

# Non-interactive Flashing using dboxflasher

## Table of contents

1 Introduction.....	2
2 General.....	2
3 A script for automated flashing.....	2
4 Building, or downloading, dboxflasher.....	3
5 Performance.....	3

## 1. Introduction

The present article describes a method for non-interactive flashing of dBoxes using a server with DHCP and TFTP capacities, like a Linux computer.

The use of the u-boot as flashing tool was introduced by Homer in "[dboxflasher mit/ohne Nullmodem-Kabel](#)", however without sources. It was checked in to CVS by yours truly, who also wrote Makefile rules (in newmake) and the script described later.

A serial connection ("nullmodem cable") is not required. However, troubleshooting without it is fairly meaningless.

## 2. General

The dboxflasher really "is" nothing else than a configuration file for the u-boot. This file is checked in in the Tuxbox CVS, [u-boot\\_flasher\\_dbox2.h](#). The interesting part is the boot command, which reads:

```
#define CONFIG_BOOTCOMMAND \
"tftp \"dboxflash.img\";protect off 10020000 107fffff; " \
"erase 10020000 107fffff;cp.l 100000 10020000 1F8000;reset"
```

This means: first, get the file `dboxflash.img` from the TFTP-server. Then make the appropriate area in the flash memory (everything except for the first 128KiB, which is used by the BR boot loader) writable and erase it. Finally, copy the just downloaded file to the just erased flash area. Then reboot.

To use dboxflasher for flashing, the TFTP-server have to first provide it as the first boot-file (replacing the normal u-boot, taking the name from the DHCP-parameters), then provide `dboxflash.img` (for which the name is known). Optionally, logo files `dboxflasher-fb` and `dboxflasher-lcd` may be provided, also by the TFTP-server. Finally, the DHCP-server have to be shutdown by the time the dBox reboots, otherwise the same procedure will start all over again. Preferably, this should all co-exist with a setup of DHCP and TFTP for booting YADDs (see [this document](#)).

To automate these tasks is the object of the following section.

## 3. A script for automated flashing

The script [do-flash](#) is an attempt to satisfy these requirements. It is invoked as:

```
do-flash filename
```

where *filename* is the filename of the image file to be flashed. The script first checks the size of the file, makes a backup of possible previous u-boot, and copies the dboxflasher to its location, and the flash-file to `dboxflash.img`. Then, to be sure we are in a known state, a possibly running `dhcpd` is terminated. `dhcp` is subsequently started, and the dbox (that is assumed to have the IP-name "dbox") is rebooted using the Neutrino web API. Of course, this will reboot the dBox only if running Neutrino, otherwise the

user have to reboot the dBox manually. The script waits for two minutes. During this interval, the dBox is supposed to boot (only the DHCP-related services are required to be completed during this interval). After two minutes, the flashing is supposed not to be completed, the DHCP server is terminated, preventing the flashing to start anew. The original u-boot is then restored. In the case of interruption (with e.g. Control-C), the signal is trapped, dhcp terminated, and the original u-boot restored. The flashing is also logged to a local file.

#### **4. Building, or downloading, dboxflasher**

The dboxflasher configuration file is a part of the Tuxbox CVS. Users of the newmake branch can build it automatically within the CDK setup using the make target `dboxflasher`. This target will install dboxflasher in the tftp-directory.

[Here](#), a compiled dboxflasher can be downloaded.

#### **5. Performance**

My Nokia (2x) is flashed in approximately 2 minutes. Unfortunately, flashing my Sagem (1x) does not work that well, but takes approximately 10 minutes. I do not know why.