Converting Mellanox EMC switch to SX60xx

Version v1.12

Originally prepared by: dodgy route

Please note, this is ***not a guide and should not be used by anyone***, it is purely for my personal research purposes only.

There is no warranty, support or anything implied, as you are not doing anything more than reading this document.

If you use this document for something evil, hired goons will be dispatched at short notice to rectify the situation.

About this document

This document is a compilation of my own testing, learning from my mistakes and reading information from the EMC switch thread on STH which I will try to credit as much as possible. Thanks to mpogr for the initial guide on converting the switch, which this document is a complete replacement for, the process is nothing alike. He sparked the idea, leading to many people performing updates and progressing the document.

The reason this is not an update to mpogr's document is this process is entirely different to convert the switch to a plain Mellanox version. This document is based on my experience with the SX6012 but will ultimately work for SX6018 and SX6036.

This allows one to then perform OS updates from the CLI or GUI, perform factory resets as well as run multiple images without worry of things breaking or not working ever.

This information is available online in bits and pieces, all over the place, I merely compiled it into this document for myself.

Document revision history

2022.01.11 - v1.12 - dodgy route

- As reported by arnbju in heading get current FRU, show devs does not work at this stage, I added to doco post conversion and I could not remember if it worked at the time, so now that it's confirmed its not working its removed

2022.01.11 - v1.11 - dodgy route

- Necrotyr has successfully confirmed the EMC to SX6018 FRU changeover works and followed guide to convert fine
- Necrotyr has confirmed the document works on a previously converted SX6018 in fact, using mpogr's notes
- Reverted v1.10 changed back to MSX6018F-2SRS again post Necrotyr completing testing as the chassis is exactly that

2022.01.10 - v1.10 - dodgy route

- Fixed mistake for SX6018, had MSX6018F-2SRS for FRU, changed to MSX6018F-2SFS, air flowing out connector side

2022.01.10 - v1.09 - dodgy route

- Adding instructions for converting SX6018 with the help of Necrotyr who is hopefully going to test
- Added copies of the FRU change script I modified, based entirely on SGS original scripts to cover SX6018 all versions, once testing done will leave only the EMC one
- Modified Change FRU process to be more friendly for SX6012,SX6018 and SX6036 series
- Added information to Get current FRU heading, was present in appendix but included for sake of clarity
- Clarified PSID numbers for SX6012 and SX6018, do not have information yet for SX6036

2022.01.06 - v1.08 - dodgy route

- Created process for bootloader password disable on images version >= 3.6.5000
- Clarified update process, image delete is only done once both partitions are updated
- Clarified update process, image fetch is only done once both partitions are updated
- Updated about this document description slightly

2022.01.06 - v1.07 - dodgy route

- Updated generate licences with info from andvalb for generating one license key with multiple licences activated
- Updated genlicense topic with new single license key information for enabling ethernet
- Updated loading licenses topic with new single license key information for enabling ethernet

2022.01.05 - v1.06 - dodgy route

- Updated manufacture switches to install bootloader that flashes Mellanox U-BOOT that enables the boot menu
- Removed v1.05 section to add MLNX-OS boot menu manually using official process, now done during manufacture
- Added Appendix 5 – Mellanox U-BOOT environment variables
- Clarified the bootloader password will need to be removed on every image update if need bootloader/U-BOOT access
- Clarified the bootloader password removal needs to be done from _shell

2022.01.04 - v1.05 - dodgy route

- As per previous update, added section how to add MLNX-OS boot menu, the process fixes booting to partition 2
- Added section how to remove bootloader password so can use boot menu
- Added picture to last reboot step showing the CRC change as well as MLNX-OS boot menu availability
- Finished updating the switch software image update process and pictures as that is now working correctly

2022.01.04 - v1.04 - dodgy route

- Added run boot_mlxlinux to switch preparation
- Changed information about fan speed dropping on post conversion reboot to include actual percentages
- Started adding information on upgrade process to bring converted switch up to latest version
- Not a problem but definitely a nuisance, the update process is not booting to partition 2 automatically, but the upgrade process works great, guessing remnant of EMC U-BOOT settings, trying to resolve but help is appreciated. Believe to be related is the switch does not present a menu to boot from which partition, so believe U-BOOT related

2022.01.04 - v1.03 - dodgy route

- Send the updated FRU to the switch heading, updated text to match reading FRU for bus number
- Send the updated FRU to the switch heading, image MIA - now re-attached

2022.01.04 - v1.02 - dodgy route

- Clarify reboots are just commands run through the console
- Crop picture in firmware upload
- Change document revision history format

2022.01.04 - v1.01 - dodgy route

- Fix initial version date
- Fix some text under download mfa_extract.py heading

2022.01.03 - v1.00 - dodgy route

- Initial version

Pre requisites and preparation

List of all required hardware and software for this learning exercise. Software is listed and configuration provided if needed.

- Windows 10 machine, also tested via Linux, but process is same as this document, using Linux applications of course
- Console cable to connect between switch and machine
- Network cable to connect the Mellanox management port to an existing switch to access Win10 machine's network
- Putty
- Tftpd64
- 7zip
- Mellanox Firmware Tools (MFT)
- Python 3
- SwitchX software packages detailed in next section

Windows 10 machine preparation

ALL of these steps re required to be completed for conversion process as well as upgrading to latest MLNX-OS afterwards

Install 7zip for Windows

Link: https://www.7-zip.org/download.html

Download and configure Uniform web server

This is used to serve the images and various files needed for the conversion

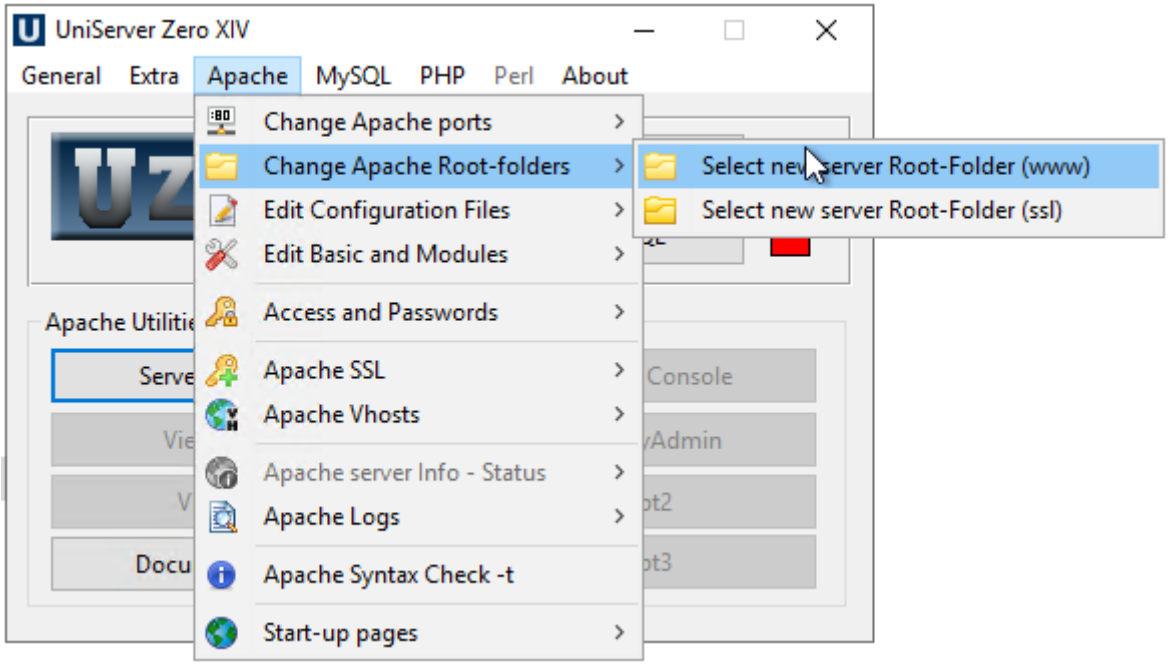
Link: https://www.uniformserver.com/

My link/version at the time of writing is:
https://sourceforge.net/projects/miniserver/files/Uniform%20Server%20ZeroXIV/14_0_3_ZeroXIV/

This is a self-extracting 7zip archive, it is not an installer. Does not touch registry/etc

On my desktop I have a Mellanox folder

- Run the exe and select the Mellanox folder, should now have a UniServerZ folder in Mellanox folder
- For the Uniform Server I have created web_share folder under Mellanox folder to serve files from



- Click Start Apache from the UniServer dialog

NOTE: As per readme, x86 version of Visual C++ Redistributable for Visual Studio 2019 is required. May already be installed

Link: https://support.microsoft.com/en-us/help/2977003/the-latest-supported-visual-c-downloads

Download and configure Tftpd64

Tftpd64 is used as a quick tftp and DHCP server. If there is DHCP in the network, no need to run the DHCP server

Link: https://pjo2.github.io/tftpd64/

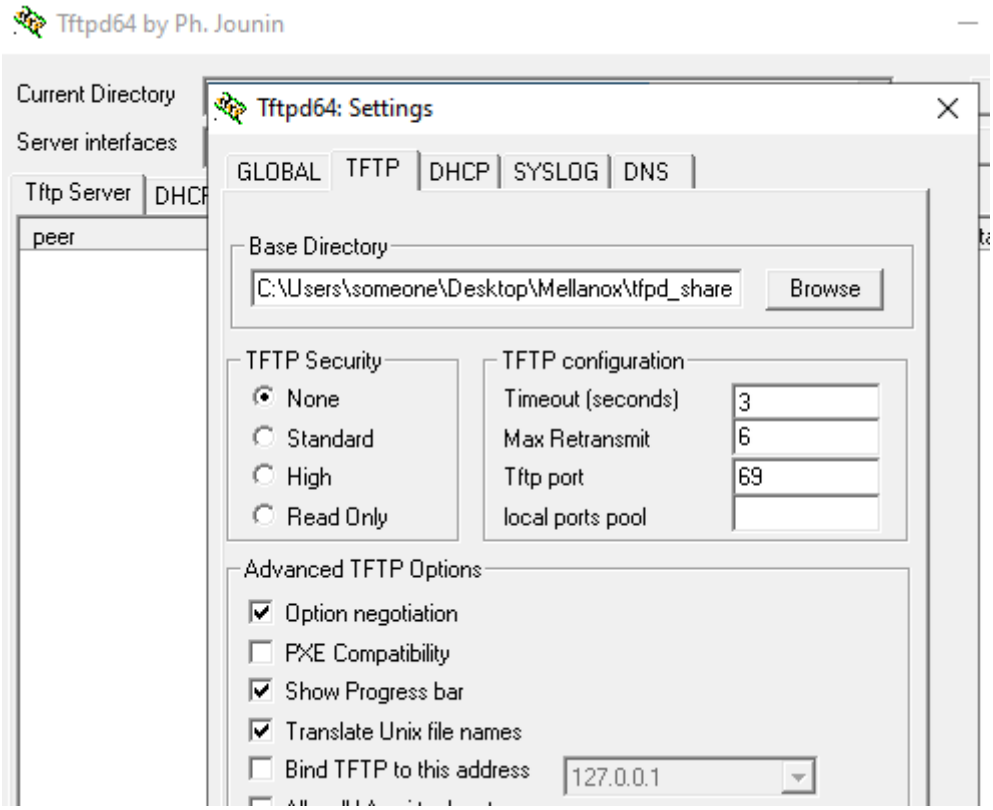
My link/version at the time of writing is: https://bitbucket.org/phjounin/tftpd64/downloads/tftpd64.464.zip

On my desktop I have a Mellanox folder

- Extract the zip to the Mellanox folder, should now have a tftpd64 folder in Mellanox folder
- For the Tftpd64 server I have created tfpd_share folder under Mellanox folder to serve files from

- In the tfpd_share folder I have a folder called “mlnx460ex” which must be lowercase as per this document

In Tftpd64 set up the base directory as tfpd_share



Set up Firewall to accept inbound connections

This is to set up Windows Defender Firewall with Advanced Security to allow connections to the Win10 machine

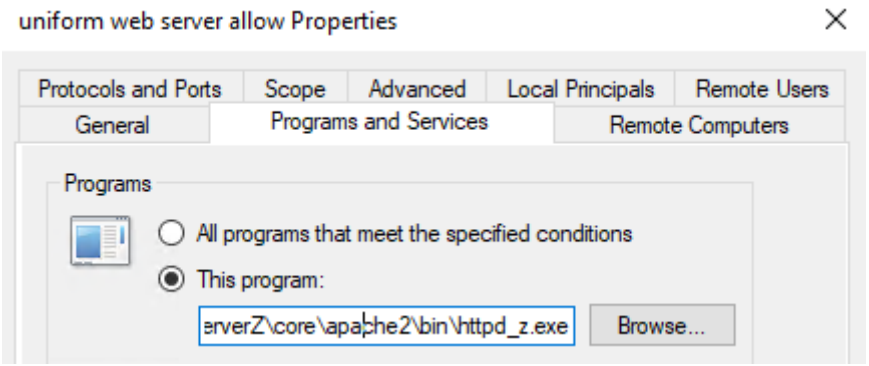
Add Uniform web server to firewall

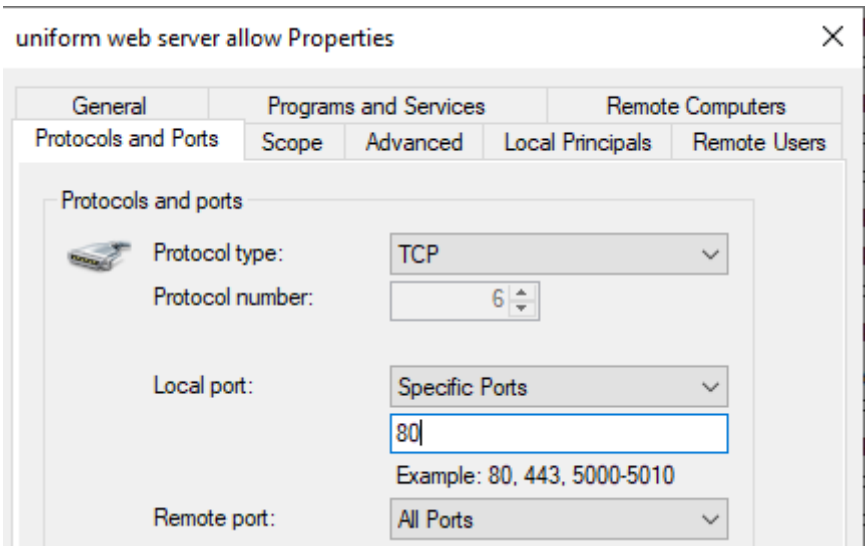
Add a new inbound rule to allow tcp port 80 to httpd_z.exe program on all profiles

NOTE: Overwrite what is in the program path. The firewall does not understand the variable name, Microsoft problems….

Original path is put in as %USERPROFILE%\Desktop\Mellanox\UniServerZ\core\apache2\bin\httpd_z.exe

Change path to C:\Users\your_user_profile_name\Desktop\Mellanox\UniServerZ\core\apache2\bin\httpd_z.exe





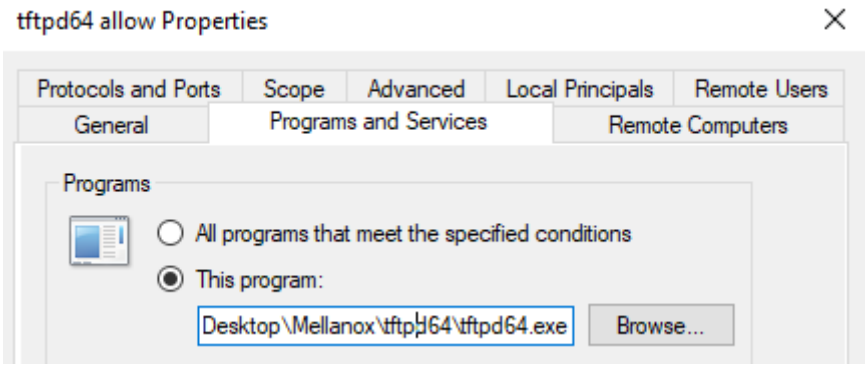
Add Tftpd64 to firewall

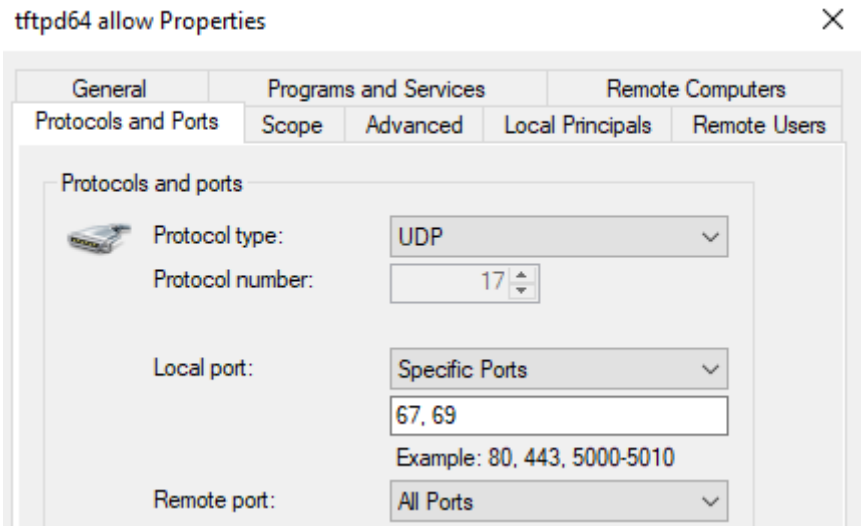
Add a new inbound rule to allow udp port 67 and 69 to tftpd64.exe program on all profiles

NOTE: Overwrite what is in the program path. The firewall does not understand the variable name, Microsoft problems….

Original path is put in as %USERPROFILE%\Desktop\Mellanox\tftpd64\tftpd64.exe

Change path to C:\Users\ your_user_profile_name\Desktop\Mellanox\tftpd64\tftpd64.exe





Download manufacturing environment

Download the 3.2.0100 version software package

Link: https://support.hpe.com/hpesc/public/swd/detail?swItemId=MTX_93f9ad4836824db99c8d873d67

From the downloaded file, extract the below files to the mlnx460ex directory and rename

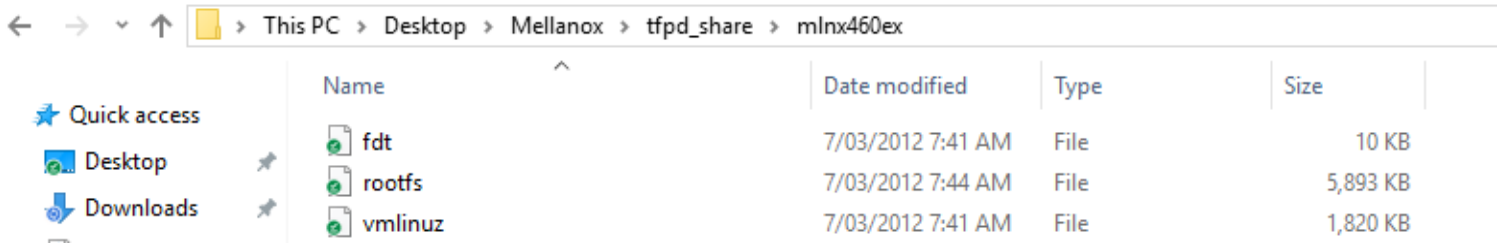
NOTE: Ensure in Windows folder options the “Hide extensions for known file types” is unchecked

fdt-PPC_M460EX-SX_3.2.0100.img **rename to** fdt

rootfs-PPC_M460EX-SX_3.2.0100.img **rename to** rootfs

vmlinuz-PPC_M460EX-SX_3.2.0100.img **rename to** vmlinuz

These files are needed for tftp to load up the manufacturing environment and my mlnx460ex folder looks like this



Prepare switch software packages

All of the below switch software packages need to be downloaded

After downloading use 7zip and extract the .img file from each package to the web_share folder, apart from 3.6.8012

3.4.0012 link: https://support.hpe.com/hpesc/public/swd/detail?swItemId=MTX_563eb88aa55a4495b6a30033a1

3.4.2008 link: https://support.hpe.com/hpesc/public/swd/detail?swItemId=MTX_6c8ee134228f4882ad517a3b4e

3.5.1006 link: https://support.hpe.com/hpesc/public/swd/detail?swItemId=MTX_1f7fc6ca8d2942b2b04bb82764

3.6.3004 link: https://support.hpe.com/hpesc/public/swd/detail?swItemId=MTX_061c24a638b44bd2826c344823

3.6.4006 link: https://support.hpe.com/hpesc/public/swd/detail?swItemId=MTX_acc6f261080046aab9935cedbc

3.6.5000 link: https://support.hpe.com/hpesc/public/swd/detail?swItemId=MTX_df68447dc2f14228a9e5839641

3.6.8010 link: https://support.hpe.com/hpesc/public/swd/detail?swItemId=MTX_f1ab10f1084d4bc7951a1ade5c

3.6.8012 link: https://www.mellanox.com/downloads/Software/image-PPC_M460EX-3.6.8012.img

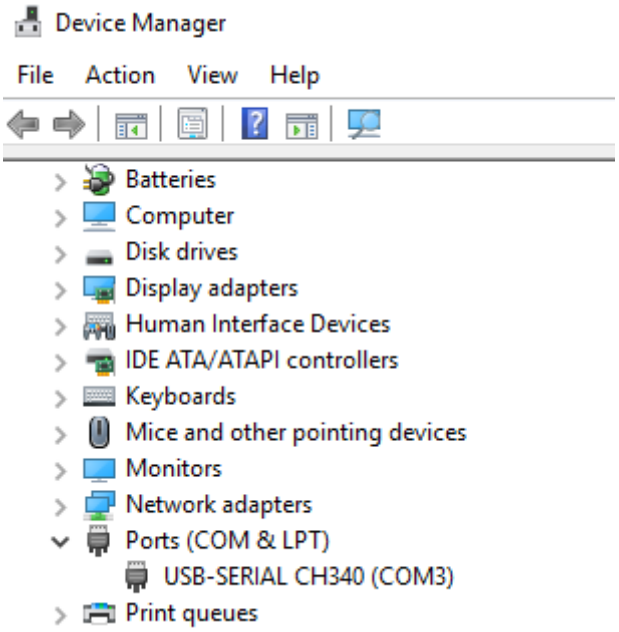
Install and configure PuTTy

This is used to access the console terminal of the switch

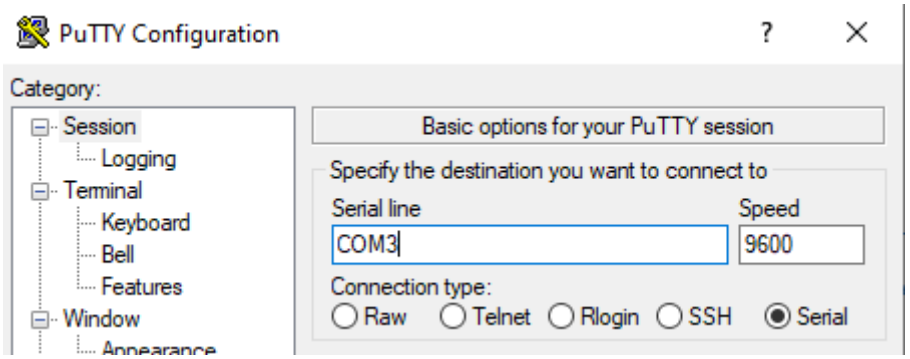
Link: https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

My link/version at the time of writing is: https://the.earth.li/~sgtatham/putty/latest/w64/putty-64bit-0.76-installer.msi

Find the COM port needed to use in PuTTy in the Windows device manager



Use this port in PuTTy, change the connection type to Serial and change COM1 to COM3 as per device manager.



Install Python3

Link: https://www.python.org/downloads/windows/

My link/version at the time of writing is https://www.python.org/ftp/python/3.10.1/python-3.10.1-amd64.exe

Install Mellanox Firmware Tools (MFT)

Link: https://www.mellanox.com/products/adapter-software/firmware-tools

My link/version at the time of writing is https://www.mellanox.com/downloads/MFT/WinMFT_x64_4_18_0_106.exe

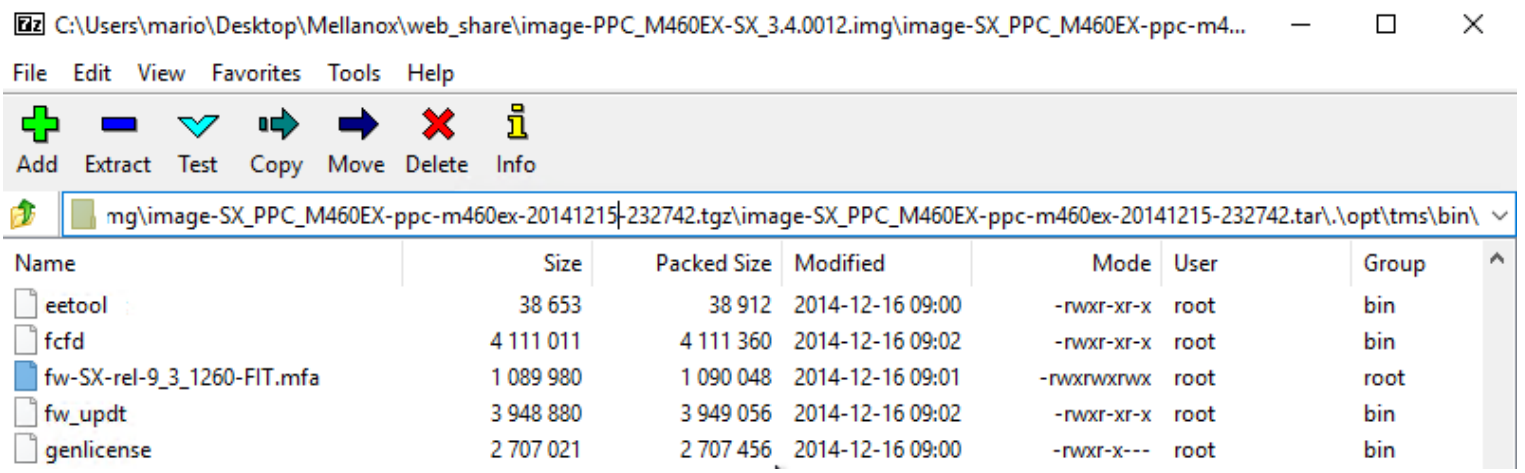
Prepare SX60xx firmware

Extract firmware update

Prepare firmware update based on image-PPC_M460EX-SX_3.4.0012.img as that is what we will manufacture with

Open image-PPC_M460EX-SX_3.4.0012.img with 7zip and get into the base folder structure

From the image extract file fw-SX-rel-9_3_1260-FIT.mfa which is in .\opt\tms\bin folder to the web_share folder



Download mfa_extract.py firmware extraction script

This is created by a BeTeP on STH

Link: https://github.com/BeTeP-STH/mft-scripts

Direct link to script: https://raw.githubusercontent.com/BeTeP-STH/mft-scripts/master/mfa_extract.py

The scripts are backed up to Appendix 1 – mfa_extract.py

I store mine in the web_share folder inside the Mellanox folder as the extracted file is generated there

Change mfa_extract.py script

Edit mfa_extract.py script, line 16, change from FLINT = “mstflint” to FLINT = “flint.bat” (quotes…)

```
1  #!/usr/bin/python3
2
3  # tested with
4  # mstflint -v
5  # mstflint, mstflint 4.6.0. Git SHA Hash: 375120d
6
7  import sys
8  import os
9  import subprocess
10 import struct
11 import zlib
12 import lzma
13 import re
14 import configparser
15
16 FLINT = "mstflint"
17
```

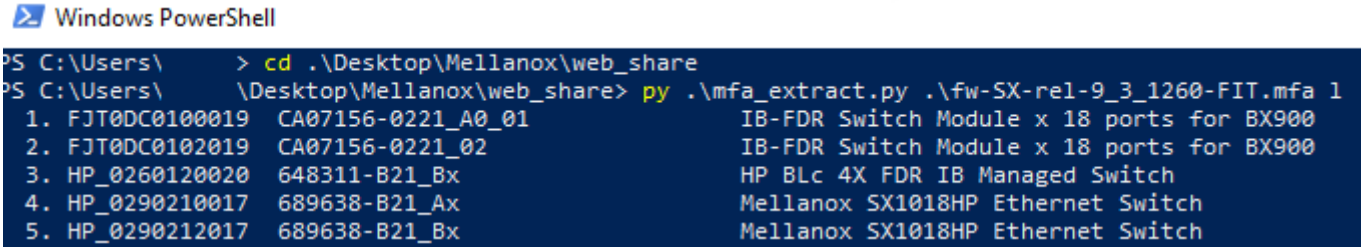
```
1  #!/usr/bin/python3
2
3  # tested with
4  # mstflint -v
5  # mstflint, mstflint 4.
6
7  import sys
8  import os
9  import subprocess
10 import struct
11 import zlib
12 import lzma
13 import re
14 import configparser
15
16 FLINT = "flint.bat"
17
```

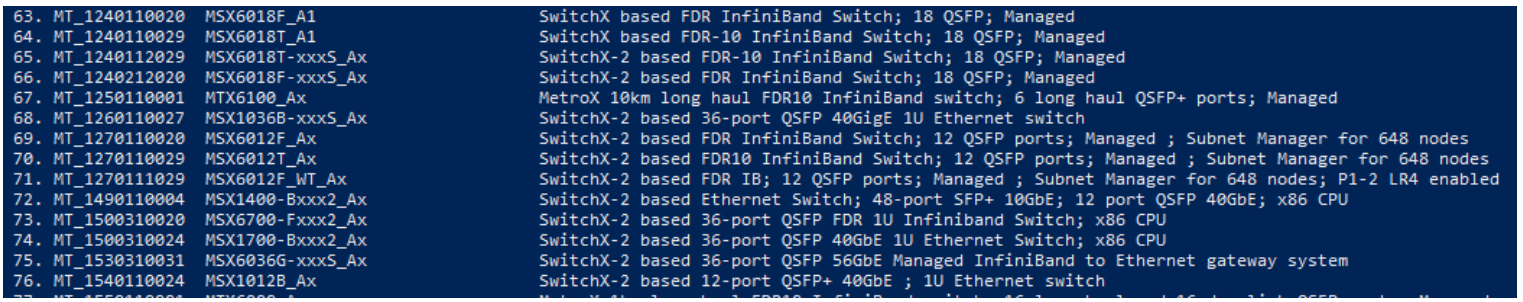
Get available firmware

In regular user privileges PowerShell, the below commands will print the list of all firmware PSID available

cd .\Desktop\Mellanox\web_share

py .\mfa_extract.py .\fw-SX-rel-9_3_1260-FIT.mfa l





We need the switch PSID, here is a list of PSID matched to switch model

SX6012: MT_1270110020 (Line 69 in screenshot)

SX6018: MT_1240212020 (Line 66 in screenshot)

SX6036: Yet to be determined?

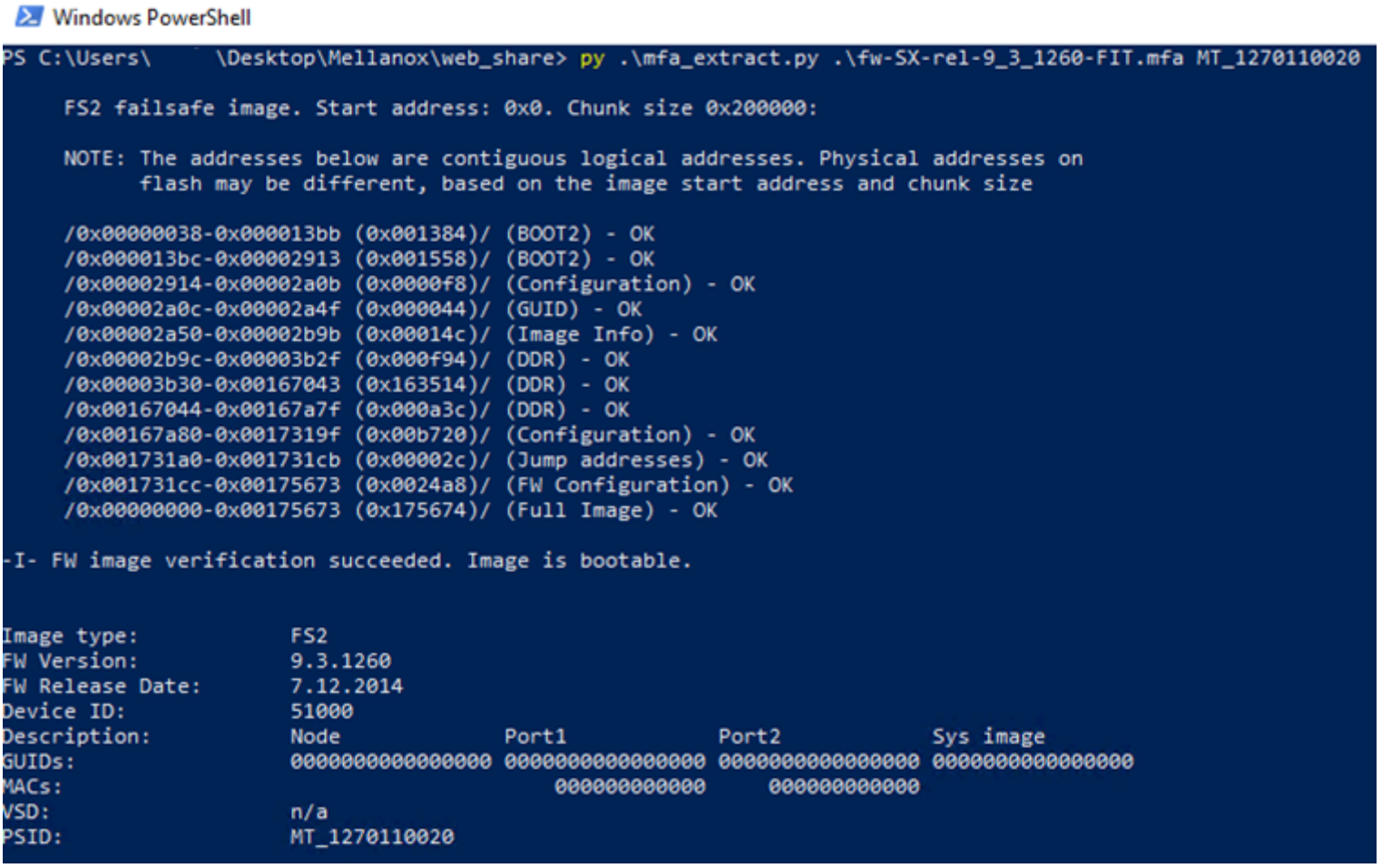
Extract firmware .bin file

After running the list command, change it as per below to extract using the needed PSID determined above for switch type

SX6012: py .\mfa_extract.py .\fw-SX-rel-9_3_1260-FIT.mfa MT_1270110020

SX6018: py .\mfa_extract.py .\fw-SX-rel-9_3_1260-FIT.mfa MT_1240212020

SX6036: Yet to be determined?



This will leave a new .bin file in the web_share folder

Rename it to MSX6012_9.3.1260.bin for later consumption, or something consistent to use in other commands in document

EMC switch preparation

Plug into switch console port and use putty or similar to connect to the terminal

Boot up switch and in U-BOOT press any key when asked to stop the booting process

At this stage take a backup of the environment variables, copy the output to a text file on the Win10 machine

printenv

Use below commands to set up U-BOOT to be able to load the manufacturing environment

setenv mfg_ramdisk_size 180224

setenv mfg_extra_args ramdisk=262144

setenv ipaddr 192.168.7.49

setenv netmask 255.255.255.0

setenv gatewayip 192.168.7.253

setenv serverip 192.168.7.77

setenv autostart no

setenv autoload no

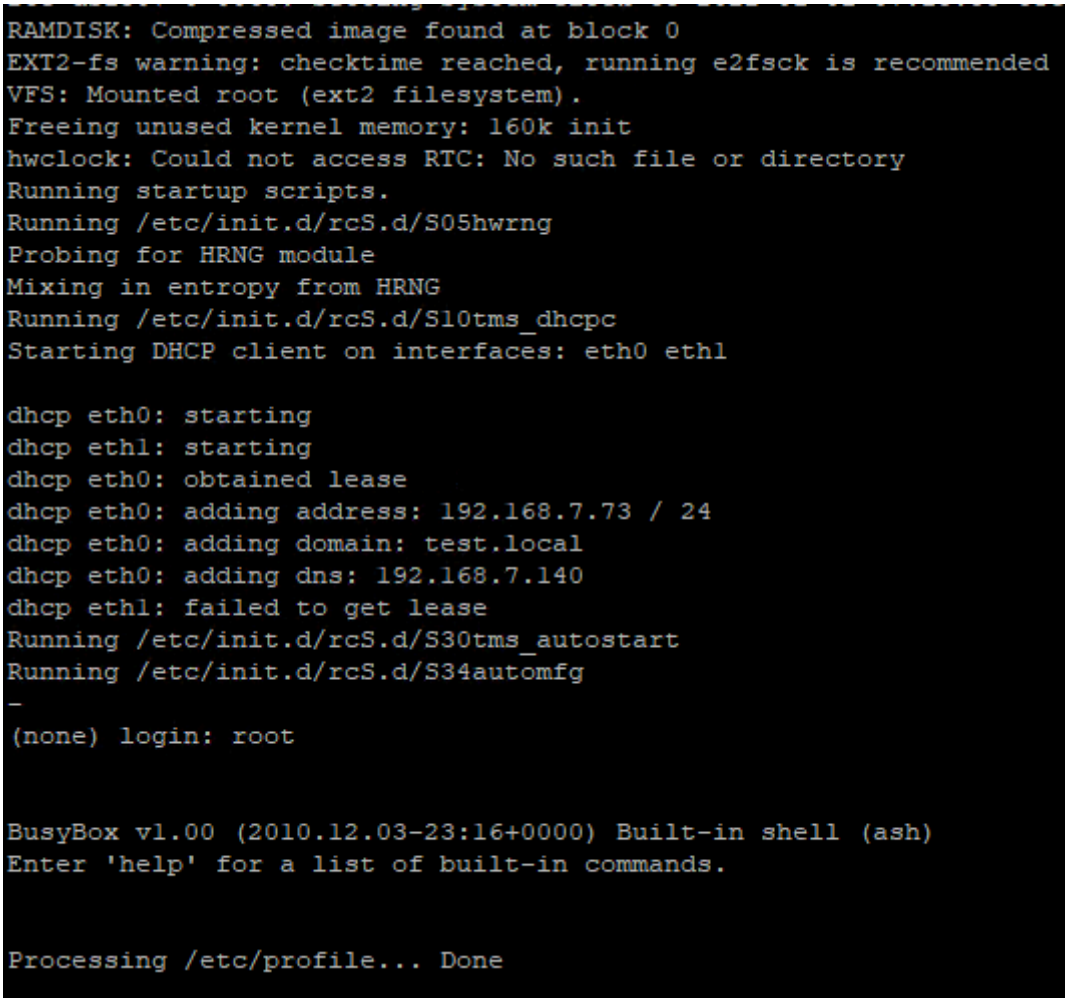
run boot_mlxlinux (this also runs a saveenv, so no need to run it after executing this)

Load manufacturing environment

Provided the previous section is completed can then load up the manufacturing environment

run mfg_nodhcp

Once loaded this will ask to log in, simply entering root in the login field will let you log in to the Linux environment



Genlicense

While in the manufacturing environment, I am going to deal with the ethernet and shell access licences

This information is gathered from information in the EMC thread on how to get the license keys using nothing but the information available through the genlicense binary alone. Thanks to crash_maxed and lambdafunction

Thanks to andvalb from STH for supplying the command on how to generate one license key with multiple licences activated

These licenses appear to be generic and not tied to anything by the look of things, so are here for my own convenience

It does look like it is possible to generate license keys for the switch specifically but I have not had time to experiment yet

However, Appendix 3 – genlicense section on how to generate the licences

Generic licence keys

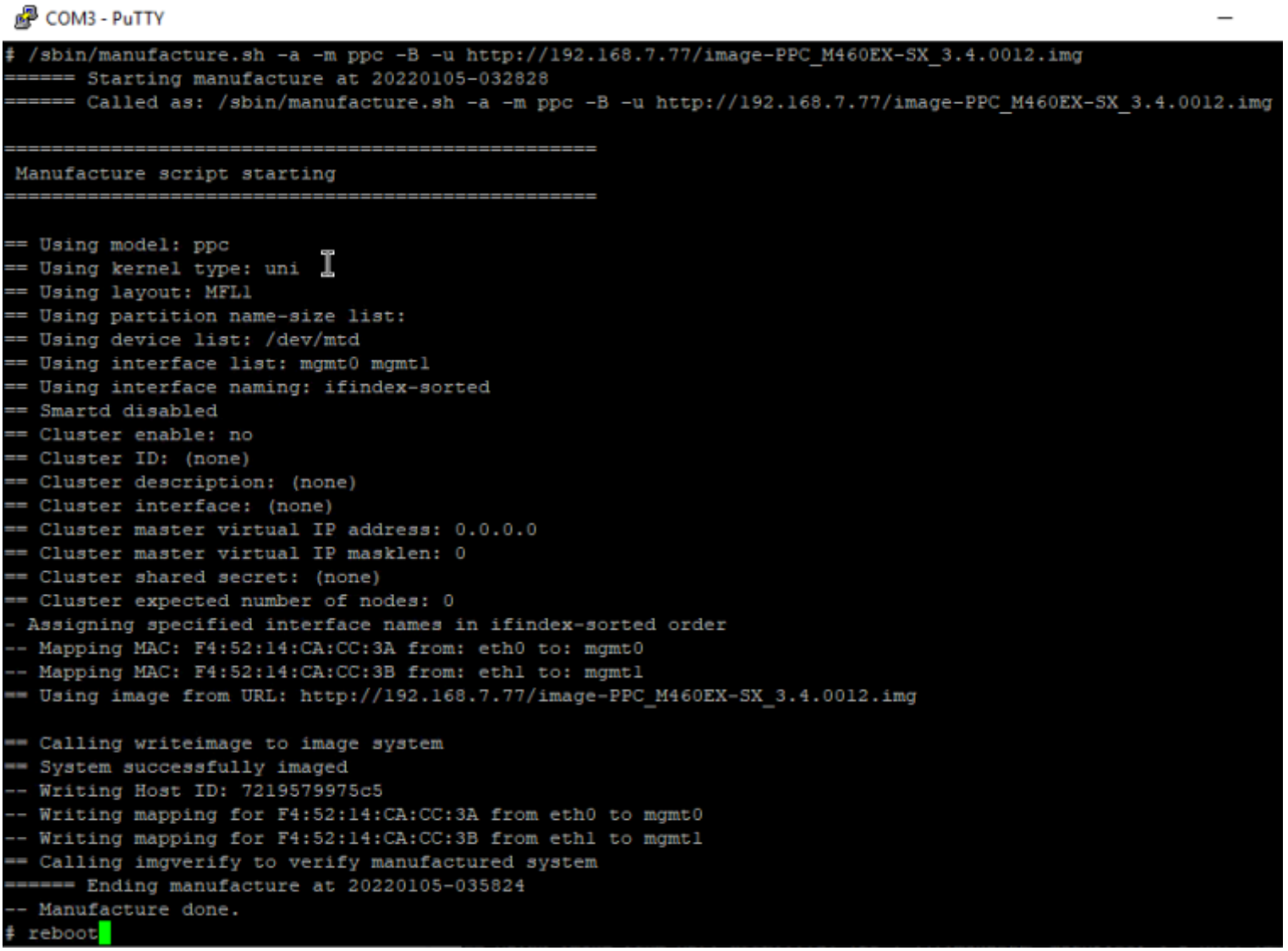
LK2-EFM_SX-5M11-5K11-5T11-88A1-BBD0-JP82-X – Enabled ethernet, L2 ethernet, L3 ethernet

LK2-RESTRICTED_CMDS_GEN2-88A1-NEWD-BPNB-1 – Enables _shell command

Remanufacture switch!

This operation will configure the bootloader/U-BOOT to work with the 2 partitions, so can run 2 images at once. It will also flash the original Mellanox U-BOOT and reload completely fresh U-BOOT variables, all of the EMC variables will be gone at this point on the next reboot and the MLNX-OS menu will be functional as it is part of the non EMC branded U-BOOT

/sbin/manufacture.sh -a -m ppc -B -u http://192.168.7.77/image-PPC_M460EX-SX_3.4.0012.img



This step will take quite a while, it will be around 30 – 60 minutes and there is no progress indicators on the really slow parts when it gets to “Calling writeimage to image system” and “Calling imgverify to verify manufactured system” parts

When the message “Manufacture done” displays, enter reboot and let it boot into MLNX-OS without any interaction

U-BOOT observation

EMC U-BOOT prior to the manufacturing operation U-BOOT on switch startup shows

- U-Boot 2009.01 SX_PPC_M460EX SX_3.2.0330-82-EMC ppc (Feb 27 2013 - 12:13:42)

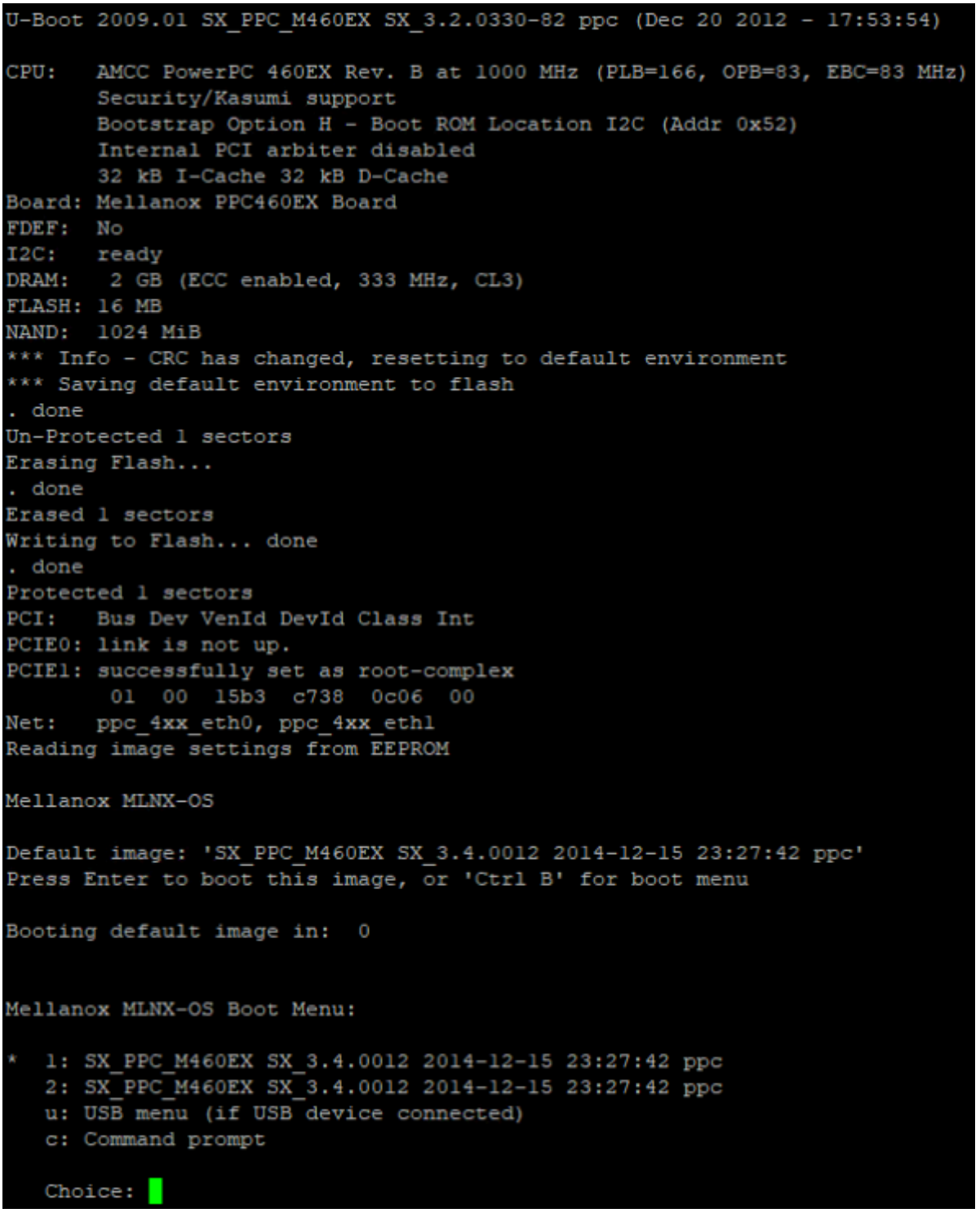
Post manufacture with Mellanox U-BOOT

- U-Boot 2009.01 SX_PPC_M460EX SX_3.2.0330-82 ppc (Dec 20 2012 - 17:53:54)

A copy of the freshly built switch’s U-BOOT variables is in Appendix 5 – Mellanox U-BOOT environment variables

Screenshot of boot post manufacture, showing new U-BOOT as well as the MLNX-OS boot menu

NOTE: This reboot only will contain messages that CRC has changed due to U-BOOT being flashed to Mellanox version



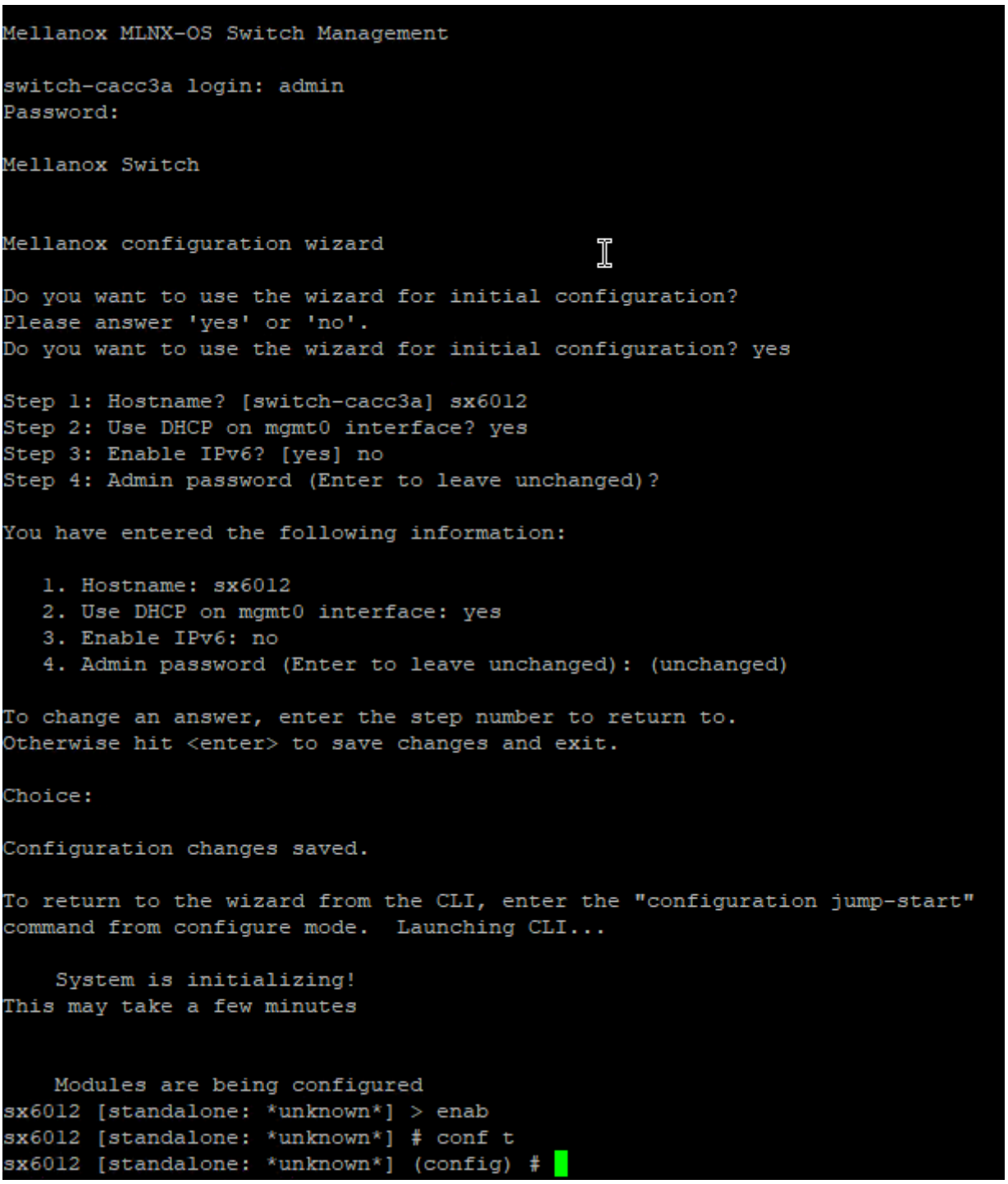
Post manufacturing reboot

After the reboot we are greeted with the MLNX-OS prompt

Login details are admin / admin

After a little while it will prompt for the wizard to do initial setup

After the wizard the configuration of the modules will take quite a while again.



Loading the licences

Enter below commands after completion

enable

configure terminal

license install LK2-EFM_SX-5M11-5K11-5T11-88A1-BBD0-JP82-X

license install LK2-RESTRICTED_CMDS_GEN2-88A1-NEWD-BPNB-1

configuration write

```
switch-cacc3a [standalone: master] > enable
switch-cacc3a [standalone: master] # conf t
switch-cacc3a [standalone: master] (config) # license install LK2-EFM_SX-5M11-5K11-5T11-88A1-BBD0-JP82-X
License was installed successfully. Please wait 1 minute before further configurations.
switch-cacc3a [standalone: master] (config) # license install LK2-RESTRICTED_CMDS_GEN2-88A1-NEWD-BPNB-1
switch-cacc3a [standalone: master] (config) # configuration write
switch-cacc3a [standalone: master] (config) # show licenses
License 1: LK2-EFM_SX-5M11-5K11-5T11-88A1-BBD0-JP82-X
   Feature:          EFM_SX
   Description:      Generic SX license
   Valid:            yes
   Active:           yes
   Eth enabled:      true
   Full Eth L2 enabled: true
   Eth L3 enabled:   true

License 2: LK2-RESTRICTED_CMDS_GEN2-88A1-NEWD-BPNB-1
   Feature:          RESTRICTED_CMDS_GEN2
   Description:      Access to restricted system functionality
   Valid:            yes
   Active:           yes
switch-cacc3a [standalone: master] (config) #
```

Change FRU from EMC to Mellanox

First we need the script by SGS from Appendix 2 – FRU conversion scripts, go to the correct section for switch type

Original link to where these originated from:
https://forums.servethehome.com/index.php?threads/beware-of-emc-switches-sold-as-mellanox-sx6xxx-on-ebay.10786/post-287882

This will convert the FRU from EMC to SX6012, SX6018 and SX6036.

Testing has been successful on SX6012 and SX6018, but I do not have a SX6036 for testing on

Create the FRU modification script

This is created by a SGS on STH for SX6012 switches and I modified them for SX6018 and pending SX6036

From terminal, remembering to change to the switch type we are dealing with, here I used emc_to_6012:

_shell

touch emc_to_6012

vi emc_to_6012

```
sx6012 [standalone: *unknown*] > enable
sx6012 [standalone: *unknown*] # conf t
sx6012 [standalone: *unknown*] (config) # _shell
[admin@sx6012 ~]# touch emc_to_6012
[admin@sx6012 ~]# vi emc_to_6012
```

push i to insert text

COM3 - PuTTY

```

~
-- INSERT --
```

Copy and paste the required switch model script from Appendix 2 – FRU conversion scripts, ensuring the quotes "" are correct

NOTE: This is critical it is copied 100% correctly, without it, and without backup, will be unable to load the switch systems

Once finished making sure all the lines and starts and ends of lines are EXACTLY as per original, press ESC to exit insert mode

COM3 - PuTTY

```
   dd if=/dev/zero bs=16 count=256 of="$2" 2> /dev/null

   dd if="$1" bs=16 count=12 of="$2" conv=notrunc 2> /dev/null
   dd if="$1" bs=16 count=5 of="$2" skip=12 seek=14 conv=notrunc 2> /dev/null

   printf "\x20" | dd of="$2" bs=1 seek=1 count=1 conv=notrunc 2> /dev/null
   printf "\x00" | dd of="$2" bs=1 seek=5 count=1 conv=notrunc 2> /dev/null

   printf "\x05\x0E\x02\x14\x06\x16\x07" | dd of="$2" bs=1 seek=15 count=7 conv=
notrunc 2> /dev/null

   printf "\x00\x1A\x00\x03\x05\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00" | dd of="$2" bs=1 s
eek=192 count=32 conv=notrunc 2> /dev/null
   printf "\x00\x12\x00\x01\x06\x00\x00\x00\x00\x01\x00\x00\x02\x88\x04\x04\x02\
x02\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x0A\x00\x01\x07\
x00\x00\x00\x00\x02\x10\x00\x00\x00\x00\x00" | dd of="$2" bs=1 seek=320 count=48
 conv=notrunc 2> /dev/null

   printf "\x4D\x53\x58\x36\x30\x31\x32\x46\x2D\x32\x42\x46\x53\x00" | dd of="$2
" bs=1 seek=64 count=14 conv=notrunc 2> /dev/null
~
~
```

Press SHIFT + : and type wq to save changes and quit vi

```
"emc_to_6012" 14L, 1055C written
[admin@sx6012 ~]#
```

To recheck the saved file, run the vi command again and this time SHIFT + : and type q! if no changes to exit without saving

Get current FRU

If converting from an earlier converted switch, the bus number may be different for the device bus number, going from 1 to 8, so we you may need to change the 1 to 8 if necessary

/opt/tms/bin/mellaggra _read_fru 1 0x51 1000 fru_backplate.bin

```
[admin@sx6012 ~]# /opt/tms/bin/mellaggra _read_fru 1 0x51 1000 fru_backplate.bin
Read fru 1 - 0x51, sz - 4096 to file fru_backplate.binx51 1000 fru_backplate.bin
[admin@sx6012 ~]#
```

Convert exported FRU to SX60xx

Run command to convert the FRU from EMC to SX60xx

sh emc_to_6012 "fru_backplate.bin" "fru_patched.bin"

```
[admin@sx6012 ~]# sh emc_to_6012 "fru_backplate.bin" "fru_patched.bin"
[admin@sx6012 ~]# ls -la
total 48
drwxr-xr-x 2 admin root     0 Jan  1 11:25 .
drwxr-xr-x 6 admin root     0 Jan  1 09:00 ..
-rw------- 1 admin root   162 Jan  1 10:07 .bash_history
-rw-r--r-- 1 admin root    18 Dec 15  2014 .bash_logout
-rw-r--r-- 1 admin root   176 Dec 15  2014 .bash_profile
-rw-r--r-- 1 admin root   176 Dec 15  2014 .bashrc
-rw------- 1 admin root   498 Jan  1 11:21 .cli_history
-rw------- 1 admin root     0 Jan  1 09:01 .cli_history.lock
-rw-r--r-- 1 admin root   210 Dec 15  2014 .cshrc
-rw-r--r-- 1 admin root 35201 Dec 15  2014 .gdbinit
-rw-r--r-- 1 admin root   [illegible]
-rw-r--r-- 1 admin root  1055 Jan  1 10:41 emc_to_6012
-rw-r--r-- 1 admin root  4096 Jan  1 11:24 fru_backplate.bin
-rw-r--r-- 1 admin root  4096 Jan  1 11:25 fru_patched.bin
[admin@sx6012 ~]#
```

Backup the original and new FRU to TFTP server

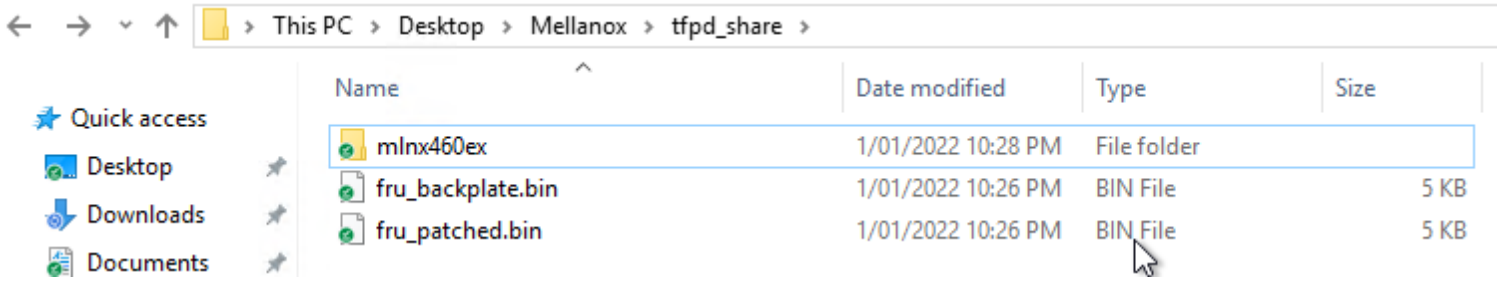
Backup the fru files to server to have, just in case as well as for interest's sake. This will store them in the Win10 machine

tftp 192.168.7.77 -c put fru_backplate.bin

tftp 192.168.7.77 -c put fru_patched.bin

```
[admin@sx6012 ~]# tftp 192.168.7.77 -c put fru_backplate.bin
[admin@sx6012 ~]# tftp 192.168.7.77 -c put fru_patched.bin
[admin@sx6012 ~]#
```

This copies the files to the root of the base Tftpd64 folder



Send the updated FRU to the switch

Run command to send the converted SX60xx FRU to the switch, changing the device bus number 1 to 8 only if needed.

You should know which bus was used when exploring it from get current FRU step

/opt/tms/bin/mellaggra _write_fru 1 0x51 1000 fru_patched.bin

```
ed.bin@switch-cacc3a ~]# /opt/tms/bin/mellaggra _write_fru 1 0x51 1000 fru_patch
[admin@switch-cacc3a ~]#
```

Firmware update

This section details the change of the switch firmware from EMC to Mellanox

Get the device path

First we need to get the device

mst status

This tells us /dev/mst/mt51000_pciconf0 is the device we will be dealing with

```
[admin@sx6012 ~]# mst status
MST modules:
------------
    MST PCI module loaded
    MST PCI configuration module loaded

MST devices:
------------
/dev/mst/dev-i2c-0               - Embedded I2C master
/dev/mst/dev-i2c-1               - Embedded I2C master
/dev/mst/mt51000_pciconf0        - PCI configuration cycles access.
                                   domain:bus:dev.fn=0001:81:00.0 addr.reg=88 data.reg=92
                                   Chip revision is: 02
[admin@sx6012 ~]#
```

It can also be confirmed with lspci

```
[admin@sx6012 ~]# lspci
0000:40:00.0 PCI bridge: Unknown device aaa0:bed0 (rev 01)
0001:80:00.0 PCI bridge: Unknown device aaa1:bed1 (rev 01)
0001:81:00.0 InfiniBand: Mellanox Technologies Unknown device c738 (rev 02)
```

Get current firmware information

Lets get the current firmware information

flint -override_cache_replacement -d /dev/mst/mt51000_pciconf0 q

Ignore formatting that gets like that when pasted into console

COM3 - PuTTY

```
f0 qin@switch01 ~]# flint -override_cache_replacement -d /dev/mst/mt51000_pcicon

-W- Firmware flash cache access is enabled. Running in this mode may cause the firmware to hang.

-W- Running quick query - Skipping full image integrity checks.

Image type:      FS2
FW Version:      9.9.1260
Device ID:       51000
Description:     Node             Sys image
GUIDs:           f452140300217560 f452140300217560
Description:     Base             Switch
MACs:                f45214217560     f452142175c0
VSD:             n/a
PSID:            EMC1270110020
[admin@switch01 ~]#
```

Backup EMC firmware

flint -override_cache_replacement -d /dev/mst/mt51000_pciconf0 ri EMC-old.bin

```
 ri EMC-old.bin]# flint -override_cache_replacement -d /dev/mst/mt51000_pciconf0

-W- Firmware flash cache access is enabled. Running in this mode may cause the firmware to hang.
[admin@sx6012 ~]#
```

Verify old EMC firmware file

flint -i EMC-old.bin q

COM3 - PuTTY

```
[admin@switch01 ~]# flint -i EMC-old.bin q

-W- Running quick query - Skipping full image integrity checks.

Image type:       FS2
FW Version:       9.9.1260
Device ID:        51000
Description:      Node             Sys image
GUIDs:            f452140300217560 f452140300217560
Description:      Base             Switch
MACs:                 f45214217560     f452142175c0
VSD:              n/a
PSID:             EMC1270110020
[admin@switch01 ~]#
```

Upload old EMC firmware to TFTP server

tftp 192.168.7.77 -m binary -c put EMC-old.bin

```
[admin@sx6012 ~]# tftp 192.168.7.77 -m binary -c put EMC-old.bin
[admin@sx6012 ~]#
```

Backup firmware ini file

This wont actually be used in the conversion process, but just nice to have as a backup

flint -override_cache_replacement -d /dev/mst/mt51000_pciconf0 dc > EMC1270110020.ini

Ignore formatting that gets like that when pasted into console

```
 dc > EMC1270110020.ini -override_cache_replacement -d /dev/mst/mt51000_pciconf0
```

Upload firmware ini to TFTP server

tftp 192.168.7.77 -c put EMC1270110020.ini

```
[admin@switch01 ~]# tftp 192.168.7.77 -c put EMC1270110020.ini
[admin@switch01 ~]#
```

Download SX60xx firmware to switch

In the preparation steps I extracted the firmware image to the web_share folder, it is now time to download it

curl -O http://192.168.7.77/MSX6012_9.3.1260.bin

```
[admin@switch01 ~]# curl -O http://192.168.7.77/MSX6012_9.3.1260.bin
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 1493k  100 1493k    0     0  1897k      0 --:--:-- --:--:-- --:--:-- 1907k
[admin@switch01 ~]#
```

Verify new firmware bin file

flint -i MSX6012_9.3.1260.bin q

```
[admin@sx6012 ~]# flint -i MSX6012_9.3.1260.bin q

-W- Running quick query - Skipping full image integrity checks.

Image type:       FS2
FW Version:       9.3.1260
Device ID:        51000
Description:      Node             Sys image
GUIDs:            0000000000000000 0000000000000000
Description:      Base             Switch
MACs:                 000000000000     000000000000
VSD:              n/a
PSID:             MT_1270110020
```

Flash new firmware to switch

flint --override_cache_replacement --allow_psid_change -d /dev/mst/mt51000_pciconf0 -i ./MSX6012_9.3.1260.bin b

Ignore formatting that gets like that when pasted into console

```
ev/mst/mt51000_pciconf0 -i ./MSX6012_9.3.1260.bin bent --allow_psid_change -d /d

-W- Firmware flash cache access is enabled. Running in this mode may cause the firmware to hang.

    Current FW version on flash:  9.9.1260
    New FW version:               9.3.1260

    Note: The new FW version is not newer than the current FW version on flash.

 Do you want to continue ? (y/n) [n] : y


    You are about to replace current PSID on flash - "EMC1270110020" with a different PSID - "MT_1270110020".
    Note: It is highly recommended not to change the PSID.

 Do you want to continue ? (y/n) [n] : y
Burning FS2 FW image without signatures - %010
```

Get new/current firmware information

Lets get the newly uploaded firmware information

flint -override_cache_replacement -d /dev/mst/mt51000_pciconf0 q

Ignore formatting that gets like that when pasted into console

```
f0 qin@switch01 ~]# flint -override_cache_replacement -d /dev/mst/mt51000_picon

-W- Firmware flash cache access is enabled. Running in this mode may cause the firmware to hang.

-W- Running quick query - Skipping full image integrity checks.

Image type:       FS2
FW Version:       9.3.1260
Device ID:        51000
Description:      Node             Sys image
GUIDs:            f452140300217560 f452140300217560
Description:      Base             Switch
MACs:                 f45214217560     f452142175c0
VSD:              n/a
PSID:             MT_1270110020
[admin@switch01 ~]#
```

Remove the bootloader password so U-BOOT is accessible

Thanks to necr from STH where I saw this

Process for image versions under 3.6.5000

NOTE: This operation will have to be performed on every new image update **if you need** to get into the bootloader/U-BOOT, can skip the backup process. I did mine to test on 3.4.0012 and then when I installed 3.6.5000 which allowed to disable it.

Log into _shell first using

enable

conf t

_shell

Backup the existing password, or view it using

/opt/tms/bin/mddbreq /config/db/initial query get - /system/bootmgr/password

Ignore formatting that gets like that when pasted into console

```
stem/bootmgr/password~]# /opt/tms/bin/mddbreq /config/db/initial query get - /sy
Name: /system/bootmgr/password
        Attrib: value
        Type: string
        Value: $1$yCoib8pn$vSaWSssw2kl7iOJRIdmcw/
```

The text is: 1yCoib8pn$vSaWSssw2k17iOJRIdmcw/

Remove the bootloader password

/opt/tms/bin/mddbreq /config/db/initial set modify - /system/bootmgr/password string ''

eetool -a bf -s UBPASSWD=""

```
ystem/bootmgr/password string ''s/bin/mddbreq /config/db/initial set modify - /s
[admin@switch-cacc3a ~]# eetool -a bf -s UBPASSWD=""
[admin@switch-cacc3a ~]#
```

Verify the bootloader password is removed

Using mlxi2c, from _shell

mlxi2c show fru /CPU

```
[admin@switch-cacc3a ~]# mlxi2c show fru /CPU
Jan 04 08:12:51 INFO    LOG: Initializing SX log with STDOUT as output file.

------ Block 0 ------
-I- SN          : "MT1349X00764"
-I- PN          : "SA002203"
-I- REV         : "A2"
-I- BID         : 0
-I- CPUT        : "PPC 460EX"
-I- OUI         : 2c9
-I- MAC0        : f45214cacc3a
-I- MAC1        : f45214cacc3b
-I- HWNAME      : "M460EX"

------ Block 1 ------
-I- MFG         : 0
-I- DHCP        : 0
-I- S1STATE     : 1
-I- S2STATE     : 2
-I- S1OSNAME    : "SX_PPC_M460EX SX_3.4.0012 2014-12-15 23:27:42 ppc"
-I- S2OSNAME    : "SX_PPC_M460EX SX_3.4.0012 2014-12-15 23:27:42 ppc"
-I- S1OSARGS    : "  img_id=1 quiet loglevel=4 panic=1"
-I- S2OSARGS    : "  img_id=2 quiet loglevel=4 panic=1"
-I- UBPASSWD    : ""
```

Process for image versions 3.6.5000 and above

NOTE: This operation can be performed once and it should keep, I did mine when I updated to 3.6.5000 which disables it

enable

configure terminal

boot bootmgr password 7 ""

write memory

show bootvar

```
switch-cacc42 [standalone: master] (config) # boot bootmgr password 7 ""
switch-cacc42 [standalone: master] (config) # write memory
switch-cacc42 [standalone: master] (config) # show bootvar
Installed images:

  Partition 1:
  PPC_M460EX 3.6.4006 2017-07-03 16:17:35 ppc

  Partition 2:
  PPC_M460EX 3.6.5000 2017-11-10 18:14:29 ppc

Last boot partition: 2
Next boot partition: 2

Serve image files via HTTP/HTTPS: no

No boot manager password is set.

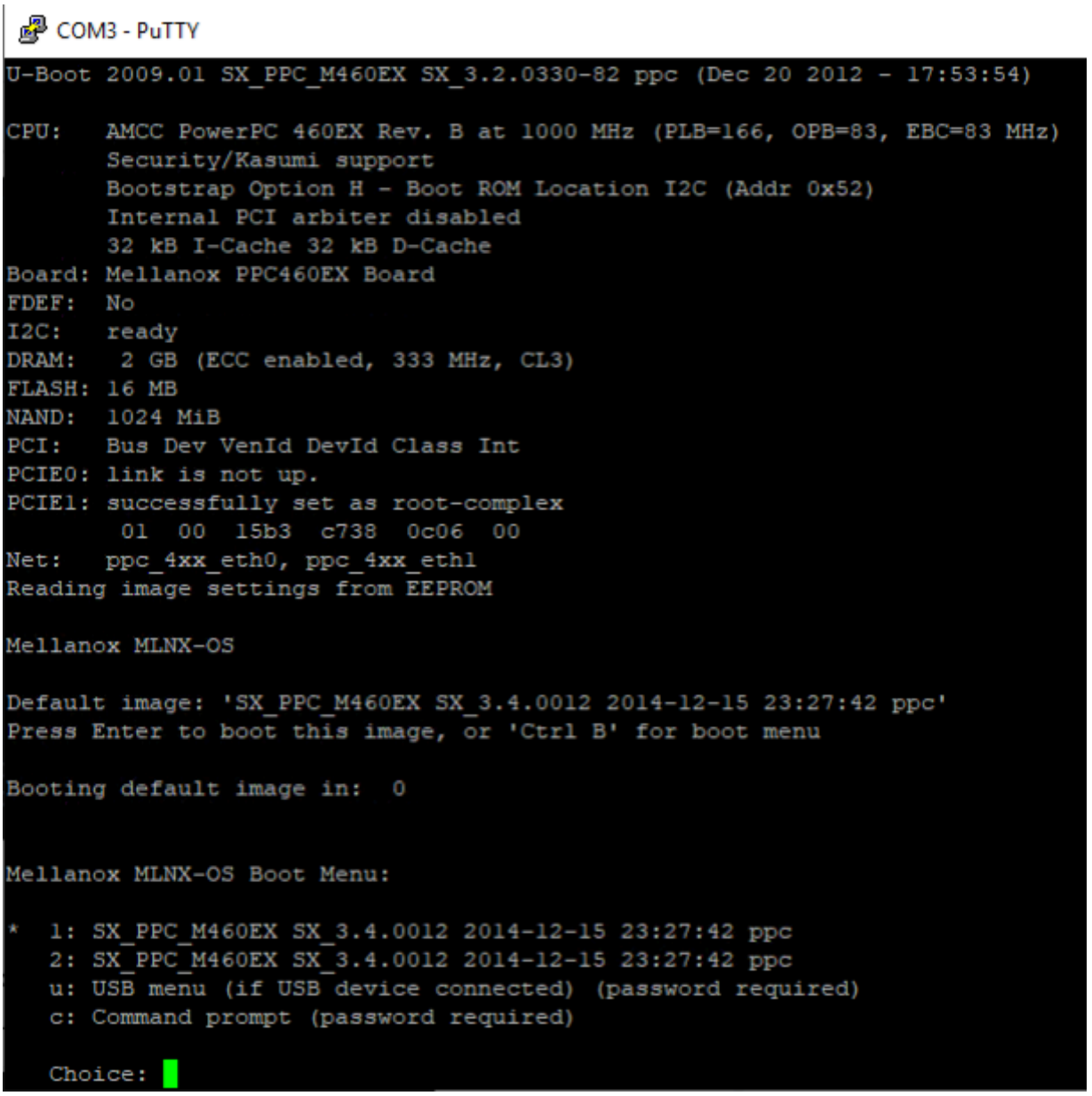
Image signing: trusted signature always required
Admin require signed images: yes

Settings for next boot only:
   Fallback reboot on configuration failure: yes (default)
switch-cacc42 [standalone: master] (config) # show version concise
PPC_M460EX 3.6.5000 2017-11-10 18:14:29 ppc
switch-cacc42 [standalone: master] (config) #
```

Reboot

This is it, enter reboot to…. reboot, and after that the switch conversion process is completed.

The fans will ramp down quickly in two stages once the switch finishes loading up after the reboot, first to 60% and then 40%.



Upgrade image versions to 3.6.8012

Run the update process through the switch software packages detailed in Prepare switch software packages heading

Rinse and repeat this, this is pulled from the recommended upgrade paths and based on available packages

show images

```
switch-cacc42 [standalone: master] (config) # show images
Installed images:

  Partition 1:
  SX_PPC_M460EX SX_3.4.0012 2014-12-15 23:27:42 ppc

  Partition 2:
  SX_PPC_M460EX SX_3.4.0012 2014-12-15 23:27:42 ppc

Last boot partition: 1
Next boot partition: 1

No image files are available to be installed.

Serve image files via HTTP/HTTPS: no

No image install currently in progress.

Boot manager password is set.

Image signing: trusted signature always required
Admin require signed images: yes

Settings for next boot only:
   Fallback reboot on configuration failure: yes (default)
```

image delete <tab to autocomplete>

NOTE: This is only done once both partitions are updated, don't do this between partition updates as will need to re-download

```
switch-cacc3a [standalone: master] (config) # image delete image-PPC_M460EX-3.4.2008.img
switch-cacc3a [standalone: master] (config) #
```

image fetch http://192.168.7.77/image-PPC_M460EX-3.xxx.img

Note: This is only done once per image version and will allow you to perform update on both partitions

```
switch-cacc42 [standalone: master] (config) # image fetch http://192.168.7.77/image-PPC_M460EX-3.4.2008.img
 100.0%  [##################################################################################################]
```

show images

```
switch-cacc42 [standalone: master] (config) # show images
Installed images:

  Partition 1:
  SX_PPC_M460EX SX_3.4.0012 2014-12-15 23:27:42 ppc

  Partition 2:
  SX_PPC_M460EX SX_3.4.0012 2014-12-15 23:27:42 ppc

Last boot partition: 1
Next boot partition: 1

Images available to be installed:

  image-PPC_M460EX-3.4.2008.img
  PPC_M460EX 3.4.2008 2015-06-12 11:48:41 ppc

Serve image files via HTTP/HTTPS: no

No image install currently in progress.

Boot manager password is set.

Image signing: trusted signature always required
Admin require signed images: yes

Settings for next boot only:
   Fallback reboot on configuration failure: yes (default)
```

image install <tab to autocomplete>

```
switch-cacc42 [standalone: master] (config) # image install image-PPC_M460EX-3.4.2008.img
Step 1 of 4: Verify Image
 100.0%  [##########################################################################################################]
Step 2 of 4: Uncompress Image
 100.0%  [##########################################################################################################]
Step 3 of 4: Create Filesystems
 100.0%  [##########################################################################################################]
Step 4 of 4: Extract Image
 100.0%  [##########################################################################################################]
switch-cacc42 [standalone: master] (config) #
```

image boot next

```
switch-cacc42 [standalone: master] (config) # image boot next
switch-cacc42 [standalone: master] (config) #
```

show images

NOTE: the password still says it is set but it is nulled now, should be removable completely in a later 3.6.xxxx image

```
switch-cacc42 [standalone: master] (config) # show images
Installed images:

  Partition 1:
  SX_PPC_M460EX SX_3.4.0012 2014-12-15 23:27:42 ppc

  Partition 2:
  PPC_M460EX 3.4.2008 2015-06-12 11:48:41 ppc

Last boot partition: 1
Next boot partition: 2

Images available to be installed:

  image-PPC_M460EX-3.4.2008.img
  PPC_M460EX 3.4.2008 2015-06-12 11:48:41 ppc

Serve image files via HTTP/HTTPS: no

No image install currently in progress.

Boot manager password is set.

Image signing: trusted signature always required
Admin require signed images: yes

Settings for next boot only:
   Fallback reboot on configuration failure: yes (default)
switch-cacc42 [standalone: master] (config) #
```

configuration write

```
switch-cacc42 [standalone: master] (config) # configuration write
switch-cacc42 [standalone: master] (config) #
```
3

reload

```
switch-cacc3a [standalone: master] (config) # reload


Rebooting...


System shutdown initiated -- logging off.
```

On the reboot it will boot to partition 2 and you will be able to repeat the above steps again, to update partition 1

After both partitions are updated, you can then move onto the next firmware version

```
U-Boot 2009.01 SX_PPC_M460EX SX_3.2.0330-82 ppc (Dec 20 2012 - 17:53:54)

CPU:   AMCC PowerPC 460EX Rev. B at 1000 MHz (PLB=166, OPB=83, EBC=83 MHz)
       Security/Kasumi support
       Bootstrap Option H - Boot ROM Location I2C (Addr 0x52)
       Internal PCI arbiter disabled
       32 kB I-Cache 32 kB D-Cache
Board: Mellanox PPC460EX Board
FDEF:  No
I2C:   ready
DRAM:   2 GB (ECC enabled, 333 MHz, CL3)
FLASH: 16 MB
NAND:  1024 MiB
PCI:   Bus Dev VenId DevId Class Int
PCIE0: link is not up.
PCIE1: successfully set as root-complex
        01  00  15b3  c738  0c06  00
Net:   ppc_4xx_eth0, ppc_4xx_eth1
Reading image settings from EEPROM

Mellanox MLNX-OS

Default image: 'PPC_M460EX 3.4.2008 2015-06-12 11:48:41 ppc'
Press Enter to boot this image, or 'Ctrl B' for boot menu

Booting default image in:  2
```

show images (from updated partition 2)

```
switch-cacc3a [standalone: master] (config) # show images
Installed images:

  Partition 1:
  SX_PPC_M460EX SX_3.4.0012 2014-12-15 23:27:42 ppc

  Partition 2:
  PPC_M460EX 3.4.2008 2015-06-12 11:48:41 ppc

Last boot partition: 2
Next boot partition: 2

Images available to be installed:

  image-PPC_M460EX-3.4.2008.img
  PPC_M460EX 3.4.2008 2015-06-12 11:48:41 ppc

Serve image files via HTTP/HTTPS: no

No image install currently in progress.

Boot manager password is set.

Image signing: trusted signature always required
Admin require signed images: yes

Settings for next boot only:
   Fallback reboot on configuration failure: yes (default)
```

show asic-version (from updated partition 2)

This shows that the firmware was automatically updated to what is in MLNX-OS software image 3.4.2008, which is 9.3.3180

```
switch-cacc3a [standalone: master] (config) # show asic-version
================================================
Module              Device              Version
================================================
MGMT                SX                  9.3.3180
```

Appendix

Appendix 1 – mfa_extract.py

```
#!/usr/bin/python3

# tested with
# mstflint -v
# mstflint, mstflint 4.6.0. Git SHA Hash: 375120d

import sys
import os
import subprocess
import struct
import zlib
import lzma
import re
import configparser

FLINT = "flint.bat"

def read_buff(fn, n=-1):
    with open(fn, 'rb') as f:
        return f.read(n)

def lzma_decompress(buf):
    decomp = lzma.LZMADecompressor(memlimit=0x10000000)
    try:
        return decomp.decompress(buf)
    except lzma.LZMAError:
        pass
    return b''

def save_bin(fn, buff, bin_off, bin_len):
    decomp = lzma.LZMADecompressor(memlimit=0x10000000)
    with open(fn, 'wb') as f:
        try:
            f.write(decomp.decompress(buff)[bin_off:bin_off + bin_len])
        except lzma.LZMAError:
            pass
    return decomp.eof

def parse_mtoc(buff, compressed, offset, size):
    mtoc = {}
    if compressed:
        off = 0
        while off < len(buff):
            a, b, c = struct.unpack_from('>32sB1xH', buff, off)
            psid = a.decode('ascii').strip('\0')
            _, pn, _, desc = buff[off+40:off+180].decode('latin1').strip('\0').split('\0')[0:4]
            mtoc[psid] = { 'pn': pn, 'desc': desc, 'off': [struct.unpack_from('>IHH', buff,
36+off+c+40*i) for i in range(b)]}
            off += 36 + 40 * b + c
    return mtoc

def mfa_extract(mfaname, psid):
    SECTIONS = {}
    BUFFER = bytearray(read_buff(mfaname))
    if b'MFAR' != BUFFER[0:4]:
```

```
        return 1
    off = 16
    for i in range(3):
        a,b,c,d = struct.unpack_from('>B2xBI4s', BUFFER, off)
        off += 8
        SECTIONS[a] = {'offset': off, 'size': c, 'compressed': b and (d == b'\xFD7zX'), 'buff':
memoryview(BUFFER[off:off+c]) }
        if SECTIONS[a]['compressed'] and i < 2:
            SECTIONS[a]['buff'] = lzma_decompress(SECTIONS[a]['buff'])
        off += c

    MTOC = parse_mtoc(**SECTIONS[1])
    if MTOC.get(psid):
        fn = "{}.bin".format(psid)
        for moff in MTOC[psid]['off']:
            off, size = struct.unpack_from('>ii', SECTIONS[2]['buff'], moff[0])
            if size > 0:
                break
        if save_bin(fn, SECTIONS[3]['buff'], off, size):
            print(subprocess.check_output([FLINT, '-i', fn, 'v']).decode('ascii'))
            print(subprocess.check_output([FLINT, '-i', fn, 'q']).decode('ascii'))
            return 0

    else:
        for i, psid in enumerate(sorted(MTOC.keys()), 1):
            print('{i:>3}. {psid:15s}{pn:33s}{desc}'.format(i=i, psid=psid, **MTOC[psid]))
        return 0
    return 1

if __name__ == "__main__":
    if len(sys.argv) != 3:
        print("Usage:\n\t{0} firmware.mfa <PSID>\t - to extract\n\t{0} firmware.mfa l|list\t -
to list".format(*sys.argv))
        sys.exit(2)
    sys.exit(mfa_extract(*sys.argv[1:]))
```

Appendix 2 – FRU conversion scripts

Check carefully for switch model, ie SX6012, SX6018 or SX6036 (upcoming)

SX6012

EMC SX6012 to MSX6012F-2BFS

```
dd if=/dev/zero bs=16 count=256 of="$2" 2> /dev/null

dd if="$1" bs=16 count=12 of="$2" conv=notrunc 2> /dev/null
dd if="$1" bs=16 count=5 of="$2" skip=12 seek=14 conv=notrunc 2> /dev/null

printf "\x20" | dd of="$2" bs=1 seek=1 count=1 conv=notrunc 2> /dev/null
printf "\x00" | dd of="$2" bs=1 seek=5 count=1 conv=notrunc 2> /dev/null

printf "\x05\x0E\x02\x14\x06\x16\x07" | dd of="$2" bs=1 seek=15 count=7 conv=notrunc 2>
/dev/null

printf
"\x00\x1A\x00\x03\x05\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x0
0\x00\x00\x00\x00\x00\x00\x00" | dd of="$2" bs=1 seek=192 count=32 conv=notrunc 2> /dev/null
printf
"\x00\x12\x00\x01\x06\x00\x00\x00\x00\x01\x00\x00\x02\x88\x04\x04\x02\x02\x00\x00\x00\x00\x00\x0
0\x00\x00\x00\x00\x00\x00\x00\x00\x00\x0A\x00\x01\x07\x00\x00\x00\x00\x02\x10\x00\x00\x00\x00\x0
0" | dd of="$2" bs=1 seek=320 count=48 conv=notrunc 2> /dev/null

printf "\x4D\x53\x58\x36\x30\x31\x32\x46\x2D\x32\x42\x46\x53\x00" | dd of="$2" bs=1 seek=64
count=14 conv=notrunc 2> /dev/null
```

MSX6012F-2BFS to MSX1012F-2BFS

```
if [ "$1" != "$2" ]; then
   dd if="$1" bs=16 count=256 of="$2" 2> /dev/null
fi
printf "\x4D\x53\x58\x31\x30\x31\x32\x42\x2D\x32\x42\x46\x53\x00" | dd of="$2" bs=1 seek=64
count=14 conv=notrunc 2> /dev/null
printf "\x03\xF4" | dd of="$2" bs=1 seek=162 count=2 conv=notrunc 2> /dev/null
printf "\x31" | dd of="$2" bs=1 seek=166 count=1 conv=notrunc 2> /dev/null
printf "\x06" | dd of="$2" bs=1 seek=329 count=1 conv=notrunc 2> /dev/null
```

MSX1012F-2BFS to MSX6012F-2BFS

```
if [ "$1" != "$2" ]; then
   dd if="$1" bs=16 count=256 of="$2" 2> /dev/null
fi
printf "\x4D\x53\x58\x36\x30\x31\x32\x46\x2D\x32\x42\x46\x53\x00" | dd of="$2" bs=1 seek=64
count=14 conv=notrunc 2> /dev/null
printf "\x17\x7C" | dd of="$2" bs=1 seek=162 count=2 conv=notrunc 2> /dev/null
printf "\x36" | dd of="$2" bs=1 seek=166 count=1 conv=notrunc 2> /dev/null
printf "\x01" | dd of="$2" bs=1 seek=329 count=1 conv=notrunc 2> /dev/null
```

SX6018

EMC SX6018 to MSX6018F-2SRS

```
dd if=/dev/zero bs=16 count=256 of="$2" 2> /dev/null

dd if="$1" bs=16 count=12 of="$2" conv=notrunc 2> /dev/null
dd if="$1" bs=16 count=5 of="$2" skip=12 seek=14 conv=notrunc 2> /dev/null

printf "\x20" | dd of="$2" bs=1 seek=1 count=1 conv=notrunc 2> /dev/null
printf "\x00" | dd of="$2" bs=1 seek=5 count=1 conv=notrunc 2> /dev/null

printf "\x05\x0E\x02\x14\x06\x16\x07" | dd of="$2" bs=1 seek=15 count=7 conv=notrunc 2>
/dev/null

printf
"\x00\x1A\x00\x03\x05\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x0
0\x00\x00\x00\x00\x00\x00\x00" | dd of="$2" bs=1 seek=192 count=32 conv=notrunc 2> /dev/null
printf
"\x00\x12\x00\x01\x06\x00\x00\x00\x00\x01\x00\x00\x02\x88\x04\x04\x02\x02\x00\x00\x00\x00\x00\x0
0\x00\x00\x00\x00\x00\x00\x00\x00\x00\x0A\x00\x01\x07\x00\x00\x00\x00\x02\x10\x00\x00\x00\x00\x0
0" | dd of="$2" bs=1 seek=320 count=48 conv=notrunc 2> /dev/null

printf "\x4D\x53\x58\x36\x30\x31\x38\x46\x2D\x32\x53\x52\x53\x00" | dd of="$2" bs=1 seek=64
count=14 conv=notrunc 2> /dev/null
```

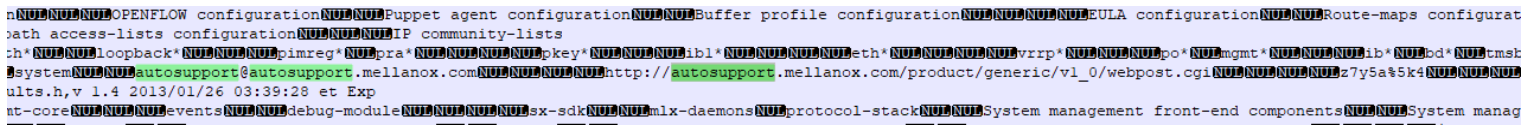
Appendix 3 – genlicense

Read genlicense encoded secret

Thanks to lambdafunction

From the switch image image-PPC_M460EX-SX_3.4.0012.img open the base folder and extract \opt\tms\bin\genlicense

Using Notepad++ or similar to explore the contents, search for autosupport where some interesting characters show up



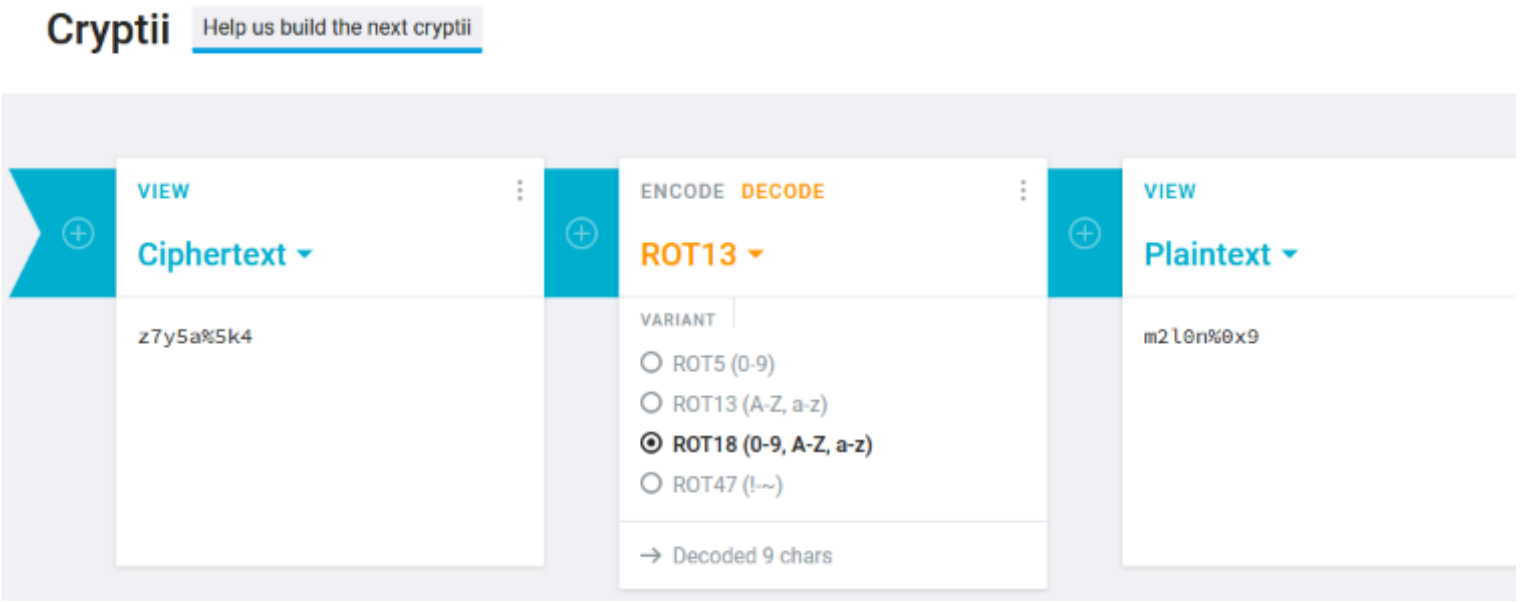
Interesting characters: z7y5a%5k4

Decode interesting characters

The characters are encoded in ROT18 which is a combination of ROT13 for letters and ROT5 for numbers

Armed with this information and the interesting characters for the genlicense secret open https://cryptii.com/pipes/rot13-decoder and pick ROT18

The secret turns out to be m2l0n%0x9 after decoding the ROT18



Generate licences

From the manufacturing environment, genlicense commands can be used to generate ethernet and _shell licences, additional license keys can be generated as shown.

Thanks to andvalb from STH for supplying the command on how to generate one license key with multiple licences activated

```
Informational option IDs:
   48 (efm_sx_max_num_hca_ports): Maximum number of HCA ports supported by this EFM SX license
   50 (efm_sx_active_ports): Active ports number supported by this EFM SX license
   51 (efm_sx_full_l2_enabled): Full Eth L2 enabled by this EFM SX license
   52 (efm_sx_ib_enabled): IB enabled by this EFM SX license
   53 (efm_sx_eth_enabled): Eth enabled by this EFM SX license
   54 (efm_sx_gw_ports): GW ports number supported by this EFM SX license
   55 (efm_sx_max_ufm_ports): Maximum number of UFM ports supported by this EFM SX license
   56 (efm_sx_ib_speed_sw_limit): IB port SW speed limit enabled by this EFM SX license
   57 (efm_sx_eth_speed_sw_limit): Eth port SW speed limit enabled by this EFM SX license
   58 (efm_sx_l3_enabled): Eth L3 enabled by this EFM SX license
```

Here is the commands I used to generate the licences for my experiment, note the secret that's gained from genlicense binary

/opt/tms/bin/genlicense 2 EFM_SX m2l0n%0x9 -o 53 true -o 51 true -o 58 true

/opt/tms/bin/genlicense 2 RESTRICTED_CMDS_GEN2 m2l0n%0x9

```
o 51 true -o 58 true ~]# /opt/tms/bin/genlicense 2 EFM_SX m2l0n%0x9 -o 53 true -
LK2-EFM_SX-5M11-5K11-5T11-88A1-BBD0-JP82-X
9admin@switch-cacc3a ~]# /opt/tms/bin/genlicense 2 RESTRICTED_CMDS_GEN2 m2l0n%0x
LK2-RESTRICTED_CMDS_GEN2-88A1-NEWD-BPNB-1
```

Appendix 4 – mlxi2c interesting dumps

Some interesting information obtainable from mlxi2c command

Get the device bus number

This is needed for backing up and updating the FRU

mlxi2c show devs -v

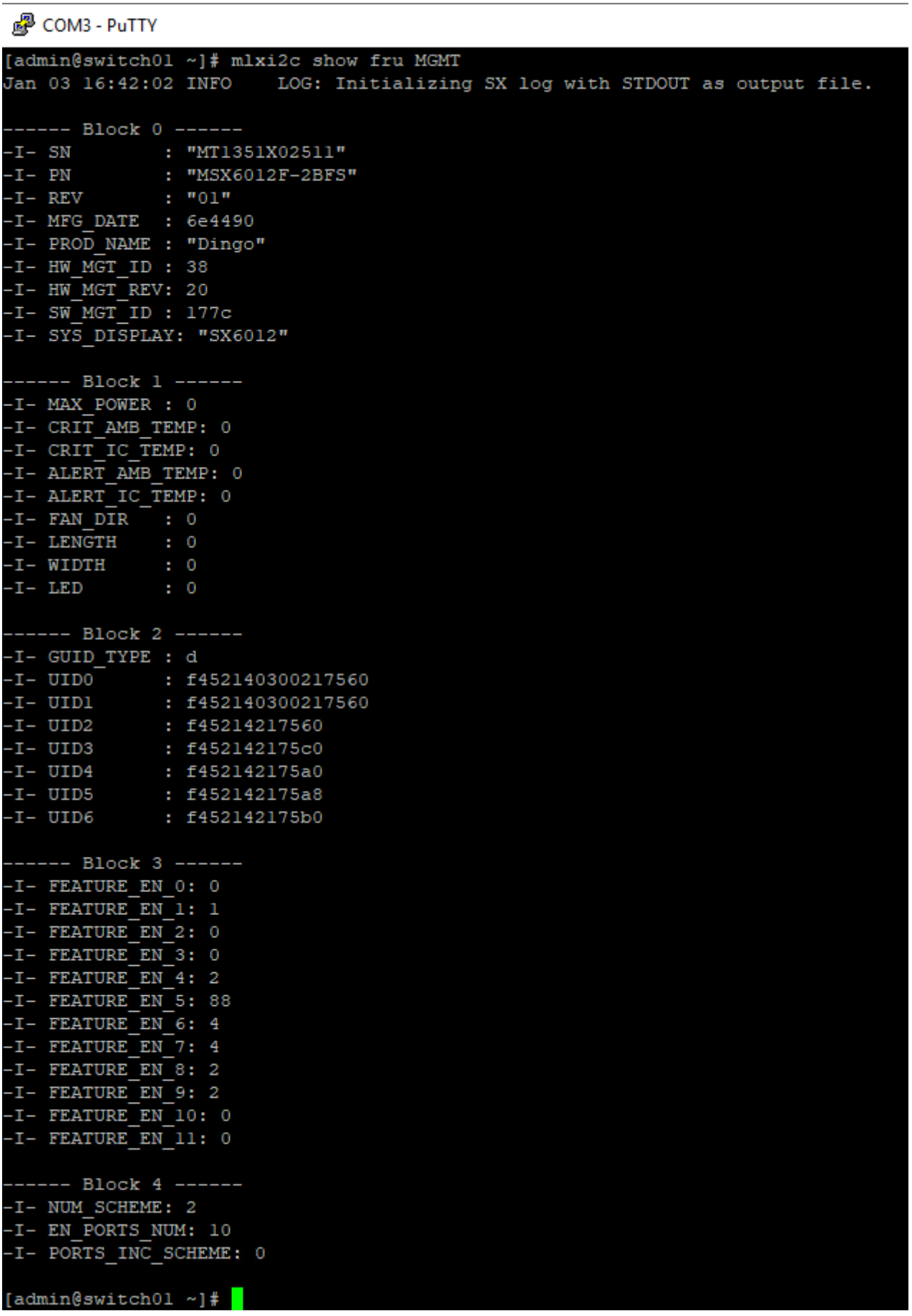
On line FRU_EEPROM, Slv Addr 0x51, note the Bus Num, 8 in this case, this is where the emc_to_6012 script works

It appears the change in firmware moves it from Bus Num 1 on EMC firmware to Bus Num 8 when converted?

```
[admin@switch-cacc42 ~]# mlxi2c show devs -v
Jan 02 13:07:09 INFO    LOG: Initializing SX log with STDOUT as output file.
 Name                Bus Num   Handle Slv Addr Addr Width  PRES  Type         Desc
 /MAIN_SW                  1   0xff00   0x70         0     Yes   PCA9548      "I2C Switch 8 legs"
 /SX                       2   0xff01   0x48         4     Yes   SX           "SX VPI Switch"
 /CPLD_TOR                 5   0xff02   0x60         1     Yes   CPLD_TOR     "Lattice CPLD TOR, I2C access, addr_width 1"
 /QSFP_TEMP1               7   0xff03   0x4a         1     Yes   LM75         "Thermal Monitor"
 /QSFP_TEMP2               7   0xff04   0x49         1     Yes   LM75         "Thermal Monitor"
 /QSFP_TEMP3               7   0xff05   0x4c         1     Yes   LM75         "Thermal Monitor"
 /BOARD_MONITOR            7   0xff06   0x2e         1     Yes   ADM1024      "Thermal Monitor"
 /CURR_MONITOR             7   0xff07   0x3f         1     Yes   ADM1191      "Digital Power Monitor"
 /FRU_EEPROM               8   0xff08   0x51         2     Yes   24LC32       "Serial EEPROM 4k bytes"
 /CPU_BOARD_MONITOR        0   0xff09   0x2e         1     Yes   ADM1024      "Thermal Monitor"
 /MGMT/FAN1                0   0xff0a   0x00         1     Yes   SX_FAN_2DRWR_FIX  "SX FAN 2x2 Fixed"
 /MGMT/FAN2                0   0xff0b   0x00         1     Yes   SX_FAN_2DRWR_FIX  "SX FAN 2x2 Fixed"
 /MGMT/FAN3                0   0xff0c   0x00         1     Yes   SX_FAN_2DRWR_FIX  "SX FAN 2x2 Fixed"
 /MGMT/FAN4                0   0xff0d   0x00         1     Yes   SX_FAN_2DRWR_FIX  "SX FAN 2x2 Fixed"
 /MGMT/PS1                 0   0xff0e   0x00         1     Yes   PS_FIX       "Power Supply Fixed"
 /MGMT/PS2                 0   0xff0f   0x00         1     Yes   PS_FIX       "Power Supply Fixed"
 /CPU/FRU_EEPROM           0   0x0000   0x50         2     Yes   24LC32       "Serial EEPROM 4k bytes"
```

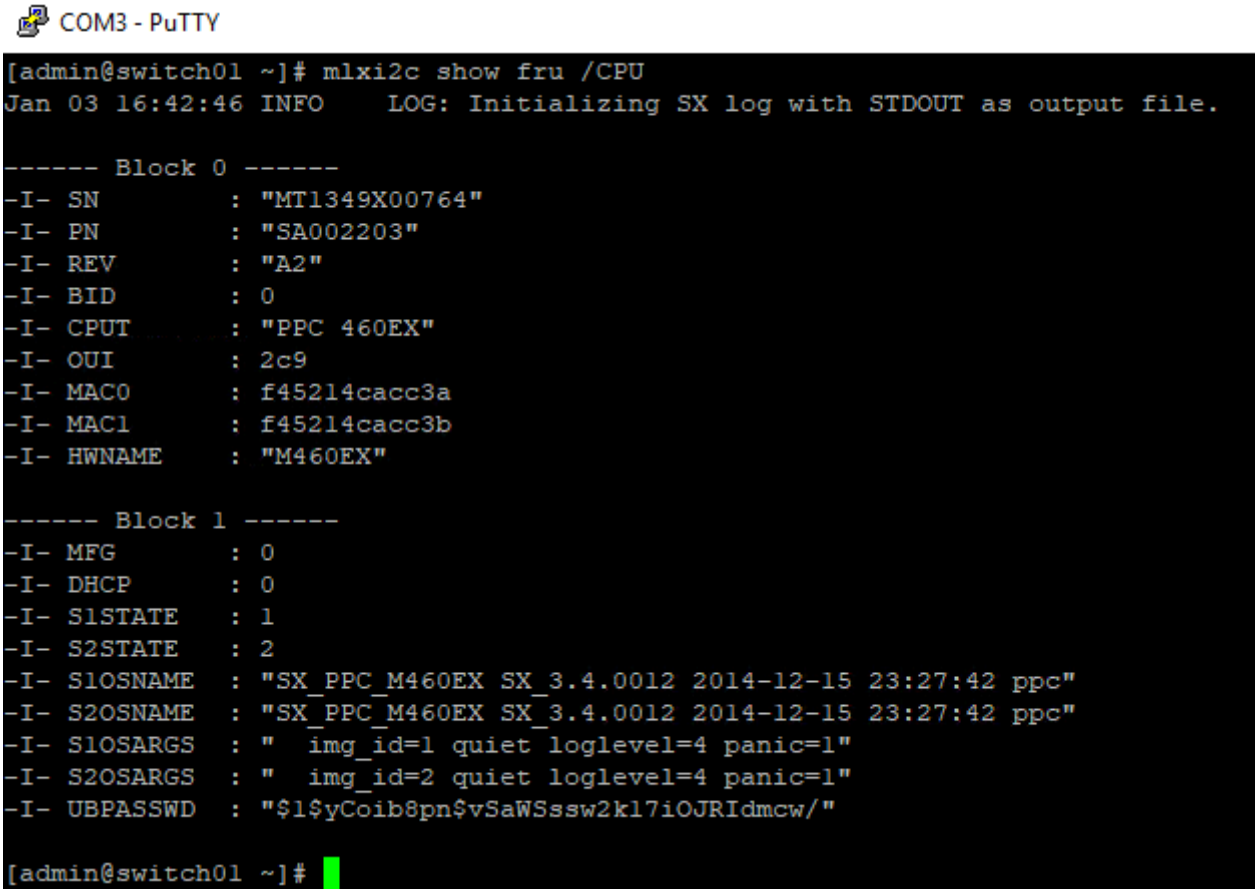
Show MGMT FRU details

mlxi2c show FRU MGMT



Show CPU FRU details

mlxi2c show fru /CPU



Appendix 5 – Mellanox U-BOOT environment variables

This is from a freshly manufactured switch

=> printenv

bootdelay=5

baudrate=9600

loads_echo=

autoload=n

hostname=mlnx460ex

netdev=eth0

nfsargs=setenv bootargs root=/dev/nfs rw nfsroot=${serverip}:${rootpath}

ramargs=setenv bootargs root=/dev/ram rw

addip=setenv bootargs ${bootargs} ip=${ipaddr}:${serverip}:${gatewayip}:${netmask}:${hostname}:${netdev}:off panic=1

addtty=setenv bootargs ${bootargs} console=ttyS0,${baudrate}

addmisc=setenv bootargs ${bootargs}

initrd_high=30000000

kernel_addr_r=400000

fdt_addr_r=800000

ramdisk_addr_r=C00000

hostname=mlnx460ex

bootfile=mlnx460ex/uImage

ramdisk_file=mlnx460ex/uRamdisk

rootpath=/opt/eldk/ppc_4xxFP

flash_self=run ramargs addip addtty addmisc;bootm ${kernel_addr} ${ramdisk_addr} ${fdt_addr}

flash_nfs=run nfsargs addip addtty addmisc;bootm ${kernel_addr} - ${fdt_addr}

net_nfs=tftp ${kernel_addr_r} ${bootfile}; tftp ${fdt_addr_r} ${fdt_file}; run nfsargs addip addtty addmisc;bootm ${kernel_addr_r} - ${fdt_addr_r}

net_self_load=tftp ${kernel_addr_r} ${bootfile};tftp ${fdt_addr_r} ${fdt_file};tftp ${ramdisk_addr_r} ${ramdisk_file};

net_self=run net_self_load;run ramargs addip addtty addmisc;bootm ${kernel_addr_r} ${ramdisk_addr_r} ${fdt_addr_r}

fdt_file=mlnx460ex/mlnx460ex.dtb

flash_self_old=run ramargs addip addtty addmisc;bootm ${kernel_addr} ${ramdisk_addr}

flash_nfs_old=run nfsargs addip addtty addmisc;bootm ${kernel_addr}

net_nfs_old=tftp ${kernel_addr_r} ${bootfile};run nfsargs addip addtty addmisc;bootm ${kernel_addr_r}

load=tftp 200000 mlnx460ex/u-boot.bin

update=protect off 0xFFFA0000 FFFFFFFF;era 0xFFFA0000 FFFFFFFF;cp.b ${fileaddr} 0xFFFA0000 ${filesize};setenv filesize;saveenv

upd=run load update

dhcp_vendor-class-identifier=bootmfg:hwname:mlnx460ex:

clear_filesize=setenv filesize

mfg_dir=mlnx460ex

mfg_args=setenv bootargs root=/dev/ram rw ramdisk_size=${mfg_ramdisk_size} ${mfg_extra_args}

mfg_common_args=run addtty addmisc

mfg_load=tftp ${kernel_addr_r} ${mfg_root}${mfg_dir}/${mfg_kernel_file};tftp ${fdt_addr_r} ${mfg_root}${mfg_dir}/${mfg_fdt_file};tftp ${ramdisk_addr_r} ${mfg_root}${mfg_dir}/${mfg_ramdisk_file}

mfg_nodhcp=echo "Manufacture will TFTP from directory ${mfg_root}${mfg_dir}, and boot";echo; run clear_filesize ; run mfg_load;if test 0${filesize} -gt 0; then echo Booting mfg ; run mfg_args mfg_common_args;bootm ${kernel_addr_r} ${ramdisk_addr_r} ${fdt_addr_r} ; else ; echo Failed mfg load ; fi

mfg=echo "Manufacture will DHCP, TFTP from directory ${mfg_root}${mfg_dir}, and boot";echo; dhcp; run clear_filesize ; run mfg_load;if test 0${filesize} -gt 0; then echo Booting mfg ; run mfg_args mfg_common_args;bootm ${kernel_addr_r} ${ramdisk_addr_r} ${fdt_addr_r} ; else ; echo Failed mfg load ; fi

menu_file=menu.img

menu_load=tftp ${menu_addr_r} ${mfg_root}${mfg_dir}/${menu_file}; if test $? -ne 0; then run clear_filesize ; echo Download failed ;fi

menu_usb_load_ext2=usb start; ext2load usb ${mfg_usb_dev}:${mfg_usb_part} ${menu_addr_r} ${mfg_usb_root}${mfg_usb_dir}/${menu_file};

menu_usb_load_fat=usb start; fatload usb ${mfg_usb_dev}:${mfg_usb_part} ${menu_addr_r} ${mfg_usb_root}${mfg_usb_dir}/${menu_file};

menu_usb_load=if test "x${mfg_usb_fstype}" = "xext2"; then run menu_usb_load_ext2 ; else ; run menu_usb_load_fat ; fi

menu_usb=run clear_filesize ; run menu_usb_load ; if test 0${filesize} -gt 0; then autoscr ${menu_addr_r}; else ; echo Failed menu load ; fi

menu_nodhcp=run clear_filesize ; run menu_load ; if test 0${filesize} -gt 0; then autoscr ${menu_addr_r}; else ; echo Failed menu load ; fi

menu=dhcp ; run clear_filesize ; run menu_load ; if test 0${filesize} -gt 0; then autoscr ${menu_addr_r}; else ; echo Failed menu load ; fi

fw_file=u-boot.bin

fw_load=tftp ${fw_addr_r} ${mfg_root}${mfg_dir}/${fw_file}; if test $? -ne 0; then run clear_filesize ; echo Download failed ;fi

fw_usb_load_ext2=usb start; ext2load usb ${mfg_usb_dev}:${mfg_usb_part} ${fw_addr_r} ${mfg_usb_root}${mfg_usb_dir}/${fw_file};

fw_usb_load_fat=usb start; fatload usb ${mfg_usb_dev}:${mfg_usb_part} ${fw_addr_r} ${mfg_usb_root}${mfg_usb_dir}/${fw_file};

fw_usb_load=if test "x${mfg_usb_fstype}" = "xext2"; then run fw_usb_load_ext2 ; else ; run fw_usb_load_fat ; fi

fw_update_raw=protect off 0xFFFA0000 FFFFFFFF;erase 0xFFFA0000 FFFFFFFF;cp.b ${fw_addr_r} 0xFFFA0000 ${filesize};cmp.b ${fw_addr_r} 0xFFFA0000 ${filesize};setenv filesize; saveenv

fw_usb_update=run clear_filesize ; run fw_usb_load ; if test 0${filesize} -gt 0; then run fw_update_raw ; else ; echo Failed update load ; fi

fw_update_nodhcp=run clear_filesize ; run fw_load ; if test 0${filesize} -gt 0; then run fw_update_raw ; else ; echo Failed update load ; fi

fw_update=dhcp ; run clear_filesize ; run fw_load ; if test 0${filesize} -gt 0; then run fw_update_raw ; else ; echo Failed update load ; fi

boot_common_args=run addtty addmisc

mfg_usb_dir=mlnx460ex

mfg_usb_load_ext2=usb start; echo "Loading ${mfg_kernel_file}";ext2load usb ${mfg_usb_dev}:${mfg_usb_part} ${kernel_addr_r} ${mfg_usb_root}${mfg_usb_dir}/${mfg_kernel_file};echo "Loading ${mfg_fdt_file}"; ext2load usb ${mfg_usb_dev}:${mfg_usb_part} ${fdt_addr_r} ${mfg_usb_root}${mfg_usb_dir}/${mfg_fdt_file};echo "Loading ${mfg_ramdisk_file}"; ext2load usb ${mfg_usb_dev}:${mfg_usb_part} ${ramdisk_addr_r} ${mfg_usb_root}${mfg_usb_dir}/${mfg_ramdisk_file}

mfg_usb_load_fat=usb start; echo "Loading ${mfg_kernel_file}";fatload usb ${mfg_usb_dev}:${mfg_usb_part} ${kernel_addr_r} ${mfg_usb_root}${mfg_usb_dir}/${mfg_kernel_file};echo "Loading ${mfg_fdt_file}"; fatload usb ${mfg_usb_dev}:${mfg_usb_part} ${fdt_addr_r} ${mfg_usb_root}${mfg_usb_dir}/${mfg_fdt_file};echo "Loading ${mfg_ramdisk_file}"; fatload usb ${mfg_usb_dev}:${mfg_usb_part} ${ramdisk_addr_r} ${mfg_usb_root}${mfg_usb_dir}/${mfg_ramdisk_file}

mfg_usb_load=if test "x${mfg_usb_fstype}" = "xext2"; then run mfg_usb_load_ext2 ; else ; run mfg_usb_load_fat ; fi

mfg_usb=echo "Manufacture will load from USB directory ${mfg_usb_root}${mfg_usb_dir}, and boot"; echo; run clear_filesize ; run mfg_usb_load; if test 0${filesize} -gt 0; then echo Booting mfg ; run mfg_args mfg_common_args;bootm ${kernel_addr_r} ${ramdisk_addr_r} ${fdt_addr_r} ; else ; echo Failed mfg load ; fi

fw_addr_r=400000

menu_addr_r=B00000

pciconfighost=1

pcie_mode=RP:RP

autoload=no

boot_usb_ext2_loc_1=run usb_args_loc_1 boot_common_args;echo "Loading ${boot_kernel_file}";ext2load usb ${boot_usb_dev}:${boot_usb_part_loc_1} ${kernel_addr_r} ${boot_usb_root}${boot_usb_dir}/${boot_kernel_file};echo "Loading ${boot_fdt_file}";ext2load usb ${boot_usb_dev}:${boot_usb_part_loc_1} ${fdt_addr_r} ${boot_usb_root}${boot_usb_dir}/${boot_fdt_file};bootm ${kernel_addr_r} - ${fdt_addr_r}

boot_usb_ext2_loc_2=run usb_args_loc_2 boot_common_args;echo "Loading ${boot_kernel_file}";ext2load usb ${boot_usb_dev}:${boot_usb_part_loc_2} ${kernel_addr_r} ${boot_usb_root}${boot_usb_dir}/${boot_kernel_file};echo "Loading ${boot_fdt_file}";ext2load usb ${boot_usb_dev}:${boot_usb_part_loc_2} ${fdt_addr_r} ${boot_usb_root}${boot_usb_dir}/${boot_fdt_file};bootm ${kernel_addr_r} - ${fdt_addr_r}

boot_usb_fat_loc_1=run usb_args_loc_1 boot_common_args;echo "Loading ${boot_kernel_file}";fatload usb ${boot_usb_dev}:${boot_usb_part_loc_1} ${kernel_addr_r} ${boot_usb_root}${boot_usb_dir}/${boot_kernel_file};echo "Loading ${boot_fdt_file}";fatload usb ${boot_usb_dev}:${boot_usb_part_loc_1} ${fdt_addr_r} ${boot_usb_root}${boot_usb_dir}/${boot_fdt_file};bootm ${kernel_addr_r} - ${fdt_addr_r}

boot_usb_fat_loc_2=run usb_args_loc_2 boot_common_args;echo "Loading ${boot_kernel_file}";fatload usb ${boot_usb_dev}:${boot_usb_part_loc_2} ${kernel_addr_r} ${boot_usb_root}${boot_usb_dir}/${boot_kernel_file};echo "Loading ${boot_fdt_file}";fatload usb ${boot_usb_dev}:${boot_usb_part_loc_2} ${fdt_addr_r} ${boot_usb_root}${boot_usb_dir}/${boot_fdt_file};bootm ${kernel_addr_r} - ${fdt_addr_r}

mfg_kernel_file=vmlinuz

mfg_ramdisk_file=rootfs

mfg_ramdisk_size=180224

mfg_fdt_file=fdt

mfg_usb_dev=0

mfg_usb_part=1

mfg_usb_fstype=fat

mfg_usb_root=/

boot_kernel_file=vmlinuz

boot_fdt_file=fdt

boot_usb_dev=0

boot_usb_part_loc_1=2

boot_usb_part_loc_2=3

boot_usb_root_loc_1=/dev/sda5

boot_usb_root_loc_2=/dev/sda6

usb_args_loc_1=setenv bootargs root=${boot_usb_root_loc_1} ro reset_button=${reset_button} rootdelay=8 ${image_kernel_args} ${extra_args}

usb_args_loc_2=setenv bootargs root=${boot_usb_root_loc_2} ro reset_button=${reset_button} rootdelay=8 ${image_kernel_args} ${extra_args}

jffs2_args=setenv bootargs root=${rootdev} rootfstype=jffs2 ro reset_button=${reset_button} ${image_kernel_args} ${extra_args}

flash_jffs2=run jffs2_args boot_common_args;bootm ${kernel_addr} - ${fdt_addr}

ethaddr=F4:52:14:CA:CC:3A

eth1addr=F4:52:14:CA:CC:3B

stdin=serial

stdout=serial

stderr=serial

reset_button=0

ethact=ppc_4xx_eth0

ver=U-Boot 2009.01 SX_PPC_M460EX SX_3.2.0330-82 ppc (Dec 20 2012 - 17:53:54)

bm_mfgmenu_allowed=0

bm_netboot_allowed=0

pn=SA002203

hwname=M460EX

location=1

kernel_addr=0xff000000

ramdisk_addr=-

fdt_addr=0xff1e0000

image_kernel_args= img_id=1 quiet loglevel=4 panic=1

rootdev=/dev/mtdblock6

bootcmd=run flash_jffs2

Environment size: 9023/16380 bytes