Synology - Flash uboot ROM

This guide describes a method how to restore u-boot if the flash has been completely erased. This situation can arise if there was a power loss during u-boot upgrade or you accidentally deleted the boatloader while playing with the u-boot commands like I did. There are many reasons why the DiskStation fails to boot, but as long as the u-boot bootloader partition is not destroyed, you can repair the flash image using the u-boot serial flash subsystem. This guide is for all who are sure that u-boot has been corrupted and that there is no other way to recover.

The solution is to physically replace the corrupted flash chip and restore its original content. I will be using a Dangerous Prototypes Buspirate for serial console access and to flash the firmware externally onto a brand new flash IC. This of course will void your warranty as you have to replace the old chip with the new one. In case you are not experienced with soldering you will risk destroying the mainboard, but in my opinion this is still better than throwing the bricked unit away. The following steps involve the preparation of the flash.rom image, programming and replacing the old chip. The new installation will be verified with the serial console.

1. Download and extract firmware image

This guide is for a DS213j, but should work equally on other NAS.

https://global.download.synology.com/download/DSM/release/6.2/23739/DSM_DS213j_23739.pat

The pat file is an uncompressed tar archive: $ tar -xvf DSM_DS213j_4458.pat The files we need are:
- uboot_DS213jv10.bin: u-boot
- zImage: kernel
- rd.bin: initrd

2. Prepare the flash.rom image

The size of the flash chip on the DS213j is 64Mbit, so we need an 8MB large image. We first recall the flash layout to get the right offsets:
00000000:000bffff RedBoot (uboot) 000c0000:0038ffff zImage 00390000:007cffff rd.gz 007d0000:007dffff vender 007e0000:007effff RedBoot_Config 007f0000:007fffff FIS_directory

Create an empty flash image of 8 MB size filled with zeros: $ dd if=/dev/zero of=flash.rom bs=1024 count=8192 Copy the uboot image to the beginning of the file not truncating the rest: $ dd if=uboot_DS213jv10.bin of=flash.rom conv=notrunc Now copy the kernel image at the right offset. dd cannot directly use hexadecimal values. These have to be converted by the shell: $ dd if=zImage of=flash.rom bs=1 seek=$((0xc0000)) conv=notrunc Next is the initrd image: $ dd if=rd.bin of=flash.rom bs=1 seek=$((0x390000)) conv=notrunc Then comes the vender partition. This is not provided in the download of the firmaware as this is unique to your DiskStation and written only once in the factory. It contains the serial number and the MAC address, which can be

found on the sticker at the back of the case. The DiskStation will boot without the vender information, but without the MAC address it will assign a random MAC at each boot. This will annoy DHCP servers as they serve a new IP on every reboot. Automatic firmware updates and the DS Assistant rely on the MAC address as well.

For the next step you will need the serial number and original MAC from the sticker and a hex editor (I use hexeditor). First we create an empty 64 kB vender partition: $ dd if=/dev/zero of=vender.img bs=1024 count=64 The MAC address is stored in the first 6 bytes and the 7th byte contains a checksum. The checksum is calculated by taking the sum of the bytes in the MAC address. The following C program can be used to calculate the checksum. The checksum is stored in an unsigned char, which is not large enough to store the total. It will flow over give the rest, which is larger than multiples of 256 bytes:
#include <stdio.h> int main() { unsigned char chksum = 0; int i = 0; unsigned char rgAddr[6] = {0x00, 0x11, 0x32, 0xFF, 0xFF, 0xFF}; for (i = 0; i < 6; i++) { chksum += rgAddr[i]; } printf("Checksum: 0x%X\n", chksum); return 0; } The first three bytes are the Synology part, the last three bytes needs to be changed to your MAC address. Compile with $ gcc -o checksum mac_checksum.c This MAC example 00:11:32:FF:FF:FF will give a checksum of 0x40.

Insert these bytes into the image: $ hexeditor -b vender.img 00000000 00 11 32 FF FF FF 40 The serial number is written like this. For example:
SN=1350LAN009999,CHK=999
00000020 53 4e 3d 31 33 35 30 4c 41 4e 30 30 31 39 39 39 00000030 2c 43 48 4b 3d 39 39 39
Use an ascii to hex converter to generate the hex values. The checksum is calculated on next reboot automatically and can be inserted later. We use a dummy of 999 for now. Once u-boot is working again, you can flash the updated vender image from u-boot (see below how to do this). Now insert the vender data into the flash image: $ dd if=vender.img of=flash.rom bs=1 seek=$((0x7d0000)) The flash.rom image is now ready to be flashed onto the chip. The RedBoot_Config and FIS_directory partitions are not needed.

3. Flash the rom image to the new chip

The flash chip on the DS213j is a Micron 25Q064A 13E40 64Mbit serial flash. I bought mine here
https://hbe-shop.de/Art-2253680-MICRON-N25Q064A13ESE40E-FLASH-SERIELL-64MB-3V-8WSOIC



What you also need is a SOIC/DIL 8POL adapter PCB to connect the IC to a breadboard:
https://hbe-shop.de/Art-1136592-ARIES-08-350000-11-RC-ADAPTER-SOIC-DIL-8POL

For the SPI programmer and 3.3V TTL-USB Level shifter I use a Dangerous Prototypes Buspirate:
http://dangerousprototypes.com/docs/Bus_Pirate

The Buspirate is a multi purpose debugging and prototyping device that has SPI, JTAG, UART and

I2C modes. Solder the IC onto the SOIC adapter and then you can use breadboard to connect the Buspirate with the following circuit (http://flashrom.org/Bus_Pirate):

Image not found or type unknown

For the actual programming of the flash.rom image I used flashrom (http://flashrom.org/Flashrom), which supports the Buspirate programmer directly: $ sudo flashrom -c "N25Q064..3E" -V -p buspirate_spi:dev=/dev/ttyUSB0,pullups=on -w flash.rom Now we are set to replace the flash chip on the main board.

4. Replace the flash chip on main board

This is the most difficult part and if you are inexperienced with a soldering iron you should get some practise by de-soldering IC's from old motherboards or scrap electronics. Did I mention that touching the DiskStation PCB with a soldering iron will void your warranty? To de-solder the flash chip the trick is to apply plenty of solder over all pins. The pins will be connected through a thick layer of solder and this will distribute the heat over the pins and keep the solder liquid long enough. This looks messy at first, but it is then possible to apply heat to both sides of the chip simultaneously and lift the IC from the PCB without damaging the pins. If you were successful in de-soldering the flash, you could even spare yourself the hassle of buying a new chip and flashing the old one. I use de-soldering wick to clean the pins of the IC and the pads on the PCB. Then I remove residue flux from the PCB with residue cleaner or isopropanol using an old toothbrush. With the same technique you have to de-solder the IC from the SOIC adapter. Before soldering the reprogrammed chip back in, make sure the pads are free from old solder and shiny clean. Then fix the chip on the board with tape and apply very little solder to two opposite pins first, and afterwards apply solder to the remaining pins.

5. Connecting a serial console

Information on how to connect to the serial console can be found here:

http://www.fr.netbsd.org/ports/sandpoint/instSynology.html

The serial console of the DS213j is available at the 6 pin jumper next to the 3 pin fan connector. Connect the BusPirate pins GND, MISO and MOSI to the jumper pins 2, 4 and 6.

Image not found or type unknown

Pin Function BusPirate
1 3.3 V
2 GND GND
4 TX MISO (RX)

6 RX MOSI (TX)

Start your favourite terminal program and put the BusPirate into UART transparent bridge mode. $ script -f -c "picocom -b 115200 /dev/ttyUSB0" session.log Now it is time to power on the DiskStation and watch the output on the console. If the re-flashing was successful, you should see u-boot loading kernel and initrd images. If you still have DSM installed on the hard drives, the DiskStation should boot normally now, otherwise it will stop at the initrd prompt. You could use the DS Assistant to do a fresh DSM install. Congratulations, you are almost there!

6. Flashing the vendor image using u-boot

The DiskStation will complain that the serial number checksum does not match the one provided (999) and prints the correct checksum on the console. Write it down and update the vender.img with the correct serial number using hexeditor.

EDIT: The checksum of the 13 digit serial number equals to the sum of the individual character bytes, for example SN=1350LAN009999,CHK=744

Now reboot and interrupt the boot process with CTRL-C to get to the u-boot prompt. Copy the vender.img to a TFTP server and set the Diskstation IP with ipaddr and the TFTP server address with serverip. Then transfer the image to memory address 0x2000000: Marvell>> setenv ipaddr 192.168.1.50 Marvell>> setenv serverip 192.168.1.10 Marvell>> tftpboot 0x2000000 vender.img Now initialise the flash subsystem, erase mtd3 partition and write the 64 kB vender.img to the correct address 0x7d0000: Marvell>> sf probe 0 Marvell>> sf erase 0x7d0000 10000 Marvell>> sf write 0x2000000 0x7d0000 0x10000 Marvell>> reset The DiskStation is now restored to a fully working condition. If further problems arise, it is always a good idea to connect the serial console first to see if u-boot is still working. From here you can manually flash new kernel and initrd images or fix problems with the network.

Have a good day!
Alexander

https://community.synology.com/enu/forum/17/post/69287

Revision #1
Created 11 February 2026 13:21:02 by Admin
Updated 24 October 2026 14:12:36 by Admin