

Alvis-CSI Firmware Upgrade

HOW TO

Doc. No. 1412-1-HCA-1020-1

Rev. 1.1



Copyright

© Copyright 2009, Odin TeleSystems, Inc. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written consent of Odin TeleSystems Inc., 800 East Campbell Road, Suite 334, Richardson, Texas 75081, U. S. A.

Trademarks

Odin TeleSystems, the Odin Logo, OTX, and Alvis-CSI are trademarks of Odin TeleSystems Inc., which may be registered in some jurisdictions. Other trademarks are the property of their respective companies.

Changes

The material in this document is for information only and is subject to change without notice. While reasonable efforts have been made in the preparation of this document to assure its accuracy, Odin TeleSystems Inc., assumes no liability resulting from errors or omissions in this document, or from the use of the information contained herein.

Odin TeleSystems Inc. reserves the right to make changes in the product design without reservation and notification to its users.

Warranties

THE PRODUCT AND ITS DOCUMENTATION ARE PROVIDED “AS IS” AND WITHOUT WARRANTY OF ANY KIND. ODIN TELESYSTEMS EXPRESSLY DISCLAIMS ALL THE WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR PARTICULAR PURPOSE. ODIN TELESYSTEMS DOES NOT WARRANT THAT THE FUNCTIONALITY OF THE PRODUCT WILL MEET ANY REQUIREMENTS, OR THAT THE OPERATIONS OF THE PRODUCT WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT DEFECTS WILL BE CORRECTED. FURTHERMORE, ODIN TELESYSTEMS DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING THE USE OF THE PRODUCT OR ITS DOCUMENTATION IN TERMS OF THEIR CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY ODIN TELESYSTEMS OR ODIN TELESYSTEMS’ AUTHORIZED REPRESENTATIVE SHALL CREATE A WARRANTY. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY.

UNDER NO CIRCUMSTANCE SHALL ODIN TELESYSTEMS INC., ITS OFFICERS, EMPLOYEES, OR AGENTS BE LIABLE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS, PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THE PRODUCT AND ITS DOCUMENTATION, EVEN IF ODIN TELESYSTEMS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT WILL ODIN TELESYSTEMS’ LIABILITY FOR ANY REASON EXCEED THE ACTUAL PRICE PAID FOR THE PRODUCT AND ITS DOCUMENTATION. SOME JURISDICTIONS DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL AND CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY.



Table of Contents

1	Abstract.....	4
2	Overview	4
2.1	Connecting Alvis-CSI board to a host pc	4
2.2	NAND Flash structure	5
2.3	Firmware upgrade package.....	6
2.4	Firmware upgrade parameters	7
3	Firmware upgrade process.....	8
3.1	Setting the parameters	8
3.2	Running the “flash.bat” script	8
3.2.1	U-Boot Upgrade	9
3.2.2	Kernel and File System Upgrade.....	11
3.2.3	Firmware Upgrade with Add-ons	11
4	Reference documents	12
5	Glossary.....	13

1 Abstract

This document provides a HOWTO description of Odin TeleSystems' Alvis-CSI product cards firmware upgrade. 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 framework (OTX).

2 Overview

This section describes the necessary information and preparations before starting the firmware upgrade.

2.1 Connecting Alvis-CSI board to a host pc

For firmware upgrade you need an Alvis-CSI board connected to a host pc via Ethernet and serial ports. To make the connection please follow these steps (See Figure 1, Figure 2):

1. Plug the power adapter into a wall socket (100V-240V AC, 50Hz-60Hz).
2. Plug the power cable into the connector labeled "POWER" on the main unit.
3. Connect the console cable to connector labeled "CONSOLE" on the main unit.
4. Plug the other end of the console cable into a COM port of a PC.
5. Connect a cross-over Ethernet cable to the connector labeled "ETHERNET" on the main unit.
6. Plug the other end of the cross-over Ethernet cable into the Ethernet card of the PC.



Figure 1

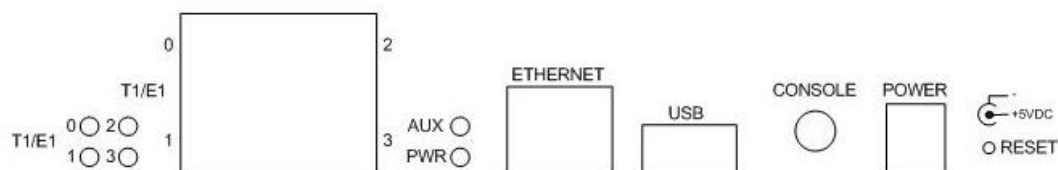


Figure 2

2.2 NAND Flash structure

The NAND Flash is divided into 4 sections:

- **Boot loader (U-Boot)** - is a small program which is loaded first, and starts when you power on the Alvis-CSI or reset it.
- **Kernel** - the main component of the Linux operating system.
- **Root file system (read only)** – the file system that is contained on the same partition on which the root directory is located, and on which all the other file systems are mounted as the system is booted up.
- **User space (read/write)** – the space where the user can write files.

The Alvis-CSI can be populated with two types of the NAND memory (See Figure 3). Each type has a dedicated version of U-Boot:

- **Small** (64Mb)
- **Large** (128Mb)

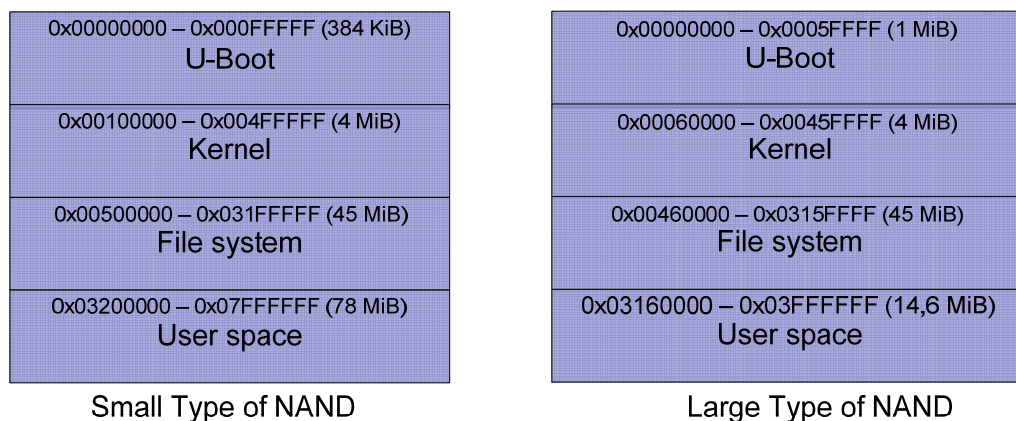


Figure 3

By default, “uboot = large” for all Alvis boards (See the processor subsystem on the Alvis-CSI board in Figure 4).

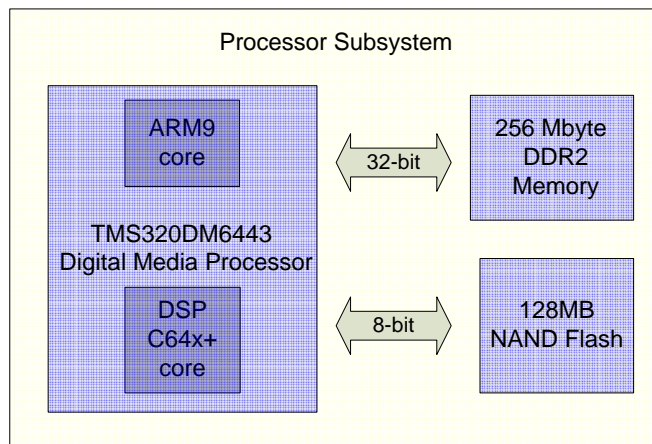


Figure 4

If you specify the wrong NAND type you will get the following warning message:

“Warning: U- Boot version is wrong. You have upgraded small (large) U-Boot while you need large (small)! Are you sure you want to continue? [Y/N]”

2.3 Firmware upgrade package

The firmware on the Alvis-CSI is upgraded via a firmware upgrade package which is represented as an archive that contains a script written in the Python script language and firmware split in a few different files. Unpack the archive. As an example, we will use “factory-1.6-20090505.rar” firmware upgrade package.

The following files and folders are represented:

- **flash.conf** – configuration file, where parameters for the firmware upgrade are stored
- **flash.bat** – executable script that runs the firmware upgrade
- **flash.py** – Python script (is used if a pc has Python interpreter)
- **ubootvars** – file with U-boot variables that are used after upgrading process
- **~variables.backup** – file containing the saved U-Boot variables that will be restored at the end of firmware upgrade
- **dist/** - folder includes compiled files which are needed to run Python script on a pc that has no Python interpreter
- **dvflasher/** - contains the “dvflasher.exe” utility which is used to upgrade the U-Boot; it is not used if we upgrade U-Boot without JP2 switching (see “-2” parameter)
- **logs/** - log files folder
- **pool/** - the folder includes:
 - “uboot.large.bin” or “uboot.small.bin” (depending on NAND memory type) files that contain the upgrade of U-Boot; it includes also “uboot.large.oci” or “uboot.small.oci” if we upgrade U-Boot without JP2 switching
 - uImage – kernel upgraded
 - uImage.softecc – kernel used to upgrade U-Boot



- uImage.gadget – special kernel which can be upgraded instead of uImage. After the firmware upgrading you can make connection between Alvis-CSI and host pc just using usb cable (usb gadget mode).
- “firmware-jffs2.oci” and “firmware-cramfs.oci” files contain file system that is used
- minimal.iso – minimal file system used just during the upgrading process to upgrade the U-Boot and file system contained at “firmware-jffs2.oci” or “firmware-cramfs.oci”
- **tftpd/** - the folder contains “tftpd32.exe” file that can be used as an external tftp (by default, tftp is internal)

It is recommended to use “flash.bat” script for the firmware upgrade because “flash.py” script can be run only on the pc with the Python interpreter and also pc can have different versions of interpreter, different Python modules, etc.

2.4 Firmware upgrade parameters

You can use following firmware upgrade parameters from the command line (See Figure 5):

- **-h** – to display the list of parameters as the help
- **-m** – to set the MAC address
- **-s** – to set the serial number of board
- **-i** – to set the IP address
- **-p** – to set the serial port
- **-c** – to use arguments from the configuration file
- **-u** – to upgrade firmware with the old version of U-Boot
- **-k** – to save U-Boot variables (MAC, serial number, IP, etc.) and recover them after firmware upgrading
- **-2** – to upgrade U-Boot without JP2 switching
- **-a** – to upgrade firmware with Add-on(s)
- **-r** – to recover U-Boot variables saved at the previous try (if loaded with “-k”)

Davinci flashing

Usage: flash.exe [arguments]

The list of possible arguments:

-m, --mac <arg>	Use <arg> mac address
-s, --serial <arg>	Use <arg> serial number
-i, --ipaddr <arg>	Use <arg> ip address
-p, --port <arg>	Use serial port <arg> (com1, 0,...)
-c, --conf <arg>	Use <arg> config file
-u, --nouboot	Skip flash upgrad
-k, --keep	Keep users uboot variables
-a, --add <file>	Upgrade with add-on <file> (multiple)
-2, --nojp2	Do not switch JP2
-r, --recover	Recover U-Boot variables backup

Figure 5

These parameters will be described in further detail below.

3 Firmware upgrade process

3.1 Setting the parameters

First you need to define com port of the pc we will use while firmware upgrade. The serial port number can be set in the configuration file “flash.conf” (parameter “port”) or by the command line option “-p”. In our example we will use “com1” port. So we set “port=com1” in the configuration file or use the command line “flash.bat -p com1”.

Next you should configure your network connection. By default, during the firmware upgrade the Alvis-CSI device has IP address 10.0.1.2 and the host pc running the firmware upgrade package has IP address 10.0.1.1. You can configure these IP addresses in the configuration file if your network is different. The network settings of the configuration file are:

- **server** - IP address of the host PC
- **ipaddr** - IP address of the Alvis-CSI during the firmware upgrade process
- **netmask** - subnet mask

Also you need to be sure that no tftp servers and firewalls are running on the host pc.

Define what firmware to upgrade. The configuration file parameter “firmware” is responsible for that. There are 2 options:

- **“firmware-cramfs.oci”** uses cramfs file system which is faster to upgrade but is not used on the memory with bad blocks.
- **“firmware-jffs2.oci”** uses jffs2 read only file system which successfully deals with bad blocks. The default firmware is jffs2.

3.2 Running the “flash.bat” script

After setting all the necessary parameters we are ready to run the “flash.bat” script. The following console window will appear (See Figure 6):

```
C:\flashing>flash.bat
Davinci flashing
Enter mac address: 00-50-C2-81-70-54
Enter serial number: DM1R151001023

Going to upgrade u-boot...
Please turn off the power on the board then remove the JP2 jumper, turn on the
power again and press "Enter"
```

Figure 6



You will be asked to enter the valid MAC address and serial number for Alvis-CSI. The other way to set the needed MAC address and serial number is the command line with an options “-m” for MAC address and “-s” for serial number. Example:

```
flash.bat -m 00-50-C2-81-70-54 -s DM1R151001023
```

If these parameters appear not valid, you will have the error message: “Variable “mac” (“serial”) does not match pattern”. The result of the successful input is illustrated in Figure 6.

Now we are ready to begin the firmware upgrade.

The firmware upgrade process goes through the following stages:

- U-Boot upgrade
- Linux Kernel upgrade
- File System upgrade
- Firmware upgrade with user’s Add-ons

You can skip the U-Boot firmware upgrade and keep your old one by using the command line option “-u”. Example:

```
flash.bat -u
```

3.2.1 U-Boot Upgrade

U-Boot stores variables for Linux boot and user variables. The file “ubootvars” contains the parameters of the U-Boot. Among them there are system variables which should not be modified and user variables (e.g. IP address) which can be customized. You can also add additional custom variables to this file.

The following is a list of the most commonly modified variables:

- **server** – IP address of the PC from which the firmware upgrade will be downloaded (if tftp is internal, this is the IP address of the PC where the firmware upgrade program is running)
- **ipaddr** – IP address that will be used after firmware upgrading (it has to be placed in the one subnet with the server)
- **netmask** – the subnet mask
- **gatewayip** – IP address of the network gateway (by default, is not used)

It is possible to keep the user variables unchanged during the firmware upgrade. To use this feature run “flasher.bat” with command option “-k”. If this option is provided, variables from the list below (and variables not listed in the “ubootvars” file) are saved and not overwritten by the firmware upgrade:

- fstype
- bootfile
- fsimage
- ethaddr

- serialno
- dmpno
- ipaddr
- hostip
- hosteth
- netmask
- gateway
- serverip
- nfsserver
- nfsdir

By default, U-Boot is upgraded via “dvflasher.exe” utility which is placed in the folder called “dvflasher/” of the given package. It is fast process but it requires switching JP2 jumper on Alvis-CSI motherboard (i.e. the enclosure has to be opened).

If you are asked to start the U-Boot upgrading with turning off the power on the Alvis-CSI board and removing the JP2 jumper, please do it.

JP2 is a two-position jumper that determines the boot mode for the DMP on Alvis-CSI. JP2 is populated (closed) by default.

The configuration options are shown in the table below.

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

The location of JP2 is shown in Figure 7.



Figure 7

Then turn the power on and press “Enter”. You will see the firmware upgrade process on the display.

If U-Boot was upgraded without errors you will see the following message:
“Operation completed successfully. U-Boot is written.”



Then put the JP2 jumper back and reset the board. The firmware upgrade will continue.

There is an experimental method of writing U-Boot from Linux with raw mode and without switching JP2 jumper. You should add “-2” in the command line to use this method. However, this method is not always possible. It requires a working U-Boot on the flash, from which you can load.

If you see a timeout warning (“T T T...” symbols displayed on the screen) you should check the Ethernet connection (cable, settings, etc).

3.2.2 Kernel and File System Upgrade

On this step the firmware upgrade utility transmits the Linux kernel by downloading it via TFTP. This process will take a bit time. Then the minimal file system image transmits. After this the Linux is booted.

Inside the Linux update utility transmits full file system that is contained in the files “firmware-cramfs.oci” or “firmware-jffs2.oci”. After the transmitting it extracts and writes into the flash memory. Upgrade utility determines if the user space is empty or corrupt and copies the default configuration into the user space. After initialization of user space upgrade utility transmits and extracts user add-ons (if given).

If the firmware upgrade is finished successfully you will be invited in MontaVista Linux and will receive the message: “Firmware upgrade is finished successfully” (See Figure 8).

```
Starting OpenBSD Secure shell server: sshd
Starting network management service : done
Starting web server: thttpd

MontaVista(R) Linux(R) Professional Edition 4.0 (0501140)

10.0.1.2 login:

Firmware upgrade is finished successfully

C:\flashing>
```

Figure 8

3.2.3 Firmware Upgrade with Add-ons

The file system on the Alvis-DMP consists of a read-only main file system (over 45Mb) and a read-write user space (over 5Mb for small type of NAND and over 70Mb for large type). It is possible to upgrade any add-on package to the user space during the firmware upgrading.

Below is a list of folders which have read-write access:



- /jffs
- /etc
- /opt
- /home
- /root
- /usr/local

You can make a “tar.gz” archive containing tree structure so that any file in this archive has absolute path within these folders. If you use the command option “-a” with this archive placed in the folder called “pool/” (pointed at configuration file), this archive will be extracted to the user space during the firmware upgrade.

For example, if you want to display a “Hello, world!” message during the loading, create a file “opt/helloworld/hi”, where “hi” is the executable file that displays “Hello, world!” on the screen. Then make a link “etc/rc.d/rc3.d/S01hi” pointing to the file “/opt/helloworld/hi”.

These two folders (“opt” and “etc”) you should pack recursively in the “hello.tar.gz” file and put the archive in the “pool/” folder.

Then you can upgrade firmware with this add-on:

```
flash -a hi.tar.gz .
```

Note that you can use any number of add-ons in one command line:

```
flash -a hi1.tar.gz -a hi2.tar.gz ... .
```

If you have a table of serial numbers and corresponding MAC addresses and you can skip entering the MAC address. The MAC address can be found from this table by using configuration file option “sfile” which equals the string with table file name, delimiter symbol and table column number. The firmware upgrade script will try to find a matching serial number in this table and use the corresponding MAC address from the given column row.

4 Reference documents

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

- Alvis-CSI Technical Description (Odin document # 1111-1-HCA-1020-1)
- 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)



5 Glossary

- OTX – Odin Telecom FrameworX
- DMP – Digital Media Processor
- Alvis – From the Norse mythology - The dwarf Alvis wanted to marry Thrud (daughter of Thor) but Thor tricked him into being above ground when the sun came up, turning him into stone. Alvis is also a product family based on Texas Instruments DaVinci processor.
- NAND – “NOT-AND”
- TFTP – Trivial File Transfer Protocol
- U-Boot – Universal Bootloader