Recovering a dead MMI 3G unit

By Keldo Gliana – Q2 2014 – Version 2

Document Summary:

If you're reading this document because you have probably had unsuccessful firmware upgrade and you have killed your MMI 3G or 3G Plus unit, it probably doesn't start at all or just comes up with a black screen, you also get no background light on button control panel.

Purpose:

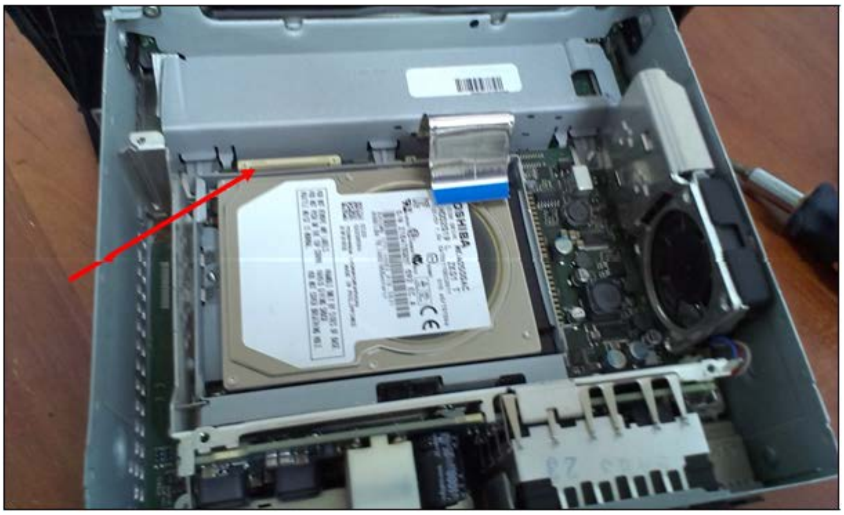
The purpose of this document is to help you attempt to recover the dead MMI unit, by accessing the unit RS232 terminal communication. If required you can also use the 'zmodem' or IPL (Initial Program Load) techniques to push the OS image back into the MMI's flash.

Scope:

All information in this document refer to the MMI 3G 9307/9308. However, in theory all other MMI variants should be the same… but use at your own risk.

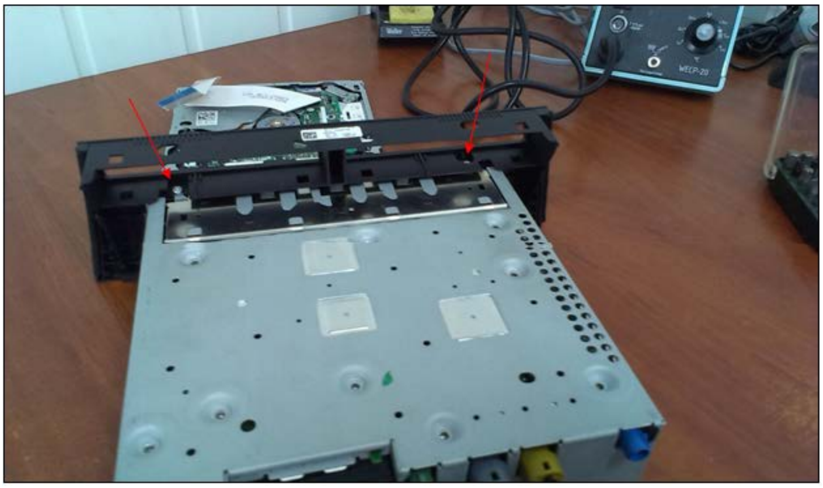
1. Open MMI unit by unscrewing all the screws at top and bottom
2. Opening the top cover

3. Remove the hard drive

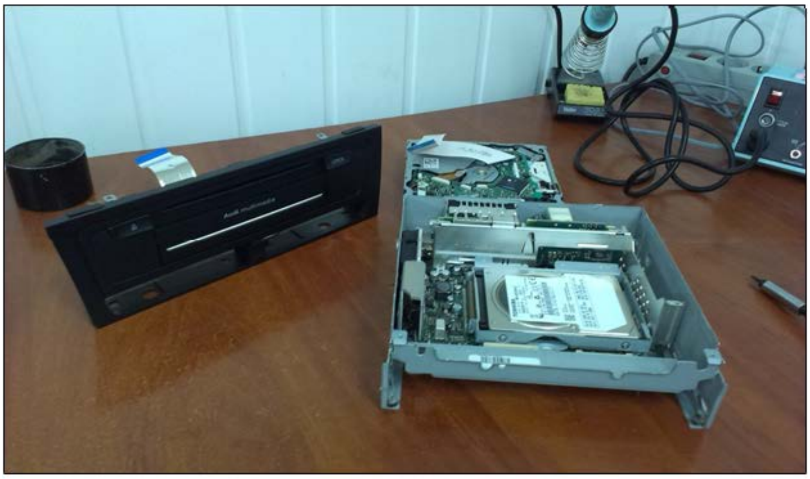


There is a small pin that holds the drive, just gently push in the clip
(where the arrow is pointing in the above picture) and the drive should pop out.

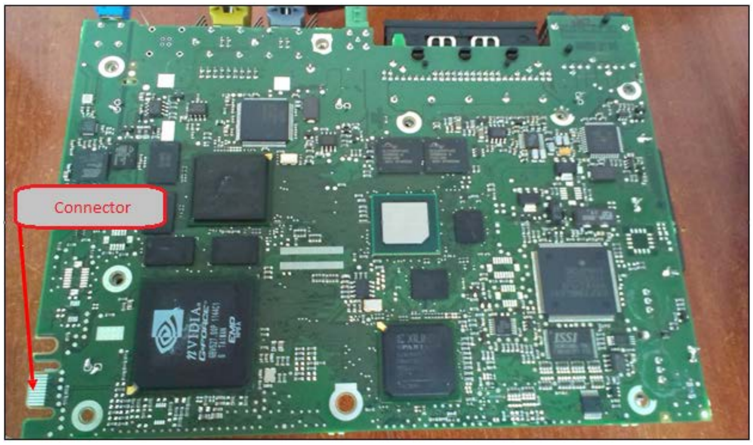
4. Remove the flexi cable that comes from the front panel (blue end).
5. If you haven't done so already, flip the unit up-side-down, and remove the two screws that hold facia panel, see below:



6. Now go ahead and remove the facia panel.

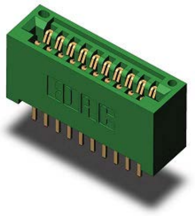


7. By now you should be at the stage shown above, if not, make your way to achieve it.
8. Now go ahead and remove the motherboard. You will have two tabs to push in to release the motherboard (they secure the front of the board), also try to avoid misplacing thermal pastes.
9. The motherboard is out, and it looks like this:



We're interested in the connector highlighted in the above picture

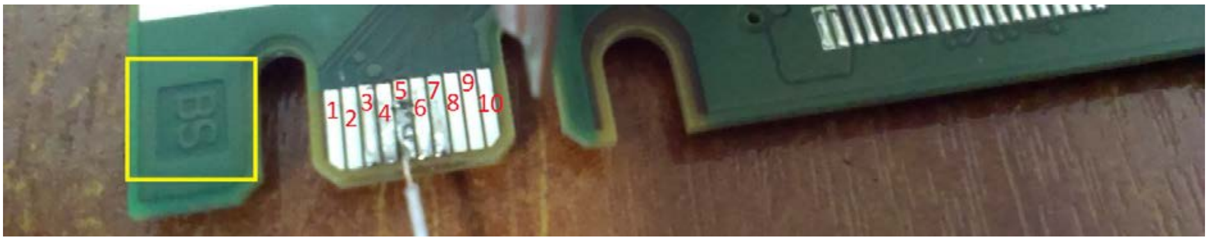
10. Either find yourself one of these Card Edge Slot Connectors:

Card Edge PCB slot connector	Card Edge connector for ribbon cable	Card Edge IDC connector
		

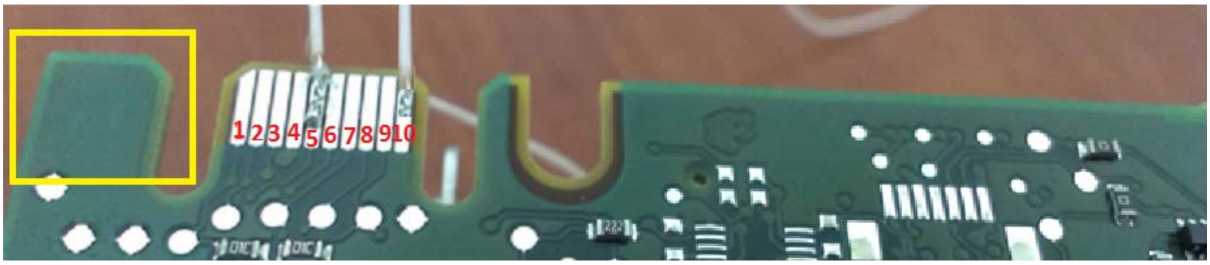
OR

Do things the cheap and ugly way by soldering wires directly to the board. The choice is yours but you must follow the schematics below, so let's identify which side is which:

This is TOP:



This is BOTTOM:

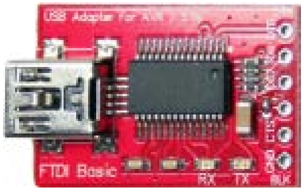


Please note: In German, it was once common practice to label the *Bestückungsseite* (assembly side) with **BS**, and the *Lötseite* (solder side) with **LS**. Fortunately this is no longer appropriate as technical environment where English is more common, the abbreviation "BS" may be misunderstood as "bottom side"! We therefore **recommend** a uniform inscription using **TOP** and **BOTTOM** to indicate the upper and lower sides.

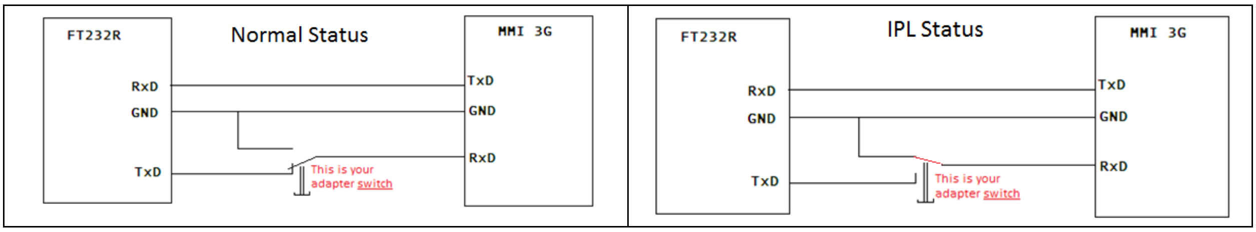
Function	Pin
RxD	Top – Pin 5
TxD	Bottom – Pin 5
Ground	Bottom – Pin 10

If you don't have a computer with COM port (RS232) you will need a good quality USB → UART adapter that has FT232RL Chip, I recommend something like this:

OR



The main thing to remember is connect the MMI's RxD and TxD in mirrored way when connecting to your RS232 adapter\cable, this to allow mouth to ear type of communication. More **importantly**, you need to make a switch in the circuit to block the RxD channel on the MMI. This is required for IPL procedure, something like this:



11. Configure your COM port with the following settings:

Port: **1 – 4** (just to simplify things)

Speed: **115200** for unit variant 9303, 9304 and others. (9302 max speed has to be 57600)

Data bits: **8**

Stop bits: **1**

Parity: **none**

Flow control: **XON / XOFF**

12. Get yourself a good terminal application such as TeraTerm v3.13

13. Connect it all up and start Troubleshooting:

If the unit MMI is not completely dead then try to troubleshoot before you try to perform IPL flashing:

1. Restart MMI using the 3 reboot buttons.
2. At this stage the Terminal Window may be empty, but you can press «Enter» and a prompt should appear " => "

Now you can run your commands recover, your commands will depend on what it needs:

```
IPL: Audi 3G 9308 (SH7785) / QNX 6.3.0 [PL_qnx_ipl-3g_9308-sh7785_08253A, built 2008/06/18_15-16-25_UTC]

CPU: SH7785 (>= 5th cut, clk mode 20. 4xDDR), 792MHz
RAM: 512 MB (probed, DBCONF=e30002)
FLASH: Spansion S29GL512N (32bit): 512 sectors x 256K
FPGA: PL_MMI3G_FPGA_9308_D1_09324A (0ms)
Entering CLI, type '?' or 'help' for help

=> help
? - alias for 'help'
bconf - display boot manager configuration
bios - start BIOS
boot - boot image in memory/flash
diag_p1j - diagnostic for the PLL1 clock jitter problem
download - download IFS over serial line (sendnto protocol)
echo - echo args
fpga - load FPGA from image in memory/flash
go - jump to 'addr'
help - print online help
hwconf - dump hardware configuration
md - memory display
memperf - memory performance test (benchmark)
memtest - memory test
mt - alias for 'memtest'
pci - lists pci devices
qinfo - display QNX build information
rsupdate - query read shifter update mode (workaround for SH7785 limitation #48)
rz - alias for 'zmodem'
scan - scan flash for images (FPGA, IFS, BIOS packet)
version - print program version
zmodem - download IFS or FPGA over serial line (Z-Modem protocol)

=> boot 780000
starting image @80780000 ...
Startup: , built 2011/07/19_09-41-26_UTC
RAM : 128 MB @0x08000000
RAM : 128 MB @0x40000000
NVRAM: 28 MB @0x5e400000
RAM : 228 MB @0x50000000
FLASH: 128 MB @0x00000000
System page at phys:0800a000 user:0800a000 kern:8800a000
Starting next program at v8804aae0
Welcome QNX6.3.2 (QNXLabel RL_qnx_os_632_PSP3_08041A)- Harman/Becker MMI3G Build 9307/08 D1-12385A
```

14. Understanding what stuff in Terminal mean:

When MMI loads after a reboot it goes into boot mode and tries to search the memory header for ifs-root and fpga, when these are not found then it triggers itself into Emergency Firmware update. The Terminal window should provide you enough information to learn what stage you have reached and where to begin your troubleshooting, a line similar to the one highlighted below is a good example:

```
starting image @680000 ...
Startup: , built 2011/07/19_09-41-26_UTC
RAM : 128 MB @0x08000000
RAM : 128 MB @0x40000000
NVRAM: 28 MB @0x5e400000
RAM : 228 MB @0x50000000
FLASH: 128 MB @0x00000000
IPL: Audi 3G 9308 (SH7785) / QNX 6.3.0 [PL_qnx_ipl-3g_9308-sh7785_08253A, built 2008/06/18_15-16-25_UTC]
CPU: SH7785 (>= 5th cut, clk mode 20. 4xDDR), 792MHz
RAM: 512 MB (probed, DBCONF=e30002)
FLASH: Spansion S29GL512N (32bit): 512 sectors x 256K
FPGA: PL_MMI3G_FPGA_9308_D1_09324A (0ms)

*** Old-fashioned mark found on IFS @0x1c0000: .zero=0x3000001e
*** Old-fashioned mark found on IFS @0x680000: .zero=0x10000000

starting image @680000 ...
Startup: , built 2011/07/19_09-41-26_UTC
RAM : 128 MB @0x08000000
RAM : 128 MB @0x40000000
NVRAM: 28 MB @0x5e400000
RAM : 228 MB @0x50000000
FLASH: 128 MB @0x00000000
Fail
```

How to read the above example: in short it says the MMI failed to start when it was attempting address the launch image 0h680000, and that's the problem as a good IFS should be located at the address 0h780000. So an easy fix for this would be to boot the unit with command:

boot 780000

An incident like this may occur due to the fact the MMI was flashed with a custom firmware that had IFS from the address 0h680000 and it was all working well, but at a later stage it was re-flashed with another firmware that had IFS 0h780000 address. So the shifting never occurred correctly, thus it killed the unit.

If the unit happens to get fix by booting it with address **boot** 780000 then a permanent fix may be applied with command:

flashit –a 0x680000 –e 1000

15. What is the best shortcut to recover?

It's absolutely essential to try to bring up the unit to Emergency Firmware update mode, with no or a very minimal flashing works. It's known that the majority of MMI units to have 128MB on board flash card, the utilisation of this storage is used as below:

Image No.	Image Address	Image Name	Start	End	Size
1.	0x00000000	IPL	63264	262144	256kb
2.	0x00040000	FPGA	746472	786432	768 kb
3.	0x00100000	FPGA EMERGENCY	746472	786432	768 kb
4.	0x001C0000	IFS EMERGENCY	4,666,284	6,029,312	5888kb (5MB)
5.	0x00780000	IFS - ROOT	36866456	50855936	49664kb (48MB)
6.	0x03800000	EFS EXTEND	23,068,672	58,720,256	57344kb (56MB)
7.	0x04E00000	EFS SYSTEM	20971520	20971520	20480kb (20MB)
8.	0x06200000	EFS PERSISTENCE	30,408,704	31,457,280	29696kb (29MB)
9.	0x07f00000	BIOS			
10.	0x08000000	END OF FLASH			

So if the integrity of Emergency Firmware update on the flash is healthy then you should always try to boot into it first. To do this, boot the system from the address 1c0000 by running the following command:

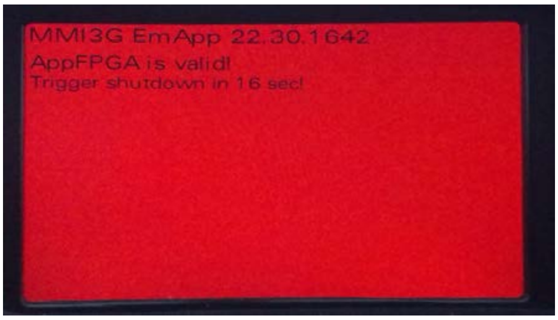
boot 1c0000

After you run the above command you will need to insert firmware SD card into the MMI unit, make sure the firmware is designed for this particular unit. If it all works well you will see something similar to this in your Terminal Windows:

```
=> boot 1c0000
starting image @801c0000 ...
Startup: PL_qnx_startup-sh7785_08362A, built 2008/09/02_07-53-12_UTC
RAM : 256 MB @0x08000000
FLASH: 128 MB @0x00000000
System page at phys:0800a000 user:0800a000 kern:8800a000
Starting next program at v88046840
------------------------------------------------------
Welcome to QNX 6.3.2 (RL_qnx_os_632_PSP3_08124A) MMI3G Audi EMERGENCY SETUP 9308 D1 09431A
starting pci driver
*** init FPGA registers ***
starting dev-ipc for IOC
initialize IPC Driver
starting network driver
starting atapi driver - setup device /dev/cam0
starting devc-pty
Path=0 - XILINX MMI3G_8
target=0 lun=0 CD-ROM(5) - FJ-TEN DVD-ROM DV-05FT2 Rev: 1B20
Path=1 - XILINX MMI3G_8
target=1 lun=0 Direct-Access(0) - TOSHIBA MK4050GA Rev: TG10
starting flash driver
starting SDCard driver
Unable to access "/dev/io-net/en5" (2)
starting with no network driver
===========================================================
Emergency IFS is started. Emergency app tries to restore the application.
===========================================================
49169 running /usr/bin/devg-NVTIRM
DMA is Disabled
dma disabled, interrupt thread priority = 100
49171 running /sbin/io-media-nvidia
49172 running /usr/sbin/io-display
49173 running /usr/apps/MMI3GEmergencyApp
===========================================================
Emergency shell started, fix your system and restart again
===========================================================
discplayer_init: IN
nvidia_render_manager_init: IN
INvmRenderManager::GetNvmRenderManager (&pRenderManager);
er = pRenderManager->Init(oConfig);
# App FPGA valid 1
Emergency FPGA valid 1
Disabling IOC watchdog
EmergencyApp-MMI3G: version 22.30.1642, date of build 14.10.2009

<-- A LOT MORE TEXT I HAVE TRIMMED HERE -->
** DeveloperMode is off!!!! **
** The system will shutdown soon! **
** DO NOT FLASH WITH SCRIPTS IN THIS MODE!!! **
***********************************************************
** You can enable DeveloperMode like this: **
** echo dummy > /HBpersistence/DeveloperMode **
***********************************************************
Set DisplayStatus to 2
OnOff: process IPC event HWsample 209 HWvariant 7 FinalShutdown 0
Enabling IOC watchdog
Initiate shutdown (after BEM timeout)...
OnOff: process IPC event HWsample 209 HWvariant 7 FinalShutdown 0
```

Somewhere towards the end, before you see Enabling IOC watchdog text, the MMI display turns red, with the words, everything is in order, MMI copies few files and then it’s switched off automatically.



MMI3G EmApp 22.30.1642
AppFPGA is valid!
Trigger shutdown in 16 sec!

If the firmware is copied successfully back from SD into the flash, then you’re done… your MMI is back alive.

Sometimes the time of 60 seconds given to perform a certain process is too short, if you need to extend this just run the following command:

echo dummy > /HBpersistence/DeveloperMode

16. How to flash certain images from a source such as SD card?

If you know what you're doing then you probably realised that it's best to only flash the missing images, for example to flash ifs-root.ifs from SD card, you would use the following command:

```
flashit 0x00680000 -a -f / mnt/sdcard10t11/ifs-root.ifs
```

More syntaxes on flashit are listed below:
-a address start address in hex the file should be flashed to
-d if set, all necessary sectors will be erased
-e size erase size number of bytes
-f file name of the file to be flashed
-l step [obsolete / not supported: stepwidth of progress procentage]
-p device path to raw flash device (default: / dev/fs0; NAND is / dev/etfs1)
-x do not verify data written
-v increase verbosity

17. If the unit MMI is completely dead then you will need to initiate IPL process:

a. Turn on adapter's switch (or somehow connect MMI's RxD to Adapter's ground)
b. Restart MMI using the 3 reboot buttons.
c. Turn off Adapter's switch (remove the connection between MMI's RxD and ground)
d. MMI will now go into boot mode IPL.

> Please note that IPL is always at address 0 and it only receives data when MMI's RxD terminal port is connected to GND port at the reboot. Otherwise MMI searches the memory header for ifs-root and fpga, when these are not found then it triggers MMI unit into Emergency Firmware update.

If IFS emergency is completely dead, the MMI screen will remain black. When that happens you will get '#' prompt in the terminal window. Form here you will need to delete header FPGA by running command:

```
flashit-a 40000 -e in 1000
```

The above command will erase the FPGA image, and on the next MMI reboot the unit is forced into disaster recovery mode, this will enable IPL process to begin. Now you will be required to push a small version of QNX ifs image in order to boot it

I have put this document together from collecting a lot of information from many different online forums; PM conversations, offline notes, hands-on experience with aim to help you recover your bricked MMI and I hope it helps.

END OF DOCUMENT