

Utilize Ronetix Development Tools Virtual Machine to build OpenEmbedded-classic for PM9G45

Demonstration



Acknowledgement

December, 2011

Ronetix has made every attempt to ensure that the information in this document is accurate and complete. However, Ronetix assumes no responsibility for any errors, omissions, or for any consequences resulting from the use of the information included herein or the equipment it accompanies. Ronetix reserves the right to make changes in its products and specifications at any time without notice. Any software described in this document is furnished under a license or non-disclosure agreement. It is against the law to copy this software on magnetic tape, disk, or other medium for any purpose other than the licensee's personal use.

Ronetix Development Tools GmbH

Waidhausenstrasse 13/5

1140 Vienna

Austria

Tel: +43-720-500315

+43-1-956-3138

Fax: +43-1-8174-955-3464

Web: www.ronetix.at

E-Mail: info@ronetix.at

Acknowledgments:

ARM, ARM7, ARM9, ARM11 are trademarks of ARM Ltd. Windows, Win32, Windows CE are trademarks of Microsoft Corporation. Ethernet is a trademark of XEROX. All other trademarks are trademarks of their respective companies.

Copyright© 2010-2011 Ronetix Development Tools GmbH

All rights reserved.

Contents

How to play with OE-classic on PM9G45 using the Ronetix Development Tools VM . . . 4
Reference 7

Change log

December 2011	- Initial release.
---------------	--------------------

This demonstration is based on [OpenEmbedded ReadMe](#) and aims to show you how to utilize the provided Virtual Machine to build and work with OpenEmbedded-classic on SK-PM9G45.



The things you will need are:

- disk space of 20GiB for the tar.bz2 archive of the VM on the host PC
- disk space of 50GiB to play the VM
- one COM port (COM1 and so on, or ttyS0 and so on) on the host PC
- Ronetix Development Tools Virtual Machine
- installed VMware Player
- SK-PM9G45 board
- Ethernet switch/hub or vacant LAN port on the host PC

How to play with OE-classic on PM9G45 using the Ronetix Development Tools VM

At first, some cables have to be connected. Connect the SK-PM9G45 to an Ethernet together with the host PC. The VM Ethernet adapter is bridged to the Ethernet of the host PC and gets its IP from a DHCP server. Static IP can be set, too.

A serial console to the board will be needed, too. To be able to attach the COM port to the VM you have to be ensure that no software is using the intended COM port.

Attach the serial port to the VM if it is not - the symbol  means that serial port is attached, and the presence of the cross  shows disconnected serial port. The default serial port to be attached is /dev/ttyS0, also named COM1 in DOS/Windows environment. Which serial port to attach is selected in the VM(virtual machine) configuration, which configuration is editable in VMware player.

Now it is time to play the VM. It will take around 10 minutes until it gets ready to play with SK-PM9G45. From now on all commands and activities have to be done in the VM if something else is not said.

Edit the PEEDI configuration file and change the IP address of `CORE0_PATH` parameter to match the IP of the VM.

Open the "Terminal Emulator"(its icon is on the desktop) and in it type

```
dev@ronetix:~$ screen /dev/ttyS0 115200
```

to create serial terminal console on the ttyS0/COM1. And do not type `dev@ronetix:~$` , this is the command prompt.

In the same Terminal Emulator window make a new tab with keyboard combination Ctrl-Shift-T(or in the menu File->New tab). In which new tab telnet to your PEEDI programming box as shown(substitute that IP with the one of your PEEDI).

```
dev@ronetix:~$ telnet 192.168.10.42
```

On the VM there are anonymous FTP and TFTP servers with base directory pointed at /var/lib/tftpboot. The files to write using PEEDI have to be located at that base directory. Using hard links to the created images save disk space. The files have already been build so no need to rebuild them. The next five commands move the files to that base directory.

```
dev@ronetix:~$ cd /var/lib/tftpboot/pm9g45
dev@ronetix:/var/lib/tftpboot/pm9g45$ ln -fL /home/dev/usr/src/oe/build
/kaeilos/tmp/deploy/eglibc/images/ronetix-pm9g45/at91bootstrap.bin
at91bootstrap.bin
dev@ronetix:/var/lib/tftpboot/pm9g45$ ln -fL /home/dev/usr/src/oe/build
/kaeilos/tmp/deploy/eglibc/images/ronetix-pm9g45/u-boot-ronetix-pm9g45.bin
u-boot.bin
dev@ronetix:/var/lib/tftpboot/pm9g45$ ln -fL /home/dev/usr/src/oe/build
/kaeilos/tmp/deploy/eglibc/images/ronetix-pm9g45/uImage-ronetix-pm9g45.bin
uImage
dev@ronetix:/var/lib/tftpboot/pm9g45$ ln -fL /home/dev/usr/src/oe/build
/kaeilos/tmp/deploy/eglibc/images/ronetix-pm9g45/custom-extra-image-ronetix
-pm9g45.jffs2 ce-rootfs.jffs2
```

To program the images go to the PEEDI telnet console and run the next commands.

To program the SK-PM9G45 with all images, its NAND memory have to be empty.

```
PM9G45> f s 0
Selected FLASH section:  BOOTSTRAP
PM9G45> f e
Performing mass erase ...
```

Next, the AT91Bootstrap have to be written:

```
PM9G45> f p pm9g45/at91bootstrap.bin bin 0x0
++ info: Programming image file:  tftp://192.168.10.69/pm9g45/
at91bootstrap.bin
++ info: Programming using agent, buffer = 32768 bytes
++ info: At absolute address:  0x00000000
programming at 0x00000000

++ info:  successfully programmed 6.15 KB in 0.05 sec
```

The U-Boot programming follows:

```
PM9G45> f s 1
Selected FLASH section:  U-BOOT
PM9G45>
PM9G45> f p pm9g45/u-boot.bin bin 0x20000
++ info: Programming image file:  tftp://192.168.10.69/pm9g45/u-boot.bin
++ info: Programming using agent, buffer = 32768 bytes
++ info: At absolute address:  0x00020000
programming at 0x00020000
programming at 0x00028000
programming at 0x00030000
programming at 0x00038000
programming at 0x00040000
programming at 0x00048000
programming at 0x00050000

++ info:  successfully programmed 211.76 KB in 0.57 sec
```

Time to put the Linux kernel in the NAND:

```
PM9G45> f s 2
Selected FLASH section:  KERNEL
PM9G45> f p pm9g45/uImage bin 0x200000
++ info: Programming image file:  tftp://192.168.10.69/pm9g45/uImage
++ info: Programming using agent, buffer = 32768 bytes
++ info: At absolute address:  0x00200000
programming at 0x00200000
programming at 0x00208000
...
programming at 0x003E0000
programming at 0x003E8000

++ info:  successfully programmed 1.91 MB in 5.82 sec
```

The last image to program is the Root file system:

```
PM9G45> f s 3
Selected FLASH section:  ROOTFS
PM9G45> f p pm9g45/ce-rootfs.jffs2 bin 0x400000
++ info: Programming image file:  tftp://192.168.10.69/pm9g45/ce-rootfs.jffs2
++ info: Programming using agent, buffer = 32768 bytes
++ info: At absolute address:  0x00400000
programming at 0x00400000
programming at 0x00408000
...
programming at 0x02A70000
programming at 0x02A78000

++ info:  successfully programmed 38.50 MB in 116.95 sec
```

Finally, detach the target from the PEEDI either with detaching the JTAG cable or with PEEDI detach command.

```
PM9G45> ta de
++ info: power fail detected
++ info: no target power
The target is now detached
PM9G45>
```

The PEEDI is detached from the target and the SK-PM9G45 is ready to go.

The SK-PM9G45 have to be power-cycled to boot properly. Power cycle it. It will take 2-3 minutes to load completely and the GPE login screen will appear.

The touch pointer have to be calibrated manually. In the terminal window with running screen do this:

```
dev@ronetix:~$ /etc/init.d/gpe-dm stop
Stopping GPE display manager:  gpe-dm
dev@ronetix:~$ ts_calibrate
```

and follow the pointer on the screen then reboot.

These commands have to be run manual, after each reboot so that the X has its calibration data.

```
dev@ronetix:~$ export DISPLAY=localhost:0.0
dev@ronetix:~$ . /etc/pointercal.xinput
```

The X touch input (xinput) needs adjustments, software fixes, the framebuffer touchscreen (ts_lib) is more precise.

Initially the login screen will ask you to create an user. Fill in the fields and confirm with OK button.

A few seconds and the desktop manager GPE will be usable.

To manipulate screen saver and balnd pattern see "Screen blanking, screen saver and screen DPMI" in the [OpenEmbedded ReadMe](#)

Reference

OpenEmbedded ReadMe

<http://download.ronetix.info/boards/README/README-OpenEmbedded.ronetix>

PM9G45 datasheet

<http://download.ronetix.info/boards/doc/PM9G45/SK-PM9G45-datasheet.pdf>