***Boot***

## Table des matières



# 1 Introduction

Voici la documentation de PPCMon.

# 2 Booter depuis le firmware PPCMon

Valeurs d'origines :

| | | |
|---|---|---|
| inet_host | IP number of this target | 192.168.1.168 |
| inet_bplane | IP on BpNet (optional) | 0.0.0.0 |
| inet_bootserver | IP number of TFTP and NFS server | 192.168.1.9 |
| inet_gateway | Default gateway (optional) | 192.168.1.3 |
| inet_nameserver | Nameserver : DNS (optional) | 192.168.1.3 |
| inet_protocol | do we use the BOOTP protocol | arpa |
| inet_mount | NFS mount point on server machine | 192.168.1.9 :/opt/CES/devkit/target8 |
| inet_ethermode | | auto |

| | | |
|---|---|---|
| boot_flags | Not used to by Linux | a |
| boot_device | Boot from local ethernet | le |
| boot_filename | Filename in tftp for the kernel image | /tftpboot/zImage8 |
| boot_rootfs | Mount NFS root filesystem | nfs |
| boot_delay | Should be > 0 | 3 sec |
| boot_address | Load address should be a multiple of 16kB | 1000000 |
| boot_size | Memory used by Linux (0=all) | 0 |
| boot_fast | Must be disabled | n |
| boot_filetype | Must be 'binary' for zImage | binary |
| boot_cpunr | Required on VME : Node id for bpNet | 0 |
| boot_mode | ppcmon VS auto | auto |
| boot_line | Linux kernel parameters | ex : ip= : : : : :eth1 |

# 3 Configurations particulières

## 3.1 C18 : banc de test Corona

| | |
|---|---|
| inet_bootserver | 192.168.1.9 |
| inet_mount | 192.168.1.9 :/opt/sugarhat/devkit/target18 |

| | |
|---|---|
| boot_filename | /tftpboot/zImage-RIOC4068 |
| boot_delay | 3 sec |
| boot_line | |

## 3.2 C19 : module 20 tirroirs

| | |
|---|---|
| inet_bootserver | 192.168.1.9 |
| inet_mount | 192.168.1.9 :/opt/sugarhat/devkit/target19 |

| | |
|---|---|
| boot_filename | /tftpboot/zImage19cameraTest |
| boot_line | |

# 4 Utilisation des drivers trigers

Lors du chargement des drivers des cartes triggers, on obtient un message d'erreur relatif aux paramètres. Si l'on creuse avec 'dmesg' précise qu'une erreur de collision empêche le driver 'trigger' de se charger. Pour y remédier, il faut positionner le paramètre `diag_cpci` à `y`.

```
ppcmon> set diag_cpci
--> y
```

# 5 Utilisation de cartes à partir du 2ème segment

Pour booter avec des cartes au delà du premier segment :

```
ppcmon> set diag_cpci
--> y
ppcmon> set diag_xpci
--> y
```

# 6 Plusieurs cartes processeurs

Pour booter avec 2 cartes controleur, si des cartes étant présentes au delà du premier segment alors il faut ajouter l'option du noyau `cpci_full_enum` sinon la deuxième carte refuse de booter. Par ailleurs cette option est aussi requise pour l'unique processeur du châssis trigger.

```
ppcmon> set boot_line
--> cpci_full_enum
```

**Remarque :** On ne sait pas encore bouter avec des cartes controleur au delà du premier segment du châssis 21 slot.

# 7 Charger les drivers au démarage

Toutes les opération décrites ci-dessous se passent sur cible.

- compiler les drivers

```
$ telnet camera16
# su roche
$ cd ~/SBig/Driver/TmgTs/
$ make
tmgts.h:20:26: error: linux/config.h: No such file or directory
tmgts.h:25:27: error: linux/wrapper.h: No such file or directory
```

- supprimer les inclusion des fichier *linux/config.h* et *linux/wrapper.h*

```
$ make
```

- ranger les drivers dans */lib/modules/2.6.24.7-rt21-shl-3.2.3*

```
# mkdir /lib/modules/2.6.24.7-rt21-shl-3.2.3/papat2.6
# cp ~roche/SBig/Driver/TmgTs/tmgts.ko /lib/modules/2.6.24.7-rt21-shl-3.2.3/papat2.6/
```

- mettre à jour la liste de dépendance des modules */lib/modules/.../modules.dep*

```
# depmod -a
# modprobe tmgts
# modprobe -r tmgts
```

- créer le fichier */etc/sysconfig/module/xxx.modules* avec les droits d'execution.

```
# grep -n 'sysconfig/modules' -B1 -A2 /etc/rc.sysinit
151-# Load other user-defined modules
152:for file in /etc/sysconfig/modules/*.modules ; do
153-  [ -x $file ] && $file
154-done

# cat > /etc/sysconfig/modules/emilie.modules << FIN
> modprobe trig_drv26
> modprobe tmgts
> FIN

# chmod +x /etc/sysconfig/modules/emilie.modules
# /etc/sysconfig/modules/emilie.modules
# lsmod
# reboot
# lsmod
```

En fait, mieux vaut utiliser le fichier */etc/rc.local* pour les drivers développés car il permet de charger des drivers depuis les */home* montés via NFS.