps should be taken in order to make a bootable VxWorks floppy:

1. Put the AAL1GATOR-32/TEMUX distribution CD-ROM into your host machine CD-ROM drive.
2. Open Windows Explorer and change the CD-ROM directory to /SOFTWARE/TARGET/UTIL/MKVXBOOT .
3. Put the blank floppy disk into your floppy drive and label it as "VxWorks Hard Disk Boot". Make sure the floppy disk is not write protected.
4. Doubleclick on mvxwbd.bat file. This action will open MS-DOS window and start making the bootable VxWorks floppy.
5. Once the MS-DOS window has been closed, the VxWorks bootable floppy disk is ready and the user can remove it from the host machine floppy drive. The file bootrom.sys should exist on the bootable floppy.

Note: the MS-DOS window may not close automatically after the MS-DOS task has been finished. In that case the MS-DOS box will have "Finished..." at its top left corner. The user should close the window manually.

6.3.2 Software Binaries Installation

1. Reboot the target platform in MS-DOS.
2. Insert the distribution CD-ROM into the target machine CD-ROM drive.
3. Set the current directory to D:\ (assuming that CD-ROM drive letter is D:).
4. Run the `install.bat <source drive> <destination drive>` program which will copy the necessary files to the target platform hard drive.

After the automatic installation is complete the VxWorks directory structure will exist and all the software necessary for target system operation will be installed.

6.3.2.1 Self-test

After the software is installed on the target system, power it down and:

1. Add the AAL1GATOR-32/TEMUX Development Kit board to the desired slot.

2. Insert the VxWorks boot diskette.
3. Power up the target system. Make sure that BIOS on the target machine is first looking for the bootable floppy and then the bootable hard drive. Once the boot sequence has finished, the AAL1-32/TEMUX \$ > command prompt will show up on the target machine screen.
4. Type the test command "al32SelfTest %a,%b" where:
 - %a is the slot number of the AAL1GATOR-32/TEMUX CHIPSET card
 - %b is the "animate" mode parameter. Animate mode displays the register writes in a visible manor to the terminal. This parameter can be 0 or 1 (1 for animate mode).

This self-test performs two board level tests. The first test uses the pseudo-random sequence generator and detector to test the line side of the TEMUX. A PRBS 23 pattern is transmitted by the generator into the DS3 LIU. The DS3 LIU is configured for loopback using the jumper (J5). Alternatively, an external loopback cable can be used to loopback the data, but the cable is not provided. The receive line interface of the TEMUX receives the data and verifies that the pattern is being received without errors.

The second test transmits cells with framed DS3 AIS data. The S/UNI-DUPLEX then loops the cells back on the UTOPIA interface. The test then checks the AAL1gator for dropped cells and sequence number errors. The data is then transmitted across the SBI bus and verified by the TEMUX.

Once the test has finished the following message will appear on the target machine:

```
AAL1-32/TEMUX board in slot[%d] self-test PASSED!
```

5. Configure the board for COMET-QUAD operation by shorting jumper J8. Type the test command "al32SelfTest_CometQ %a,%b" where:
 - %a is the slot number of the AAL1GATOR-32/TEMUX card
 - %b is the "animate" mode parameter. Animate mode steps through each of the writes to the registers at a visible pace and displays the values on the terminal. This parameter can be 0 or 1 (1 for animate mode).

Note that the %b parameter is optional and defaults to 0.

Type al32SelfTest_CometQ with the PCI slot number containing AAL1GATOR-32/TEMUX board in it.

The AAL1gator-32/COMET-QUAD Self Test verifies the ability of the development kit to configure the AAL1gator-32, COMET-QUAD, and S/UNI-DUPLEX for correct operation. This test also verifies the datapaths between the three devices. The test configures the COMET-QUAD to overwrite the entire frame transmitted on the lowspeed link with PRBS data. The AAL1gator-32 is then configured to circuit emulate the 4 unframed T1 links over ATM cells. The ATM cells are then transmitted across the UTOPIA bus where they are looped back by the S/UNI-DUPLEX.

6.3.3 Software Source Code Installation

The following steps are required to install the software development environment and are specific to the Tornado 2.0 VxWorks environment. Users installing the code on a different environment should use this as a reference only.

1. Modify your `config.h` by hard coding the `#include "prjparams.h"` file at the bottom of the file of your BSP. Also modify `sysLib.c` to include the file `sysPci.c` provided in the BSP directory. Also in `sysLib.c` add the following code fragment that is indicated below to the function `sysHwInit`.

Existing code:

```
#if defined (INCLUDE_FEI) || defined (INCLUDE_FEI_END)
    sys557PciInit();
#endif /* INCLUDE_FEI || INCLUDE_FEI_END */
```

Add the following code:

```
/* PMC PCI support */
#ifdef INCLUDE_PMC_PCI_LIB
    sysfPmcAdapterPciInit ();
#endif /* INCLUDE_PMC_PCI_LIB */
```

Existing Code:

```
#ifdef INCLUDE_LN_97X_END
    sysLan97xPciInit ();
#endif /* INCLUDE_LN_97X_END */
```

2. Add the files `sysPci.c` and `sysPci.h` provided to the BSP directory, e.g. `D:\tornado2\target\config\pc486`.
3. Compile VxWorks and create a boot floppy.
4. Create a Tornado project. Select "Create a bootable VxWorks Image". Fill in the Project, location and description. "Add to a New or Existing Workspace". In the next dialog box when asked, "Would you like to base your project on:" select "A BSP" and select the BSP directory from above. VxWorks will then generate a project. You can now build a VxWorks image from the workspace. Please, see the Tornado2 documentation for more details.
5. Create the root directory for your project (e.g. `c:\project\al32Board`).
6. Create the following subdirectories in your project directory:
 - `al32`
 - `temux`
 - `duplex`
 - `Cometq`
 - `shared`Note that the directory names can be different than suggested.
7. Unzip the driver source code from the distribution CD-ROM to the corresponding directory on the host hard drive.

8. If necessary, modify the directory variable names in the makefiles, to reflect your actual driver and directory names.
9. Compile the source files using the makefiles supplied with each software driver. Note that the shared component doesn't have a makefile and therefore it is not necessary to compile it.

6.3.4 Software Updates On The Target System

Updating the software files on the target system is easy and straightforward.

The first step is to boot the target machine into MS-DOS.

If the device driver has to be upgraded, the user has to copy the software chip driver over the old one in the `c:\vxworks\drv` directory.

If the script has to be upgraded, the user should copy the new script over the old one in the `c:\vxworks\scripts` directory.

Adding a new software chip driver requires copying the new driver into the `c:\vxworks\drv` directory and adding the driver into the load list of drivers in the `startup.vxs` script.

Adding a new script requires copying the new script into the `c:\vxworks\scripts` directory and possibly adding the script into the list of startup scripts in the `startup.vxs` script.

6.4 Software on the Remote Command Machine

As mentioned earlier there are no special requirements for the software to run on the Remote Command Machine.

In order to communicate with the target system the terminal software must exist on the Remote Command Machine and has to be properly configured.

6.4.1 Remote Command Machine Terminal Configuration

There are many terminal programs on the market. The example in this document will show (step by step) how to properly configure Windows HyperTerminal.

After opening HyperTerminal the first step is to configure the properties of the current session.

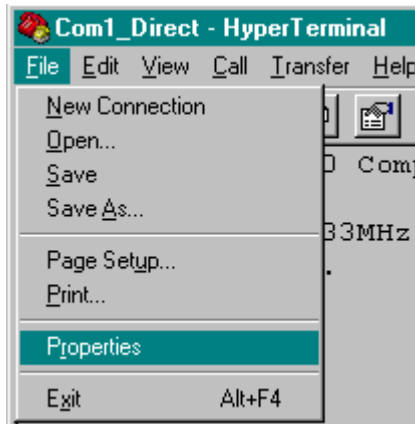


Figure 5 - Setting up HyperTerminal Properties

After clicking on Properties, the HyperTerminal “Properties” dialog box will show up.

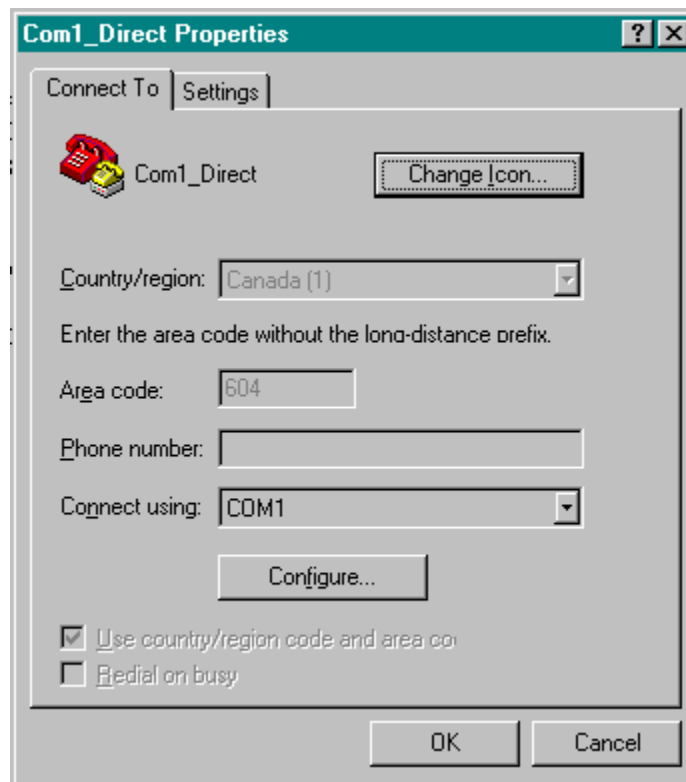


Figure 6 - HyperTerminal Properties Box

We will assume here that COM1 is the port that is free to use for communication between the host system and target machine. If COMx is the communication port that is free, the user should select it from the dropdown list and click on the Configure button.

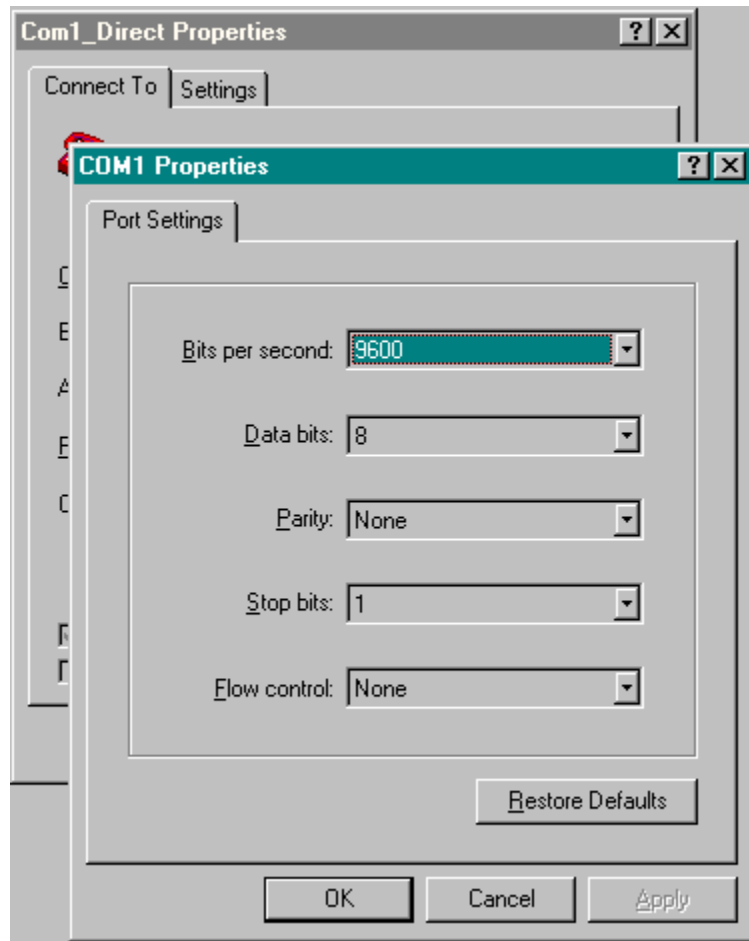


Figure 7 - COM port settings

The "Port Settings" dialog will show up. The serial port configuration should be exactly the same as shown in Figure 7. After setting up the port, click OK. The COMx properties dialog box will disappear.

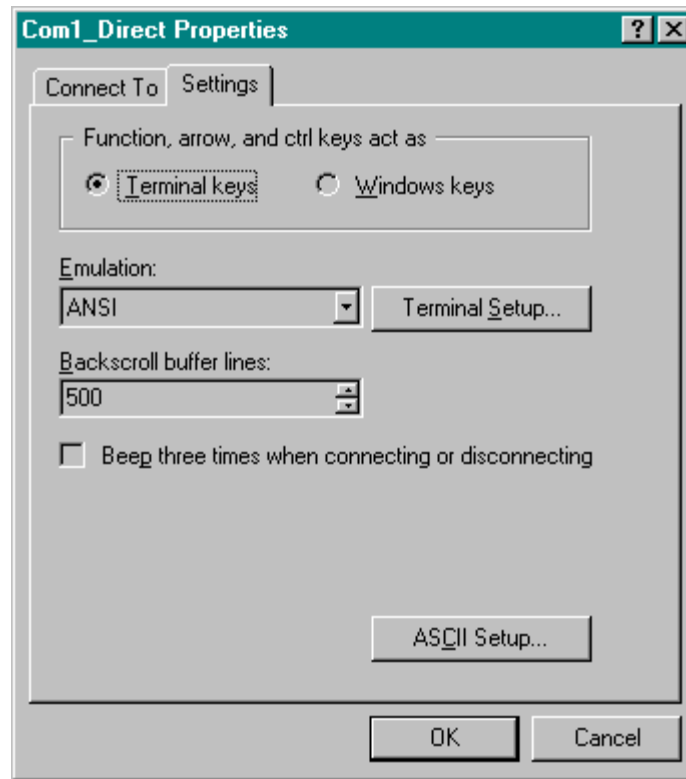


Figure 8 - HyperTerminal Settings Dialog

Click on the “Settings” tab and select “Terminal keys”, “ANSI” Emulation and enter “500” Backscroll buffer lines.

Click on the “ASCII Setup...” button.

Once the ASCII Setup Dialog box shows up check:

- Echo typed characters locally
- Force incoming data to 7-bit ASCII
- Wrap lines that exceed terminal width

The following have to be unchecked:

- Send line ends with line feeds
- Append line feeds to incoming line ends
- Line and character delay should be set to 0 milliseconds.

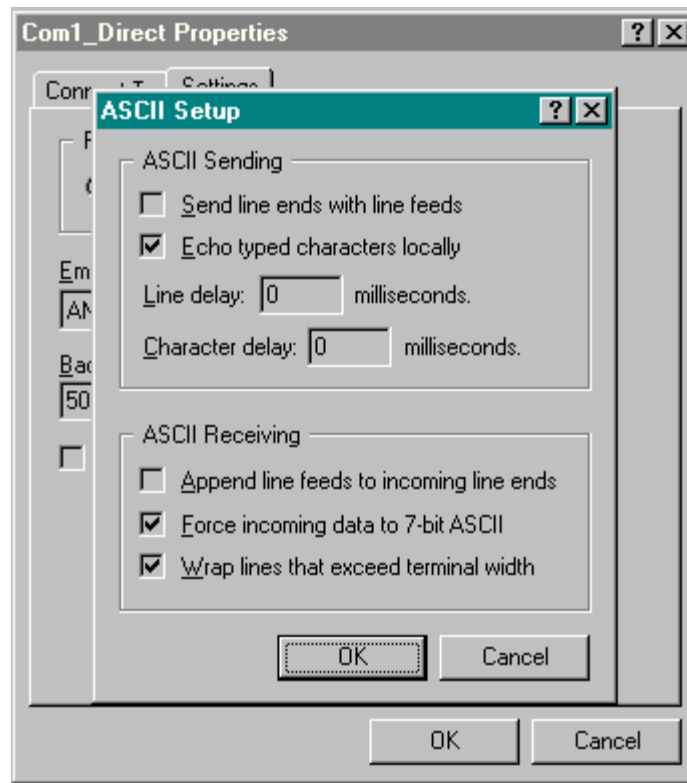


Figure 9 - ASCII Setup Dialog

Once setup has been completed click OK and the “ASCII Setup” dialog box will disappear.

Click OK on the session properties dialog box.

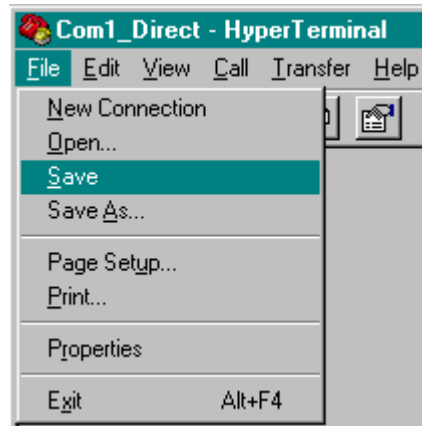


Figure 10 - Save the properties

The user must save the properties in order for them to come into effect (as shown on Figure 10). Otherwise, HyperTerminal will keep the old settings. After the terminal configuration is complete, the Remote Command Machine is ready to communicate with the target system.

6.4.2 Other Software On The Remote Command Machine

Simple terminal software might not be enough for long-term testing and development of the target system.

The user may choose to use some other off-the-shelf tool, like TCL, or build their own application in order to extend the development and testing capabilities of the entire system.

7 TARGET SYSTEM STARTUP

Once the software installation on the target system has been finished, the target system is ready to run. The user should ensure that the BIOS bootup sequence on the target machine is "A: then C:".

In order to run the system the user has to do the following:

- Insert the VxWorks boot floppy into the target machine floppy drive.
- Reboot the target platform (by either switching the power or pressing reset button).

After VxWorks is finished its boot up sequence, the VxWorks Shell command prompt should show up on the terminal platform.

If there is no terminal screen on the target platform the user can see when the boot up sequence is finished by watching the Terminal Window on the Remote Command Machine. If the target system is ready, a message similar to the following will show up:

```
Welcome to the AAL1GATOR-32/TEMUX Board Command Parser
```

8 SYSTEM COMMANDS EXECUTION OVERVIEW

8.1 About VxWorks Shell

The VxWorks shell is not like other OS shells. The user is not launching executables from the VxWorks shell prompt.

Given the right configuration of the VxWorks kernel, the user is actually making the function calls from the VxWorks shell prompt. Therefore, the VxWorks shell is a debug shell.

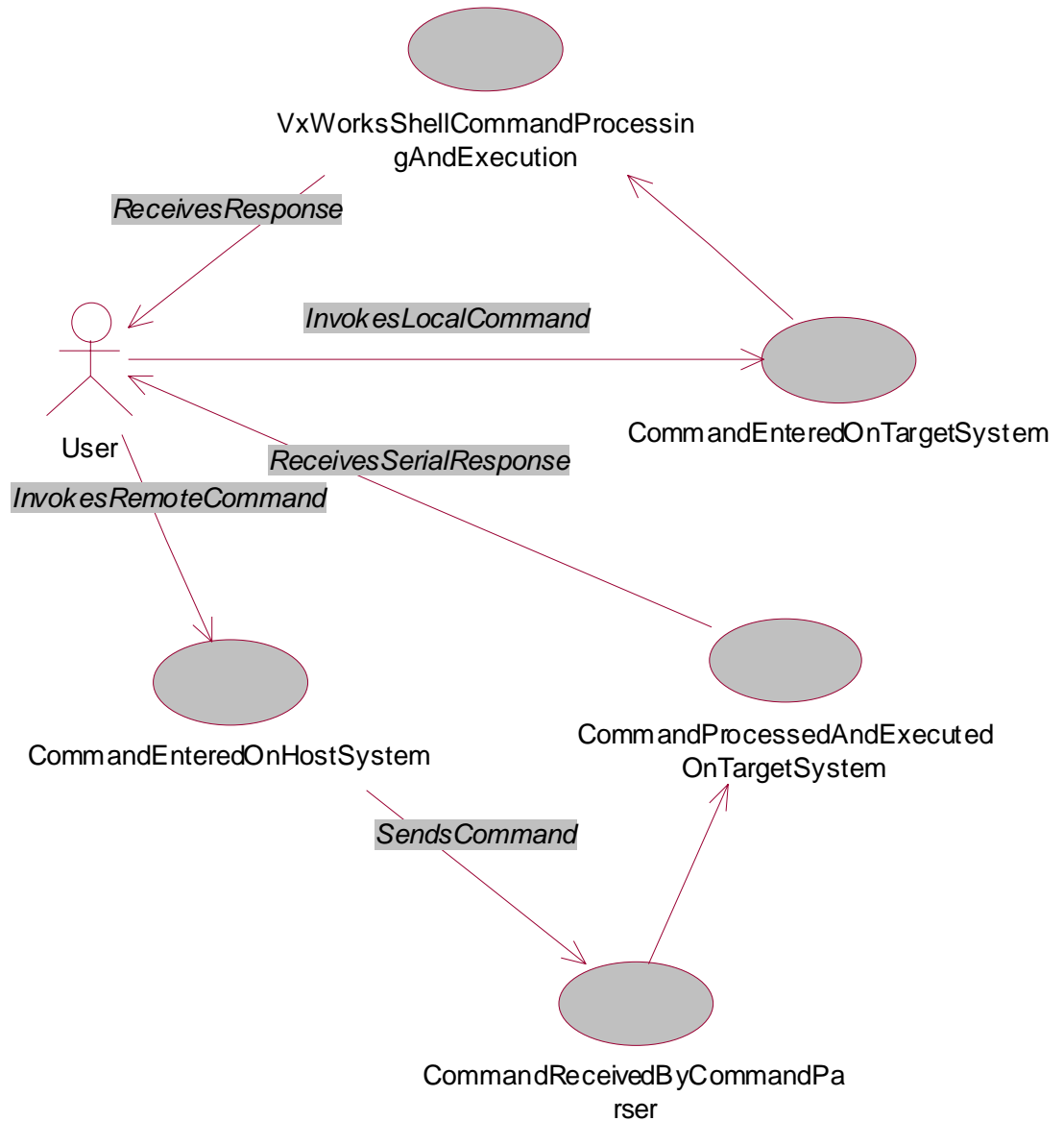


Figure 11 – User command invocation

8.2 Commands Invoked At The Host (Remote) System

The distribution CD-ROM contains the command parser application.

Command Parser:

- takes the input from the serial link
- parses the command
- finds the function that is responsible for the command execution (if possible)
- takes the response from the same function
- then returns the response over the serial link to the user

There are a limited number of commands that can be sent over the serial link to the target machine and be properly executed on the target system.

The set of the commands is predefined by the software driver(s) in the system. Once loaded, the software driver will register its set of commands to the command parser. Once the commands have been received over the serial link, only commands registered with the command parser will be executed.

For all other inputs, the command parser will return a general error to the user that it could not find the command in its command list.

8.3 Commands Invoked From The Target System

This situation is somewhat different. The commands entered at the VxWorks command shell are not going through the command parser – instead these commands will be function calls to any of the VxWorks exposed functions in the system.

When a command is received over the serial link, the command parser will call a function that will call the underlying driver functions.

In this situation the driver functions have to be called from the VxWorks command shell directly since the command parser entries can't be properly accessed.

8.4 Synchronous mode of operation

The software executing on the target system works in a synchronous mode of operation. Therefore, once a command is invoked, the target system will not respond to further commands until a response has been returned to the user. At that point in time the user can invoke a different development kit command.

9 GLOSSARY

HOST	Used interchangeably with the Remote Command Machine
Target Machine	Hardware platform where AAL1GATOR-32/TEMUX Development Kit Board resides.
BSP	Board Support Package
OS	Operating System
BIOS	Basic Input Output System

10 APPENDIX A: VXWORKS SHELL SCRIPTS

VxWorks shell scripting is a very powerful tool for target system application debugging.

The VxWorks shell is not a classic OS shell – it is more of a debug shell. If configured properly, the VxWorks kernel can expose all the functions and variables in the system to the user (local and global).

Figure 12 - Example of the VxWorks shell 'lkup' command

```
-> lkup "al3E" <ENTER>
al3EnhancedActivateChannel      0x05e54f08 text      (al32.o)
al3ErrIsr                       0x05e3df2c abs        (al32.o)
al3EnableSBIAAlarm             0x05e59e40 text      (al32.o)
al3EnableDS3AISCells           0x05e59cf0 text      (al32.o)
al3ErrStat                     0x05e3cf24 abs        (al32.o)
al3ErrChan                     0x05e3dc74 abs        (al32.o)
al3EvalReset                   0x05e640f8 text      (al32.o)
al3EnhancedActivateChannelUnstr 0x05e54b78 text      (al32.o)
al3ErrDpr                      0x05e3e174 abs        (al32.o)
al3ErrInit                     0x05e3d418 abs        (al32.o)
al3ErrApi2                     0x05e3c554 abs        (al32.o)
al3ErrApi                      0x05e3c408 abs        (al32.o)
al3ErrDiag                     0x05e3d844 abs        (al32.o)
al3ErrUtil                     0x05e3ce00 abs        (al32.o)
al3EnhancedActivateChannelStr   0x05e54d38 text      (al32.o)
value = 0 = 0x0
->_
```

Figure 12 demonstrates an example of the VxWorks shell command, which lists all the functions and variables in the system that contain the "al3E" string (the command is case sensitive).

In this example, VxWorks lists the names of the functions and variables, where they are located and the module that function or variable belongs to.

At any given point in time the user can do something like:

```
al3EvalReset 0xfa000000
```

'0xfa000000' is the parameter that has to be supplied to the al3EvalReset function. One must have access to the source code or source code documentation to know which parameters have to be supplied to a particular function.

Therefore, the VxWorks shell syntax is the same syntax that has to be used in the module source code.

Instead of typing separate commands at the command prompt, the user can put all the commands into the shell script and call the script as:

```
-> < script.txt
```

where ‘->’ is the VxWorks command prompt sign and “< script.txt” is telling VxWorks shell to take the input from the script.txt file.

11 APPENDIX B: CONTENT OF THE STARTUP.VXS

STARTUP.VXS is a special file: it is executed automatically upon the startup of the system. The name of the file is arbitrary and is specified in the default boot line during the compilation of the VxWorks boot code.

```
#####  
# 06 Jun 2000  
# File: startup.vxs  
# Script file for VxWorks boot from the HDD. This assumes that default  
# boot line points to the HDD.  
#####  
  
# Give FDD a name.  
usrFdConfig(0,0,"a:")  
  
# Give HDD a name.  
usrAtaConfig(0,0,"c:")  
  
# Set the environment.  
< c:/vxworks/scripts/VxEnv.txt  
  
# Load temux, aallgator-32 and duplex driver from the HDD.  
ld (1,0,"c:/VxWorks/drv/temux.o")  
ld (1,0,"c:/VxWorks/drv/al32.o")  
ld (1,0,"c:/VxWorks/drv/duplex.o")  
  
# Start the driver APIs: temux, aallgator32 and duplex.  
cpAppMain  
cpAl3Init  
cpDpxInit  
  
# Set the shell prompt  
shellPromptSet ("AAL1-32/TEMUX $ >")
```

Figure 13 – Example STARTUP.VXS script

This script will attach the drive letters to the floppy and hard disks, execute another script (VxEnv.txt), load the object modules (temux.o, al32.o and duplex.o) from the hard drive and start all the modules by calling their main entries: cpAppMain, cpAl3Init and cpDpxInit.

At the end, the VxWorks shell prompt is set (system function call to **shellPromptSet**).

12 APPENDIX C: MS-DOS FILES NEEDED FOR THE AUTOMATIC INSTALLATION

There is a minimum of five MS-DOS files required for the automatic software installation:

- io.sys - MS-DOS core file for the bootable HDD
- msdos.sys - MS-DOS core file for the bootable HDD
- command.com - MS-DOS kernel
- choice.com - required during automatic installation procedure
- mscdex.exe - MS-DOS CD-ROM support

13 APPENDIX D: DRIVER MAKEFILE

There are certain environment dependencies in the application software makefiles that the user has to be aware of:

WIND_BASE. This is specific to the Tornado 2 development system environment. If the build process is running outside of the Tornado 2 development environment, WIND_BASE has to be set to the root directory of the Tornado 2 development environment (e.g. SET WIND_BASE=d:\bin\tornado2).

MY_APP. Root directory of the applications. This environment variable has to point to the root directory of the AAL1GATOR-32/TEMUX software chip drivers.

MY_TARG. Points to the Tornado 2 **target** subdirectory.

Note that all of the AAL1GATOR-32/TEMUX software chip drivers have to be compiled separately (every software chip driver has its own makefile).

14 APPENDIX E: HOW TO RECREATE TORNADO 2 BUILD ENVIRONMENT

On the supplied distribution CD-ROM the user has everything that is necessary to recreate the Tornado 2 build environment. In this way the VxWorks kernel can be rebuild and new user options can be added.

The following steps install the software development environment and are specific to the Tornado 2.0 VxWorks environment. Users installing the code on a different environment should use this as a reference only.

- Modify your config.h by hard coding the #include "prjparams.h" file at the bottom of the file of your BSP.
- Modify sysLib.c to include the file sysPci.c provided in the BSP directory (put #include "sysPci.c" statement at the end of the include section of the sysLib.c file above the sysModel function).
- In the sysLib.c, add the following code fragment to the function sysHwInit:

Existing code:

```
#if defined (INCLUDE_FEI) || defined (INCLUDE_FEI_END)
    sys557PciInit ();
#endif /* INCLUDE_FEI || INCLUDE_FEI_END */
```

Add the following code:

```
/* PMC PCI support */
#ifdef INCLUDE_PMC_PCI_LIB
    sysfPmcAdapterPciInit ();
#endif /* INCLUDE_PMC_PCI_LIB */
```

Existing Code:

```
#ifdef INCLUDE_LN_97X_END
    sysLan97xPciInit ();
#endif /* INCLUDE_LN_97X_END */
```

- Add eight more DUMMY_MMU_ENTRY entries to the sysPhysMemDesc structure. There are a two sections where this has to be added:

within #if (VM_PAGE_SIZE == PAGE_SIZE_4KB) section

and within #else section.

Add the sysPci.c and sysPci.h files provided to the BSP directory.

15 APPENDIX F: TARGET SYSTEM SELF-TEST

The target system self test can be run from the VxWorks shell. Type the test command "a132SelfTest %a, %b" where:

- %a is the PCI slot number of the AAL1GATOR-32/TEMUX CHIPSET board
- %b is the "animate" mode parameter. Animate mode displays the register writes in a visible manor to the terminal. This parameter can be 0 or 1 (1 for animate mode).

This self-test performs two board level tests. The first test uses the pseudo-random sequence generator and detector to test the line side of the TEMUX. A PRBS 23 pattern is transmitted by the generator into the DS3 LIU. The DS3 LIU is configured for loopback using the jumper (J5). Alternatively, an external loopback cable can be used to loopback the data, but the cable is not provided. The receive line interface of the TEMUX receives the data and verifies that the pattern is being received without errors.

The second test transmits cells with framed DS3 AIS data. The S/UNI-DUPLEX then loops the cells back on the UTOPIA interface. The test then checks the AAL1gator for dropped cells and sequence number errors. The data is then transmitted across the SBI bus and verified by the TEMUX.

You will see status information on the screen that indicates the progress of the self-test. Once the test has been finished the following message on the target machine screen will appear:

```
AAL1-32/TEMUX board in slot [%d] self-test PASSED!
```

To run the AAL1gator-32/COMET-QUAD self-test please, do the following:

Configure the board for COMET-QUAD operation by shorting jumper J8. Type the test command "a132SelfTest_CometQ %a, %b" where:

- %a is the slot number of the AAL1GATOR-32/TEMUX card
- %b is the "animate" mode parameter. Animate mode steps through each of the writes to the registers at a visible pace and displays the values on the terminal. This parameter can be 0 or 1 (1 for animate mode).

Note that the %b parameter is optional and defaults to 0.

Type a132SelfTest_CometQ with the PCI slot number containing AAL1GATOR-32/TEMUX board in it.

The AAL1gator-32/COMET-QUAD Self Test verifies the ability of the development kit to configure the AAL1gator-32, COMET-QUAD, and S/UNI-DUPLEX for correct operation. This test also verifies the datapaths between the three devices. The test configures the COMET-QUAD to the overwrite the entire frame transmitted on the lowspeed link with PRBS data. The AAL1gator-32 is then configured to circuit emulate the 4 unframed T1 links over ATM cells. The ATM cells are then transmitted across the UTOPIA bus where they are looped back by the S/UNI-DUPLEX.

16 APPENDIX G: TARGET CONSOLE APPLICATION LEVEL DEBUG MESSAGES

During the development cycle of the product it is extremely important to know what is happening in the system at any given point of time.

The debugging messages will appear on the target system console.

AAL1GATOR-32/TEMUX Development Board drivers offer the following application level debug message settings:

- **dbgAppLogOn** - turn on application level debug messages.
- **dbgAppLogOff** - turn off application level debug messages.
- **dbgAppLogLvlNone** - no application level debug messages.
- **dbgAppLogLvlVerb** - verbose application level debug messages. In this mode all application level debug messages will be displayed on the terminal console.
- **DbgAppLogLvlMed** - medium application level debug messages. In this mode only more significant application level debug messages will appear on the terminal console.
- **dbgAppLogLvlErr** - error application level debug messages. In this mode only critical application level debug messages will appear on the terminal console.
- **dbgAppLogLvlCstmr** - customer application level debugging messages. In this case only the customer application level debug messages will appear on the terminal console.
- **getAppLogLvl** – returns the application debug messages level (NONE, VERB, MED, ERR, CSTMR).
- **getAppLog** – returns the application debug messages status (ON or OFF).

The application level debug messages must be turned ON in order for application debug messages to appear on the target system console (dbgAppLogOn). The application debug message level can be set at the target console or from the remote machine.

17 APPENDIX H: AUTOEXEC.BAT EXAMPLE

This following is the minimal configuration of the **autoexec.bat** for the automatic software installation on the target system:

```
@ECHO OFF
PROMPT $p$g
PATH=C:\

rem #
rem # The following line was added by NEC ATAPI Installation
rem #
c:\cdrom\mscdex.exe /d:neccd /m:10
```


18 APPENDIX I: CONFIG.SYS EXAMPLE

The following is the minimal configuration of the **config.sys** for the automatic software installation on the target system:

```
FILES=30
BUFFERS=80
SHELL=C:\COMMAND.COM c:\ /e:6000 /p

rem #
rem # The following line was added by NEC ATAPI Installation
rem #
device = c:\cdrom\nec_ide.sys /d:neccd /v
```

The CD-ROM driver, `nec_ide.sys` has to be replaced with the MS-DOS CD-ROM device driver for the particular CD-ROM drive located on the target platform.

19 APPENDIX J: DRIVER SOURCE CODE AND BINARIES FILES LIST

AAL1GATOR-32 Software Driver Source Files
./aal1gator-3/makefile
./aal1gator-3/driver/inc/al3_api.h
./aal1gator-3/driver/inc/al3_chan.h
./aal1gator-3/driver/inc/al3_dev.h
./aal1gator-3/driver/inc/al3_diag.h
./aal1gator-3/driver/inc/al3_dpr.h
./aal1gator-3/driver/inc/al3_hw.h
./aal1gator-3/driver/inc/al3_init.h
./aal1gator-3/driver/inc/al3_isr.h
./aal1gator-3/driver/inc/al3_mdb.h
./aal1gator-3/driver/inc/al3_que.h
./aal1gator-3/driver/inc/al3_rtos.h
./aal1gator-3/driver/inc/al3_stat.h
./aal1gator-3/driver/inc/al3_util.h
./aal1gator-3/driver/inc/types.h
./aal1gator-3/driver/src/al3_api.c
./aal1gator-3/driver/src/al3_api2.c
./aal1gator-3/driver/src/al3_chan.c
./aal1gator-3/driver/src/al3_diag.c
./aal1gator-3/driver/src/al3_dpr.c
./aal1gator-3/driver/src/al3_hw.c
./aal1gator-3/driver/src/al3_init.c
./aal1gator-3/driver/src/al3_isr.c
./aal1gator-3/driver/src/al3_que.c
./aal1gator-3/driver/src/al3_rtos.c
./aal1gator-3/driver/src/al3_stat.c
./aal1gator-3/driver/src/al3_util.c
./aal1gator-3/ginc/al3_cmn.h
./aal1gator-3/parser/inc/cp_init.h
./aal1gator-3/parser/inc/cp_usr2.h
./aal1gator-3/parser/src/cp_init.c
./aal1gator-3/parser/src/cp_pv2.c
./aal1gator-3/parser/src/cp_test2.c
./aal1gator-3/parser/src/cp_usr2.c
./aal1gator-3/test/inc/inceval.h
./aal1gator-3/test/inc/hw_pci.h
./aal1gator-3/test/inc/rtos_vxw.h
./aal1gator-3/test/src/eval.c
./aal1gator-3/test/src/hw_pci.c

AAL1GATOR-32 Software Driver Source Files

./aal1gator-3/test/src/rtos_vxx.c

Table 3 - AAL1GATOR-32 Source Files List

TEMUX Software Driver Source Files
./temux/makefile
./temux/driver/inc/temux.h
./temux/driver/inc/tmx_api.h
./temux/driver/inc/tmx_app.h
./temux/driver/inc/tmx_dev.h
./temux/driver/inc/tmx_dpr.h
./temux/driver/inc/tmx_hw.h
./temux/driver/inc/tmx_isr.h
./temux/driver/inc/tmx_mdb.h
./temux/driver/inc/tmx_rtos.h
./temux/driver/inc/tmx_util.h
./temux/driver/src/tmx_api.c
./temux/driver/src/tmx_app.c
./temux/driver/src/tmx_diag.c
./temux/driver/src/tmx_dpr.c
./temux/driver/src/tmx_hw.c
./temux/driver/src/tmx_isr.c
./temux/driver/src/tmx_rtos.c
./temux/driver/src/tmx_stat.c
./temux/driver/src/tmx_util.c
./temux/parser/inc/cp_app.h
./temux/parser/inc/cp_cmds.h
./temux/parser/inc/cp_debug.h
./temux/parser/inc/cp_error.h
./temux/parser/inc/cp_loop.h
./temux/parser/inc/cp_pci.h
./temux/parser/inc/cp_serial.h
./temux/parser/inc/cp_usr1.h
./temux/parser/inc/cp_usr2.h
./temux/parser/inc/cp_usr3.h
./temux/parser/inc/cp_utils.h
./temux/parser/src/cp_app.c
./temux/parser/src/cp_cmds.c
./temux/parser/src/cp_debug.c
./temux/parser/src/cp_error.c
./temux/parser/src/cp_loop.c
./temux/parser/src/cp_pci.c
./temux/parser/src/cp_pv1.c

TEMUX Software Driver Source Files
./temux/parser/src/cp_pv2.c
./temux/parser/src/cp_serial.c
./temux/parser/src/cp_test1.c
./temux/parser/src/cp_usr1.c
./temux/parser/src/cp_usr2.c
./temux/parser/src/cp_usr3.c
./temux/parser/src/cp_utils.c
./temux/test/inc/app.h
./temux/test/inc/eval.h
./temux/test/inc/hw_pci.h
./temux/test/inc/rtos_vxw.h
./temux/test/src/app.c
./temux/test/src/eval.c
./temux/test/src/hw_pci.c
./temux/test/src/rtos_vxw.c

Table 4 - TEMUX Source Files List

DUPLEX Software Driver Source Files
./duplex/makefile
./duplex/driver/inc/dpx.h
./duplex/driver/inc/dpx_api.h
./duplex/driver/inc/dpx_err.h
./duplex/driver/inc/dpx_hw.h
./duplex/driver/inc/dpx_rtos.h
./duplex/driver/inc/dpx_test.h
./duplex/driver/src/dpx.c
./duplex/driver/src/dpx_api.c
./duplex/driver/src/dpx_hw.c
./duplex/driver/src/dpx_rtos.c
./duplex/driver/src/dpx_test.c

Table 5 - DUPLEX Source Files List

COMET-QUAD Software Driver Source Files
./cometq/makefile
./cometq/driver/inc/cmq_api.h
./cometq/driver/inc/cmq_defs.h
./cometq/driver/inc/cmq_err.h
./cometq/driver/inc/cmq_fns.h
./cometq/driver/inc/cmq_hw.h
./cometq/driver/inc/cmq_pci.h
./cometq/driver/inc/cmq_rtos.h

COMET-QUAD Software Driver Source Files
./cometq/driver/inc/cm_q_strs.h
./cometq/driver/inc/cm_q_typs.h
./cometq/driver/src/cm_q_api.c
./cometq/driver/src/cm_q_diag.c
./cometq/driver/src/cm_q_frm.c
./cometq/driver/src/cm_q_hw.c
./cometq/driver/src/cm_q_inbd.c
./cometq/driver/src/cm_q_intf.c
./cometq/driver/src/cm_q_isr.c
./cometq/driver/src/cm_q_pci.c
./cometq/driver/src/cm_q_rtos.c
./cometq/driver/src/cm_q_ser.c
./cometq/driver/src/cm_q_sig.c
./cometq/driver/src/cm_q_stats.c
./cometq/driver/src/cm_q_util.c
./cometq/parser/inc/cp_cm_q_error.h
./cometq/parser/inc/cp_cm_q_main.h
./cometq/parser/src/cm_q_app.c
./cometq/parser/src/cm_q_app.h
./cometq/parser/src/cm_q_debug.c
./cometq/parser/src/cm_q_debug.h

Table 6 - DUPLEX Source Files List

SHARED Software Driver Source Files
./shared/inc/hw.h
./shared/inc/shr_idnt.h
./shared/inc/shr_type.h

Table 7 - SHARED Source Files List

Tornado 2 BSP Add-Ons
Prjparams.h
syspci.c
syspci.h

Table 8 - Tornado 2 BSP Add-On Files

Executables
al32.o
temux.o

Executables
duplex.o
cmq_drv.o
VxWorks
VxWorks.sym
startup.vxs
pppcfg.txt
vxenv.txt
install.bat

Table 9 - Executables List

NOTES

CONTACTING PMC-SIERRA, INC.

PMC-Sierra, Inc.
105-8555 Baxter Place Burnaby, BC
Canada V5A 4V7

Tel: (604) 415-6000

Fax: (604) 415-6200

Document Information: document@pmc-sierra.com

Corporate Information: info@pmc-sierra.com

Application Information: apps@pmc-sierra.com

(604) 415-4533

Web Site: <http://www.pmc-sierra.com>

None of the information contained in this document constitutes an express or implied warranty by PMC-Sierra, Inc. as to the sufficiency, fitness or suitability for a particular purpose of any such information or the fitness, or suitability for a particular purpose, merchantability, performance, compatibility with other parts or systems, of any of the products of PMC-Sierra, Inc., or any portion thereof, referred to in this document. PMC-Sierra, Inc. expressly disclaims all representations and warranties of any kind regarding the contents or use of the information, including, but not limited to, express and implied warranties of accuracy, completeness, merchantability, fitness for a particular use, or non-infringement.

In no event will PMC-Sierra, Inc. be liable for any direct, indirect, special, incidental or consequential damages, including, but not limited to, lost profits, lost business or lost data resulting from any use of or reliance upon the information, whether or not PMC-Sierra, Inc. has been advised of the possibility of such damage.

© 2000 PMC-Sierra, Inc.

PMC-2001068

Issue date: December 2000