RD4740 PAVO Quick Start

Revision: 1.1
Date: Sep 2009

北京君正集成电路有限公司
Ingenic Semiconductor Co. Ltd

RD4740 PAVO Quick Start

Release history

Date	Revision	Change
Sep 2007	1.0	First release
Sep 2009	1.1	Add USB boot section and delete JDI boot section

Disclaimer

This documentation is provided for use with Ingenic products. No license to Ingenic property rights is granted. Ingenic assumes no liability, provides no warranty either expressed or implied relating to the usage, or intellectual property right infringement except as provided for by Ingenic Terms and Conditions of Sale.

Ingenic products are not designed for and should not be used in any medical or life sustaining or supporting equipment.

All information in this document should be treated as preliminary. Ingenic may make changes to this document without notice. Anyone relying on this documentation should contact Ingenic for the current documentation and errata.

Ingenic Semiconductor Co., Ltd.
Room 108, Information Center Block A
Zhongguancun Software Park, 8 Dongbeiwang west Road,
Haidian District, Beijing China, 100193
Tel: 86-10-82826661
Fax: 86-10-82825845
Http: //www.ingenic.cn

Content

RD4740 PAVO Quick Start

1 Overview

JZ4740 is an embedded processor introduced by Ingenic Semiconductor Co., Ltd, which integrates a high performance and low power 32-bit MIPS core. Its main frequency is 360MHz. JZ4740 supports MMU, with 16K instruction cache and 16K data cache, and also integrates many peripherals. JZ4740 supports many Embedded Operating Systems such as Linux™, WinCE™, etc. This document will introduce how to use the reference design board PAVO and help readers gain a quick start of Linux and WinCE.

2　　　Quick start Linux

2.1　Compile U-Boot

Compile as the following step and generate the **u-boot-nand.bin**:

    Firstly, prepare the U-Boot source:

     # tar –xjf u-boot-1.1.6.tar.bz2

     # cd u-boot-1.1.6

     # gzip –cd ../u-boot-1.1.6-jz-yyyymmdd.patch.gz | patch –p1

    And then Compile as following steps:

     $ make pavo_nand_config

     $ make

2.1.1　Use USB tool program u-boot

Before programming u-boot, now introduce the way of PAVO board boot from usb device:

1） Press **SW5** and **reset** key, if success you can see a new usb device on PC.

2） The SDRAM is 64MB on PAVO board. The NandFlash type is K9G8G08U0M. Rename **USBBoot_PAVO.cfg** to **USBBoot.cfg** in the toolkit. **USBBoot_PAVO.cfg** includes default configuration of the PAVO board.

3） Run **usb_boot.exe**, if the connection is correct, execute **list** command you can see the number of device is not zero.

4） Execute **boot** command, the following information can be seen：

     USBBoot :> boot 0

     Checking state of No.0 device: Unboot

     Now booting No.0 device:

     Download stage one program and execute at 0x80002000: Pass

     Download stage two program and execute at 0x80c00000: Pass

     Boot success!

     Now configure No.0 device:

     Now checking whether all configure args valid:

     Current device information: CPU is Jz4740

     Crystal work at 12MHz, the CCLK up to 336MHz and PMH_CLK up to 112MHz

     Total SDRAM size is 16 MB, work in 4 bank and 16 bit mode

     Nand page size 2048, ECC offset 28, bad block ID 127, use 1 plane mode

     Configure success!

Execute programming command:

     USBBoot :> nerase 0 8 0 0

     USBBoot :> nprog 0 u-boot-nand.bin 0 0 –n

To here, the U-Boot has been programmed to target board.

The serial port's configuration is ：57600bps, 8N1

Now, connect the serial port of target board to PC, start HyperTerminal and set parameters. And then hardware reset the target board, start the U-Boot, there will be start message of U-Boot on HyperTerminal console.

2.2   Compile and start Linux

Firstly, compile and generate **uImage** as following step:

    $ tar xjf linux-2.6.24.3.tar.bz2
    $ cd linux-2.6.24.3
    $ gzip -cd ../linux-2.6.24.3-jz-yyyymmdd.patch.gz | patch –p1
    $ make pavo_defconfig
    $ make uImage

2.2.1   Use USB tool program uImage

Use usb boot tool program uImage is similar to program u-boot. Because reading uImage from NAND to SDRAM is through u-boot, so please be sure that the corresponding of the NAND configuration file between u-boot and USB boot tool.

Execute programming command:

    USBBoot :> nprog 2048 uImage 0 0 –n

To here, the uImage has been programmed to target board too.

Execute as following to program YAFFS2 file:

    USBboot :> nprog 4096 root.yaffs2 0 0 -o

After power on the board, there will be output on the console via serial port and LCD panel. After a moment, the demo of a QTOPIA application will be launched.

3   Quick start WINCE

3.1 Use USB download Bootloader

1．When you get BSP，in the PLATFORM\VENUS\Bin\USBBOOT\JZ4740directory,there is a compiled boot.nb0,execute USB_Boot.exe, you will see a command window, the following information can be seen：

> Welcome!
> USB Boot Host Software!
> USB Boot Software current version: 1.4b
> Handling user command.
> USBBoot :>

2. Insert USB. if the device is not recognized，you need to install the driver that is placed in this director. Press **SW5** key，then press **reset** key，then loosen **reset** key，you will see the device icon on the PC. Enter "Boot 0" in the command line，then press enter key, the following information can be seen：

> USBBoot :> boot 0
> Checking state of No.0 device: Unboot
> Now booting No.0 device:
> Download stage one program and execute at 0x80002000: Pass
> Download stage two program and execute at 0x80c00000: Pass
> Boot success!
> Now configure No.0 device:
> Now checking whether all configure args valid:
> Current device information: CPU is Jz4740
> Crystal work at 24MHz, the CCLK up to 336MHz and PMH_CLK up to 112MHz
> Total SDRAM size is 16 MB, work in 4 bank and 16 bit mode
> Nand page size 2048, ECC offset 3, bad block ID 127, use 1 plane mode
> Configure success!
> USBBoot :>

3. Enter "nprog 0 boot.nb0 0 0 –n" in the command line, the following information can be seen：

> USBBoot :> nprog 0 boot.nb0 0 0 -n
> Programing No.0 device...
> Erasing No.0 device No.0 flash...... Finish!
> Operation end position : 1
> Force erase ,no bad block infomation !
> Total size to send in byte is :262144
> Image type : without oob
> It will cause 2 times buffer transfer.
> No.1 Programming... Finish! Checking...

Now, we have already downloaded the boot.nb0 in the JZ4740 board successfully.

RD4740 PAVO Quick Start

3.2 Start WinCE

1. first we can use ⅹ ⅹ ⅹ.msi.msi or VENUS.cec file to install BSP, if we use VENUS.cec file ,the step is followed :
Select Manage Catalog Items from the File toolbar in the PB environment ,you will see a dialog box, then import relevant cec file .the picture is showed
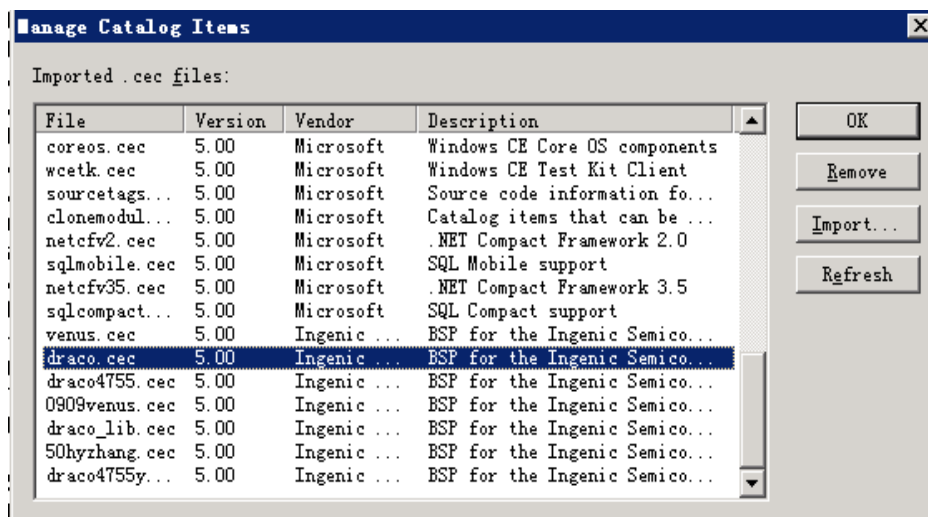
2. Open VENUS.bat file, define project name:
    SET BSP_BASED_CHIP=JZ4740
    SET BSP_BASED_BOARD=PAVO
Open JZ4740_PAVO.bat file in the \ VENUS\Files\SCRIPT directory, you can add some driver selectively, for example:
    SET BSP_PAVO_WAVEDEV_INTERNAL=1
    SET BSP_PAVO_MEDIA_PLAYER=1
After modification, select "Open Release Directory" from PB menu "Build OS" to enter PB50 command line, execute "set bsp" command to confirm whether the environment variable is right.

3. When the setting is OK we execute Sysgen, there is a NK.bin in the WINCE500\PBWorkspaces\Venus\RelDir\VENUS_MIPSII_Release directory. Then select "Open Release Directory" from PB menu "Build OS" to enter PB50 command line, we don't need to change this directory, execute "getnb0.bat" command directly, a Nk.nb0 is generated in this directory.
We copy the NK.nb0 from the directory to the SD card, then reset the development board after insert the SD card to the slot.

Note: Here VENUS is project name.

  RD4740 PAVO Quick Start