The restauration of BBRam of Anritsu Scorpion VNA‘s

The mainboard of the MS46xx is a slightly changed Motorola MVME162LX board. The most manuals of the board are therefore usefull and can be found in the internet, especially:

debugger manual part 1 and part 2
diagnostics manual
MVME162FX Embedded Controller Installation and Use
MVME162LX Embedded Controller ProgrammerÕs Reference Guide

The BBRam is located frrom 0xfffc000 to 0xfffc1ffff
the last 8 bytes are the clock register with date and time

The BBRam is diveded in 3 aereas, which is a bit other than in the original MVME162LX:

1. Application Aerea
from 0xfffc000 to 0xfffc16f7

The first part is the networking aerea and only used if you configure the ethernet controller.

The application stores in the rest of this aerea the alc-cal, and front panel setups and serial number of the analyzer.

For the restauration process this aerea is out of interest, but should be set in whole to an initial value. This might be a 0xaa or just 0.

2. Debugger Aerea
from 0xfffc16f8 to 0xfffc1ef7

Here are the values of the env or env;d commands are stored. It is diveded in subsections, which are guarded each by a checksum. If you do not change the default values, this aerea is not used and the debugger and boot-software works with the default values, but the env;d command wrotes the default values to this aerea.

3. The board configuration aerea
from 0xfffc1ef8 to 0xfffc1ff6
the last byte at 0xfffcff7 is a one byte checksum.

How checksums are calculated is described in the debugger manual under the description of the cs command.

This aerea stores the values of the cnfg;m command of the debugger and uses the two fields (MVME162LX Embedded Controller ProgrammerÕs Reference Guide):

char mem_pwb[8]
char _serial[8]

to store the memory layout and the mac address of your ethernet controller. You will find the last on a label of the back side of the mainbord (a hex number that begins with 00E0A0).

```
FFFC1EF0  00 00 00 00 00 00 00 00  30 31 30 30 00 00 00 00  ........0100....
FFFC1F00  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
FFFC1F10  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

```
FFFC1F20  00 00 00 00 00 00 00 00  33 32 20 20 00 E0 A0 00  ........32  ....
FFFC1F30  04 7B 00 00 00 00 31 36  32 30 34 38 30 31 32 38  .{....1620480128
FFFC1F40  30 32 30 38 00 00 00 00  00 00 00 00 00 00 00 00  0208............
FFFC1F50  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
FFFC1F60  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
FFFC1F70  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
FFFC1F80  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
FFFC1F90  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
FFFC1FA0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
FFFC1FB0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
FFFC1FC0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
FFFC1FD0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
FFFC1FE0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
FFFC1FF0  00 00 00 00 00 00 00 35  20 22 57 12 04 02 02 19  .......5 "W.....
```

0100	Debugger Version
"32 "	Clock speed
00E0A000047B	Mac-Address
16	DRAM Size in MBytes, 40 with Option 24 installed
2048	SRAM Size
0128	DSP shared memory
02	Flash Size
08	extended Flash Size, 02 and 08 works both for me

For the restauration process you need beside Michal‘s eprom:

1. Boot Utility Disk from Anritsu. Under the subdirectory SYSTEM you will find 2 files without extension, BOOT and MON.

2. The network analyzer application. Two disks. On the first Disk you will find under SYSTEM the file APP, on the second DISK APP2.

If you start the Analyzer with a corrupted BBRam content, the boot eprom invalidates the flash memory, where BOOT, MON and APP normally reside.

So it tries first to load BOOT. Therefor you should have put the disk in place before power on.

So the boot code can load BOOT and updates the internal FLASH memory.

If this is done, BOOT is started. You now will see the message "Press 1 ....". Now you should immediatly press the Key "1" on the analyzers front panel (For that you have only a short time).

After that the boot loader‘s menu is displayed. Choose 4. and load MON then start it with 6.

At first you should zero the whole BBRam. With sd command activate the diagnostic software. Type in: cf rtc. Answer the question with N. Then put in rtc adr. Then goto back to the debugger with sd. If you now make a hexdump with md fffc000:2000;b you will see, that the whole BBRam is zeroed.

After this call cnfg;m to configure the board configuration aerea of the BBRam. The debuggers prompt to put in the different values is a bit misunderstanding. Put in the values without the surrounding ". Control your input with cnfg or better with md fffc1ef0:110;b

Then I save the default env values with env;d the default values to the BBRam. This may not be always necessary. I do it, because the automatic selftest at the start of the debugger gives a warning, that something has failed (if you repeat the self with st, you get nearer information). After saving the default env parameter to BBRam the warning disappears.

Now the BBRam has only a valid board configuration and optional the env parameters. All other aereas and unused locations in the aereas are zeroed.

At this point you can change to the original boot eprom but you must not. Michals eprom is able to replace the original eprom with one difference: the loading of the application from internal memory is a bit more quickly with the original eprom.

The last step is to power down and on, to go with "1" again into the debugger and load the application and then starting it by leaving the menu with enter.

If the application starts (and it should) set time and date and make an alc-cal or restore this calibration data from disk, if you have stored it.

The last point is to restore your serial number into the BBRam.

```
FFFC0430  00 00 00 FF FF EA 80 AA  AA AA AA AA AA AA AA AA  ................
FFFC0440  AA AA AA AA AA AA AA AA  AA AA AA AA AA AA AA AA  ................
FFFC0450  AA AA AA AA AA AA AA AA  AA AA AA AA AA AA AA AA  ................
FFFC0460  AA AA AA AA AA AA AA AA  AA AA AA AA AA AA AA AA  ................
FFFC0470  AA AA AA AA AA AA AA 00  00 04 81 00 00 00 00 2A  ...............*
FFFC0480  49 4E 56 41 4C 49 44 2A  00 00 00 00 00 00 00 41  INVALID*.......A
FFFC0490  4E 52 49 54 53 55 00 00  00 00 00 00 00 00 00 00  NRITSU..........
```

with md you will find, that the application sets it to *INVALID*. You must replace it with your serial number 0xxxxx(and zero the remaining bytes of *INVALID*. Than you must calculate a word checksum with cs fffc047b fffc0497;w and replace the old 0481 with that number.