

The Smart**FIRMWARE**

www.genesi.lu

When the Pegasos boots, then it is the SmartFirmware that initialises the hardware, performs diagnostics and starts the operating system.

Normally this happens automatically without any interaction from the user. However, it's possible to abort this process by pressing "Escape" to enter a command line interface. Through this interface it's possible to select the boot device, check which hardware is recognised by the system, and many other things. This document only explains the most commonly needed commands.



400, route d'Esch, B.P. 1443 L-1014, Luxembourg
www.genesi.lu

1. Changing videomode

Pressing F6 - F9 will change between different videomodes.

2. Command line history, tab-completion and help

Like a normal shell, it's possible to access the last command using the arrow-up key. It's also possible to write just the first characters in a command and press tab to complete it. If there's more than one match, you can press tab twice to get a list of all commands starting with the characters you've typed.

To get information about a command xyz, type
help xyz

3. Keeping the output on screen

By default, if the information being printed to the screen doesn't fit, it will quickly scroll past and you won't be able to read it. To change this, type:

18 to lines/page

This means that when more than 18 lines are being shown, the output will be halted until the user presses a key.

Perhaps a more useful setting would be to type:

-1 to lines/page

This means the output will be paused when the max page lines minus 1 is displayed, ie it will display 1 full page at a time...

4. Listing devices recognised by the system

Enter
show-devs

A long list of devices will be printed. This includes memory and the cpu but for our purposes the pci devices are the most interesting. These include not only pci cards but also IDE, USB and firewire devices as well as built-in sound and ethernet. Now enter

cd /pci

and then
ls

This will list all the pci devices. This should include ide. Enter
cd ide

and
ls

Now you'll get a list of IDE devices. Let's say this shows the following:

disk@0,0
disk@1,0

What does this mean? The first part, "disk" indicates that it's a hard drive. The first number after the @ is the IDE interface, where a 0 indicates the innermost IDE connector on the motherboard, and a 1 means it's the outermost connector. The number after the comma is the unit number on this connector - either 0 or 1. So in the case above, there is one disk on each connector.

Now, let's see what the disks contain. Type
ls /pci/ide/disk@0,0 (or adapt this to some disk in your system)

This will display something like the following:
RDB partition 0 <FFS>: <DH0> (0x444F5301)
RDB partition 1 <SFS>: <DH1> (0x53465300)

This means there are two partitions, 0 and 1. The first one has an FFS filesystem while the second uses the SFS filesystem. Then follows the device name of the partition, and last of all the dostype identified with the filesystem.

5. Listing files on the devices

Now let's say we want to know what's on the partitions. Enter

ls /pci/ide/disk@0,0:0

This will show a list of the first partition on the first device on the first ide interface.

To list subdirectories, type something like
ls /pci/ide/disk@0,0:0 devs/datatypes

6. Examining and setting SmartFirmware variables

There are a number of variables which can be set to configure the behaviour of SmartFirmware during startup. To see which variables exist and their values, type
printenv

Let's say we want to change the boot delay from 5 seconds to 1 second. This is done with
setenv auto-boot-timeout 1000

7. Booting

When booting, a boot-file and a boot-device is needed. The default is stored in the boot-device and boot-file variables. However, you can also select a boot file from the SmartFirmware prompt by typing something like
boot /pci/ide/disk@0,0:0 boot.img

This will boot from the file boot.img located on partition 0 of the first unit on the first connector.

8. Passing parameters to MorphOS during startup

When booting, parameters can be passed to MorphOS. This way you can select a different boot device and control debug output from MorphOS. Normally this output is sent via the serial port. It can then be captured by a terminal program on another computer, which makes it possible to keep the debug output even if the Pegasos machine has crashed or hanged.

However, this is not always desirable, since you might want to use the serial port for other things, and also the serial output slows down operation. It's also possible to enable extra debug output for some parts of the system. These are not described too much here, but if you submit a bug you may be asked to enable one of these.

General options:

QDebugFlags - set debug flags for Quark (see below)

DDebugFlags - set ABox's dos debug flags (see below)

EDebugFlags - set debug flags for the ABox (see below)

BootDevice - set boot device

RamDebug - disable serial output and keep logs in resident ram instead

MaxHits - set a maximum number of hits, after which the system is halted

FullHit - enable extra debug information

BackTraceDepth - select how many stackframes are shown

NoL2Cache - turn off the L2 cache

QDebugFlags:

Init, EmulMapping, MMU, Log, JoyDebug, Exception

DDebugFlags:

Init, DosList, Segment, CreateProc, SimpleFuncs, LoadSeg, Files, Locks, RawIO, Match, Pattern, ReadArgs, Packets, DeviceProc, InternalLoadSeg

EDebugFlags:

InitResident, InitCode, FindResident, CreateLibrary, SetFunction, NewSetFunction, ChipRam, AddTask, RemTask, GetTaskAttr, SetTaskAttr, ExceptHandler, AddDosNode, PCI, LoadSeg, UnloadSeg, PPCStart, Init, Log

Some examples:

Enabling some extra debug flags

boot boot.img QDebugFlags="Init MMU" EDebugFlags="PCI"

Booting from dh1: instead of the device with highest BootPri

boot boot.img bootdevice dh1

Same, but also get the boot.img from a different device

boot /pci/ide/disk@0,0:1 boot.img bootdevice dh1

If you just want to disable serial output, use something like

boot boot.img ramdebug

If you want to make this the permanent option, set the boot-file variable

setenv boot-file boot.img ramdebug