Ingenic@

Linux X1000 development

Halley2 series

Date: Jan.13 2016

北京君正集成电路股份有限公司
Ingenic Semiconductor Co., Ltd.

君正 Ingenic

Ingenic@Linux X1000 development

Release history

Date	Revision	Revision History
Jan.13, 2015	1.0	- First released

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.
Ingenic Headquarters, East Bldg. 14, Courtyard #10, Xibeiwang East Road, Haidian Dist., Beijing, China, 100193
Tel: 86-10-56345000
Fax: 86-10-56345001
Http: //www.ingenic.cn

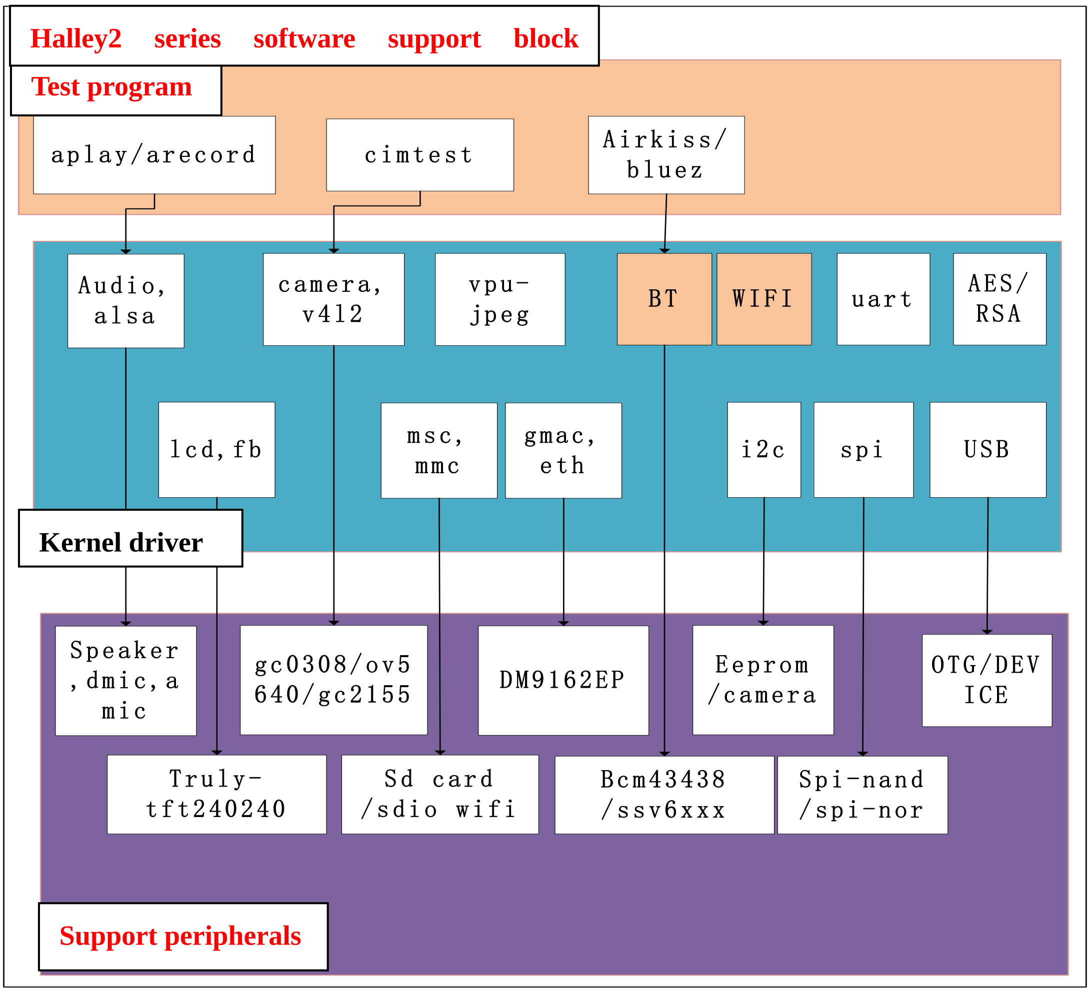
1. Introduction

1.1. Purpose and background

Ingenic processor is high integration, high performance and low power consumption of a 32-bit RISC processor.With MMU and data and instruction Cache, and plenty of peripherals, you can run the Linux operating system.This article will introduce to the readers how to config the kernel based on ingenic processor and develop apllication .it can lead developer for Linux development as soon as possible.Including the u-boot configurations, Linux3.10 kernel and drivers, file system of production and development, the configuration and application of OTA development and so on.

(Note:This document is for the halley2 series development board which mainly includes halley2 ,halley2_mini, that storage devices includes SPI - nor and SPI - nand)

The software support diagram of halley2 series as follows:



2. The development and using of u-boot

The linux kernel needs u-boot to lead.u-boot is for embedded platform providing open source bootstrap ,it provides a serial port, a variety of ways to download of ethernet , NOR and NAND flash memory and management function of environment variables .it supports network protocol stack, JFFS2 /EXT2 /FAT file system, it also supports a variety of device drivers such as MMC/SD card, USB device, LCD driver,and so on .

The u-boot source directory is in “halley2 / platform/u-boot/”.Platform for all of the Settings are configured to complete can be used directly, below is a list of several main configurations which users according to their actual needs to change.

2.1. Configure the environment variables

After entering in the u-boot,you can configure the u-boot in “include/configs/halley2.h”file simply.

2.1.1. Modify the boot parameters

Major changes in the type of file system and partition location have rootfstype and root designated respectively, and the Linux kernel starts at address 0x80800000.

#define CONFIG_BOOTARGS BOOTARGS_COMMON "ip=off init=/linuxrc rootfstype=jffs2 root=/dev/mtdblock2 rw″

#define CONFIG_BOOTCOMMAND ″sfcnor read 0x40000 0x300000 0x80800000 ;bootm 0x80800000″

2.1.2. Modify the DDR frequency

#define CONFIG_SYS_MPLL_FREQ 600000000 /*If MPLL not use mast be set 0*/

#define CONFIG_SYS_MEM_FREQ 200000000

Note:when you modified DDR frequency through CONFIG_SYS_MEM_FREQ, the value modified must be dominant frequency MPLL CONFIG_SYS_MPLL_FREQ integer times, we suggests that frequency does not exceed 200 M.

2.1.3. Modify the IP address

#define CONFIG_SERVERIP 192.168.4.13

#define CONFIG_IPADDR 192.168.4.90

#define CONFIG_GATEWAYIP 192.168.4.1
#define CONFIG_NETMASK 255.255.255.0
#define CONFIG_ETHADDR 00:11:22:33:44:55

2.1.4. Build a network file system

To build a network file system mainly modify boot parameters of u-boot BOOTARGS _COMMON , specific modification can refer to the following settings (the “user” is the user name) :

#define CONFIG_BOOTARGS
BOOTARGS_COMMON "ip=192.168.4.254:192.168.4.1:192.168.4.1:255.255.255.0
rootdelay=2 nfsroot=192.168.4.13:/home/fpga/user/rootfs rw"

Note: nfsroot=192.168.4.13:/home/fpga/user/rootfs is the shared directory of network file system .

2.2. Compile u-boot

It’s required to compile the u-boot after modify the configurations.

1. Remove the last compile

```
$ make distclean
```

2.Choice board level of supporting configurations

On the boards.cfg file has making a list of all currently supported u-boot configurations, according to the development board is the spi - nor or spi - nand to compiled:

Compile spi-nor (Halley2_mini_v2.0(SPI-nor)board,Halley2_v2.0(SPI-nor)board)

```
$ make halley2_v10_uImage_sfc_nor
```

Compile spi-nand(Has not been released nand development board at present yet)

```
$ make halley2_v10_uImage_spi_nand
```

3. Linux kernel and drivers

The kernel source code in “halley2/platform/kernel” directory.

3.1. GPIO

3.1.1. Configuration file

1. The GPIO management basic idea

All GPIO pin feet, its function has been fixed in the chip welded to the circuit board that moment,, which used in equipment IO, which used as a real GPIO, is completely determined by the board level layer at the outset, drive need only operate real GPIO.

2. The interface function

By the file "arch/mips/xburst/soc-x1000/common/gpio.c".Under GPIOLIB based interface implementation is provided by GPIOLIB call interface, the interface definition header file for the < include/Linux/gpio.h>, providing interface are:

Interface function	Function
gpio_is_valid	apply for operation onily if GPIO is effective
gpio_request	apply for GPIO
gpio_free	free GPIO
gpio_direction_input	set GPIO as input
gpio_direction_output	set GPIO as output
gpio_get_value	read the value of GPIO，when GPIO is input or output
gpio_set_value	set the value of GPIO as output,GPIO configuration for the output as the premise
gpio_to_irq	get the interrupt number through GPIO

3.1.2. Method of use

1) Judge the gpio is valid, and then apply for gpio.

2) According to configurations of using select for input or output.(the default is input)

You can read the pin level or configure output level in use. (gpio get/set the value)

3) Used as the interrupt source through gpio to get irq interrupt number, to apply to register the interrupt ISR.

a) register interrupt had better specified trigger category .(the default for the falling edge)

b) the triggered methed of interrupt can be changed by set_irq_type in use.

4) at last (remove drver)free gpio.

Example:

drivers/input/keyboard/gpio_keys.c

3.2. I2C

1. Add equipment information

Add equipment information in file "arch/mips/xburst/soc-x1000/chip-x1000/halley2/common/i2c_bus.c".

2. Registered slave device

The function named "board_base_init"which in the file named "board_base.c"in the directory "arch/mips/xburst/soc-x1000/chip-x1000/halley2/common/"will be called "i2c_register_boa rd_info" equipment registered into the i2c slave device.start the kernel,When to start the kernel, check kernel's print and look at whether it can read id successful or not,registered if read successful, otherwise not.

3.3. Spi nand

Depending on the type of storage media on the development board, select the drive way.Here is mainly the use of SPI nand, including the selection of configuration items,a modification of the partition and

usbcloner update.

Spi nand file system type is ubi.

(Note:currently published development board no nand flash, nor only)

3.3.1.Divide the partition

In the SDK source code provided in the u-boot and the kernel has to the default configuration of SPI nand partitions, the partition is as follows:

divide the partition of nand :

partition	u-boot	kernel	rootfs（ubi）	**data（ubi）**
	1M	8M	40M	**remaining space**

I The overall compilation

Current release SDK has made the overal compiler support for spi nand, if you need to use spi nand overall compilation ,you should carry out according to the following.

In the file "halley2/platform/development/device/device.mk",you should choice "MAKE_SPI_NAND=y",do Overall compile in the directory "halley2/platform/", execute the following command to compile, it generate all of the image file including u-boot, kernel, file system in "halley2/platform/out/target/product/halley2/image/"directory to support for SPI nand at last.

```
$make
```

Note:For SPI Nand storage medium, temporarily does not support the OTA update.

II Part of the compilation

U-boot：halley2_v10_uImage_spi_nand

Compile:`$ make halley2_v10_uImage_spi_nand`

Kernel：halley2_nand_v10_linux_defconfig

Compile:`$ make halley2_nand_v10_linux_defconfig`

```
$ make uImage
```

3.3.2. Changing of partition

When operate the partition to partition, you need to modify u-boot and kernel of partition information at the same time, the firmware of usbcloner must be updated to burn u-boot, kernel and file system.

1.Modified the partition information of u-boot

Location:platform/u-boot/drivers/mtd/nand/jz_spinand.c

Modify the structure:struct jz_spinand_partition

2.Modified the partition information of kernel

Location:platform/kernel/arch/mips/xburst/soc-x1000/chip-x1000/halley2/common/spi_bus.c

Modify the structure:struct mtd_partition

3.update the usbcloner

If nand partition information has been changed, the usbcloner of firmware must be up dated in front of start burner .

1) Compile the usbcloner's firmware

Enter the u-boot

```
$ make burner_x1000_lpddr
```

2) Replace the usbcloner's firmware

Using the u-boot.bin generate from u-boot replace the file named u-boot.bin of "firm ware/x1000/"

3.4. audio

Audio is an audio module using with alsa realized the function of DMIC and AIC.

3.4.1. u-boot & kernel driver configuration

3.4.1.1. u-boot configuration

u-boot use default configurations.

3.4.1.2. kernel configuration

A) Board registration

audio equipment on board level is defined in the following files:arch/mips/xburst/soc-x 1000/common/platform.c

audio board level device name :jz-asoc-aic-dma, jz-asoc-aic, icdc-d3, jz-asoc-dmic-dma, jz-asoc-aic-dmic, jz-asoc-pcm-dma, jz-asoc-pcm, spdif dump, pcm dump, dmic dump

B) Driver code

sound/soc/ingenic/icodec/icdc_d3.c

sound/soc/ingenic/icodec/pcm_dump.c
sound/soc/ingenic/icodec/dmic_dump.c
sound/soc/ingenic/asoc-v13/
sound/soc/ingenic/asoc-v12/asoc-aic-v12.c
sound/soc/ingenic/asoc-board/phoenix_icdc.c

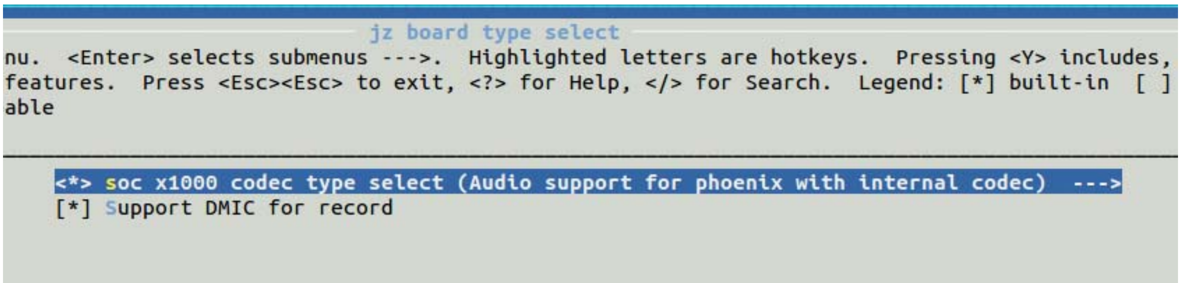
C) The following options inside of menuconfig must be enabled to make sure the audio driver can be used :

(1)AIC enabled

Device Drivers
-> Sound card support
-> Advanced Linux Sound Architecture
-> ALSA for SoC audio support
-> ASoC support for Ingenic
-> jz board type select(Audio support for phoenix with internal codec)

D) DMIC enabled

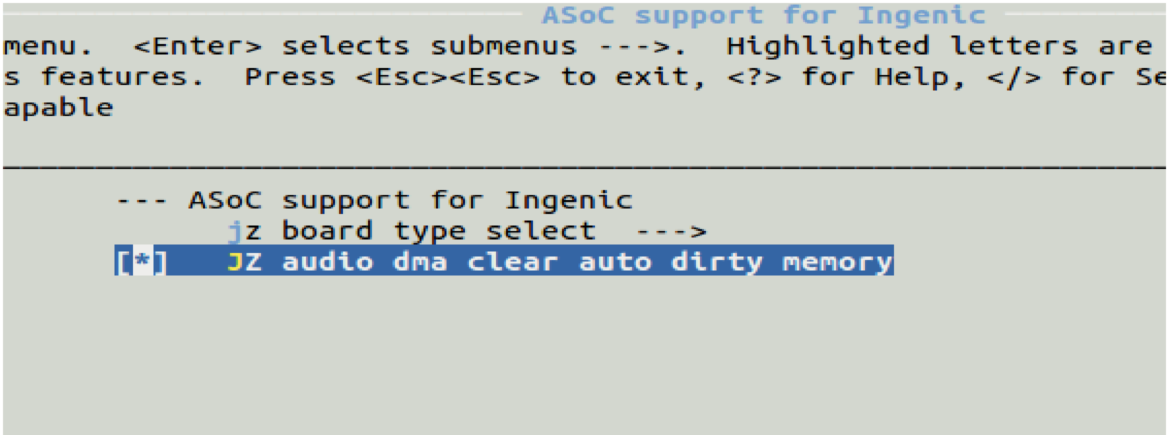
Device Drivers
-> Sound card support
-> Advanced Linux Sound Architecture
-> ALSA for SoC audio support
-> ASoC support for Ingenic
->Support DMIC for record



Note:We have two DMIC controller , one integrated inside the codec don't need a separate configuration, another one independent on the chip need separate configuration.Two controller cannot be used at the same time.

audio dma selection:

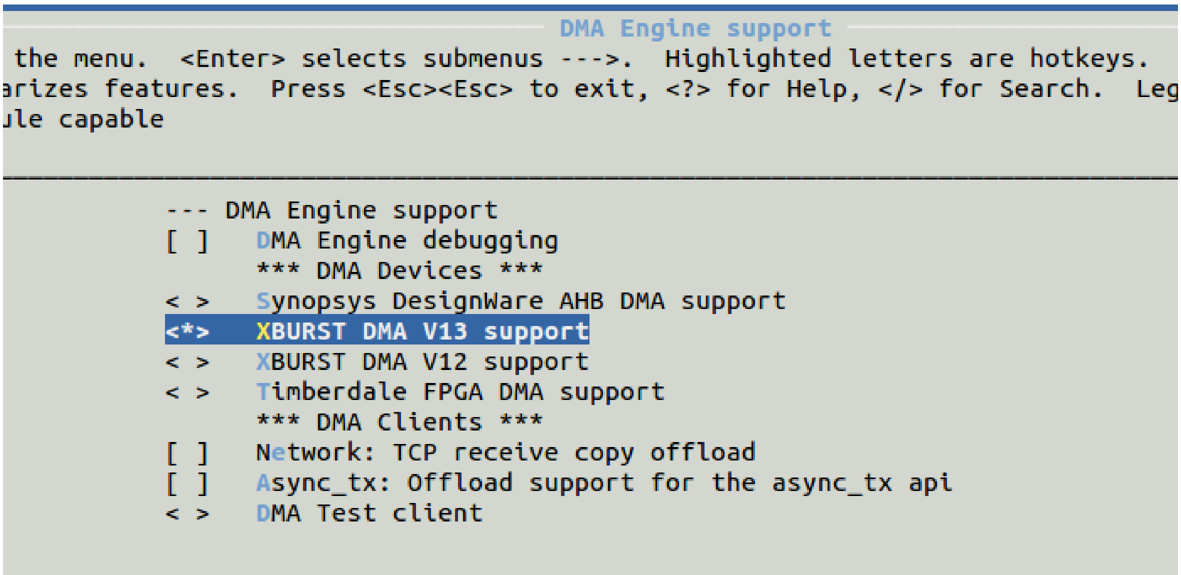
1. Device Drivers
->Sound card support
-> Advanced Linux Sound Architecture
->ALSA for SoC audio support
-> ASoC support for Ingenic
->JZ audio dma clear auto dirty memory



2. Device Drivers

->DMA Engine support

-> XBURST DMA V13 support



3.4.1.3. audio driver function verification method and process

The following (1) (2) (3) validation methods of the audio function are listed in the serial port to verify when you input command.

(1) the AIC functional verification

Setting before recording (choose to use AMIC) :

```
amixer cset numid=17,iface=MIXER,name='ADC Mux'  0
```

Simulation mike internal codec recording:

```
amixer cset numid=4,iface=MIXER,name='Digital Capture Volume' 20
amixer cset numid=6,iface=MIXER,name='Mic Volume' 3
```

```
arecord -D hw:0,0 -c 2 -f S16_LE -r 44100 -d 10 a.wav
```

Our simulation mic only supports single channel recording.Due to the different Mic has the different physical properties ,you need to adjust the Mic of two levels recording gain to the appropriate value in order to prevent from losing data..

Play the recording files according to the following commands:

```
aplay a.wav
```

Note:detailed in all test tools such as amixer arecord, aplay instructions at the end of this summary.

(2) the DMIC functional verification

DMIC recording:

```
arecord -D hw:0,2 -c 2 -f S16_LE -r 8000 -d 10 b.wav
```

Dmic support the sampling rate is 8000 and 16000.

(3) the reverberation

1. The ADC data overlay on the flow of data of DAC's channel (regard the playback path as a background data when recording) :

```
amixer cset numid=6,iface=MIXER,name='Mic Volume' 2
amixer cset numid=4,iface=MIXER,name='Digital Capture Volume' 30
amixer cset numid=17,iface=MIXER,name='ADC Mux' 0
amixer cset numid=12,iface=MIXER,name='ADC Mode Mux' 1
amixer cset numid=11,iface=MIXER,name='AIADC Mux' 2
amixer cset numid=10,iface=MIXER,name='ADC MIXER Mux' 2
amixer cset numid=9,iface=MIXER,name='mixer Enable' 1
```

2. The DAC data overlay on the flow of data of ADC's channel(regard the recording path as a background data when play)

```
amixer cset numid=6,iface=MIXER,name='Mic Volume' 4
amixer cset numid=16,iface=MIXER,name='DAC_MERCURY VMux' 0
amixer cset numid=15,iface=MIXER,name='MERCURY AIDAC MIXER Mux' 2
amixer cset numid=14,iface=MIXER,name='DAC Mode Mux' 1
amixer cset numid=13,iface=MIXER,name='MERCURY AIDAC Mux' 2
amixer cset numid=9,iface=MIXER,name='mixer Enable' 1
```

3. Paly and record:

```
arecord -D hw:0,0 -c 2 -f S16_LE -r 8000 -d 10 a.wav&
aplay a.wav &
```

NOTE:

(1)Using reverb functions need to have a certain understanding with alsa subsystem and internal codec pathways.

(2)When using above two kinds of functions, you had better to ensure that the playback and recording pathways have data can obtain the obvious effect.Namely aplay and arecord executed at the same time.the reverberation function must use the internal codec.

(3)When using dmic recording does not use hpf1, need to reduce the gain.

☆audio test tools introduction

1. amixer

amixer is a control abstract out of alsa in user mode that is used to configure the hardware, users according to their own needs to configure, the execution of the command is on a serial port.

◆Enter the following command to get the control list:

```
amixer controls
```

◆Enter the following command to get specific state of controls:

amixer cget controls

◆Enter the following command to set the corresponding controls as value:

amixer cset controls value

◆The volume of the codec mic

numid=6,iface=MIXER,name='Mic Volume'

◆High-pass filter

numid=7,iface=MIXER,name='ADC High Pass Filter Switch'

◆Selection mode of mixing

numid=10,iface=MIXER,name='ADC MIXER Mux'

◆Selection mode of recording

numid=12,iface=MIXER,name='ADC Mode Mux'

◆Select analog mic or digital mic

numid=17,iface=MIXER,name='ADC Mux'

◆Volume

numid=3,iface=MIXER,name='Playback Mixer Volume'

◆Mixing pattern

numid=11,iface=MIXER,name='AIADC Mux'

◆Choice the the way of playing

numid=14,iface=MIXER,name='DAC Mode Mux'

◆choice mixing of playing

numid=16,iface=MIXER,name='DAC_MERCURY VMux'

◆Temporary does not support

numid=16,iface=MIXER,name='DAC_TITANIUM VMux'

◆Digital mixing volume of

numid=5,iface=MIXER,name='Digital Capture Mixer Volume'

◆Digital recording volume

numid=4,iface=MIXER,name='Digital Capture Volume'

◆Digital broadcast mute

numid=8,iface=MIXER,name='Digital Playback mute'

◆mixing way of Playing

numid=15,iface=MIXER,name='MERCURY AIDAC MIXER Mux'

numid=13,iface=MIXER,name='MERCURY AIDAC Mux'

◆The volume of playing

numid=1,iface=MIXER,name='MERCURY Playback Volume'

◆no need to set up

numid=2,iface=MIXER,name='TITANIUM Playback Volume'

◆enable mixing

numid=9,iface=MIXER,name='mixer Enable'

2 arecord recording

-D This parameter is used to specify the PCM of audio equipment

hw The first parameter to specify the sound card number, the second parameter is used to specify the device number

-c is used to specify the track number

-f is used to specify the data format

-r is used to to specify the sampling frequency

-d is used to specify the recording time

--help get help

3.aplay

Parameter is set in line with arecord.

3.5. TF Card

3.5.1. Configuration of board level

1.Configuration file of Board level

arch/mips/xburst/soc-x1000/chip-x1000/halley2/common/mmc.c

among them:

“struct jzmmc_platform_data tf_pdata “is to describe the sd card equipment

“struct jzmmc_platform_data sdio_pdata”is to describe the stdio class equipment,generally refers to sdio wifi.

2.Circuit connection

sd card received MSC0, sdio-wifi received MSC1

3.5.2. TF Card driver

1.Driver file：

drivers/mmc/host/jzmmc_v12.c

2.MMC driver compilation options configuration is as follows：

Device Drivers

->MMC/SD/SDIO card support

->JZMMC_V12 MMC0

```
    [*] Enable the block layer  --->
        Bus options (PCI, PCMCIA, EISA, ISA, TC)  --->
        Executable file formats  --->
        Power management options  --->
        CPU Frequency scaling  --->
    [*] Networking support  --->
        Device Drivers  --->
        File systems  --->
        Kernel hacking  --->
        Security options  --->
    -*- Cryptographic API  --->
    [ ] Virtualization  --->
        Library routines  --->
    ---
        Load an Alternate Configuration File
```

```
        Graphics support  --->
    <*> Sound card support  --->
    [ ] HID Devices  --->
    [*] USB support  --->
    <*> MMC/SD/SDIO card support  --->
    < > Sony MemoryStick card support (EXPERIMENTAL)  --->
    [ ] LED Support  --->
    [ ] Near Field Communication (NFC) devices  --->
    v(+)

              <Select>    < Exit >    < Help >
```

```
<*>   Ingenic(XBurst)  MMC/SD Card Controller(MSC) v1.2 support
        JZMMC_V12 data transfer method select (JZMMC_V12 data transfer using MSC SDMA)  --->
[*]     JZMMC_V12 MMC0
          JZMMC_V12 GPIO function pins select (GPIO A, data with 4 bit)  --->
[ ]       JZMMC_V12 MSC0 function pin pull up enable
(24000000) msc0 max frequency
[ ]       Use PIO mode for MSC0
```

3.5.3. SD card mounted

The default file system to support the function of automatic mount SD card, SD will be mounted at the file system/MNT/SD directory, check the type the following command to see whether to automatically mount:

```
$mount
```

```
rootfs on / type rootfs (rw)
/dev/root on / type jffs2 (rw,relatime)
proc on /proc type proc (rw,relatime)
sysfs on /sys type sysfs (rw,relatime)
tmpfs on /dev type tmpfs (rw,relatime)
tmpfs on /tmp type tmpfs (rw,relatime)
devpts on /dev/pts type devpts (rw,relatime,mode=600,ptmxmode=000)
tmpfs on /dev/shm type tmpfs (rw,relatime)
/dev/mmcblk0p1 on /mnt/sd type vfat (rw,relatime,fmask=0000,dmask=0000,allow_utime=0022,codepage=cp437,iocharset=iso8859-1,shortname=mixed,e)
```

In the last row"/dev/mmcblk0p1 on/MNT/sd type vfat"means mount successfully.
Note:automatically mount script will default to TF card mounted on the first partition, if automatically mount is not successful, please check the TF card partition information, manual mounted partition.

Currently published halley2 development board does not support hot plug SD card, need to manually operate the following steps:
1.after insert SD card, restart development board.
2.carry out ls dev/mmcblk0 to check if any SD card equipment, if it has device node continued to perform the following steps, if not, repeat the first step.
3.perform the mount/dev/mmcblk0p * / MNT, hanging in your SD/MNT, pay attention to / dev/mmcblk0p * for your SD card partitions, it's depends on the individual development b oard.

3.6. USB

A）Board level registration file

Platform device registration file：arch/mips/xburst/soc-x1000/chip-x1000/halley2/common/ board_base.c

In the file "arch/mips/xburst/soc-x1000/chip-x1000/halley2/common/misc.c" application of userusb dete pin, ID pin ， bus pin

B）Driver path

The controller driver path:drivers/usb/dwc2/

C) USB Driver configuration

USB consists of both the host and device capabilities,above two functions need to be used in the menuconfig configuration, the following is otg as an example:

OTG configuration method:

Device Drivers
->USB support (USB_SUPPORT [=y])
->Support for Host-side USB (USB [=y])
->USB routime power management (autosuspend) and wakeup (USB_SUSPEND [=y])

If you need regard otg at the same time as the device and the host,you need to choose the following options:

```
< >    SL811HS HCD support
< >    R8A66597 HCD support
< >    Host Wire Adapter (HWA) driver (EXPERIMENTAL)
< >    Inventra Highspeed Dual Role Controller (TI, ADI, ...)
<*>    DesignWare Core USB 2.0 Hi-Speed On-The-Go(OTG)
         Driver Mode (Both Host and Device)  --->
[ ]      Allow use dwc2 drvvbus function pin
[*]      Allow wakeup when usb cable plug/unplug
[ ]      Board has no plug detect facility
```

3.6.1. Host

3.6.1.1. Host function configuration method

for example U disk:

(1)Device Drivers

->USB support

->USB Mass storage support

```
< >    USB Printer support
< >    USB Wireless Device Management support
< >    USB Test and Measurement Class support
       *** NOTE: USB_STORAGE depends on SCSI but BLK_DEV_SD may
       *** also be needed; see USB_STORAGE Help for more info *
<*>    USB Mass Storage support
[ ]      USB Mass Storage verbose debug
< >      Realtek Card Reader support
```

(2)Device Drivers

->SCSI device

->SCSI disk support

Because the USB Host function need the support of the upper drive, so you need to choose the SCSI drive support.

Device Drivers

->SCSI device support

3.6.1.2. Functional verification of USB

The file system default to support the USB hotplug, inserted U disk, U disk will be the mount to the default file system /mnt/sda directory,the serial port after printing as follows when you insert USB device :

```
[    22.092290] jz-dwc2 jz-dwc2: set vbus on(on) for host mode
[    22.202174] USB connect
[    22.902209] usb 1-1: new high speed USB device number 2 using dwc2
```

```
[     23.328262] usb 1-1: New USB device found, idVendor=0930, idProduct=6545
[     23.342090] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[     23.356745] usb 1-1: Product: DataTraveler 2.0
[     23.365858] usb 1-1: Manufacturer: Kingston
[     23.374570] usb 1-1: SerialNumber: C86000BDBA09EF60CA285106
[     23.412410] scsi0 : usb-storage 1-1:1.0
[     24.475824] scsi 0:0:0:0: Direct-Access     Kingston DataTraveler 2.0 PMAP PQ: 0 ANSI: 4
[     25.751099] sd 0:0:0:0: [sda] 30497664 512-byte logical blocks: (15.6 GB/14.5 GiB)
[     25.768695] sd 0:0:0:0: [sda] Write Protect is off
[     25.779135] sd 0:0:0:0: [sda] No Caching mode page present
[     25.790501] sd 0:0:0:0: [sda] Assuming drive cache: write through
[     25.807171] sd 0:0:0:0: [sda] No Caching mode page present
[     25.832513] sd 0:0:0:0: [sda] Assuming drive cache: write throug
[     25.879083] sda:sdal
[     25.895075] sd 0:0:0:0: [sda] No Caching mode page present
[     25.932372] sd 0:0:0:0: [sda] Assuming drive cache: write through
[     25.964595] sd 0:0:0:0: [sda] Attached SCSI removable disk
```

On a serial port, enter the following command to see whether to automatically mount:

```
$mount
```

```
# mount
rootfs on / type rootfs (rw)
/dev/root on / type jffs2 (rw,relatime)
proc on /proc type proc (rw,relatime)
sysfs on /sys type sysfs (rw,relatime)
tmpfs on /dev type tmpfs (rw,relatime)
tmpfs on /tmp type tmpfs (rw,relatime)
devpts on /dev/pts type devpts (rw,relatime,mode=600,ptmxmode=000)
tmpfs on /dev/shm type tmpfs (rw,relatime)
/dev/sda1 on /mnt/sda type vfat (rw,relatime,fmask=0000,dmask=0000,allow_utime=0022,codepage=cp437,iocharset=iso8859-1,s
#
#
```

In the last line "/ dev/sda1 on/mnt/sda type vfat" represents has automatically mount.

Note:

Currently published halley2 development board temporarily does not support automatic mounted file system,it need to perform the steps to manually mount:

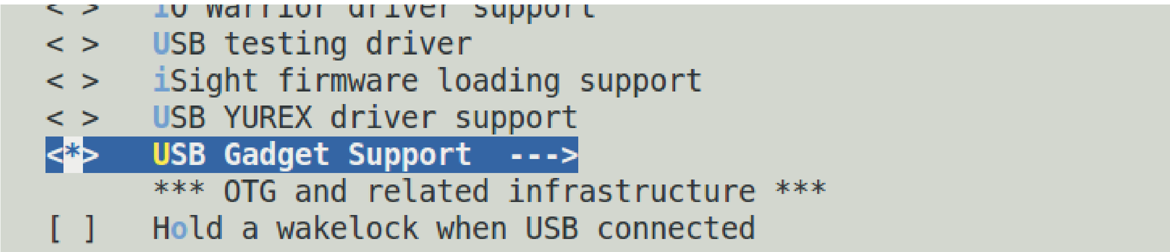
1.Carry out the ls dev/sda to check whether there is a device file, if any, you need restart insert.

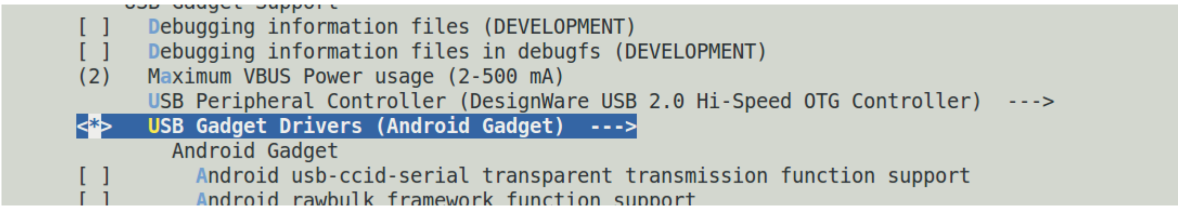
2.Carry out mount dev/sda1 /mnt mounted at /mnt dictionary.

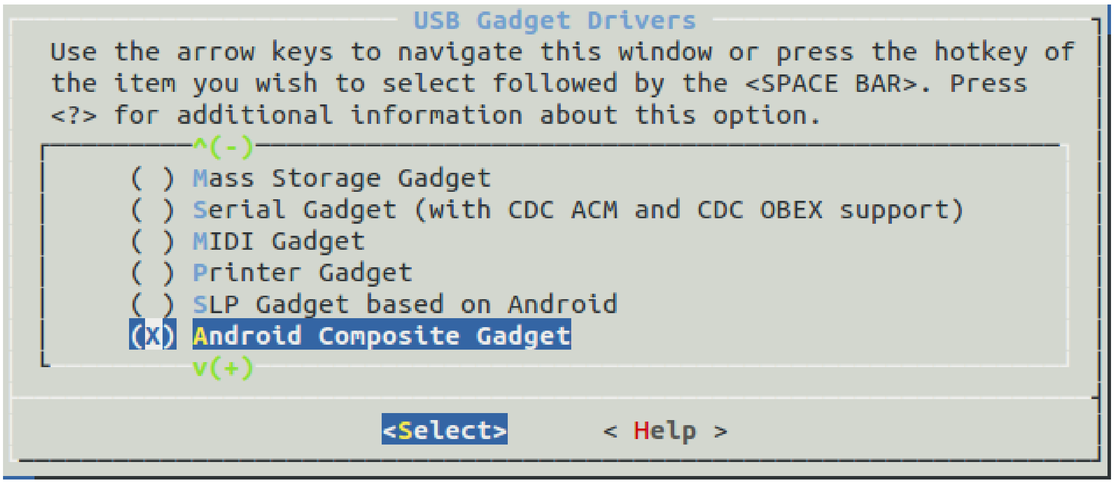
3.6.2. USB device

3.6.2.1. USB Device function configuration method

Device Drivers
  ->USB support
    ->USB Gadget support
      ->USB Gadget Drivers (Android Gadget)
        ->(X) Android Composite Gadget







The gadget function using android gadgets, android gadgets default support mass storge, adb function specific operation method with the adb as example.

3.6.2.2. USB device validation method

In a serial port input:

```
$cd /sys/class/android_usb/android0
 $echo 0 > enable
 $echo 18d1 > idVendor
 $echo d002 > idProduct
 $echo mass_storage,adb > functions
 $echo 1 > enable
```

In the pc input :

```
$adb devices
```

3.6.2.3. USB device functional verification analysis

perform the following command in pc to check usb device:

$ adb devices

```
user@user-desktop:~$ adb devices
List of devices attached
Ingenic device

user@user-desktop:~$
```

there is “ingenic device” in pc means USB device function is normal.

3.7.LCD

3.7.1. Board level registration

The file of Board level of LCD is :

arch/mips/xburst/soc-x1000/chip-x1000/halley2/common/lcd/lcd-truly_tft240240_2_e.c

3.7.2. Drvers file descriptor

Drvers file descriptor	Path
Driver code	halley2/platform/kernel/drivers/video/jz_fb_v13/jz_fb.c
Power management code	halley2/platform/kernel/drivers/video/backlight/truly_tft240240_2_e.c
Backlit code	halley2/platform/kernel/drivers/video/backlight/pwm_bl.c

3.7.3. Drive configuration method

menuconfig configuration, the choice of the following options must be selected so as to LCD can be normal used:

1.Device Drivers

-->Graphics support

-> Support for frame buffer devices

```
    [*] Networking support  --->
        Device Drivers  --->
        Firmware Drivers  --->
        File systems  --->
        Kernel hacking  --->
```

```
[*] Voltage and Current Regulator Support  --->
<*> Multimedia support  --->
    Graphics support  --->
<*> Sound card support  --->
    HID support  --->
```

```
< > Lowlevel video output switch controls
<*> Support for frame buffer devices  --->
[ ] Exynos Video driver support  --->
[*] Backlight & LCD device support  --->
```

2.Device Drivers

-->Graphics support

--><*>Backlight & LCD device support

--> Lowlevel LCD controls

SLCD TRULY TFT240240-2-E with control IC st7789s (240x240)

SLCDC USE TE SIGNAL

SLCDC CONTINUA TRANFER

Lowlevel Backlight controls

Generic PWM based Backlight Driver V13

```
<*> Support for frame buffer devices  --->
[ ] Exynos Video driver support  --->
[*] Backlight & LCD device support  --->
    Console display driver support  --->
[ ] Bootup logo  --->
<*> JZ LCDC framebuffer V1.3  --->
```

```
--- Backlight & LCD device support
<*>   Lowlevel LCD controls
< >     Platform LCD controls
[ ]     LCD panel reset enable
< >     AT070TN93 LCD Driver
< >     AUO_A043FL01V2 LCD Driver
< >     KD50G2_40NM_A2 panel(800x480)
< >     BYD BM8766U panel(800x480)
```

```
< >      KFM701A21_1A TFT Smart LCD panel(400x240)
<*>      SLCD TRULY TFT240240-2-E with control IC st7789s (240x240)
<*>      SLCDC USE TE SIGNAL
< >      USE GPIO SIMULATE TO MCU INIT
<*>      SLCDC CONTINUA TRANFER
[ ]      KD301_M03545_0317A panel(320x480)
[ ]      TM035PDH03 panel(320x480)
<*>    Lowlevel Backlight controls
< >      Generic (aka Sharp Corgi) Backlight Driver
< >      Generic PWM based Backlight Driver
<*>      Generic PWM based Backlight Driver V13
```

3.Device Drivers

-->Graphics support

--><*>JZ LCDC framebuffer V1.3

-->set lcd gpio (lcd v13 8bit slcd)

-->(X) lcd v13 8bit slcd

```
[ ] Bootup logo  --->
<*> JZ LCDC framebuffer V1.3  --->
< > JZ virtual framebuffer(just for test)  --->
(9) Vsync skip ratio[0..9]
    allocation frame buffer (alloc 3 frame buffers)  --->
```

```
                          set lcd gpio
  Use the arrow keys to navigate this window or press the hotkey of
  the item you wish to select followed by the <SPACE BAR>. Press
  <?> for additional information about this option.

                  (X) lcd v13  8bit slcd
                  ( ) lcd v13 9bit slcd
                  ( ) lcd v13 16bit slcd

                  <Select>      < Help >
```

3.8. Camera

3.8.1. Board level configuration file

Board level configuration file path is as follows:

"arch/mips/xburst/soc-x1000/chip-x1000/halley2/common/cim_board.c"

3.8.2. Drivers file descriptor

File descriptor	Board level name	Path
Controller code	jz-cim	halley2/platform/kernel/drivers/media/video/jz_camera_v13.c
Camera code	ov5640-front	halley2/platform/kernel/drivers/media/video/ov5640.c

3.8.3. The camera driver configuration method

3.8.3.1. Cim controller configuration

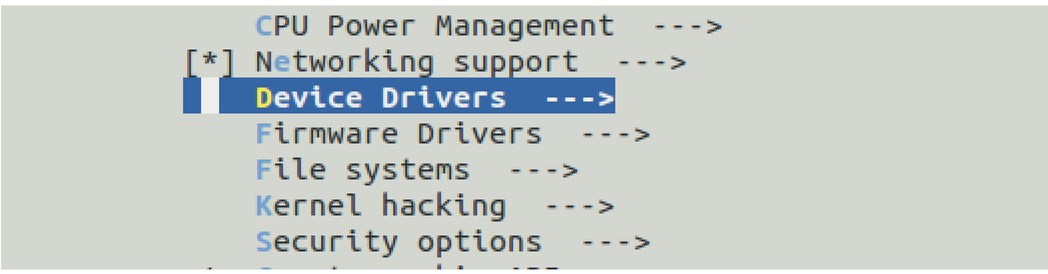
1.Device Drivers

->Multimedia support

->V4L platform devices

->Soc camera support

ingenic cim driver used on camera x1000



```
        Broadcom specific AMBA  --->
        Multifunction device drivers  --->
    [*] Voltage and Current Regulator Support  --->
    <*> Multimedia support  --->
        Graphics support  --->
    <*> Sound card support  --->
        HID support  --->
```

```
        *** Media drivers ***
    [*]   Media USB Adapters  --->
    [*]   V4L platform devices  --->
    [ ]   Memory-to-memory multimedia devices  --->
    [ ]   Media test drivers  --->
          *** Supported MMC/SDIO adapters ***
```

```
    <*>   SoC camera support
    <*>     platform camera support
    < >     ingenic cim driver used on jz4775/jz4780
    <*>     ingenic cim driver used on camera x1000
    < >   SuperH Mobile MIPI CSI-2 Interface driver
    < >   SuperH Mobile CEU Interface driver
```

2. Device Drivers

->Multimedia support

->Sensors used on soc_camera driver

->ov5640 camera support

```
< >   Cypress firmware helper routines
      *** Media ancillary drivers (tuners, sensors, i2c, frontends) ***
[*]   Autoselect ancillary drivers (tuners, sensors, i2c, frontends)
      Sensors used on soc_camera driver  --->
```

```
    < > ov5642 camera support
    <*> ov5640 camera support
    < > ov6650 sensor support
```

3.8.3.2. VPU configuration

1. Machine selection

->Soc type
->jz imem

```
    Machine selection  --->
    Endianness selection (Little endian)  --->
    CPU selection  --->
    Kernel type  --->
```

```
    System type (Ingenic Xburst based machines)  --->
<*> SOC type  --->
```

```
[ ]   early run init process
[ ]   fast test reset_dll
[*]   jz imem
[ ]   jz ota updata
```

2. Device Drivers
->Graphics support
->JZ VPU driver

```
    CPU Power Management  --->
[*] Networking support  --->
    Device Drivers  --->
    Firmware Drivers  --->
    File systems  --->
    Kernel hacking  --->
    Security options  --->
```

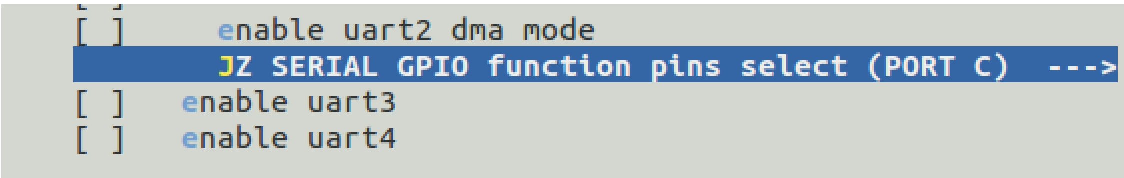
```
[*] Voltage and Current Regulator Support  --->
<*> Multimedia support  --->
    Graphics support  --->
<*> Sound card support  --->
```

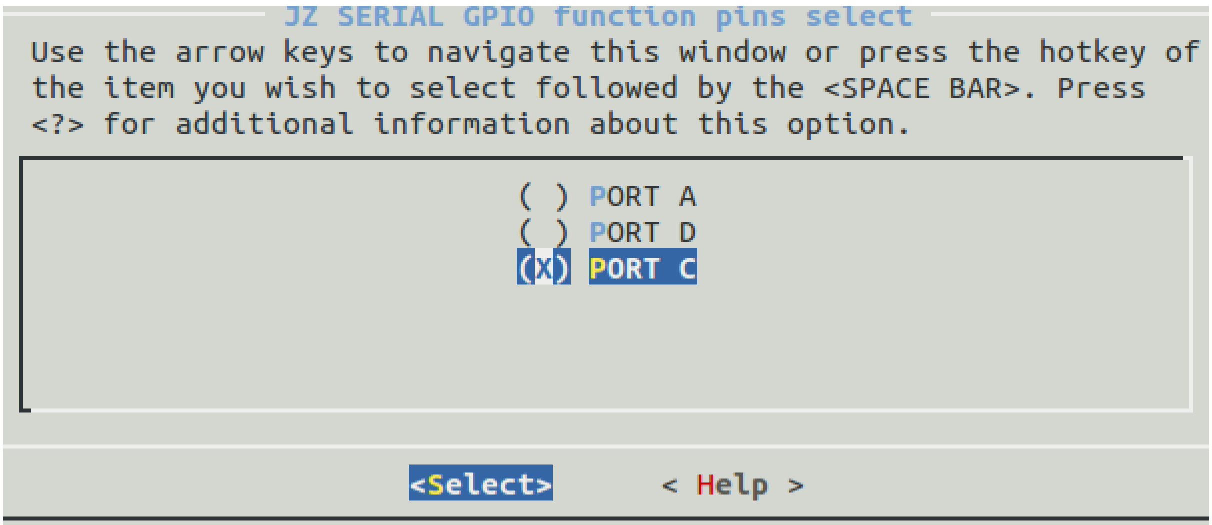
```
< > JZ virtual framebuffer(just for test)  --->
(9) Vsync skip ratio[0..9]
    allocation frame buffer (alloc 3 frame buffers)  --->
[*] JZ VPU driver  --->
```

3.8.3.3. Serial port configuration

1. Device Drivers
->Character devices

->serial drivers
->JZ SERIAL GPIO function pins select (PORT C)
->PORT C





3.8.4. Method of using

The method of Camera using is detailed in chapter 6 test case 6.1.
Note:Currently published halley2 default configurations does not support the camera, but the driver support it, if you need to use, you can manually configure 3.8.3.1, 3.8.3.2 two sections.

3.9. USB Camera

3.9.1. driver configuration method of USB Camera

Device Drivers
-> Multimedia support
->Media USB Adapters
->USB Video Class (UVC)

```
    CPU Power Management  --->
[*] Networking support  --->
    Device Drivers  --->
    Firmware Drivers  --->
    File systems  --->
    Kernel hacking  --->
    Security options  --->
```

```
    Multifunction device drivers  --->
[*] Voltage and Current Regulator Support  --->
<*> Multimedia support  --->
    Graphics support  --->
<*> Sound card support  --->
    HID support  --->
```

```
< >   V4L2 int device (DEPRECATED)
      *** Media drivers ***
[*]   Media USB Adapters  --->
[*]   V4L platform devices  --->
[ ]   Memory-to-memory multimedia devices  --->
```

```
--- Media USB Adapters
      *** Webcam devices ***
<*>   USB Video Class (UVC)
[ ]     UVC input events device support
```

3.9.2. USB camera validation and using method

The using method USB Camera detailed in chapter 6 test case 6.2.

Note:all the default configuration of halley2 released currently also include Halley2_mini_v2. 0 (SPI-nor) development board does not support usb camera, but the driver support it, if you need the function with Halley2_v2. 0 (SPI-nor), you can perform manually 3.9.1 configuration.

3.10. Sleep and wake up

System if idle for a long time, you can let it into sleep mode.In this mode,most of the modules of the system are placed in a low power mode, DRAM in the refresh mode and save the program in running scene, only keep RTC clock working to wake up the system.

3.10.1. Sleep and wake up configuration, using method

Power management options
->Suspend to RAM and standby
Run-time PM core functionality
Log time spent in suspend

3.10.2. Sleep and wake up validation method

To ensure that the above listed 3.10.1 configuration was right selected after the, recompile the kernel, burn, start, execute the following commands on the serial port last to make the system into sleep mode.(you can't input in serial when it in sleep mode)

```
$echo mem >/sys/power/state
```

Press the power button to wake up.

3.11. Voice trigger

3.11.1. Voice trigger introduction

Voice trigger (Voice dormancy awakening) mainly using the Linux system and characteristic of ingenic processor support sleep and wake up,it aims to make your plan to achieve better effect of energy saving .

Voice trigger's code can be divided into two parts: the first part of the code for the speech recognition test,the second part code for voice wake up.

file description:

the firmware code of Voice dormancy and awakened:drivers/char/voice_wakeup_v13/
voice recognition test driver code:drivers/char/jz_wakeup_v13.c
DMIC drives available to other applications:drivers/char/jz_dmic_v13.c
Test case code:tools/wakeup-test/wakup.c
The file for voice comparison needed to test cases:tools/wakeup-test/ivModel_v21.irf
The entry file of system after dormancy:arch/mips/xburst/soc-x1000/common/pm_p0.c

3.11.2. Voice trigger driver configuration method

1.in the directory"drivers/char/voice_wakeup_v13/wakeup_module/",you can carry out make clean ,then perform ./mkmodule.sh.
2.Device drivers

--> Character devices

--> Ingenic Dmic Driveri v13

Ingenic Voice Wakeup Driver V13

```
    [*] Networking support  --->
        Device Drivers  --->
        Firmware Drivers  --->
        File systems  --->
        Kernel hacking  --->
```

```
[*] Network device support  --->
    Input device support  --->
    Character devices  --->
<*> I2C support  --->
[ ] SPI support  --->
    Qualcomm MSM SSBI bus support  --->
```

```
[ ] Ingenic Dmic Driver
[*] Ingenic Dmic Driveri v13
[*]   Ingenic Voice Wakeup Driver V13
[ ] Ingenic Test second refresh for watch(test driver.)
```

after configuration the options above,you can execute the following command to compile:

$ make uImage

after the completion it will appear the uImage for voice trigger in the directory "platform/kernel/arch/mips/boot/".

3.11.3. Validation method

Copy the the file wakeup and ivModel_v21.irf in the directory "halley2/platform/kernel/tools/wakeup-test/"to the system.

Enter the following command on the serial port (background):

$./wakeup ivModel_v21.irf &

```
# ./wakeup ivModel_v21.irf &
# open file[ivModel_v21.irf], ok!
open file[/dev/jz-wakeup] ok![ 2019.532480] enable wakeup function

open file[/sys/class/jz-wakeup/jz-wakeup/wakeup] ok!
read don[ 2019.546428] module open open_cnt = 1
e!!!!
#### begin read !!!!!
```

When it appears the information as above ,you can audio input on the audio module "ling xi ling xi", the serial port in print as follows:

```
WKUPOK[ 2026.691348] sys wakeup ok!--wakeup_timer_handler():158, wakeup_pending:1
[ 2026.705626] [Voice Wakeup] wakeup by voice.
################ret:9, wakeup_ok
sh: input: not found
#########read ok!, wakeup ok!
#### begin read !!!!!
```

Make the development board entering a sleep state (for specific operation reference 3. 10.2) , then through audio "ling xi ling xi" to wake up the system.

3.12. AES-RSA

3.12.1. The driver name and path

The driver name and path of AES-RSA is “jz_security”
(1)Device node:/dev/jz_security
Using the following ways to manipulate the driver:
open(/dev/jz-security, ...) → ioctrl(xxx...) → ioctl(xxx...) → ...→ ioctl(xxx) -->close(device)
(2)driver file and path:
“platform/kernel/drivers/misc/jz_security/”

3.12.2. Driver configuration

Device Drivers
 -->Misc devices
 -->JZ SECURITY Driver(AES && RSA)

```
        Power management options  --->
        CPU Power Management  --->
    [*] Networking support  --->
        Device Drivers  --->
        Firmware Drivers  --->
        File systems  --->
        Kernel hacking  --->
        Security options  --->
```

```
< > Parallel port support  --->
[*] Block devices  --->
    Misc devices  --->
< > ATA/ATAPI/MFM/RLL support (DEPRECATED)  --->
    SCSI device support  --->
< > Serial ATA and Parallel ATA drivers  --->
```

```
[ ] JZ V13 EFUSE Driver
[ ] JZ V11_IRDA Driver
[*] JZ SECURITY Driver(AES  && RSA)
[ ]   DEBUG of JZ_SECURITY DRIVER (NEW)
[ ] ingenic efuse support  --->
```

3.12.3. IOCTL command definition

#define SECURITY_INTERNAL_CHANGE_KEY (0xffff0010)
/*set up AES-KEY*/
#define SECURITY_INTERNAL_AES (0xffff0020)
/*AES encrypt or decrypt*/
#define SECURITY_RSA (0xffff0030)
/*RSA encrypt or decrypt*/

3.12.4. Driver struct description

The following driver structure defines path:/halley2/platform/kernel/tools/security-test
(1)Define the AES - KEY structure is as follows:

struct rsa_aes_packet {
unsigned short oklen; //old AES-KEY len (unit:word)
unsigned short nklen; // new AES-KEYlen (unit:word)
unsigned int * okey; // old AES-KEYaddr
unsigned int * nkey;// new AES-KEYaddr
};

You can also use an array to store the AES-KEY, specific definition is as follows:

unsigned int user_key[9]=
{
    0x00040004,
    /*bit31~16: old key length ,bit15~bit0:new key length*/
    0x1,0x2,0x3,0x4, /*old AES-KEY*/
    0x4, 0x5,0x6,0x7, /*new AES-KEY*/
}

Note:Initialize the AES-KEY, "old AES-KEY" "new AES-KEY" should be set to the same initial value.

(2) the RSA parameter structure

User use RSA encrypt and decrypt AES-KEY ",the structure "struct rsa_param" is to describe RSA , the concrete structure is as follows:

struct rsa_param {
unsigned int in_len; //input data length (units:word)

```
unsigned int key_len;     //private or public key length(units:word)
unsigned int n_len;             //n length(units:word)
unsigned int out_len;      //ouput length(units:word)
unsigned int *input;             //input data buffer
unsigned int *key;        /*Ku or KR*/ buffer
unsigned int *n;          /*N*/buffer
unsigned int *output;      //encrypted data or decrypted data buffer
unsigned int mode;        //mode: 1,encrypt 0,decrypt
};
```

(3) the CHANGE-KEY structure

```
struct change_key_param {
int len;                            //rsa_enc_data len in bytes.
int *rsa_enc_data;                  //okey_len,nkey_len,okey,nkey
int *n_ku_kr;                       //NKU or NKR buffer, for rsa decrypt (62 words)
int init_mode;                      //init key 1;init key, 0:change key
};
```

Rsa_enc_data is the data encrypted after the structure "struct rsa_aes_packet" with rac of , the NKU or NKR of encrypted data.

(4) AES encryption and decryption parameters

At present we only support the ECB mode, AES-KEY size of 128 bit support four word of AES encryption or decrypt at a time.

```
struct aes_param {
unsigned int in_len;                  //input data length (units:word)
unsigned int key_len;               //private or public key length(units:word)
unsigned int n_len;                       //n length(units:word)
unsigned int out_len;               //ouput length(units:word)
unsigned int *input;                      //input data buffer
unsigned int *key;                  /*Ku or KR*/ buffer
unsigned int *n;                    /*N*/buffer
unsigned int *output;               //encrypted data or decrypted data buffer
unsigned int mode;                  //mode: 1,encrypt 0,decrypt
};
```

4 Programming guide

1)Open the device and initialize the AES controller.
2)Use the correct format of AES-KEY and encrypted RSA.
3)Using the results in step 1, set the AES-KEY to the CPU.
4)Perform the AES encryption or decryption.

If you want to change the AES-KEY, please carry out the step 2, step 3, we should pay attention to "1" in step 2 as the initial KEY value, "0"as the KEY value which was changed.

3.12.5. USER API

(1)rsa encryption and decryption
int do_rsa(unsigned int fd, //file node
unsigned int orig_aes_len, // original AES_KEY length
unsigned int * rsa_key, //KU or KR
unsigned int * n,
unsigned int * input, // input_data
unsigned int *output, //encrypted or decrypted data
unsigned int mode) ; //mode: 0:encryption 1:decryption
(2)set AES-KEY or change the AES-KEY
int setup_aes_key(int fd, //file node
unsigned int *key, //encrypted AES-KEY by using RSA
int len, // length of key
unsigned int *nku_kr, //NKU ok NKR(62 words)
int init_mode) ; //init_mode: 1:init code , 0:change code
(3)AES encryption and decryption
nt do_aes(int fd, //file node
unsigned int *input, //input data
unsigned int in_len, //input and output data length(unit:word)
unsigned int *output, //output data(encrypted data or decrypted data)
unsigned int mode); //mode :0,encryption 1:decryption

4. The root file system of linux

The platform default to use jiffs2 file system (storage medium for spi nor flash) see details in chapter 4.4.1.

In conditions that the file "platform/development/device/device.mk " select the "MAKE _SPI_NAND" variable , using the ubi file system (spi nand flash storage medium) detailed in this chapter 4.1.2.

4.1. Make the file system

4.1.1. Jiffs2 file system

1. Select the type of file system

In the platform default to use jiffs2 file system,So there is no need to do any choice, you can compile.

2. make the file system

enter the directory of platform,you can carry out the command as follows:

```
$ make
```

In the directory out/target/product/halley2/image/ to generate the image file system.Jiffs 2, the default size is 12.5 M.

4.1.2. Ubi file system

1.Select the type of file system

In the directory of platform, enter the directory development/device/ edit the file na med device. mk select MAKE_SPI_NAND variables.

e.g. MAKE_SPI_NAND=y

2.make the file system

Enter the platform directory, execute the following commands:

```
$ make
```

When you carry out make,it will generate the image file named system.ubi using the system-ubi.tar in the directory development/rootfs/ubi.

Note:because the default size of nand is 128M, the partition of rootfs maximum is 11 5M, so the size of the image file system must not exceed 115M.

the released version of Halley2 temporary does not support ubi file system.

4.2. File system extended

According to 4.1 the attached file system can be made with platform, if you need to add a new application or function, you can extend on the basis of the attached file system.

In the directory “out/target/product/halley2/system”(system.tar extract here),you can add the extension application file in corresponding directory file.

After adding the extension file, execute the following command in the platform directory to recompile.

```
$ make
```

Note:The extended file system should be packaged for use for a long time to replace the system.tar in the directory of “development/rootfs” or backup in time.Otherwise make distclean will delete the directory out/target/product/ of all the things.

Packaging to replace the file system tar package:

i:Enter the directory out/target/product/halley2/system/

$ cd out/target/product/halley2/system/

ii:packaging

$ tar cvf ../system.tar ./*

iii:relpace

If you need to replace, you can use your new system.tar to replace the old system.tar in rootfs directory.

4.2.1. Jiffs2 file system

If the nor flash was replaced, the storage space changed,you can change the default size of the file system image though modify to the ROOTFS_JFFS2_SIZE variable in the file development/device/device.mk, at the same time erasing the size change may modify by ROOTFS_JFFS2_NORFLASH_ERASESIZE variables here.

ROOTFS_JFFS2_NORFLASH_ERASESIZE:= 0x8000

ROOTFS_JFFS2_SIZE:= 0xc80000 #The file system image size represent in hexadecimal

Note:At the time of extending file system, the size of the file system may increase.Jffs2 c ompression ratio of 2:1 in theory, so the size of the file system does not exceed 25M.

4.2.2. Ubi file system

When the development board built-in nand can't satisfy your needs, replacement of the size of the nand flash. according to the nand data book which be replaced, determine the page size, and the size of the logical block erasure, and the maximum number of logic block erasure,you can modify the file in the development/device/device.mk, specific modification parameters is as follows:

```
#nand flash config
ROOTFS_UBIFS_LEBSIZE     := 0x1f000
ROOTFS_UBIFS_MAXLEBCNT := 2048
ROOTFS_UBIFS_MINIOSIZE := 0x800
```

Note:Halley2 development board released without a nand flash, only for nor flash.

5. OTA

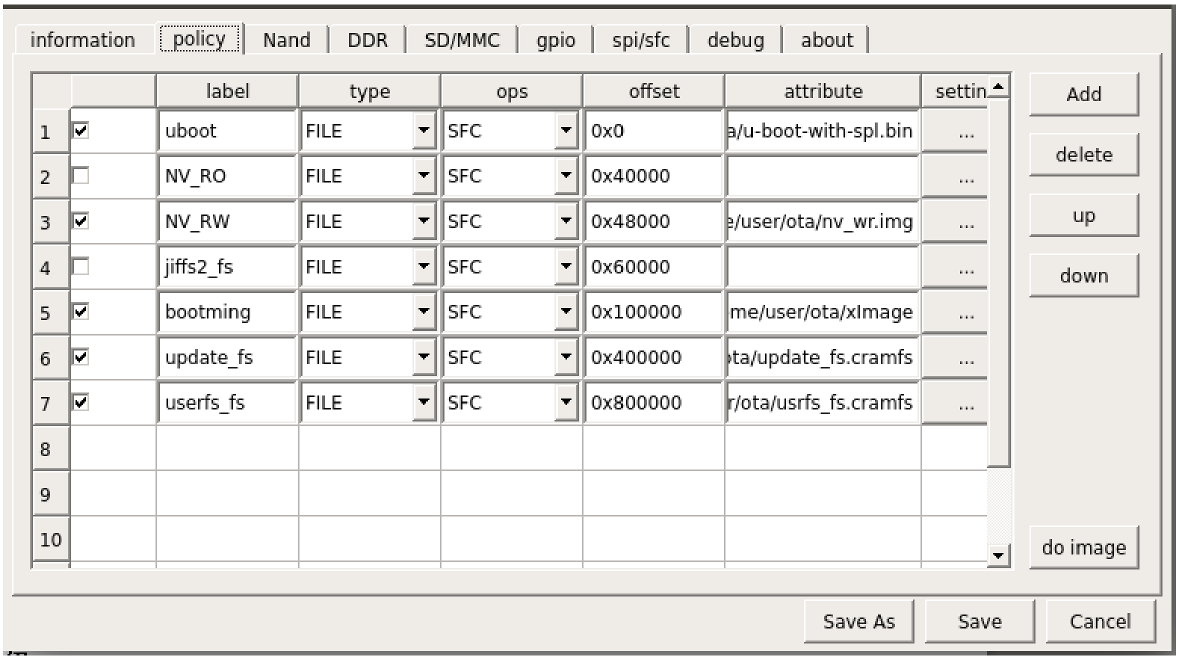
Note :Halley 2 version is not supported OTA currently.

5.1. Environment prepare

5.1.1.usbcloner partition configuration

choose "x1000_sfc_ota_lpddr_linux. CFG" in"board" option of "Information" option in the config

.



Among them:

1.u-boot burn address is 0x0.

2.NV_RO burn address is 0x40000

3.NV_RW burn address is 0x48000

4.jiffs2 burn address is 0x60000

5. xImage burn address is 0x100000

6. updatefs burn address is 0x400000

7. usrfs burn address is 0x800000

Note:jiffs2 is no need to burn

5.1.2.Make update image

1.Modify the configuration parameters

(1)Enter in directory “halley2/platform/development/device” ,vi device.mk file.

(2)Add “y” after “export MAKE_OTA=” as follows.

```
export DEVICE_NAME:=phoenix
export PRODUCT_NAME:=product
export OUT_DIR=out
export OUT_HOST=host
export OUT_TARGET=target
export SYSTEM-OTA=system_ota/ota_fs
export SYSTEM-OTA-USR=system_ota/ota_usr_fs
export MAKE_OTA=y
```

5.2. Compile

enter the following command to compile as a whole in the plateform directory.

```
$ make install
```

After compilation it will generated upgrade image required in halley2/platform/out/target/product/halley2/image.

It will generate the corresponding image is as follows:

File name	Function
u-boot-with-spl.bin	Boot image
xImage	Kernel image
update_fs.cramfs	update_fs image（file system used upgrade）
usrfs_fs.cramfs	userfs_fs image（file system used for user）

5.3. Make bin file of NV_RW partition

Carry out "dd if=/dev/zero of=xxx.bin bs=1024 count=96" command,The generation of xxx.bin to burn to NV_RW partitions to clear NV_RW area operations.

Note:Before the test, you need to clear NV_RW partition.

5.4. Make the upgrade package

1.put the new kernal, updatefs_fs usrfs_fs corresponding image for upgrading in halley2/platform/development/ota/updatezip/image.

File name	Function
u-boot-with-spl.bin	Boot image
xImage	Kernel image
update_fs.cramfs	update_fs image（file system used upgrade）
usrfs_fs.cramfs	userfs_fs image（file system used for user）

There is updated Scripts named update.script in “halley2/platform/development/ota/updatezip/split/scrip”.

2.in the directory “halley2/platform/development/ota”,carry out mkzip.sh make updated Scripts to Upload to server,the first parameter to the version, the second parameter to upload updates of the url.

e.g. `$./mkzip.sh 20150808 http://192.168.1.200/ota/download-bliu`

5.5. Upgrade

After system running, connect the wifi, input wr_flag URL, read the upgrade of nv flag bit, .enter the following command on a serial port:

```
e.g. $ wr_flag http://192.168.1.200/ota/download-bliu
```

After the upgrade completed, suggest to delete the last info and updates so as not to affect the next time to make new upgrade package .

Note:

1)If the input url is invalid, you can use alternate url, if both the url is invalid,you can use the default url.

2)After burn ota code, if you want to burn other code ,please select erase all item that is "config - > spi/SFC - > all erase" in the burn tool to burn normal.

6. The test cases

6.1. Camera test

After open the serial port,electricity to start the development board, into the "/testsuit/cim/" directory, input the following command in the current directory to use camera function:

1)take photo

```
$./cimutils -I 0 -C -v -x 320 -y 240
```

2)Preview

```
$./cimutils -I 0 -P -w 320 -h 240
```

3)take picture and preview

```
$./cimutils -I 0 -P -v -l -w 320 -h 240
```

Note:Execute the following command to view the specific parameters in the current directory:

```
$./cimutils --help
```

If the camera take photo successful the serial will print as follows:

```
write vdma chn
open clock!
write huffman table!
write quantization table!
write regs!
refresh cache
start vpu
poll:-----bslen = 8721
out
------mmapfd = 0
picture taked!!!!!!!![ 1076.618560] ov5640 0-003c: stream down



----------------------CIM TEST END ----------------------
VAE unmap successfully done!
------mmapfd = 0
```

After taking photo successful with camera, it will be generate test3.jpg file in the current directory path ,you can enter the following command look out the photo you have taked:

`$ adb pull /testsuit/cim/test3.jpg .`

After execute the command successfully,you can review images if correct or not in local.

6.2. USB Camera test

After open the serial port, electricity to start the development board, into the "/ testsuit/grab/" directory, input following command in the current directory to use USB Camera function:

`$./grab -w 320 -h 240 -c 10 -r 5 -y`

Note:Execute the following command in the current directory to view the specific parameters:

`$./grab --help`

If the USB Camera take photograph successful then serial print as follows:

After take photos successfully it will generate "p-x.jpg" photo files in the current path , the "x" for the "0 ~ 9".Enter the following command to look at pictures available on PC :

Below to view the "p-0.jpg" as an example:

```
width = 320, height = 240
input_yuv = 1
grab version 0.1.4
video /dev/video0
coc --1
{ pixelformat = ''YUYV'', description = ''YUV 4:2:2 (YUYV)'' }
coc --2
ddddddddddddddCurrent data format information:
        width:320
        height:240
running......
forever = 0,fcount=10
oooooooooo 120 frame time = 32 429496733696ns/frame
yuv
savejpeg.c:get_pictureYUYV:293
yuv
savejpeg.c:get_pictureYUYV:293
yuv
savejpeg.c:get_pictureYUYV:293
yuv
savejpeg.c:get_pictureYUYV:293
yuv
savejpeg.c:get_pictureYUYV:293
yuv
savejpeg.c:get_pictureYUYV:293
yuv
savejpeg.c:get_pictureYUYV:293
yuv
savejpeg.c:get_pictureYUYV:293
yuv
savejpeg.c:get_pictureYUYV:293
yuv
savejpeg.c:get_pictureYUYV:293
[   25.541770] dwc2 dwc2: Unlink after no-IRQ?  Controller is probably using the wrong IRQ.
close fd 3
```

```
$adb pull /testsuit/grab/p-0.jpg .
```

After execute the command successfully you can review images "p-0.jpg" in local.

6.3. WI-FI connection test

6.3.1. Halley2_v2. 0 (SPI-nor) development board using method

The update of airkiss support for multiple protocol (airkiss, cooee),it also changed the way to use it at the same time, concrete operation is as follows:

1.Install com. broadcom. cooeedemo-v1.4.0.apk on the Android mobile.

2.Open the wi-fi on phone and connected to an AP router.

3.Run BrcmCooee applications.

4.Input in BrcmCooee SSID name (the first input), the AP router's key (the second input) (to ensure the connection router of BrcmCooee applications and mobile phone is the same one).

5.Start the halley2 development board,input the following command on the serial port end.

```
$ airkiss
```

when output "Easy setup target library v3.2.0"in the serial port,click the send button on mobile application interface, halley2 development board will receive the SSID and keys, and began to connect AP router.

6.You can ping www.baidu.com or gateway to test network connectivity.

Due to the use for the first time have been generated configuration files which path is "/etc/wpa_supplicant.conf".So it will automatically connect to the Internet for the next time, if the configuration file which stored in the wifi name and password does not exist or is not correct,it will not be able to connect to the Internet, you can re-execute airkiss to configure the network.

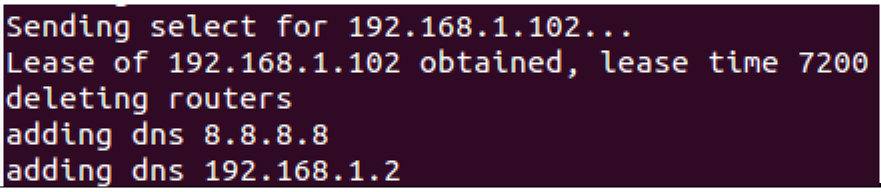
6.3.2.Halley2_mini_v2. 0 (SPI-nor) development board using method

1.Install AirKissDebugger.apk on the Android mobile.
2.Open the wi-fi on phone and connected to an AP router.
3.Run AirKissDebugger applications.
4.Input in AirKissDebugger SSID name (the first input), the AP router's key (the second input)and the AESKey(the third input),the default value of AESKey is "0123456789123456",There needs to be filled as "Wechatiothardwav".(to ensure the connection router of BrcmCooee applications and mobile phone is the same one)

5.input airkiss command in terminal.

```
$ airkiss
```

6.when the terminal display as shown in below means it is connected to the Internet successfully, you can ping www.baidu.com or gateway to test whether the network connectivity.



6.it requires the user to manually perform the following steps when performing airkiss insmod problem if it is in the two development board of halley2 mini nor or halley2 mini nand

a. cd the directory of kernel:carry out make modules.

b. cp drivers/net/wireless/ssv6xxx/ssv6051.ko ~/

c. adb push ~/ssv6051.ko /opt/modules

d. carry out airkiss again, repeat step 4 and step 5

6.4. Bluez test

1.Start bluez

When the development board was electric,you can start execution "bt_enable"to start the bluez, wait about 10 s to reappear the prompt "#".Start to finish.At this time using a mobile phone search bluetooth devices which can be found called "BlueZ" (executable pairing operation).

2. Sending files to the device

Execute the following steps:

sdptool add OPUSH

There will be the following message:

OBEX Object Push service registered

At this time match a mobile phone and development board(if there is a matching before this operation,you should cancel the current matching first, then to match)

Continue to execute the following commands

obex_test -b local 9

When appears the following message:

Using Bluetooth RFCOMM transport

OBEX Interactive test client/server

\>

Send “s” command:

(please get ready to send files, avoid wait too long, there is about 1 s operation delay)

\>s

Then send any files to the BlueZ equipment, when prompts a Timeout while doing OBEX_HandleInput () again input "s"

\> s

If you want to continue to upload files, please from "local 9 # obex_test - b" command for start until the final step.

3.Device receives the file

It began to receive files when appears the following information, until the file received complete (receive files in the current directory to store)

connect_server()

server request finished!

server_done() Command (00) has now finished

OBEX_HandleInput() returned 12
OBEX_HandleInput() returned 990

Note: Halley2_mini_v2. 0 (SPI - nor) development board does not support bluetooth function, Halley2_v2. 0 (SPI - nor) development board of bluetooth is still has a problem.

7. Common problems and solutions

7.1. Ubuntu Oracle VM VirtualBox virtual machine burning questions

7.1.1.The problem

If you use Oracle VM VirtualBox in Ubuntu environment installed Windows XP virtual machine , when you add a USB device, it appears the phenomenon that device name garbled , details are as follows:

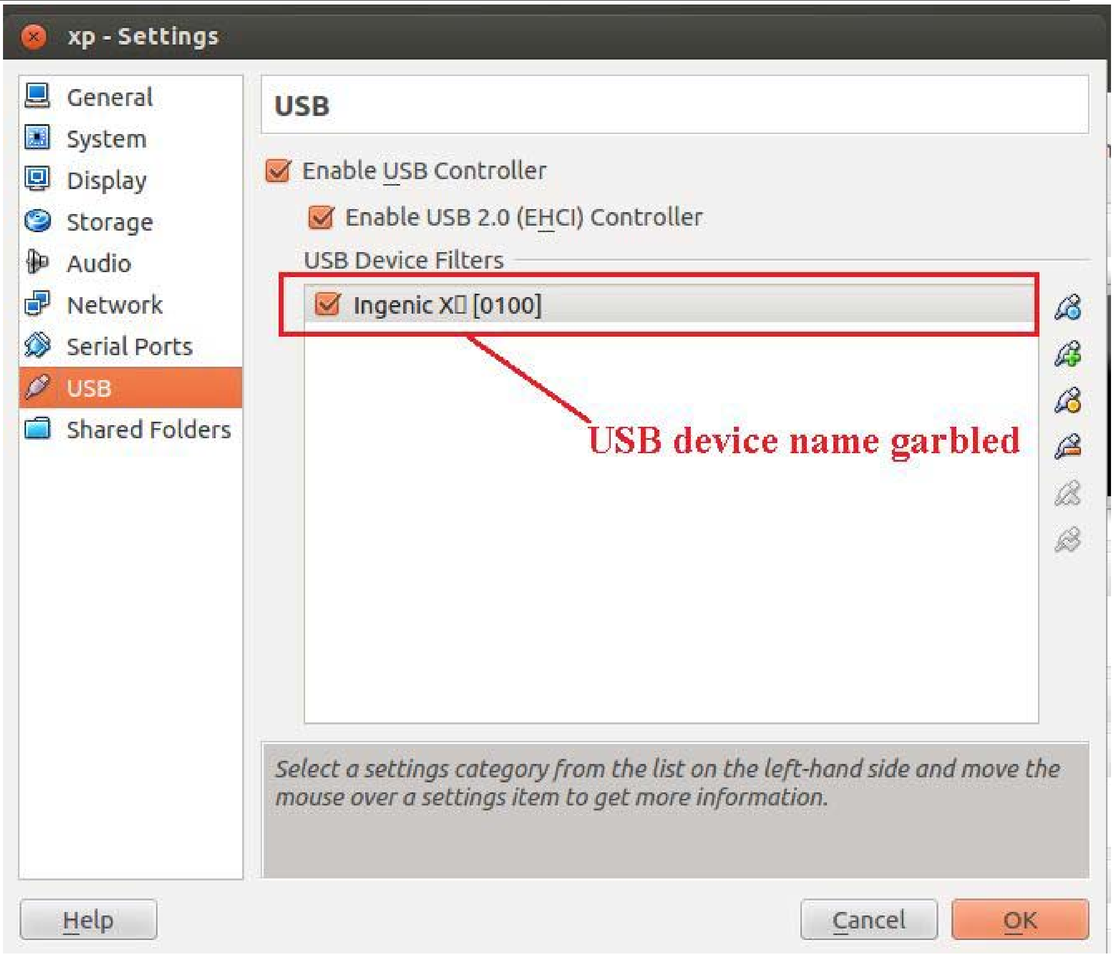


Figure 1

In order to ensure the normal use, it is suggested that once appear the above situation, just manually modify X1000 USB device description (look at 7.1.2) in this chapter, otherwise, the virtual machine will not be able to start machine,it will appear the following situations:

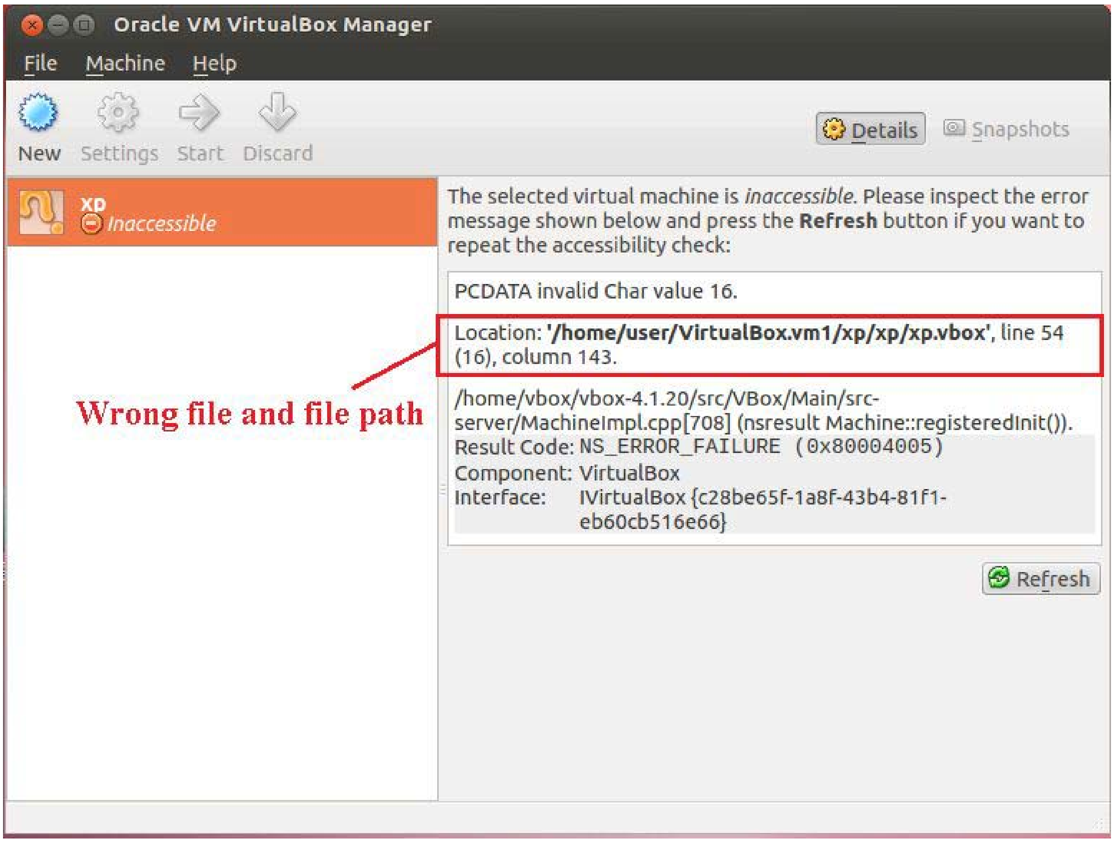


Figure 2

7.1.2.Solution

In the 7.1.1 have error message in figure 2, according to the information of graph that open the file"/home/user/VirtualBox.vm1/xp/xp/xp.vbox" under the environment of Ubuntu, delete the line 54.

Figure 3

Save the file after modified.Restart the virtual machine and add the USB device again, the X1000 USB device description information interface as shown in the figure below:

Figure 4

The above the option of the contents of the "Name" changed to "ingenix X1000", you can clear contents of the "Product" option to ensure it is empty, modified as follows, press the "OK" to save it.

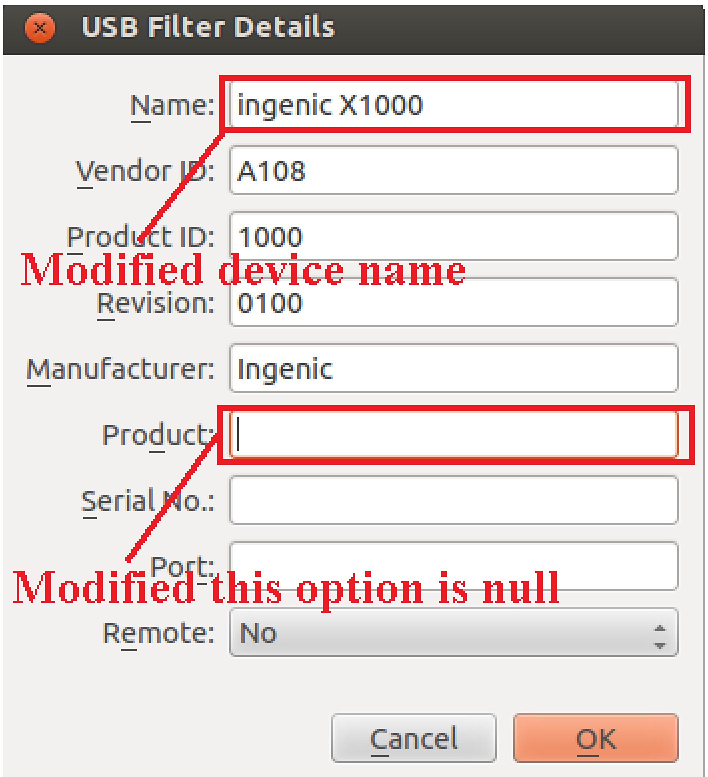


Figure 5

7.2. The problem of the adb under Ubuntu

7.2.1. Problem

If your adb in Ubuntu environment cannot be used, the specific phenomenon are as follows:

```
user@user-FMVDB2A0C1:~$ adb push hello /
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
error: insufficient permissions for device
user@user-FMVDB2A0C1:~$
```

7.2.2. Solution

the terminal execute the following command under the super user permissions:

adb kill-server

adb start-server

Exit the super user afterm command execution is completed, , the adb can be normal used after operation is completed.

8. The source code to compile

8.1. The source code directory structure

The SDK directory structure:

```
        * README：              //README for this platform
        * burnertools           //usbcloner
        * docs                  //documents and README
        * toolchains:           //crossover toolchain
        * platform：            // all source files platform development need
           |--kernel            //kernel code
           |--u-boot            //u-boot code
           |--development       //platform related content
           |      |--device     //board level related configuration
           |                    e.q. kernel and0 u-boot configuration files
           |      |--rootfs     //zip file of file system
           |      |--testsuit   //all the executable test program
           |
           |--Makefile          //the overall build script
```

```
|--tools
|     ├── android    //adb code and tool of generating the adb
|     |— build       //Makefile and tools of making file system
|--out                //the file output directory of compiled
      |--target/product/halley2
            |--image     //u-boot-with-spl.bin、uImage、system.jffs2
            |--testsuit  //the generated test procedures
            |--tools     // generation tool program
            |--system    //decompression of the file system
```

8.2. The overall compilation

Configuration of development board (halley2_nor halley2_nand, halley2_mini_nor, halley2_mini_nand)With halley2 mini nor development board as example,depending on the type of development board, modify the file platform/device/device. Mk.

export BOARD_NAME=halley2_mini_nor

I.in the directory of halley2,carry out the follows command:

```
$ cd platform
$ source tools/build/source.sh
```

II. Compile

```
$ make
```

Compilation is complete, in the current directory the out/target/product/halley2/image/ generate three image files in the following table:

File name	function
u-boot-with-spl.bin	U-boot image
uImage	Kernel image
system.jffs2	File system image

III.install

If you want to install the test program and tools into the system,you can perform the following commands:

```
$ make install
```

This command will also compile all project source code.

8.3. Part of the compilation

First of all, in halley2 directory, execute the following command:

```
$ cd platform
$ source tools/build/source.sh
```

A. Compile the bootloader

In the directory of platform,execute the following commands:

```
$ cd u-boot
```

$ make halley2_v10_uImage_sfc_nor (both of two board development)

It will generate u-boot-with-spl.bin in current directory.

B. Compile the kernel

In the directory of platform,execute the following commands:

$ cd kernel

$make halley2_nor_v10_linux_defconfig (Halley2_v2.0(SPI-nor) development board)

$make halley2_mini_nor_v10_linux_defconfig(Halley2_mini_v2.0(SPI-nor)development board)

$ make uImage

It will generate uImage in the directory “arch/mips/boot/”.

Note:Test host can install Ubuntu 12.04 32 bit and 64 bit Ubuntu 12.04, Ubuntu 14.04 32 bit and 64 bit Ubuntu 14.04, this document is under in the test host installed Ubuntu 14.04 64 bit environment operation.