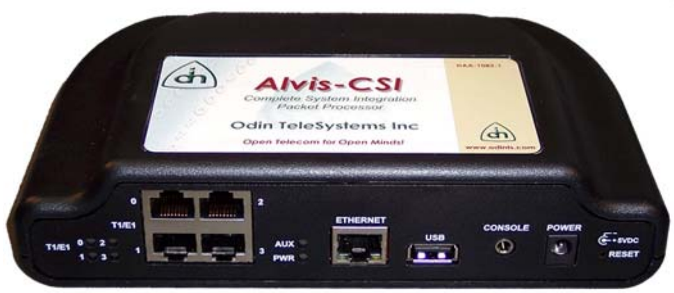
Odin TeleSystems Inc.



Alvis-CSI
Technical Description

Doc. No. 1111-1-HCA-1020-1

Rev. 1.5

Table of Contents

1 Abstract

This document provides a technical description of Odin TeleSystems' Alvis-CSI (Complete System Integration) product cards. This presentation is targeted to systems integrators and application developers who are developing telecommunications systems and/or software applications for Alvis-CSI utilizing the Odin Telecom frameworX (OTX). The purpose of this document is to provide the needed information about the hardware to allow software developers to efficiently integrate Alvis-CSI into their overall system under design.

For information on how to develop host applications utilizing the OTX Hardware Device Driver Application Programming Interface (API), please refer to the "Programmer's Guide for OTX Hardware API" document (Odin TeleSystems Inc. document number 1411-1-SAA-1006-1).

2 Alvis-CSI Overview

The Alvis-CSI is a member of the Odin Telecom frameworX (OTX) product family. It is supported by the OTX device driver (Linux only) and accessed through the OTX Hardware Application Programming Interface (API).

Alvis-CSI is a stand-alone system designed to run custom application for TDM and IP applications. Alvis-CSI has one Digital Media Processor a.k.a System-On-Chip (SOC).

Alvis-CSI is currently available in 2 different options:

- Alvis-4-CSI: 1 TMS320DM6443 System On Chip (SOC) processor and 4 T1/E1 transceivers (4 transmit and 4 receive).
- Alvis-4M-CSI: 1 TMS320DM6443 System On Chip (SOC) processor and 4 T1/E1 receivers (no transmit).

This document is applicable to both Alvis-CSI versions.

The DM6443 device contains two cores. One core is a C64x+ DSP code rated at 4752 MIPS. The other core is an ARM9 core clocked at 297 MHz. Both cores share the same memory map and can very efficiently process Digital Media, such as voice and video encoding and decoding.

TDM data can be routed between the processor (DMP) and the Quad T1/E1 device on the Alvis-CSI board. After the data has been encoded or decoded by the DSP core of an Alvis processor it can easily be packaged by the ARM9 core (running MontaVista Linux or Windows CE) and transferred to and from the Ethernet port. Ethernet packages can leave the Alvis board via a 100BaseT Ethernet port.

TDM data can also be decoded and encoded directly by the T1/E1 Line interface device (e.g. HDLC receiver/send functionality).

The DSP core on the Alvis daughter board can also be used to run Odin provided standard DSP applications or they can be used to run customized user applications. Alvis-CSI is delivered with a number of Odin's Standard Program Modules (SPM) that provide supports for many common telecom applications; such as audio and video encoding and decoding, tone detection and generation, and HDLC sending and receiving.

For custom DSP application development, Alvis-CSI supports the Texas Instruments development tools, such as Code Composer Studio. These tools can be purchased directly from Texas Instruments or from any of their distributors.

For more information on custom DSP application development, please refer to the "Programmer's Guide for OTX C64x+ DSP Software Development Kit" (Odin document number 1412-1-SAA-1014-1).

Equipped with the appropriate OTX software modules, Alvis-CSI can be utilized in a variety of Voice over IP (VoIP), TVoIP, Soft-switch, Trans-coding and Signaling System #7 (SS#7) applications.

3 Installation

The Alvis-CSI is supplied with a power adapter and a console cable (in addition to the main unit). Figure 1 shows the main unit of Alvis-4-CSI with all external interfaces connected.

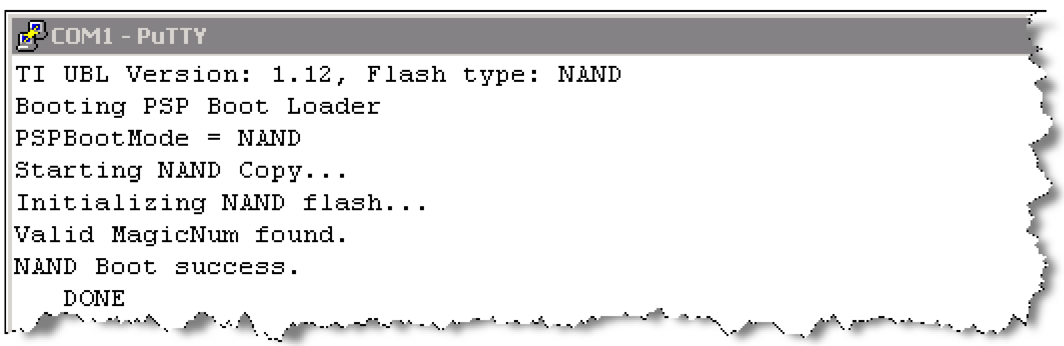
Figure 1

The power adapter has different inserts that will match the wall-power plug in most regions of the world. See Figure 2 below.

Figure 2

To install the Alvis-CSI follow these steps:

1. Remove the protective insert from the power adapter.
2. Insert the proper wall-power insert into the power adapter.
3. Plug the power adapter into a wall socket (100V-240V AC, 50Hz-60Hz).
4. Connect the console cable to connector labeled “CONSOLE” on the main unit. Plug the other end of the cable into a COM port of a PC.
5. Run a terminal emulator program (e.g. PUTTY) on the PC configured for 115200 baud, 8 data bits, 1 stop bit, no parity.
6. Plug the power cable into the connector labeled “POWER” on the main unit.
7. You should see boot messages on the console. See Figure 3.
8. Once the boot process completes you should see a login prompt. Log in as ‘root’. The password is blank (no password). See Figure 4.



```
COM1 - PuTTY
TI UBL Version: 1.12, Flash type: NAND
Booting PSP Boot Loader
PSPBootMode = NAND
Starting NAND Copy...
Initializing NAND flash...
Valid MagicNum found.
NAND Boot success.
   DONE
```

Figure 3

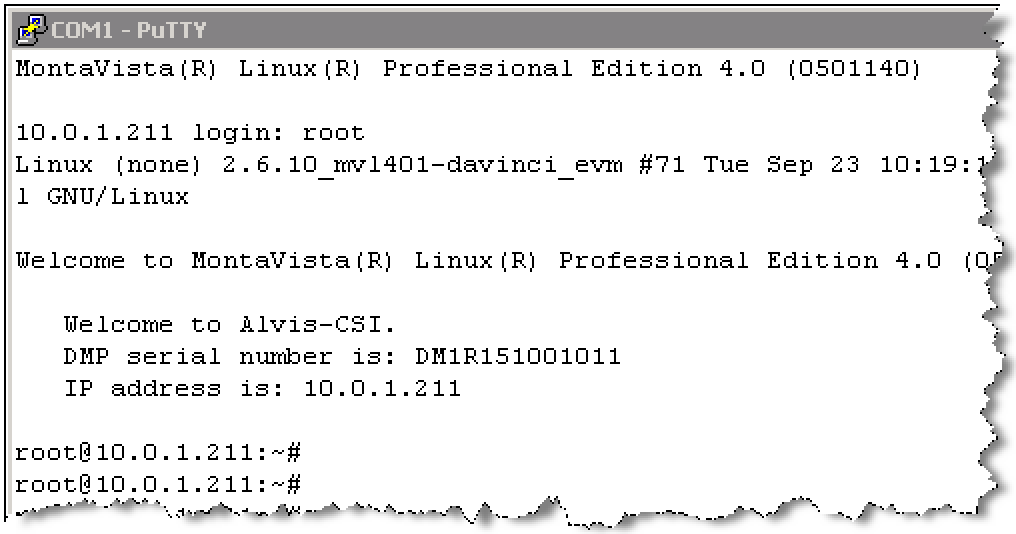


Figure 4

4 System Architecture

The overall system architecture can be best described and understood through different architectural views or aspects. This document explores the systems architecture from the following angles:

1. Physical Specifications View. Provides the dimensions of the board.
2. External Interface View: The external interface view describes the external interfaces of the adapter board, and how they are connected to the various internal devices and modules.
3. Data Architecture View: The data architecture view illustrates how the Time - Division Multiplexed (TDM) serial data is connected and transferred through the board.
4. Control Architecture View: The control architecture view describes how the internal devices and modules can be controlled by the host processor.
5. Clock Architecture View: The clock architecture view specifies what clocking and synchronization options are available, how clocking is derived, and how it distributed to the various devices.
6. Logical Subsystem View: The logical subsystem view describes the logical design subsystems in the system. Each subsystem can comprise hardware, firmware and driver or on-board processor software.

It is important to note that one device within the board can be involved in several of these views, each view describing how one aspect of the device interfaces with other devices.

4.1 Physical Specifications

The main unit of Alvis-CSI can be placed on a desktop it can be wall-mounted. The physical dimensions of Alvis-CSI are shown in Figure 5 and Table 1.

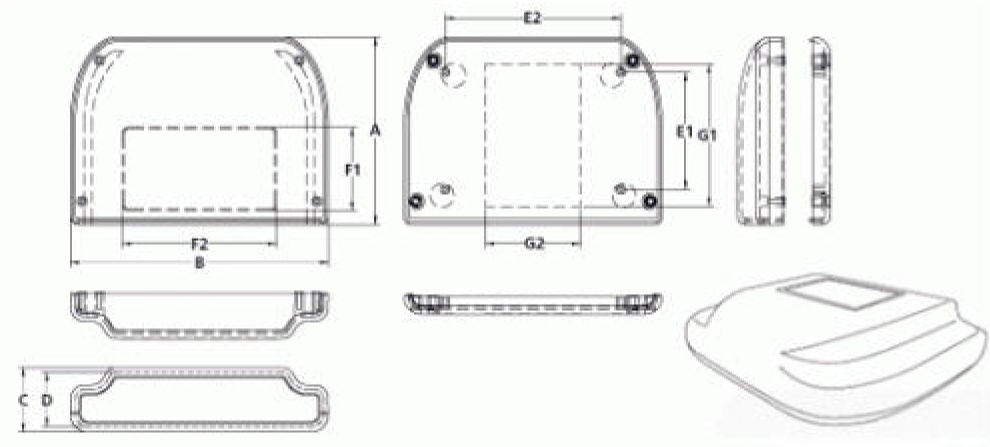


Figure 5

Enclosure Measurements:

	Measurement (inches)
A	5.000
B	7.500
C	1.750
F1	2.281
F2	4.531
G1	3.030
G2	2.030

Table 1

Enclosure material information:

Material: Flame retardant ABS plastic
Color: Black
Surface: Textured finish
Flame Rating: ULs best flame rating of 94-5VA.

4.2 External Interfaces

The external interfaces are accessed through the front panel of the main unit. The different variants of the Alvis-CSI all have different front panels. See below.

4.2.1 Front Panel for Alvis-4-CSI

The front panel for the Alvis-4-CSI is show in Figure 6

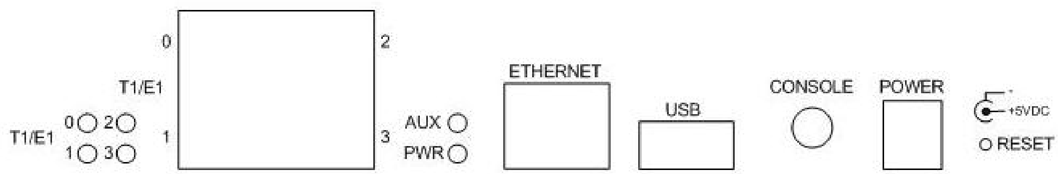


Figure 6

4.2.2 Front Panel for Alvis-4M-CSI

The front panel for the Alvis-4M-CSI is show in Figure 7.

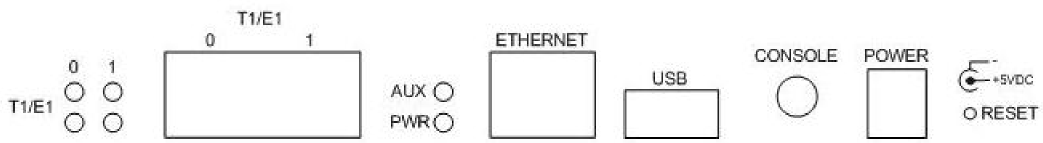


Figure 7

4.2.3 Power Connector

The Alvis-CSI is powered from a 5V DC supply. The power connector is located on the front panel (labeled “POWER”). A DC power adapter (HMA-1167-1) is supplied with the Alvis-CSI product.

If another power adapter is used (other than HMA-1167-1), please make sure it can supply adequate current, and that the power plug has the correct polarity (the center pin is the positive +5V voltage).

See also chapter 8.

4.2.4 T1/E1 Connectors

The E1/T1 interfaces are accessed through the front panel of the main unit. The different variants of the Alvis-CSI all have different E1/T1 connector configurations. See below.

4.2.4.1 T1/E1 Connectors on Alvis-4-CSI

The Alvis-4-CSI has 4 RJ-45 connectors. Each connector uses two pairs (one for transmit and one for receive). They are labeled 0, 1, 2, 3 (one connector for each T1/E1 Line Interface ()). See Figure 8.

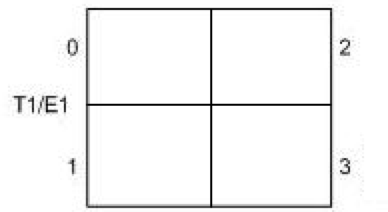


Figure 8

4.2.4.2 T1/E1 Connectors on Alvis-4M-CSI

The Alvis-4M-CSI has 2 RJ-45 connectors. Each connector uses to pairs (one for receive in the near direction, and one for receive in the far direction). The left-most T1/E1 connector is connected to Li0Rx and Li1Rx. The right-most T1/E1 connector is connected to Li2Rx and Li3Rx. See Figure 9.

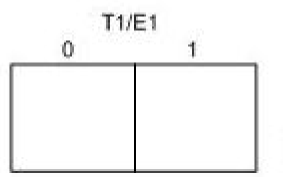


Figure 9

4.2.5 Console Connector

The Alvis-CSI board has a console (serial port) connector (3.5 mm plug) on the front panel (labeled “Console”).

The console can be used to debug and control both cores of the Digital Media processor.

The serial port cable (HMA-1154-1) is included with the Alvis-CSI unit.

4.2.6 Ethernet connector

The Ethernet (100BaseT) connector is accessible via the connector labeled “ETHERNET” on the front panel.

The pinout for the 100BaseT Ethernet connector is shown in Table 2.

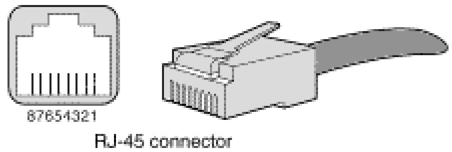


Figure 10

Pin #	Signal Name	Function
1	BI_DA+	Bi-directional pair +A
2	BI_DA-	Bi-directional pair -A
3	BI_DB+	Bi-directional pair +B
4	-	-
5	-	-
6	BI_DB-	Bi-directional pair -B
7	-	-
8	-	-

Table 2

Note that the 100BaseT Ethernet connector be connect to the slower Ethernet speeds (10BaseT) as well as 100BaseT.

4.2.7 USB Connector

The Alvis-CSI boards have a USB 2.0 connector accessible from the front panel. The USB is configured host mode, but can be changed to device/slave mode by changing a jumper setting (see chapter 4.3.1.1).

4.2.8 Indicators

The Alvis-CSI boards contain several LEDs for status and debugging purposes. The locations of the LEDs are shown in the chapters below.

The dual colored (Red/Green) AUX LED can be controller by a user application running on the DMP. Please see Table 3 and the OTX_ATTR_BRD_LED_STATE attribute in the OTX HW API for further details.

LED value	LED color
0	Off
1	Constant Red
2	Constant Green

Table 3

The T1/E1 LEDs indicate the following states:

LED color	Indication
Off	No T1/E1 application is running on the board. The board device is not opened.

Constant Red	Error indication (Alarm or Framing error)
Constant Green	No errors. Link status is good.

4.2.8.1 Alvis-4-CSI Indicators

The Alvis-4-CSI LED indicators are shown in Figure 11:

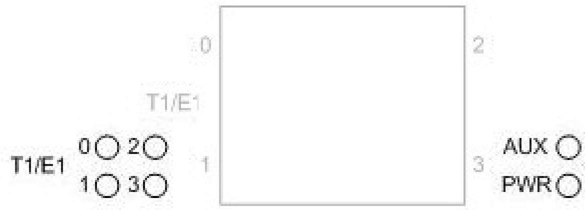


Figure 11

4.2.8.2 Alvis-4M-CSI Indicators

The Alvis-4M-CSI LED indicators are shown in Figure 12:

Figure 12

4.3 Internal interfaces

The internal interfaces (debug connectors and configuration jumpers) are accessed by disassembling the enclosure. This is done by removing the 6 screws from the bottom of the enclosure.

4.3.1 Configuration Jumpers

The Alvis-CSI has a set of configuration jumpers that determines its mode of operation.

Note:

It is important to remove the power plug (see chapter 4.2.3) before changing any of the configuration jumpers.

4.3.1.1 USB Host/Device mode selection (JP1)

The USB port on Alvis-CSI can operate either in host (default) or slave mode. When the port is configured in host mode you can connect slave device like thumb/flash drives and web cameras to the USB port. When the port is operated on device/slave mode the Alvis-CSI can be connected to another host device (like a laptop PC).

JP1 is a three-position jumper that determines the operation mode of the USB port.

Jumper in Pin 1-2 - Host mode (default)
Jumper in Pin 2-3 - Device/Slave/Gadget mode

The location of JP1 is show in Figure 13.

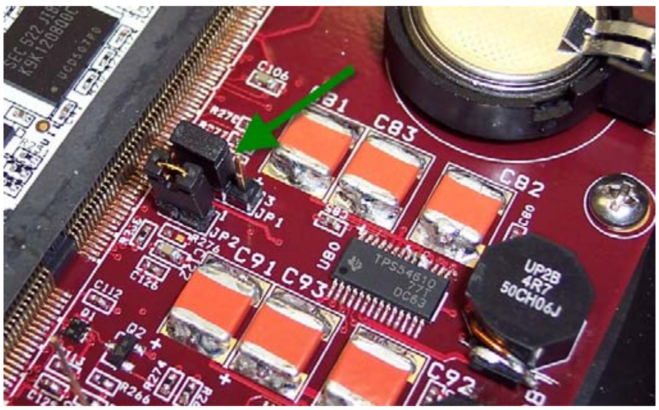


Figure 13

4.3.1.2 Boot mode (JP2 and JP3)

JP2 and JP3 are two-position jumpers that determine the boot mode for the DMP on Alvis-CSI. JP2 is populated (closed) by default. JP3 is also populated (closed) by default.

The configuration options are shown in the table below.

JP2	JP3	Function
Closed	Closed	Boot from NAND Flash (default)
Open	Closed	Boot from UART (to reload A-boot)
Closed	Open	DSP Selfboot (for use with CodeComposer Studio)
Open	Open	N/A

The location of JP2 is show in Figure 14.

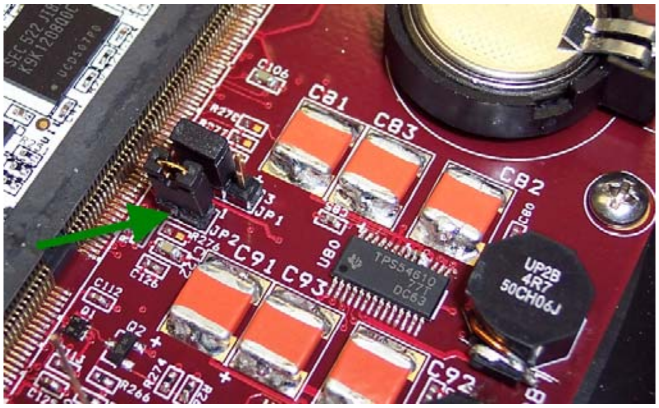


Figure 14

The location of JP3 is shown in Figure 15.

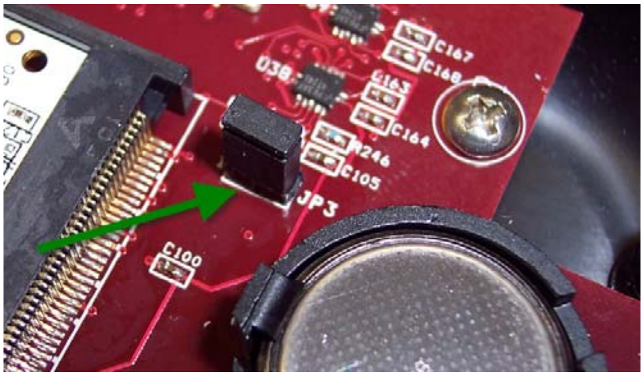


Figure 15

4.3.2 Battery for the RTC (Real-Time-Clock)

The RTC (Real-Time-Clock) on Alvis-CSI has a backup power supply in the form of a battery. The battery is located in the battery socket labeled BT1. See Figure 16.

Figure 16

Battery Type: BR2032 (3V)

The battery is mounted in the socket with the '+' side facing down.

The battery supplied with the Alvis-CSI has an expected capacity of 225 mAh. Current is only drawn from the battery if the power connector (see chapter 4.2.3) is disconnected (unpowered). The battery should last for 2 years in the unpowered state, and much longer (up to 7 years) in the powered state.

To replace the battery push the plastic tab on the left, and pry the battery out of the socket with the help of a pointy plastic (non-conductive) device.

The date and time of the RTC can be set using the 'hwclock' Linux command. The RTC uses UTC time (independent of the TZ setting in Linux). See man pages for hwclock (8) for further details.

An example of setting the date and time in the RTC and then transferring that time to the system time (the 'date' command) is shown below:

```
root@(10.0.1.150):/tmp# date
Sun Feb  1 00:24:54 CST 2004
root@(10.0.1.150):/tmp# hwclock --set --date "10/2/08 13:08:25"
root@(10.0.1.150):/tmp# hwclock --hctosys
root@(10.0.1.150):/tmp# date
Thu Oct  2 13:09:12 CDT 2008
root@(10.0.1.150):/tmp#
```

4.3.3 JTAG Connector (via OTX NIC board)

The Alvis-CSI contains several JTAG chains which are used for board testing and board configuration during the manufacturing process. The JTAG chains are accessed from the Hermod-JTAG connector marked J2 on the Alvis-CSI PCB. Note that the enclosure has to be unassembled in order to access the J2 connector.

TMS#	Devices
TMS1	Quad T1/E1 device
TMS2	RTC Processor
TMS3	DMP device
TMS4	Ethernet PHY

Table 4

The TMS3 chain covers the DMP devices exclusively. A DSP software DSP emulator can connect to chain to facilitate development and debugging of customized DSP applications. To connect to the standard 13-pin JTAG connector from the DSP emulator to the OTX NIC board, the Hermod-JTAG-Alvis-CSI adapter board (HMA-1166-1) is needed. The Hermod-JTAG-Alvis-CSI board plugs onto the JTAG connector (J2), and the DSP Emulator connects to the Hermod-JTAG-Alvis-CSI board.

Note that you can not use the DSP Emulator with any other Hermod-JTAG board other than the one (Hermod-JTAG-Alvis-CSI) that is specifically designed to be used with Alvis-CSI.

4.4 Data Architecture

4.4.1 TDM Data Path

The 4 T1/E1 data streams are multiplexed into one 8.196 MHz TDM highway (128 timeslots) by the Quad T1/E1 device. This 8.196 MHz The TDM data can be transferred to/from the DMP via the Burst device (see chapter 7.1.5).
The timeslot mapping of the burst highway is shown in Table 5. E1 data streams have 32 timeslots. T1 data streams have 24 timeslots but are mapped into 32 timeslots (1 through 24) by the Line interface device. TS0 of the 32 Li timeslots in T1 mode contains the DL-bit in the most significant bit.

Li0	Li timeslots [32]	Burst highway timeslots [128]
0	TS0, TS1, TS2, …, TS31	TS0, TS4, TS8, … , TS124
1	TS0, TS1, TS2, …, TS31	TS1, TS5, TS9, … , TS125
2	TS0, TS1, TS2, …, TS31	TS2, TS6, TS10, … , TS126
3	TS0, TS1, TS2, …, TS31	TS3, TS7, TS11, … , TS127

Table 5

4.4.2 Packet Data Path

The data packets travel via a 100BaseT Ethernet connector at the front panel of the Alvis-CSI unit.

4.5 Control Architecture

The controlling application in the Alvis-CSI is normally a user mode application running in the ARM core of the DMP device. This application would call OtxConnectLib() to connect to the OTX library to control the 4 line interfaces and other physical and logical devices hosted by the Alvis-CSI board.

For further details of OTX Programming please refer to the Programmer’s Guide for OTX Hardware Driver and the demo program supplied with the Linux Alvis SDK.

4.6 Clock Architecture

The Alvis-CSI can be clocked from 5 different sources.

- OTX_CLOCK_SOURCE_LOCAL_0 - LIs synchronized to Li0 receive span
- OTX_CLOCK_SOURCE_LOCAL_1 - LIs synchronized to Li1 receive span

- OTX_CLOCK_SOURCE_LOCAL_2 - LIs synchronized to Li2 receive span
- OTX_CLOCK_SOURCE_LOCAL_3 - LIs synchronized to Li3 receive span
- OTX_CLOCK_SOURCE_INTERNAL - LIs synchronized to on-board oscillator

The clock source set by calling the OtxBrdSetClocks() function.

4.7 Logical Subsystem

The logical subsystem view describes the logical design subsystems within the Alvis-CSI board. Each subsystem can comprise hardware, firmware, and driver or on-board processor software. The Alvis-CSI comprises of only one subsystem:

- Processor Subsystem

4.7.1 Processor Subsystem

The processor subsystem on the Alvis-CSI board contains one Texas Instruments TMS320DM6443 Digital Media Processor (DMP). These processors are sometimes also referred to as “System On Chip” since they are equipped with two different cores; one DSP C64x+ core and one ARM9 core.

The DSP core is clocked at 597 MHz and with an 8 lane wide instruction cache it is rated at 4752 C64x+ MIPS. The ARM9 core is clocked at 297 MHz.

Each DMP is connected to 256 Mbytes of DDR2 memory and a 64 MByte NAND Flash memory device for persistent storage. Both the DSP core and the ARM core share access to the memory devices. Both cores operate in the same memory space.

The processor subsystem is depicted in Figure 17.

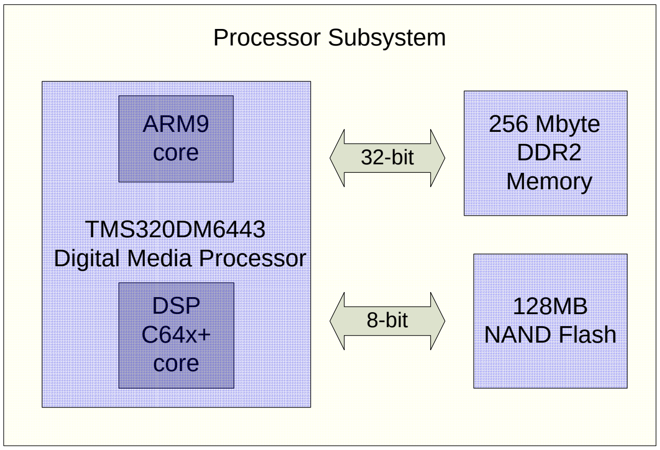


Figure 17

In addition, the processor subsystem provides two single-color LEDs which are normally controlled by the OTX driver. These LEDs (labeled ‘A’ and ‘B’) are located on the white Alvis-DMM board which is inside the main unit.

The processor LEDs indicate the following states:

LED A (red)	LED B (green)	Indication
Off	Off	No power or A-boot failed to start.
Off	On-steady	A-boot started, and the processor is booting
On-steady	On-steady	The OTX driver has loaded
On-steady	On-blinking	The T1/E1 interface has been configured and enabled

5 Web Interface

The Alvis-CSI unit can be configured through a standard web interface. To access the web interface simply type the IP address (e.g. 10.0.1.2) of the Alvis-CSI unit in any internet browser:

Enter the root password (which is blank [no password] by default).

For more details for the Web interface see the Alvis-DMP Programmers Guide (Doc. No. 1412-1-HCA-1022-1).

6 Usage Examples

6.1 Using the SNMP features

Alvis-CSI contains an SNMP daemon that can be used to report various error conditions (if they would occur). The SNMP daemon is configured on the SNMP panel in the web interface (see chapter **Error! Reference source not found.**).

When an SNMP event is triggered the SNMP daemon sends events (traps) to a trap receiver. The following process describes how to configure the trap receiver.

1. You need a computer (SNMP manager) running Linux or Windows. This procedure describes the steps for a Linux computer.

2. Get the latest version of the net-snmp distribution from www.net-snmp.org and install it on the computer that will become the SNMP manager computer (trap receiver).

3. Configure the SNMP agent on Alvis-CSI to send SNMP traps to this host (see Trap Receiver in chapter **Error! Reference source not found.**).

4. Create a text file called ‘snmptrapd.conf’ and place the following text in that file:

   authCommunity log public
   logOption f/var/log/snmptrapd.log

5. Run the ‘snmptrapd’ daemon on the SNMP Manager computer (trap receiver):

   snmptrapd -c /path/to/snmptrapd.conf

   (The SNMP traps will be logged to the log file /var/log/snmptrapd.log by default.)

6.2 Using Alvis-CSI with RNDIS (USB Device mode)

The USB 2.0 port on the Alvis-CSI unit can be configured in Device (a.k.a. Gadget or Slave) mode. In this mode the unit can be directly connected to a laptop of PC via a male-to-male USB cable (type A connector). For instructions on how to switch the USB port to Device mode please refer to chapter 4.3.1.1.

Please follow these instructions to use Alvis-CSI with RNDIS (in Windows):

1. Make sure that the Alvis-CSI unit is flashed with a kernel operating in USB gadget mode.

2. Download the ‘RNDIS_AlvisCsi.zip’ driver package from the Odin web site. Unzip the package to a temporary directory.

3. Open the enclosure and switch the USB port to Device mode.

4. Plug the power cord in Alvis-CSI (this step might not be needed if the USB host is capable of powering the Alvis-CSI unit).

5. When Alvis-CSI has completed its booting process, plug in the male-to-male USB cable (one end connects to the Alvis-CSI connector marked USB (see Figure 7) and the other end connects to a USB port of a laptop or PC).

6. The Windows Hardware Wizard should detect a new device and it should ask for a driver. Point it to the directory where you unzipped the ‘RNDIS_AlvisCsi.zip’

driver package. The driver should install and you should see the following device in the Device Manager:

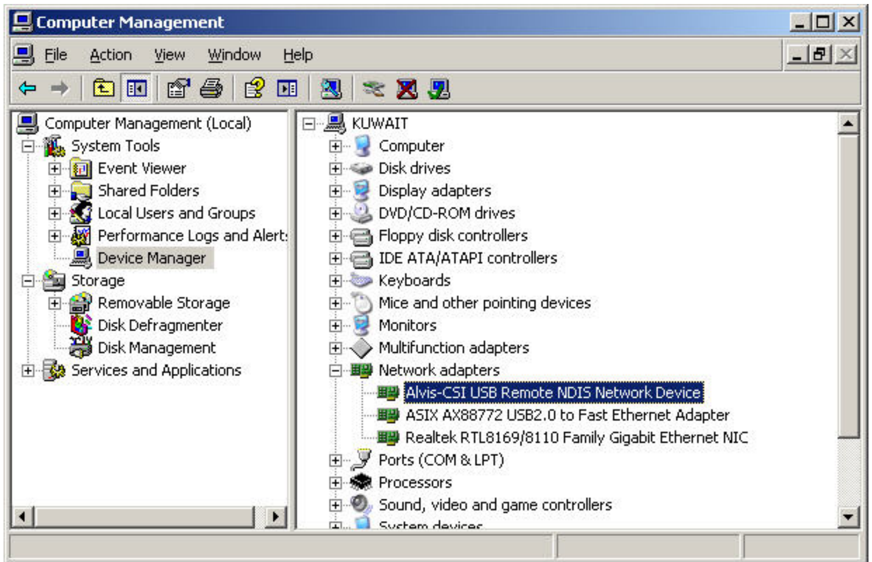


Figure 18

7. The Alvis-CSI device defaults to IP address 10.0.2.2 (and subnet mask 255.255.255.0). Running ifconfig from the Alvis-CSI console should show something similar to the following:

```
root@10.0.1.2:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:C2:81:70:4B
          inet addr:10.0.1.2  Bcast:10.0.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:19 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1721 (1.6 KiB)  TX bytes:0 (0.0 b)
          Interrupt:13

eth0:otpd Link encap:Ethernet  HWaddr 00:50:C2:81:70:4B
          inet addr:1.0.1.1  Bcast:1.255.255.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:13

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

usb0      Link encap:Ethernet  HWaddr E6:90:9E:AD:24:BF
          inet addr:10.0.2.2  Bcast:10.255.255.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

root@10.0.1.2:~#
```

Figure 19

8. If needed the IP address of the usb0 interface can be changed with ‘ifconfig’

commands or via the web interface:

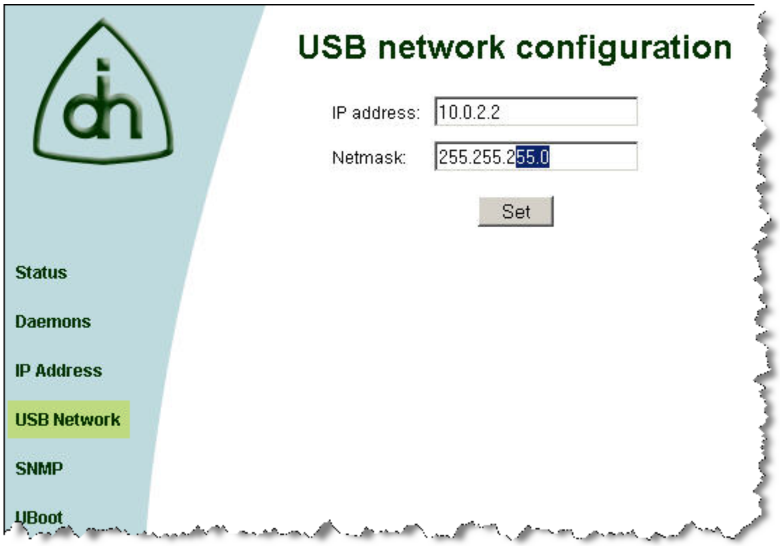


Figure 20

9. Open the Network Devices. Configure IP address and subnet mask of the Alvis-CSI Remote NDIS Network Device. For example choose IP address 10.0.2.10. See figure below:

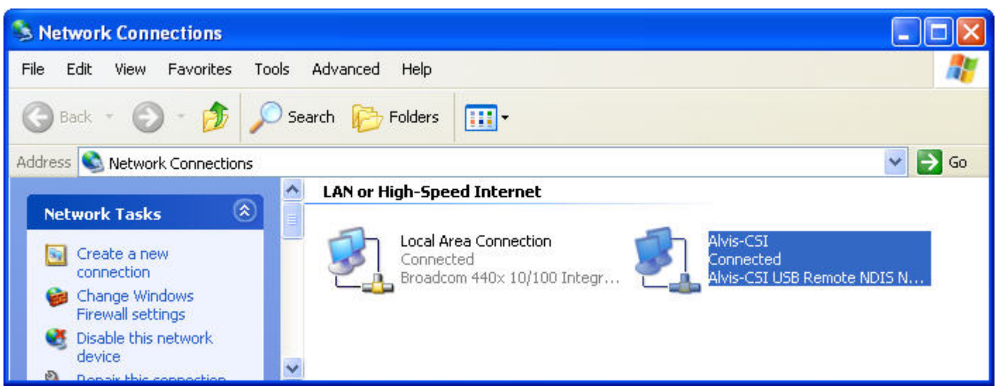
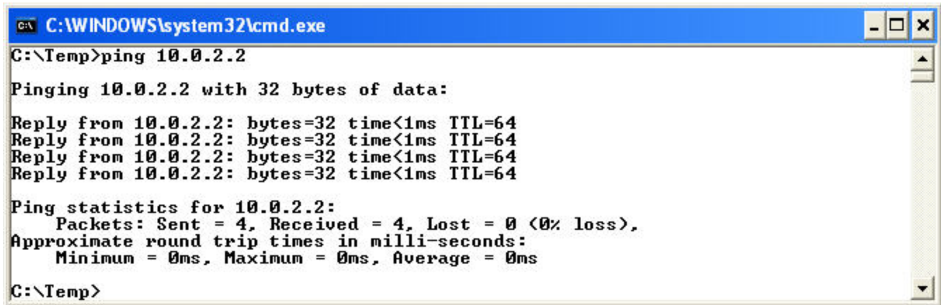


Figure 21

10. Test the interface by pinging the Alvis-CSI unit from the PC or laptop. See example below:



```
C:\WINDOWS\system32\cmd.exe
C:\Temp>ping 10.0.2.2

Pinging 10.0.2.2 with 32 bytes of data:

Reply from 10.0.2.2: bytes=32 time<1ms TTL=64
Reply from 10.0.2.2: bytes=32 time<1ms TTL=64
Reply from 10.0.2.2: bytes=32 time<1ms TTL=64
Reply from 10.0.2.2: bytes=32 time<1ms TTL=64

Ping statistics for 10.0.2.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Temp>
```

Figure 22

7 API Supported Devices

7.1 Physical Devices

7.1.1 Board Device

7.1.1.1 Alvis-4-CSI Board Device (OTX_DEVICE_DMP_ALVIS_4_CSI)

The Alvis-4-CSI board itself is considered a physical device in the OTX HW terminology. An application should always start by opening the OTX_DEVICE_DMP_ALVIS_4_CSI board device (after connecting to the OTX Library), and thereafter open any of the devices that this boards hosts providing the handle to the OTX_DEVICE_DMP_ALVIS_4_CSI board as the "parent" parameter.

The sequence number of the Alvis-4-CSI board device is always 0 (zero). The Alvis-4-CSI board hosts a number of other devices listed in the chapters below.

In order to receive any interrupts from the board device or its hosted devices the device must be enabled (see the OtxDrvEnable() function in the OTX HW API).

7.1.1.2 Alvis-4M-CSI Board Device (OTX_DEVICE_DMP_ALVIS_4M_CSI)

The Alvis-4M-CSI board itself is considered a physical device in the OTX HW terminology. An application should always start by opening the OTX_DEVICE_DMP_ALVIS_4M_CSI board device (after connecting to the OTX Library), and thereafter open any of the devices that this boards hosts providing the handle to the OTX_DEVICE_DMP_ALVIS_4M_CSI board as the "parent" parameter.

The sequence number of the Alvis-4M-CSI board device is always 0 (zero). The Alvis-4M-CSI board hosts a number of other devices listed in the chapters below.

In order to receive any interrupts from the board device or its hosted devices the device must be enabled (see the OtxDrvEnable() function in the OTX HW API).

7.1.2 Digital Media Processor (DMP) Device (OTX_DEVICE_DMP)

The Alvis-CSI board hosts one of Digital Media Processor devices OTX_DEVICE_DMP).

The sequence number of this DMP device is 0 (zero).

The DMP physical serves no meaning function at this time. It is reserved for future use.

When a DMP device is opened it is booted into its standard configuration. The boot image can be changed using the OtxDmpUploadFile() API function. The Boot image is stored in a persistent NAND flash device on the Alvis-CSI board.

The DMP device has two cores:

- ARM9 core
- DSP C64x+ core

The ARM9 core runs either Linux or Windows CE as its operating systems. The ARM core can run standard OTX libraries such as OTXRTP and OTXHDLC. It is also possible to run custom supplied libraries and applications.

The ARM9 core also controls the DSP core. It loads the DSP program into memory region for the DSP and then executes the program.

The DSP core of the DMP devices on the Alvis-CSI can also be used to run the Standard Program Modules (SPMs) provided in the OTX SDK library. These SPMs, or DSP application packages, provides supports for many common telecom applications; such as tone detection and generation, FSK, and HDLC sending and receiving.

The DSP core can also be used to run user developed custom applications. For more information on custom DSP application development, please refer to “Programmer’s Guide for OTX C64x DSP Software Development Kit” (Odin document number 1412-1-SAA-1014-1).

7.1.3 Quad T1/E1 Device (OTX_DEVICE_QUAD_T1E1)

The Quad E1/T1 device (OTX_DEVICE_QUAD_T1E1) is the host for 4 individual E1/T1 devices (OTX_DEVICE_LI_T1E1). In order to open the individual E1/T1 devices this device should be opened first. In order to get

The sequence numbers of this device is always zero (0).

For more information regarding the Ethernet switch device please see chapter 4.4.2.

7.1.4 T1/E1 Line Interface Device (OTX_DEVICE_LI_T1E1)

There are four T1/E1 Line interface devices (OTX_DEVICE_LI_T1E1) on the Alvis-CSI board. These are hosted by the Quad E1/T1 device (OTX_DEVICE_QUAD_T1E1.

The sequence numbers of these devices are 0 through 3.

7.1.5 TDM Burst Device (OTX_DEVICE_BURST)

This device is available on Alvis-CSI units of version 1.1 and later.

The TDM burst device (OTX_DEVICE_BURST) can be used to transfer the E1/T1 data stream from all for Line Interfaces to and from the board device (OTX_DEVICE_DMP_ALVIS_4_CSI and OTX_DEVICE_DMP_ALVIS_4M_CSI).

In the transmit direction the user application periodically provide a buffer data to the burst device which in turns passes it on the each Line Interface device to be transmitted on the T1/E1 span.

In the receive direction the user application is notified periodically that a buffer of received data is available to be read. The user application will then read each buffer from the Burst device.

The sequence numbers of this device is zero (0).

7.2 Logical Devices

The logical devices on the Alvis-CSI board can either be hosted by the T1/E1 Line interface device or by the DSP core.

7.3 Logical Devices hosted by the T1/E1 Line Interface

The logical devices that can be hosted by the T1/E1 device are listed below:

- OTX_LDEVICE_HDLC_SENDER
- OTX_LDEVICE_HDLC_RECEIVER
- OTX_LDEVICE_T1_DL_BIT_ACCESS
- OTX_LDEVICE_T1_ROB_BIT_ACCESS
- OTX_LDEVICE_E1_ABCD_BIT_ACCESS
- OTX_LDEVICE_E1_S_BIT_ACCESS

7.4 Logical Devices hosted by the DSP Core

The Alvis-CSI board supports a number of logical devices depending on which Codec Engine Server and which Standard Program Module that is loaded into the DSP core.

A subset of the supported devices is listed below:

- OTX_LDEVICE_VOICE_CODEC_G723_ENCODER – logical device that compresses (encodes) voice audio in 30 ms frames according to the ITU-T G.723.1 codec standard. G.723.1 is commonly used in Voice over IP (VoIP) applications.
- OTX_LDEVICE_VOICE_CODEC_G723_DECODER – logical device that

decodes voice audio according to the ITU-T G.723.1 standard.

- OTX_LDEVICE_VOICE_CODEC_G729_ENCODER – logical device that support audio data compressing (encoding) according to the ITU G.729 algorithm (Annex A and B). Operates at 6.4 kbps, 8 kbps, and 11.8 kbps. G.729AB is commonly used in Voice over IP (VoIP) applications.
- OTX_LDEVICE_VOICE_CODEC_G729_DECODER - logical device that support audio data decoding according to the ITU G.729 algorithm (Annex A and B).
- OTX_LDEVICE_TONE_DTMF_DIALER – logical device that generates DTMF dial tone sequences.
- OTX_LDEVICE_TONE_MF_DIALER - logical device for generating MF tone sequences.
- OTX_LDEVICE_TONE_DTMF_DETECTOR - logical device for detecting DTMF tones.
- OTX_LDEVICE_TONE_MF_DETECTOR - logical device for detecting MF tones.
- OTX_LDEVICE_TONE_DATA_HDLC_SENDER - logical device for sending (encoding) HDLC frames.
- OTX_LDEVICE_TONE_DATA_HDLC_RECEIVER - logical device for receiving (decoding) HDLC frames.

8 SNMP Monitoring of OTX Physical and Logical Devices

The monitoring of OTX Physical and Logical Devices is available with SNMP that provides the exchange of management information between agents located on network devices, and managers, located at the management system.

The Alvis-CSI network element includes following parts (See Figure 23):

- **MIB** (Management Information Base) describes the structure of the management data of the device subsystem and allows to set management information exchange parameters. MIB use the notation defined by ASN.1.
- **snmpd -** the SNMP agent which binds to a port and awaits requests from SNMP management software. Upon receiving a request, it processes the request, collects the requested information and/or performs the requested operation and returns the information to the sender.
- **HW Layer -** the OTX Hardware Layer.
- **user app -** the application that runs on the OTX Hardware and provides the management information's collection and transferring.
- **OTX SNMP Lib** is used to exchange management information between user app and snmpd. The library extends the functionality of SNMP agent to get the necessary information from user app. It is supported by Linux and Davinci systems.

SNMP agents request management information from the hardware using variables (such as "free memory", "name of the system", "number of active processes", etc). Snmpd accesses the Hardware Layer through the OTX SNMP Lib with the help of user app. There is an API that allows user application to set values of variables described in the MIB file with the help of OTX SNMP Lib functions (such as "OtxSnmpSetValue").

Other OTX SNMP Lib API functions (such as “OtxSnmpGetValue”) allow snmpd to request the values of the variables and then to response that monitoring information to the management software. So snmpd requests the values of counters and other scalar variables from OTX Physical and Logical Devices through OTX SNMP Lib and transfers that to client. It is possible to work with tables too.

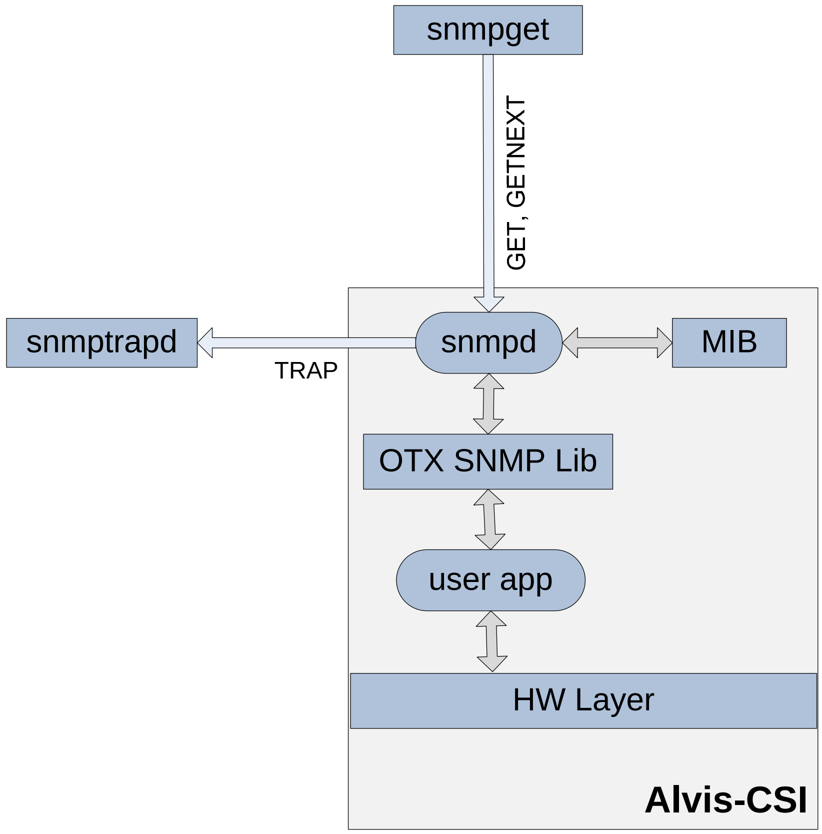


Figure 23

There is a demo MIB file “ODINTS-T1E1-LI-MIB.txt” that defines the following variables:

- **liNumber -** the number of T1E1 line interfaces (regardless of their current state) present on this system (for Alvis-CSI, liNumber = 4).
- **liTable -** the T1E1 Line Interfaces table realized as the list of interface entries. The number of entries is given by the value “liNumber”.
- **liEntry -** contains management information applicable to a particular interface (liIndex, liName, liStatus, liT1E1Mode, liLastChange, liErrors).
- **liIndex -** the unique value, greater than zero, for each T1E1 line interface in the managed system. It is recommended that values are assigned consequentially starting from 1.
- **liName -** the T1E1 line interface name.
- **liStatus -** the current operational state of the T1E1 line interface (“up” or “down”).

- **liT1E1Mode** – the current mode of the T1E1 line interface (“t1” or “e1”).
- **liLastChange -** the value of sysUpTime at the time the T1E1 line interface entered its current operational state. If the current state was entered prior to the last re-initialization of the local network management subsystem, then this object contains a zero value.
- **liErrors -** the count of the errors related with this T1E1 line interface.

As an example, please look up the following table:

liIndex	liName	liStatus	liT1E1Mode	liLastChange	liErrors
1	Li1	up	t1	0:0:00.10.00	94
2	Li2	down	e1	0:0:00.15.00	7

The user can create own MIBs, register own variables through API, and develop own applications to provide monitoring information to snmpd.

Snmpd can be configured to monitor a variable by setting a condition. If the condition holds then the trap is sent and the trap receiver registers it in the log (see chapter 5.5 and 6.1). Also user could send traps from the application with the call of API function, such as OtxSnmpSendTrap (see chapter 6.1).

9 Power

The Alvis-CSI operates from 5V DC supplied from a power adapter which connects to the connector labeled “Power” on the front panel.

Alvis-CSI is supplied with a 3A (5V DC) power adapter with international plugs. It can be plugged into a wall socket supplying AC voltage between 100 and 240 V.

The power consumption is 0.66A (with no USB devices connected), with a surge current of up to 3.0 A.

10 Certifications

Final certifications are TBD. The Alvis-CSI will be designed with the following list of planned certifications:

- FCC Part 15 (CFR47, Part 15, Subpart B)
- FCC Part 68
- CE EMC (EN61326-1 Class B Equipment, AS/NZS 2064 Class B Limits)
- Safety EN60950 and UL60950

11 Reference documents

The following documents provide further detailed information related to the Alvis-CSI board:

- Programmer’s Guide for Alvis-DMP (Odin document # 1412-1-HCA-1022-1)
- Programmer’s Guide for OTX Hardware Driver (Odin document # 1412-1-SAA-1006-1)
- Programmer’s Guide for OTX C64x+ DSP Software Development Kit (Odin document number 1412-1-SAA-1014-1)
- Alvis-CSI Product Brief (Odin document number 2020-1-HCA-1020-1)
- Alvis-CSI Asterisk Installation Guide (Odin document number 1512-1-HCA-1020-1)
- Alvis-CSI Firmware Upgrade HOWTO (Odin document number 1412-1-HCA-1020-1)

12 Glossary

- OTX – Odin Telecom FrameworX
- NIC – Network Interface Card. Refers to the OTX Base board that this ASM board is connected to. Examples of OTX NIC cards are Thor-8-PCI-Plus, Thor-2-PCI-Plus, and Thor-2-PCI-Express.
- DMP – Digital Media Processor. Refers to the dual core System On Chip processors on the Alvis board.
- DSP – Digital Signal Processor
- Li – Line Interface (T1/E1)
- SDK – Software Development Kit
- API – Application Programmer Interface
- CPU – Central Processing Unit. Refers to the host PC in this document.
- EEPROM – Electrically Erasable Programmable Read Only Memory.
- FPGA – Field Programmable Gate Array.
- LED – Light Emitting Diode
- LS – Least Significant
- MS – Most Significant