



CC9U Flash Replacement

Migration Guide

90033932_B

1 Document History

| Date | Version | Change Description |
|------------|---------|-------------------------|
| 05/29/2009 | A | Initial entry/outline |
| 05/07/2010 | B | Updated to U-Boot 1.1.4 |

2 Table of Contents

| | | |
|-----|--|----|
| 1 | Document History..... | 2 |
| 2 | Table of Contents..... | 3 |
| 3 | Introduction..... | 4 |
| 4 | Hardware compatibility | 4 |
| 5 | Software compatibility..... | 4 |
| 5.1 | Changes in U-Boot..... | 4 |
| 5.2 | Changes in LxNETES-3.2..... | 5 |
| 5.3 | Default binary images | 5 |
| 6 | Software download | 5 |
| 7 | Software installation | 6 |
| 7.1 | U-Boot source code..... | 6 |
| 7.2 | LxNETES-3.2 patches..... | 7 |
| 7.3 | Precompiled images | 8 |
| 8 | Software development | 9 |
| 8.1 | New LxNETES-3.2 projects | 9 |
| 8.2 | Migration of existing LxNETES-3.2 projects | 9 |
| 9 | Firmware update | 10 |
| 9.1 | New CC9U modules..... | 10 |
| 9.2 | Old CC9U modules | 10 |
| 9.3 | Update of rootfs from U-Boot..... | 10 |

3 Introduction

CC9U modules (former UNC90) manufactured until November 2009, populate a 16MiB NOR Flash chip S29GL128M90TFIR1 by manufacturer SPANSION.

SPANSION announced that this Flash chip is End Of Life (EOL) and no longer being manufactured.

Digi has selected two compatible NOR Flash models to substitute the current one on new CC9U modules:

- NUMONYX M29W128GH
- SPANSION S29GL128P

NUMONYX is a semiconductor company founded in 2008 by Intel and STMicroelectronics.

| |
|--|
| <p>DIGI STRONGLY RECOMMENDS YOU TO CAREFULLY READ THE PRESENT MIGRATION GUIDE TO PROPERLY SUPPORT THE NEW FLASH CHIP OF CC9U MODULES IN OLD AND NEW DEVELOPMENTS.</p> |
|--|

4 Hardware compatibility

From the hardware perspective, the new Flash chip is fully compatible and no hardware modifications are required.

5 Software compatibility

The main difference between the new Flash and the old one is that the new part has 128K erase-block size, while the old part has 64K erase-block size. The new Flash with its different erase-block size requires some software modifications to be done on U-Boot boot loader and also on LxNETES-3.2.

5.1 Changes in U-Boot

The U-Boot source tree has been completely updated to version 1.1.4. The major changes in this version for the CC9U are:

- Added support for the new flash chips
- Faster boot: doesn't require pre-loader.
- Addition of some new commands like dboot, update, envreset
- Use bit banging I2C

If you are developing a customized U-Boot boot loader image for CC9U modules, you will need to download and install the new U-Boot 1.1.4 source code to generate a customized U-Boot which supports the new flash. If you did changes in the past to the U-Boot code, you will need to merge those changes in the new source tree.

You don't need to download the U-Boot source code if not developing a customized U-Boot boot loader.

File:

- CC9U_UBOOT114_SRC_A.zip

5.2 Changes in LxNETES-3.2

The kernel needed several patches to support the new flash and work with the new U-Boot:

- Added support for the new flash chips
- Fix reading of environment variables during IP auto-configuration from EEPROM

Files:

- CC9U_LX32_FLASH_PATCH_80010022_A.zip
- CC9U_LX32_IPCONFIG.zip

5.3 Default binary images

New U-Boot, Kernel and Rootfs binary images have been created basing on the new U-Boot and LxNETES-3.2 code.

Files:

- CC9U_UBOOT114_BIN_82002254_A.zip
- CC9U_LX32_IMAGES.zip

6 Software download

To get the new software source code and compiled binaries please go to <http://www.digi.com/support> for ConnectCore 9U and patches, or use the links below to download the following files:

- http://ftp1.digi.com/support/patches/CC9U_LX32_FLASH_PATCH_80010022_A.zip
- http://ftp1.digi.com/support/patches/CC9U_LX32_IMAGES.zip
- http://ftp1.digi.com/support/patches/CC9U_LX32_IPCONFIG.zip
- http://ftp1.digi.com/support/patches/CC9U_UBOOT114_BIN_82002254_A.zip
- http://ftp1.digi.com/support/patches/CC9U_UBOOT114_SRC_A.zip

7 Software installation

The following topics contain details about the installation of the patches, the creation of LxNETES-3.2 projects for the new modules, the migration of existing LxNETES-3.2 projects, and the support for old modules.

7.1 U-Boot source code

If you are developing a customized U-Boot boot loader image for CC9U modules, you will need to install the new U-Boot 1.1.4 source code to generate a customized U-Boot which supports the new flash.

You can skip this topic if not developing a customized U-Boot boot loader.

Uncompress the U-Boot source code (for example into your \$HOME folder):



```
$ cd ~
$ unzip CC9U_UBOOT114_SRC_A.zip
```

This will create a folder called U-Boot with the new source code.

To build U-Boot, you must have the U-Boot toolchain installed (U-Boot toolchain is provided in the LxNETES-3.2 CD-ROM).



*Contrary to what the README in LxNETES-3.2 CD-ROM states, it is recommended to install the toolchain **out of the U-Boot source tree**. The U-Boot make clean process is known to delete part of the toolchain files if installed inside the same folder.*

For the instructions in this guide, it is assumed that the U-Boot toolchain from LxNETES-3.2 CD-ROM is installed into your \$HOME folder:



```
$ cd ~/
$ tar xvzf $CDROM/upstream/toolchain-uboot.tar.gz
```

This will uncompress the toolchain into a folder called *toolchain*.

To build U-Boot for the CC9U module do the following:



```
$ cd ~/U-Boot
$ export PATH=~/toolchain/bin:$PATH
$ make unc90dev_config
$ make
```



IMPORTANT NOTE: It is needed that the Flash partitions are aligned to 128K sectors (0x20000) when using the new flash. The default addresses for erasing and flashing U-Boot, rootfs, and kernel, are already aligned, but if you modify them, please check that start addresses and sizes of U-Boot scripts *cfu*, *cfk*, *cfp*, *wfu*, *wfk*, and *wfr* are a multiple of 128K.

As a result of the build process, a binary image called **u-boot-un90dev.bin** will be created.

7.2 LxNETES-3.2 patches

Uncompress the ZIP files in a temporary folder:



```
$ cd /tmp
$ unzip CC9U_LX32_FLASH_PATCH_80010022_A.zip
$ unzip CC9U_LX32_IPCONFIG.zip
```

These ZIP files contain patches for the kernel, one each:

- **CC9U_LX32_FLASH_PATCH_80010022_A.patch:** contains the code for supporting the new flash chips. This is the list of files that will be patched:
 - Makefile.in*
 - configure.ac*
 - linux/drivers/mtd/chips/cfi_probe.c*
 - linux/drivers/mtd/chips/cfi_util.c*
 - linux/include/linux/mtd/cfi.h*
- **CC9U_LX32_IPCONFIG.patch:** contains ipconfig fixes for compatibility with new U-Boot 1.1.4. This is the list of files that will be patched:
 - linux/net/ipv4/ipconfig.c*

To install the patches, go into the LxNETES-3.2 installation directory and execute the following commands:



```
$ cd ~/LxNETES-3.2
$ patch -p1 < /tmp/CC9U_LX32_FLASH_PATCH_80010022_A.patch
$ patch -p1 < /tmp/CC9U_LX32_IPCONFIG.patch
```

If you get any conflict during the patch process, you might have custom modifications to any of the patched files that the program cannot resolve. If this happens, you will need to open the conflicting patch file with an editor and merge the changes by hand to your LxNETES-3.2 installation.

After installing the patches, execute **autoconf** to generate a new **configure** script (tested with autoconf v2.65):



```
$ cd ~/LxNETES-3.2
$ autoconf
```

With these patches, any **newly created** LxNETES-3.2 project will support the flash of the new CC9U modules. Migration of existing projects is explained later in this document.



IMPORTANT NOTE: It is needed that the Flash partitions are aligned to 128K sectors (0x20000).

The default partition table is already aligned, but if you have modified it, please check that all partitions start addresses and sizes are a multiple of 128K.

7.3 Precompiled images

The file *CC9U_UBOOT114_BIN_82002254_A.zip* contains the following binary images:

- *u-boot-unc90dev.bin*: *U-Boot 1.1.4-CC9U-RevA* boot loader

This new U-Boot version boots faster than the old one, it uses I2C bit banging, and it implements some new commands.

U-Boot version *U-Boot 1.1.4-CC9U-RevA* comes already flashed in new manufactured modules which populate the new flash chip. Old modules can keep the old version *U-Boot 1.1.3 FS1a* or be upgraded to the new one.

The file *CC9U_LX32_IMAGES.zip* contains the following binary images:

- *uImage-unc90dev*: Linux kernel image

This is the factory default kernel image prepared to work in the flash of the new CC9U modules. This kernel is also suitable for old modules.

- *rootfs-unc90dev-128.jffs2*: JFFS2 root file system image with 128K erase-block size.

This is the factory default root file system image prepared to work in the flash of the new CC9U modules (with 128K erase-block size). **This image is not suitable for old modules.**

- *rootfs-unc90dev-64.jffs2*: JFFS2 root file system image with 64K erase-block size.

This is the factory default root file system image prepared to work in the flash of the old CC9U modules (with 64K erase-block size). **This image is not suitable for new modules.**

8 Software development

8.1 New LxNETES-3.2 projects

Once the LxNETES-3.2 patches are installed, a new project can be created in the standard way:



```
$ cd /home/myusername/myprojects
$ mkdir newProject
$ cd newProject
$ ~/LxNETES-3.2/configure --enable-platform=unc90dev [other-options]
```

After compilation (make) and installation (make install), two root file system images will be created in your TFTP exposed folder:

- **rootfs-unc90dev-128.jffs2**: JFFS2 root file system image with 128K erase-block size. This image is prepared to work **only in new** CC9U modules.
- **rootfs-unc90dev-64.jffs2**: JFFS2 root file system image with 64K erase-block size. This image is prepared to work **only in old** CC9U modules.

The kernel that will be created (**uImage-unc90dev**) will contain the support for the new flash and is also suitable for old modules, as long as the partitions are aligned to 128K.

8.2 Migration of existing LxNETES-3.2 projects

To add support for the new flash on existing LxNETES-3.2 projects, you need to reconfigure the project with the same options you created it in the past:



*To see the options the project was created with, see the file **config.log** before reconfiguring your project.*



```
$ cd /home/myusername/myprojects/oldProject
$ ~/LxNETES-3.2/configure --enable-platform=unc90dev [other-options]
```

After compilation and installation of the project, two root file system images will be created in your TFTPBOOT directory:

- **rootfs-unc90dev-128.jffs2**: JFFS2 root file system image with 128K erase-block size. This image is prepared to work **only in new** CC9U modules.
- **rootfs-unc90dev-64.jffs2**: JFFS2 root file system image with 64K erase-block size. This image is prepared to work **only in old** CC9U modules.

The kernel that will be created (**uImage-unc90dev**) will contain the support for the new flash and is also suitable for old modules, as long as the partitions are aligned to 128K.

9 Firmware update

9.1 New CC9U modules

New CC9U modules must be flashed with the kernel image (**uImage-unc90dev**) and the rootfs image (**rootfs-unc90dev-128.jffs2**) of either a new or a migrated LxNETES-3.2 project.

9.2 Old CC9U modules

Old CC9U modules can keep old versions of U-Boot, kernel, and root file system.

If an update is wanted, old modules can use the new U-Boot, the kernel of a new or migrated LxNETES-3.2 project, and the rootfs image **rootfs-unc90dev-64.jffs2**, which is prepared for 64K erase-block size flash.

9.3 Update of rootfs from U-Boot

U-Boot expects the root file system image to be a file called **rootfs-unc90dev.jffs2**, so you must create a symlink in your TFTPBOOT folder with this name that points to the proper file, depending on the module that you want to update.

If it is a **new** module:



```
$ cd /tftpboot
$ ln -s rootfs-unc90dev-128.jffs2 rootfs-unc90dev.jffs2
```

If it is an **old** module:



```
$ cd /tftpboot
$ ln -s rootfs-unc90dev-64.jffs2 rootfs-unc90dev.jffs2
```

Once done, you can update the rootfs normally with:



```
# run update_rootfs_tftp
```

Another possibility is to set the value of variable **ring** so that in old modules is **rootfs-unc90dev-64.jffs2** and in new modules is **rootfs-unc90dev-128.jffs2**.