

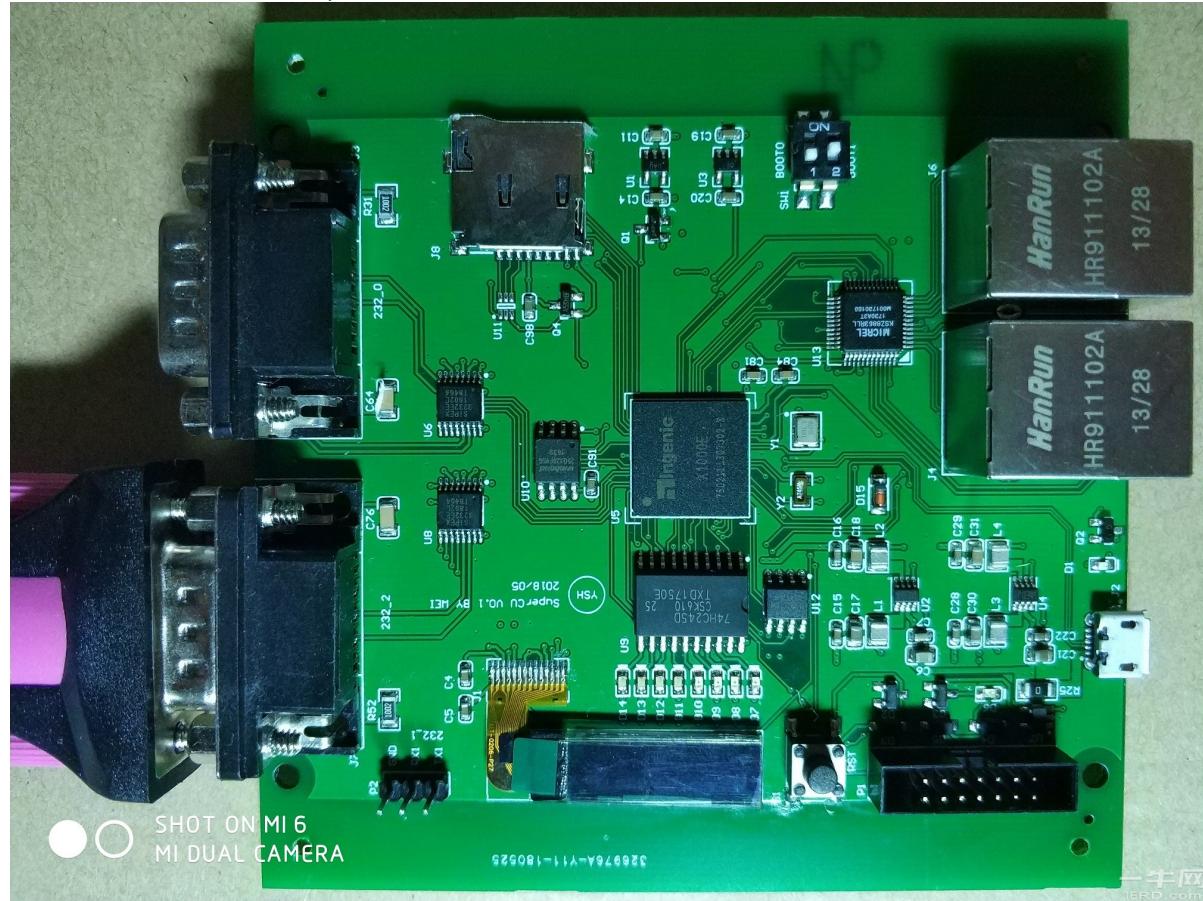
On Raspberry Pi and based on JunzhengJZ4780's CI20 board has all ran Debian8. I wondered if I could cut Debian8 and run it on a lightweight platform such as Ingenic X1000 E. First open a post and plan to toss and toss in the near future.

SayoneSayBigTooththinkroad,ForeseeccountHavebyDownSeveralindvualstepSudden:

1. Based on the existing BSP, the system can be started by TF card.
2. Package Debian 8 rootfs, instead of Buildroot in the original BSP.
3. Solve various bugs that appear.

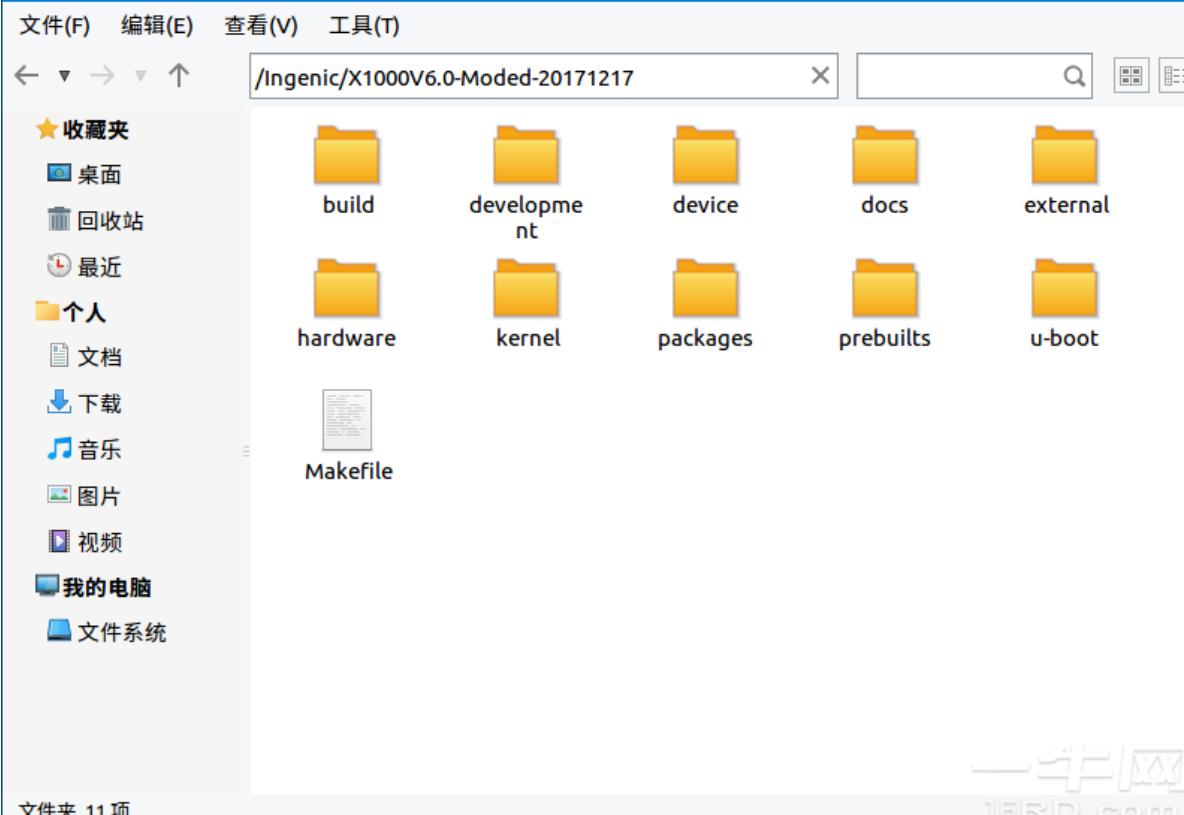
baseAtnowHaveBSPalreadythroughbecomeachievementPasspassTFCardstartmove,DownonestepwilltastetyleguideDebian8ofRootfs,baseBookAffectionconditionlikeDown:

- hardPiecesflattower:SuperCU Basic V0.1.



- softPiecesflattower:Based on Ingenic official Phoenix v6.0 BSP transplant RatCharm BSP.

X1000V6.0-Moded-20171217



```
#  
# cat /proc/cpuinfo  
system type : RatCharm  
machine : Unknown  
processor : 0  
cpu model : Ingenic Xburst V4.15 FPU V0.0  
BogoMIPS : 1196.85  
wait instruction : yes  
microsecond timers : no  
tlb_entries : 32  
extra interrupt vector : yes  
hardware watchpoint : yes, count: 1, address/irw mask: [0x0fff]  
isa : mips32rl  
ASEs implemented :  
shadow register sets : 1  
kscratch registers : 0  
core : 0  
VCED exceptions : not available  
VCEI exceptions : not available  
  
Hardware : RatCharm  
Serial : 00000000 00000000 00000000 00000000  
#
```

一牛网
16RD.com

● byDownYesshiftplantstepSudden:

1. Modify and compile uboot for TF card startup.

Because the debug serial port derived from the hardware platform used is reused for Port DUART2. Therefore, it is necessary to modify the configuration of the print serial port. Modify the main frequency configuration in include/configs/RatCharm.h in the u-boot folder. The following is the original code:

01. # define CONFIG_SYS_APLL_FREQ	1008000000	/*If APLL not use mast be set 0*/ /*If
02. # define CONFIG_SYS_MPPLL_FREQ	600000000	MPPLL not use mast be set 0*/
03. # define CONFIG_CPU_SEL_PLL	APLL	
04. # define CONFIG_DDR_SEL_PLL	MPLL	
05. # define CONFIG_SYS_CPU_FREQ	1008000000	
06. # define CONFIG_SYS_MEM_FREQ	200000000	

[Copy code](#)

The modified code is as follows:

01. # define CONFIG_SYS_APLL_FREQ	1200000000	/*If APLL not use mast be set 0*/ /*If
02. # define CONFIG_SYS_MPPLL_FREQ	600000000	MPPLL not use mast be set 0*/
03. # define CONFIG_CPU_SEL_PLL	APLL	
04. # define CONFIG_DDR_SEL_PLL	MPLL	
05. # define CONFIG_SYS_CPU_FREQ	1200000000	
06. # define CONFIG_SYS_MEM_FREQ	200000000	

[Copy code](#)

Then modify the configuration part of the print serial port under the same file, the following is the original code:

01. # define CONFIG_SYS_UART_INDEX	2
02. # if defined(CONFIG_RatCharm_BARE) defined(CONFIG_RatCharm_RTTHREAD)	
03. # define CONFIG_SYS_UART2_PD	
04. # define CONFIG_BAUDRATE	115200
05. # else	
06. # define CONFIG_SYS_UART2_PC	
07. # define CONFIG_BAUDRATE	115200
08. # endif	

[Copy code](#)

The modified code is as follows:

```
01. # define CONFIG_SYS_UART_INDEX          2
02. # if defined(CONFIG_RatCharm_BARE) || defined(CONFIG_RatCharm_RTTHREAD)
03. # define CONFIG_SYS_UART2_PD
04. # define CONFIG_BAUDRATE             115200
05. # else
06. # define CONFIG_SYS_UART2_PD
07. # define CONFIG_BAUDRATE             115200
08. # endif
```

[Copy code](#)

After the modification is completed, use the following command to compile. The compilation takes about 1 minute. After the compilation is completed, you can get the u-boot-with-spl-mbr- for TF card startup in the u-boot directory gpt.bin file.

```
01. make RatCharm_uImage_msc0 -j4
```

[Copy code](#)

2. Modify and compile the kernel used for TF card startup. (Take the make menuconfig method as an example) As mentioned earlier, since the debug serial port uses Port D's UART2, you first need to modify the corresponding configuration in the kernel configuration.

First enter the kernel directory, use make RatCharm_linux_defconfig performs the default configuration, and then changes the original default Port C port to Port D port, as follows:

```
01. | Symbol: SERIAL_JZ47XX_UART2_PD [=y] |
02. Type: boolean
03. | Prompt: PORT D
04. | Location:
05. |     -> Device Drivers
06. |         -> Character devices
07. |             -> Serial drivers
08. |                 -> ingenic jz47xx serial port support (SERIAL_JZ47XX_UART [=y])
09. |                     -> enable uart2 (SERIAL_JZ47XX_UART2 [=y])
10. |                         -> JZ SERIAL GPIO function pins select (<choice> [=y])
11. |                             Defined at drivers/tty/serial/Kconfig:1575 Depends
12. |                                 on: <choice> && SOC_X1000 [=y]
```

[Copy code](#)

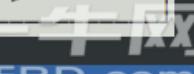
The screenshot comparison before and after modification, the above picture is before modification and the following picture is after modification:

```
.config - Linux/mips 3.10.14 Kernel Configuration
> Device Drivers > Character devices > Serial drivers -----
      Serial drivers
Arrow keys navigate the menu. <Enter> selects submenus ---.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
^(-)
[*] Console on jz47xx and compatible serial port
[*] enable uart0
[ ]   enable uart0 wakeup cpu
[ ]   enable uart0 dma mode
[ ]   enable uart1
[*] enable uart2
[ ]   enable uart2 wakeup cpu
[ ]   enable uart2 dma mode
[ ]     JZ SERIAL GPIO function pins select (PORT C) --->
[ ]   enable uart3
(+)

<Select>  < Exit >  < Help >  < Save >  < Load >
```

```
.config - Linux/mips 3.10.14 Kernel Configuration
> Device Drivers > Character devices > Serial drivers
    Serial drivers
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
^(-)
[*] Console on jz47xx and compatible serial port
[*] enable uart0
[ ]   enable uart0 wakeup cpu
[ ]   enable uart0 dma mode
[ ] enable uart1
[*] enable uart2
[ ]   enable uart2 wakeup cpu
[ ]   enable uart2 dma mode
[*] JZ SERIAL GPIO function pins select (PORT D) --->
[ ] enable uart3
I(+)

<Select> < Exit > < Help > < Save > < Load >
```



The current platform does not have a wireless communication module, so modify the corresponding configuration to turn off the wireless network and Bluetooth. There are three functions that need to be modified:

First, turn off Wireless LAN, as follows:

01. | Symbol: WLAN [=n]
02. | Type: boolean
03. | Prompt: Wireless LAN
04. | Location:
05. | -> Device Drivers
06. | -> Network device support (NETDEVICES [=y])
07. | Defined at drivers/net/wireless/Kconfig:5 Depends on:
08. | NETDEVICES [=y] && !S390 && NET [=y] Selects: WIRELESS
09. | [=n]

[Copy code](#)

The screenshot comparison before and after modification, the above picture is before modification and the following picture is after modification:

```
.config - Linux/mips 3.10.14 Kernel Configuration
> Device Drivers > Network device support
    Network device support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
^(-)
*** CAIF transport drivers ***
Distributed Switch Architecture drivers --->
[*] Ethernet driver support --->
--> PHY Device support and infrastructure --->
< > PPP (point-to-point protocol) support
< > SLIP (serial line) support
USB Network Adapters --->
[*] Wireless LAN --->
*** Enable WiMAX (Networking options) to see the WiMAX driv
[ ] Wan interfaces support --->
I(+)

<Select> < Exit > < Help > < Save > < Load >
```



```

.config - Linux/mips 3.10.14 Kernel Configuration
> Device Drivers > Network device support
    Network device support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
^(-)
    *** CAIF transport drivers ***
    Distributed Switch Architecture drivers --->
[*] Ethernet driver support --->
-* PHY Device support and infrastructure --->
< > PPP (point-to-point protocol) support
< > SLIP (serial line) support
    USB Network Adapters --->
[ ] Wireless LAN --->
    *** Enable WiMAX (Networking options) to see the WiMAX driv
[ ] Wan interfaces support --->
I(+)

<Select> < Exit > < Help > < Save > < Load >

```



16RD.com

The second place is to turn off Wireless, as follows:

```

01. | Symbol: WIRELESS [=n] |
02. Type: boolean
03. | Prompt: Wireless
04. | Location:
05. |     -> Networking support (NET [=y])
06. |     Defined at net/Kconfig:334 Depends
07. |     on: NET [=y] && !S390
08. |     Selected by: WLAN [=n] && NETDEVICES [=y] && !S390 && NET [=y]

```

[Copy code](#)

The screenshot comparison before and after modification, the above picture is before modification and the following picture is after modification:

```

.config - Linux/mips 3.10.14 Kernel Configuration
> Networking support
    Networking support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
--- Networking support
    Networking options --->
[ ] Amateur Radio support --->
< > CAN bus subsystem support --->
< > IrDA (infrared) subsystem support --->
<*> Bluetooth subsystem support --->
< > RxRPC session sockets
[*] Wireless --->
< > WiMAX Wireless Broadband support --->
<*> RF switch subsystem support --->
I(+)

<Select> < Exit > < Help > < Save > < Load >

```



16RD.com

```
.config - Linux/mips 3.10.14 Kernel Configuration
> Networking support
    Networking support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
--- Networking support
    Networking options --->
    [ ] Amateur Radio support --->
    < > CAN bus subsystem support --->
    < > IrDA (infrared) subsystem support --->
    <*> Bluetooth subsystem support --->
    < > RxRPC session sockets
    [ ] Wireless --->
    < > WiMAX Wireless Broadband support --->
    <*> RF switch subsystem support --->
    ↴(+)
<Select> < Exit > < Help > < Save > < Load >
```



Third place, closed Bluetooth Subsystem support, as follows:

01. | Symbol: BT [=n] |
02. Type: tristate
03. | Prompt: Bluetooth subsystem support
04. | Location:
05. | -> Networking support (NET [=y])
06. | Defined at net/bluetooth/Kconfig:5
07. | Depends on: NET [=y] && !S390 && (RFKILL [=n] || !RFKILL [=n]) Selects: CRC16
08. | [=y] && CRYPTO [=y] && CRYPTO_BLKCIPHER [=y] && \ | CRYPTO_AES [=y] &&
09. CRYPTO_ECB [=y] && CRYPTO_SHA256 [=y]

[Copy code](#)

The screenshot comparison before and after modification, the above picture is before modification and the following picture is after modification:

```
.config - Linux/mips 3.10.14 Kernel Configuration
> Networking support
    Networking support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
--- Networking support
    Networking options --->
    [ ] Amateur Radio support --->
    < > CAN bus subsystem support --->
    < > IrDA (infrared) subsystem support --->
    <*> Bluetooth subsystem support --->
    < > RxRPC session sockets
    [ ] Wireless --->
    < > WiMAX Wireless Broadband support --->
    <*> RF switch subsystem support --->
    ↴(+)
<Select> < Exit > < Help > < Save > < Load >
```



```
.config - Linux/mips 3.10.14 Kernel Configuration
> Networking support
    Networking support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
--- Networking support
    Networking options --->
    [ ] Amateur Radio support --->
    < > CAN bus subsystem support --->
    < > IrDA (infrared) subsystem support --->
    <*> Bluetooth subsystem support --->
    < > RxRPC session sockets
    [ ] Wireless --->
    < > WiMAX Wireless Broadband support --->
    <*> RF switch subsystem support --->
    ↴(+)
<Select> < Exit > < Help > < Save > < Load >
```



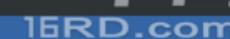
After turning off the wireless network and Bluetooth-related functions, the corresponding power control function also needs to be turned off, and two changes are required.
First, close the RF switch subsystem support, as follows:

01. | Symbol: RFKILL [=n] |
02. Type: tristate
03. | Prompt: RF switch subsystem support |
04. Location:
05. | -> Networking support (NET [=y]) | Defined
06. at net/rfkill/Kconfig:4 | Depends on: NET [=y]
- 07.

[Copy code](#)

The screenshot comparison before and after modification, the above picture is before modification and the following picture is after modification:

```
.config - Linux/mips 3.10.14 Kernel Configuration
> Networking support
    Networking support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
^(-)
    [ ] Amateur Radio support --->
    < > CAN bus subsystem support --->
    < > IrDA (infrared) subsystem support --->
    < > Bluetooth subsystem support --->
    < > RxRPC session sockets
    [ ] Wireless --->
    < > WiMAX Wireless Broadband support --->
    <*> RF switch subsystem support --->
    < > Plan 9 Resource Sharing Support (9P2000) --->
    < > CAIF support --->
    ↴(+)
<Select> < Exit > < Help > < Save > < Load >
```



Networking support

Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded <M> module < >

- ^(-)
 - [] Amateur Radio support --->
 - < > CAN bus subsystem support --->
 - < > IrDA (infrared) subsystem support --->
 - < > Bluetooth subsystem support --->
 - < > RxRPC session sockets
 - [] Wireless --->
 - < > WiMAX Wireless Broadband support --->
 - <*> RF switch subsystem support --->**
 - < > Plan 9 Resource Sharing Support (9P2000) --->
 - < > CAIF support --->
- ↓(+)

<Select> < Exit > < Help > < Save > < Load >



16RD.com

The second place, close the BCM module power The control core driver is as follows:

01. | Symbol: BCM_PM_CORE [=n]
02. | Type: tristate
03. | Prompt: BCM module power control core driver |
04. Location:
05. | -> Device Drivers | -> Misc
06. devices
07. | Defined at drivers/misc/Kconfig:545

[Copy code](#)

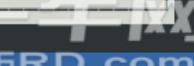
The screenshot comparison before and after modification, the above picture is before modification and the following picture is after modification:

Misc devices

Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded <M> module < >

- ^(-)
 - < > ROHM BH1780GLI ambient light sensor
 - < > BH1770GLC / SFH7770 combined ALS - Proximity sensor
 - < > APDS990X combined als and proximity sensors
 - < > Honeywell HMC6352 compass
 - < > Dallas DS1682 Total Elapsed Time Recorder with Alarm
 - [] UID based statistics tracking exported to /proc/uid_stat
 - < > BMP085 digital pressure sensor on I2C
 - < > FSA9480 USB Switch
 - <*> BCM module power control core driver**
 - [] Generic on-chip SRAM driver
- ↓(+)

<Select> < Exit > < Help > < Save > < Load >



16RD.com

.config - Linux/mips 3.10.14 Kernel Configuration
 > Device Drivers > Misc devices

Misc devices

Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded <M> module < >

^(-)

- < > ROHM BH1780GLI ambient light sensor
- < > BH1770GLC / SFH7770 combined ALS - Proximity sensor
- < > APDS990X combined als and proximity sensors
- < > Honeywell HMC6352 compass
- < > Dallas DS1682 Total Elapsed Time Recorder with Alarm
- [] UID based statistics tracking exported to /proc/uid_stat
- < > BMP085 digital pressure sensor on I2C
- < > FSA9480 USB Switch
- < > BCM module power control core driver**
- [] Generic on-chip SRAM driver

I(+)

<Select> < Exit > < Help > < Save > < Load >

Use the following command to compile after all modifications are completed. Compilation takes about 3 to 5 minutes. After the compilation is completed, you can get the ulmage file for TF card startup in the kernel/arch/mips/boot directory.

01. make ulmage -j4

[Copy code](#)

3. Package the Build Root file system in ext4 format. Since it is started with a TF card, the file system needs to be made into ext4 format. The specific steps are as follows: (1) Enter the BSP root directory and select RatCharm_norflash-user to perform a complete compilation of the BSP. (2) At this time, the system directory required to make rootfs can be found in the out/product/RatCharm path. (3) Enter the above path (do not enter the system directory) and execute the following commands in sequence:

01. dd if=/dev/zero of=system.ext4 bs=1M count=512
 02. mkfs.ext4 system.ext4
 03. sudo mount -o loop system.ext4 /mnt sudo
 04. cp -a -fr system/* /mnt sync
 05.
 06. sudo umount /mnt
 07. sudo chmod 777 system.ext4

[Copy code](#)

If it is successfully completed, the target file named system.ext4 can be obtained. Use cloner to burn the three files obtained in the above steps to realize booting from the TF card.

- AttachedsuperiorRatCharmflattowerBSPBagandalreadyEditTranslategoodofuseAtTFCardsstartmoveofitemMarkArtsPieces:

RatCharm platform BSP package:

-  X1000V6.0-Moded-20171217.part1.rar (450 MB, download times: 8)
-  X1000V6.0-Moded-20171217.part2.rar (430.67 MB, download times: 7)

Target file for TF card startup:

-  u-boot-with-spl-mbr-gpt.zip (95.18 KB, download times: 2)
-  ulmage.zip (2.4 MB, download times: 2)
-  system.zip (9.35 MB, download times: 2)

catchDownComewilltastetyleadguideDebian8ofRootfs,ForeseeccountCanHavebyDownSeveralindivualstepSudden:

1. Package the small rootfs of Debian8 and make it into ext4 format mirror.
2. Replace the previous Buildroot with Debian8's rootfs mirror.
3. Try to boot Debian 8 and solve various problems that arise.

1. Modify and use the mkdebianrfs.sh script to package the small rootfs of Debian8, which comes from the CI20 project of IMG. Because the debugging serial port used by the hardware platform used is UART2, which is different from the debugging serial port used by the CI20 platform, it needs to be modified first.

The following is the original code in the script:

```
01. # Configure a serial console.  
02. echo "T0:23:respawn:/sbin/getty -L ttyS0 115200 vt100" >> \  
03.     "${target_dir}/etc/inittab"
```

[Copy code](#)

The modified code is as follows:

```
01. # Configure a serial console.  
02. echo "T0:23:respawn:/sbin/getty -L ttyS2 115200 vt100" >> \  
03.     "${target_dir}/etc/inittab"
```

[Copy code](#)

Because the original script uses a foreign source by default, in order to avoid too long packaging time due to network speed issues, it needs to be changed to a domestic source. The following is the original code in the script:

```
01. debian_packages="locales" debian_mirror="http://ftp.uk.debian.org/debian/"  
02. debian_path="/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr /sbin:/sbin"  
03.
```

[Copy code](#)

The modified code is as follows, you can choose other domestic sources according to your actual situation:

```
01. debian_packages="locales" debian_mirror="http://mirrors.ustc.edu.cn/debian/"  
02. debian_path="/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr /sbin:/sbin"  
03.
```

[Copy code](#)

2. Install the necessary software packages for packaging.

Because mkdebianrfs.sh calls debootstrap, debootstrap needs to be installed. At the same time, because X1000E is a processor of the MIPS instruction set, in order to package on the X86 platform, a qemu emulator must be installed. Execute the following commands in sequence to complete the installation of the required software packages:

```
01. sudo apt-get update  
02. sudo apt-get install binfmt-support qemu qemu-user-static debootstrap bzip2 gedit
```

[Copy code](#)

3. Create the required directories, and complete the packaging and related settings of the Debian8 small rootfs. Execute the following commands in sequence to create the required directories and package them:

```
01. mkdir debian8-rootfs  
02. sudo ./mkdebianrfs.sh mipsel jessie debian8-rootfs
```

[Copy code](#)

At this point, the packaging of the Debian8 small rootfs will begin, which may take a few minutes to ten minutes depending on the network situation. After the packaging is complete, you will be prompted to set a password for the root account, as shown in the figure below, you need to enter it twice.

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

```
I: Configuring libp11-kit0:mipsel...
I: Configuring libgnutls-deb0-28:mipsel...
I: Configuring wget...
I: Configuring apt...
I: Configuring man-db...
I: Configuring libnetfilter-acct1:mipsel...
I: Configuring rsyslog...
I: Configuring nfacct...
I: Configuring apt-utils...
I: Configuring libirs-export91...
I: Configuring libgnutls-openssl27:mipsel...
I: Configuring iputils-ping...
I: Configuring isc-dhcp-common...
I: Configuring isc-dhcp-client...
I: Configuring tasksel...
I: Configuring tasksel-data...
I: Configuring libc-bin...
I: Configuring systemd...
I: Base system installed successfully.
```

Configuring target system...

Please enter a password for the root user:
Enter new UNIX password: █



After setting the root password, you will be prompted to select the language pack to be installed. Select the four simplified Chinese language packs shown in the figure below:

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

Package configuration

Configuring locales

Locales are a framework to switch between multiple languages and allow users to use their language, country, characters, collation order, etc.

Please choose which locales to generate. UTF-8 locales should be chosen by default, particularly for new installations. Other character sets may be useful for backwards compatibility with older systems and software.

Locales to be generated:

- [*] zh_CN GB2312
- [*] zh_CN.GB18030 GB18030
- [*] zh_CN.GBK GBK
- [*] zh_CN.UTF-8 UTF-8
- [] zh_HK BIG5-HKSCS

<Ok>

<Cancel>



Then you will be prompted to select the language pack used by the system by default, and select zh_CN.UTF-8, as shown in the figure below. This step takes a long time and takes about 3 to 5 minutes depending on the configuration of the machine.

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

Package configuration

| Configuring locales |

Many packages in Debian use locales to display text in the correct language for the user. You can choose a default locale for the system from the generated locales.

This will select the default language for the entire system. If this system is a multi-user system where not all users are able to speak the default language, they will experience difficulties.

Default locale for the system environment:

- None
- C.UTF-8
- zh_CN
- zh_CN.GB18030
- zh_CN.GBK
- zh_CN.UTF-8**

<Ok>

<Cancel>

一生双刃
1GRD.com

After the language configuration is complete, set the geographic location according to the prompts, and select Asia as shown in the following figure:

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

Package configuration

| Configuring tzdata |

Please select the geographic area in which you live. Subsequent configuration questions will narrow this down by presenting a list of cities, representing the time zones in which they are located.

Geographic area:

- Asia**
- Atlantic Ocean
- Europe
- Indian Ocean
- Pacific Ocean
- System V timezones
- US
- None of the above

<Ok>

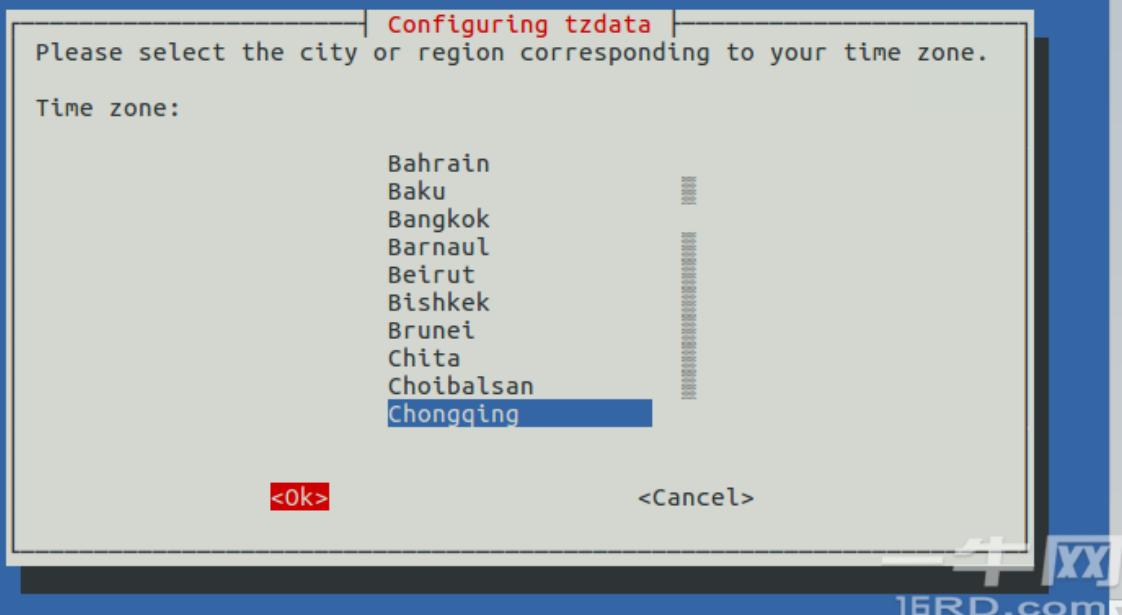
<Cancel>

一生双刃
1GRD.com

Set the time zone according to the prompts. In China, you can choose Shanghai or Chongqing. Because I am in Chengdu, I choose Chongqing nearby.

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

Package configuration



After completing all operations, enter exit according to the prompts to exit the packaging script, as shown below,

the above picture is when the packaging is completed, and the following picture is after the script is exited.

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

```
I: Configuring systemd...
I: Base system installed successfully.
```

Configuring target system...

```
Please enter a password for the root user:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Generating locales (this might take a while)...
 zh_CN.GB2312... done
 zh_CN.GB18030... done
 zh_CN.GBK... done
 zh_CN.UTF-8... done
Generation complete.
```

```
Current default time zone: 'Asia/Chongqing'
Local time is now:      Thu Jun 28 21:37:12 CST 2018.
Universal Time is now:  Thu Jun 28 13:37:12 UTC 2018.
```

Entering target system for additional configuration. Type 'exit' when done.

```
root@jdi-virtual-machine:/# exit
```

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

```
Please enter a password for the root user:  
Enter new UNIX password:  
Retype new UNIX password:  
passwd: password updated successfully  
Generating locales (this might take a while)...  
zh_CN.GB2312... done  
zh_CN.GB18030... done  
zh_CN.GBK... done  
zh_CN.UTF-8... done  
Generation complete.  
  
Current default time zone: 'Asia/Chongqing'  
Local time is now: Thu Jun 28 21:37:12 CST 2018.  
Universal Time is now: Thu Jun 28 13:37:12 UTC 2018.  
  
Entering target system for additional configuration. Type 'exit' when done.  
root@jdi-virtual-machine:/# exit  
exit  
Done!  
jdi@jdi-virtual-machine:/Ingenic$
```



Next, you need to make the packaged Debian8 Rootfs into an ext4 format image. Similar to making Buildroot's ext4 image, execute the following commands in sequence:

01. dd if=/dev/zero of=debian8-rootfs.ext4 bs=1M count=512 mkfs.ext4
02. debian8-rootfs.ext4
03. sudo mount -o loop debian8-rootfs.ext4 /mnt sudo cp
04. -a -fr debian8-rootfs/* /mnt sync
- 05.
06. sudo umount /mnt
07. sudo chmod 777 debian8-rootfs.ext4

[Copy code](#)

If it is successfully completed, you can get the target file named debian8-rootfs.ext4, replace the previous system.ext4 with this file, and use the cloner to burn to the target board.

catchDownCome will try to lead guide Debian8 and until tie Decide Mutually shut Ask question

● NSoneSecond-rate try:

COM5 - PuTTY

```
[ 0.578228] usbcore: registered new interface driver uvcvideo
[ 0.584243] USB Video Class driver (1.1.1)
[ 0.593437] jzmmc_v1.2 jzmmc_v1.2.0: card inserted, state=0
[ 0.623398] jzmmc_v1.2 jzmmc_v1.2.0: register success!
[ 0.664452] jzmmc_v1.2 jzmmc_v1.2.1: register success!
[ 0.670123] usbcore: registered new interface driver usbid
[ 0.675971] usbid: USB HID core driver
[ 0.680976] jz-asoc-aic jz-asoc-aic: Aic core probe success
[ 0.687062] jz-asoc-aic-i2s jz-asoc-aic-i2s: i2s platform probe success
[ 0.694112] icdc-d3 icdc-d3: codec icdc-d3 platform probe success
[ 0.700535] jz-asoc-pcm jz-asoc-pcm: pcm platform probe success
[ 0.707365] icdc-d3 icdc-d3: codec icdc-d3 probe enter
[ 0.712714] write extend : sreg: 3 [0 - 4], creg: 10 sdata: 180
[ 0.720088] gpio speaker enable 0
[ 0.723899] dma dma0chan4: Channel 4 have been requested.(phy id 3,type 0x06 desc a3eb0000)
[ 0.732777] dma dma0chan5: Channel 5 have been requested.(phy id 2,type 0x06 desc a3eb1000)
[ 0.741719] ingenic-alsa ingenic-alsa.0: icdc-d3-hifi <-> jz-asoc-aic-i2s mapping ok
[ 0.750366] dma dma0chan0: Channel 0 have been requested.(phy id 7,type 0x20 desc a3eb2000)
[ 0.759297] dma dma0chan1: Channel 1 have been requested.(phy id 6,type 0x20 desc a3eb3000)
[ 0.768238] ingenic-alsa ingenic-alsa.0: pcm dump dai <-> jz-asoc-pcm mapping ok
[ 0.776517] dma dma0chan7: Channel 7 have been requested.(phy id 0,type 0x04 desc a3eb4000)
[ 0.785419] ingenic-alsa ingenic-alsa.0: dmic dump dai <-> jz-asoc-dmic mapping ok
[ 0.794706] input: RatCharm Headset Jack as /devices/platform/ingenic-alsa.0/sound/card0/input0
[ 0.804343] Netfilter messages via NETLINK v0.30.
[ 0.809235] nfnl_acct: registering with nfnetlink.
[ 0.814429] ip_set: protocol 6
[ 0.817759] ip_tables: (C) 2000-2006 Netfilter Core Team
[ 0.823472] TCP: cubic registered
[ 0.826910] NET: Registered protocol family 17
[ 0.831657] Bridge firewalling registered
[ 0.836861] file system registered
[ 0.841615] using random self ethernet address
[ 0.846350] using random host ethernet address
[ 0.851287] android_usb_gadget: Mass Storage Function, version: 2009/09/11
[ 0.858485] android_usb_gadget: Number of LUNs=1
[ 0.863265] lun0: LUN: removable file: (no medium)
[ 0.868639] android_usb_gadget: android_usb ready
[ 0.873989] input: gpio-keys as /devices/platform/gpio-keys/input/input1
[ 0.881259] jz-rtc jz-rtc.0: setting system clock to 2018-06-28 17:25:36 UTC (1530206736)
[ 0.889893] ALSA device list:
[ 0.892964] #0: RatCharm
[ 0.896173] Waiting 1sec before mounting root device...
[ 1.633922] mmc0: new high speed SDHC card at address 1234
[ 1.640062] mmcblk0: mmc0:1234 SA08G 7.41 GiB
[ 1.647844] Alternate GPT is invalid, using primary GPT.
[ 1.653482] mmcblk0: p1 p2 p3 p4 p5 p6 p7 p8
[ 6.967369] EXT4-fs (mmcblk0p7): recovery complete
[ 6.974704] EXT4-fs (mmcblk0p7): mounted filesystem with ordered data mode. Opts: (null)
[ 6.983156] VFS: Mounted root (ext4 filesystem) on device 179:7.
[ 6.990477] Freeing unused kernel memory: 220K (80519000 - 80550000)
```

一牛网
16RD.com

After analysis, it is determined that the problem is caused by the fact that devtmpfs is not enabled in the kernel, so the kernel needs to be configured as follows:

01. | Symbol: DEVTMPFS_MOUNT [=y]
02. | Type: boolean
03. | Prompt: Automount devtmpfs at /dev, after the kernel mounted the rootfs
04. | Location:
 - 05. | -> Device Drivers
 - 06. | -> Generic Driver Options
 - 07. | -> Maintain a devtmpfs filesystem to mount at /dev (DEVTMPFS [=y]) |
08. | Defined at drivers/base/Kconfig:44 |
09. | Depends on: DEVTMPFS [=y] |

[Copy code](#)

jdi@jdi-virtual-machine: /Ingenic/X1000V6.0-Moded-20171217/kernel - □ X

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

.config - Linux/mips 3.10.14 Kernel Configuration

> Device Drivers > Generic Driver Options

Generic Driver Options

Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded <M> module < >

- () path to uevent helper
 - [] Maintain a devtmpfs filesystem to mount at /dev
 - [*] Select only drivers that don't need compile-time external fir
 - [*] Prevent firmware from being built
 - <*> Userspace firmware loading support
 - [*] Include in-kernel firmware blobs in kernel binary
 - () External firmware blobs to build into the kernel binary
 - [*] Fallback user-helper invocation for firmware loading
 - [] Driver Core verbose debug messages
 - [] Managed device resources verbose debug messages

<Select> < Exit > < Help > < Save > < Load >

16RD.com

jdi@jdi-virtual-machine: /Ingenic/X1000V6.0-Moded-20171217/kernel - □ X

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

.config - Linux/mips 3.10.14 Kernel Configuration

> Device Drivers > Generic Driver Options

Generic Driver Options

Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded <M> module < >

- () path to uevent helper
 - [*] Maintain a devtmpfs filesystem to mount at /dev
 - [*] Automount devtmpfs at /dev, after the kernel mounted the ro
 - [*] Select only drivers that don't need compile-time external fir
 - [*] Prevent firmware from being built
 - <*> Userspace firmware loading support
 - [*] Include in-kernel firmware blobs in kernel binary
 - () External firmware blobs to build into the kernel binary
 - [*] Fallback user-helper invocation for firmware loading
 - [] Driver Core verbose debug messages

<Select> < Exit > < Help > < Save > < Load >

16RD.com

After the modification is completed, recompile the kernel and program it into the target board, and prepare for the second attempt.

- NStwoSecond-ratetastety:

COM5 - PuTTY

```
[ 4.072036] VFS: Mounted root (ext4 filesystem) on device 179:7.
[ 4.086019] devtmpfs: mounted
[ 4.090233] Freeing unused kernel memory: 220K (80519000 - 80550000)
[ 4.455118] systemd[1]: systemd 215 running in system mode. (+PAM +AUDIT +SELINUX +IMA +SYSVINIT +LIBCRYPTSETUP +CRYPT +ACL +XZ -SECCOMP -APPARMOR)
[ 4.469520] systemd[1]: Detected architecture 'mips'.
Welcome to Debian GNU/Linux 8 (jessie)!

[ 4.518184] systemd[1]: Failed to insert module 'autofs4'
[ 4.524050] systemd[1]: Failed to insert module 'ipv6'
[ 4.532983] systemd[1]: Set hostname to <debian>.
[ 5.098539] systemd[1]: Cannot add dependency job for unit dbus.socket, ignoring: Unit dbus.socket failed to load: No such file or directory.
[ 5.112041] systemd[1]: Cannot add dependency job for unit display-manager.service, ignoring: Unit display-manager.service failed to load: No such file or directory.
[ 5.130439] systemd[1]: Expecting device dev-ttyS2.device...
[ 5.136455] systemd[1]: inotify_init() failed: Function not implemented
    Expecting device dev-ttyS2.device...
[ 5.157339] systemd[1]: Starting Forward Password Requests to Wall Directory Watch.
[ 5.165679] systemd[1]: systemd-ask-password-wall.path failed to enter waiting state: Function not implemented
[ 5.176287] systemd[1]: Failed to start Forward Password Requests to Wall Directory Watch.
[ 5.185036] systemd[1]: Unit systemd-ask-password-wall.path entered failed state.
[ 5.192989] systemd[1]: Starting Remote File Systems (Pre).
[ 5.198966] systemd[1]: inotify_init() failed: Function not implemented
[OK ] Reached target Remote File Systems (Pre).
[ 5.227261] systemd[1]: Reached target Remote File Systems (Pre).
[ 5.233789] systemd[1]: Starting Encrypted Volumes.
[ 5.239081] systemd[1]: inotify_init() failed: Function not implemented
[OK ] Reached target Encrypted Volumes.
[ 5.267272] systemd[1]: Reached target Encrypted Volumes.
[ 5.273108] systemd[1]: Starting Dispatch Password Requests to Console Directory Watch.
[ 5.281714] systemd[1]: systemd-ask-password-console.path failed to enter waiting state: Function not implemented
[ 5.292587] systemd[1]: Failed to start Dispatch Password Requests to Console Directory Watch.
[ 5.301673] systemd[1]: Unit systemd-ask-password-console.path entered failed state.
[ 5.309857] systemd[1]: Starting Paths.
[ 5.313958] systemd[1]: inotify_init() failed: Function not implemented
[OK ] Reached target Paths.
[ 5.337263] systemd[1]: Reached target Paths.
[ 5.342086] systemd[1]: Set up automount Arbitrary Executable File Formats File System Automount Point.
[ 5.352035] systemd[1]: Starting Swap.
[ 5.356053] systemd[1]: inotify_init() failed: Function not implemented
[OK ] Reached target Swap.
[ 5.377281] systemd[1]: Reached target Swap.
[ 5.381867] systemd[1]: Starting Root Slice.
[ 5.386564] systemd[1]: inotify_init() failed: Function not implemented
[OK ] Created slice Root Slice.
[ 5.407285] systemd[1]: Created slice Root Slice.
[ 5.412324] systemd[1]: Starting User and Session Slice.
[ 5.418373] systemd[1]: inotify_init() failed: Function not implemented
[OK ] Created slice User and Session Slice.
```

一牛网
16RD.com

According to the printed records, sort out the existing problems:

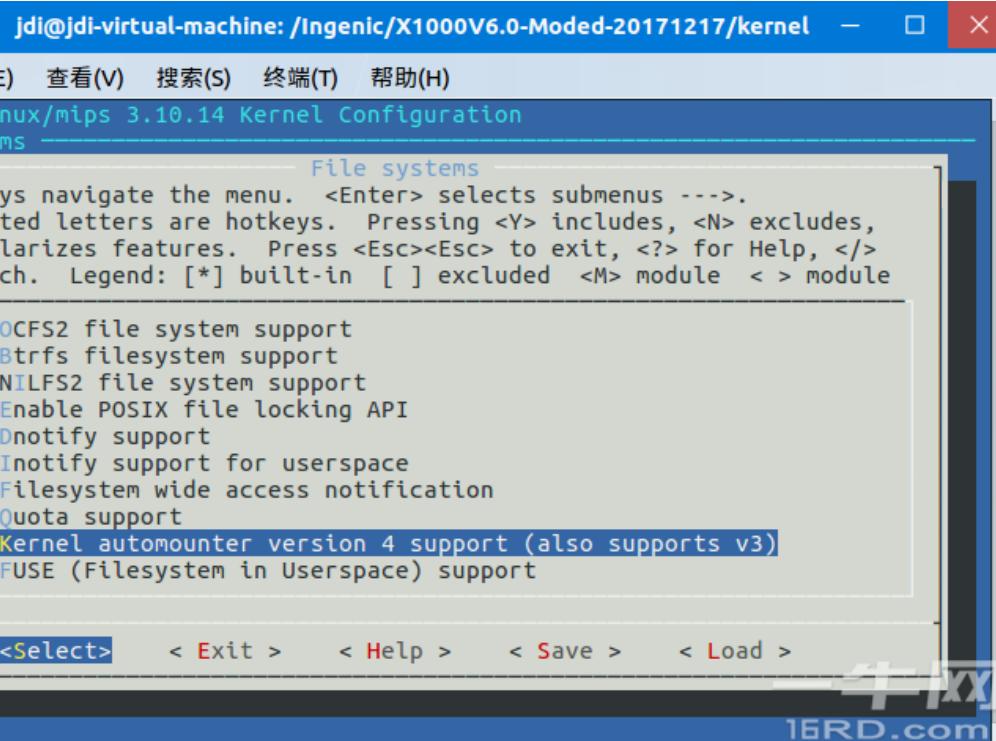
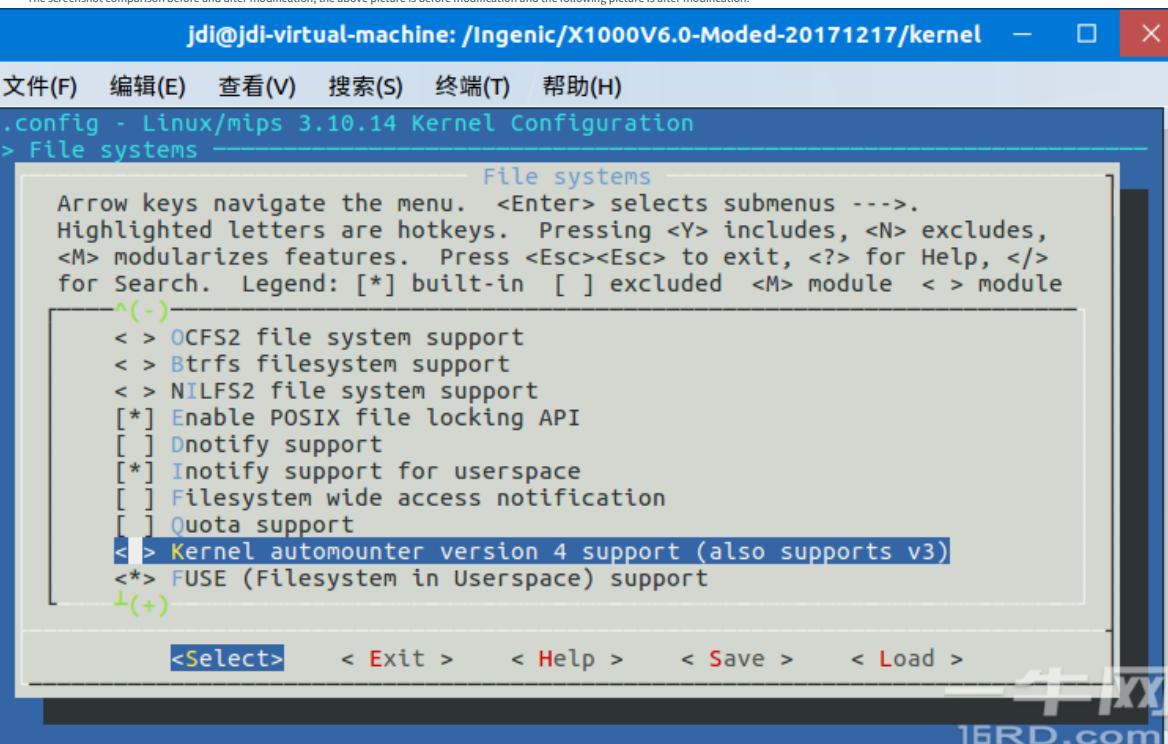
1. Problems related to autofs4;
2. Issues related to ipv6;
3. Problems related to dbus;
4. Issues related to inotify;
5. Problems related to getty (including 90 second timeout and unable to log in).

After analysis and determination, the problems related to autofs4, ipv6, and inotify are because the corresponding options are not enabled in the kernel, and the kernel needs to be configured accordingly. Problems related to dbus and getty are related to rootfs and need to be repackaged to fix the rootfs.

Based on the principle of easy first, difficult first, configure the kernel as follows to enable autofs4:

- | | | |
|-----|---|--|
| 01. | Symbol: AUTOFS4_FS [=y] | |
| 02. | Type: tristate | |
| 03. | Prompt: Kernel automounter version 4 support (also supports V3) Location: _____ | |
| 04. | | |
| 05. | -> File systems _____ | |
| 06. | Defined at fs/autofs4/Kconfig:1 | |

[Copy code](#)



Then enable ipv6, you need to configure the kernel as follows:

01. | Symbol: IPV6 [=y] |
02. Type: tristate
03. | Prompt: The IPv6 protocol
04. | Location:
05. | -> Networking support (NET [=y])
06. | -> Networking options
07. | -> TCP/IP networking (INET [=y])
08. | Defined at net/ipv6/Kconfig:6 Depends
09. | on: NET [=y] && INET [=y]

Copy code

jdi@jdi-virtual-machine: /Ingenic/X1000V6.0-Moded-20171217/kernel - X

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

.config - Linux/mips 3.10.14 Kernel Configuration
> Networking support > Networking options

Networking options

Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded <M> module < > module

^(-)
< > IP: IPsec transport mode
< > IP: IPsec tunnel mode
< > IP: IPsec BEET mode
< > Large Receive Offload (ipv4/tcp)
<*> INET: socket monitoring interface
< > UDP: socket monitoring interface
[] TCP: advanced congestion control --->
[] TCP: MD5 Signature Option support (RFC2385)
< > The IPv6 protocol --->
[*] Only allow certain groups to create sockets
L(+)

<Select> < Exit > < Help > < Save > < Load >

jdi@jdi-virtual-machine: /Ingenic/X1000V6.0-Moded-20171217/kernel - X

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

.config - Linux/mips 3.10.14 Kernel Configuration
> Networking support > Networking options

Networking options

Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded <M> module < > module

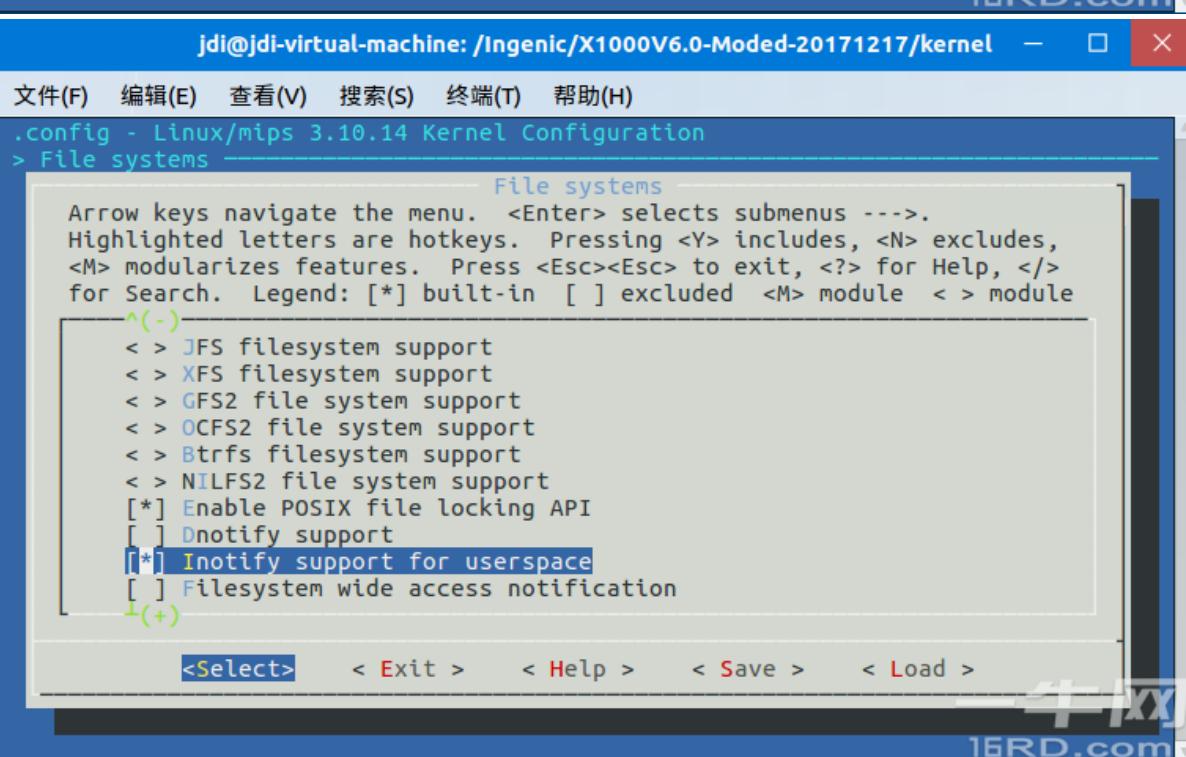
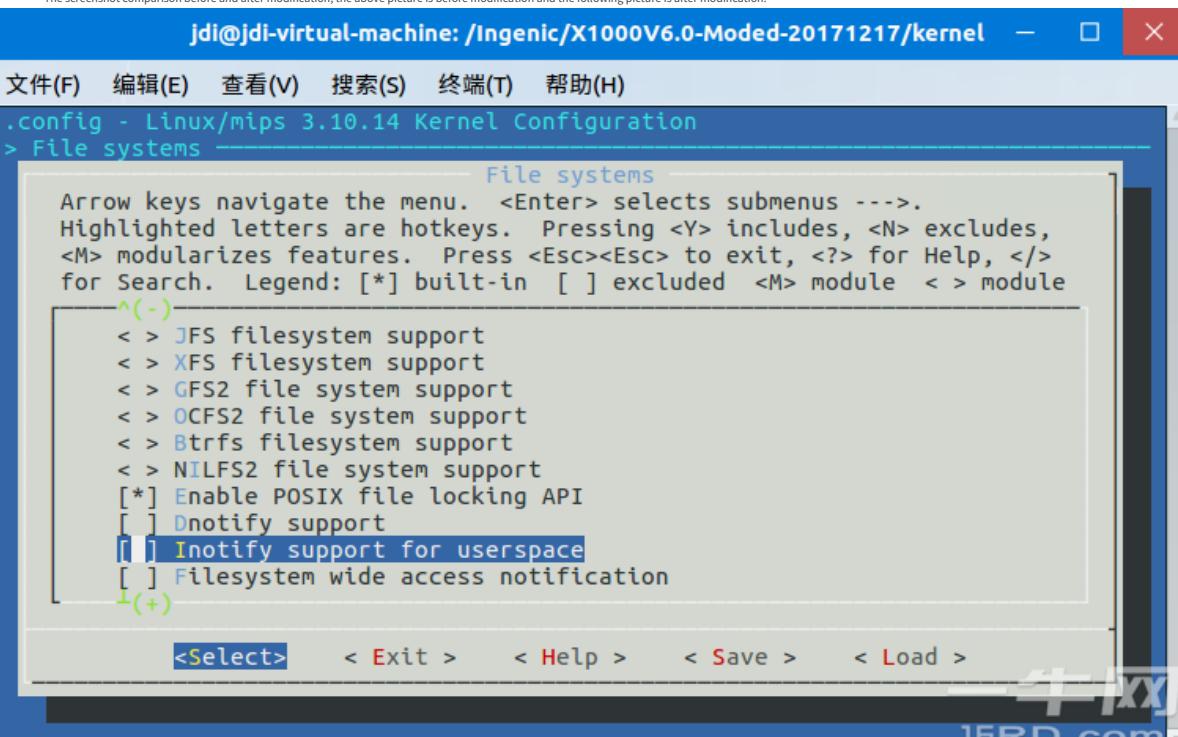
^(-)
< > IP: IPsec transport mode
< > IP: IPsec tunnel mode
< > IP: IPsec BEET mode
< > Large Receive Offload (ipv4/tcp)
<*> INET: socket monitoring interface
< > UDP: socket monitoring interface
[] TCP: advanced congestion control --->
[] TCP: MD5 Signature Option support (RFC2385)
<*> The IPv6 protocol --->
[*] Only allow certain groups to create sockets
L(+)

<Select> < Exit > < Help > < Save > < Load >

Then configure as follows to solve inotify related problems:

01. | Symbol: INOTIFY_USER [=y] |
02. Type: boolean
03. | Prompt: Inotify support for userspace |
04. Location:
05. | -> File systems | Defined at fs/notify/inotify/Kconfig:1
06. | Selects: ANON_INODES [=y] && FSNOTIFY [=y]
- 07.

[Copy code](#)



Recompile the kernel and program it to the target board, and then prepare for the third boot attempt.

● NSthreeSecond-ratetastetry:

From the print log, we can see that no errors related to autofs4, ipv6 and inotify have been reported, but a new problem has arisen: In addition to the 90-second timeout problem in the previous getty, now the wired network ifup also has a 90-second timeout.

COMS - PuTTY

```
[ OK ] Started Load/Save Random Seed.
[ OK ] Started Create Volatile Files and Directories.
[ OK ] Started Apply Kernel Variables.
[ OK ] Started Trigger Flushing of Journal to Persistent Storage.
[ 5.699079] systemd-journald[98]: Received request to flush runtime journal from PID 1
[ OK ] Created slice system-ifup.slice.
  Expecting device sys-subsystem-net-devices-eth0.device...
  Starting Update UTMP about System Boot/Shutdown...
  Starting LSB: Raise network interfaces....
  Starting Copy rules generated while the root was ro...
[ OK ] Started Copy rules generated while the root was ro.
[ OK ] Started Update UTMP about System Boot/Shutdown.
[ 6.410423] Bus Mode Reg after reset: 0x00020101, cnt=0
[ 9.529522] libphy: 0:01 - Link is Up - 100/Full
[ OK ] Started LSB: Raise network interfaces..
[ OK ] Reached target System Initialization.
[ OK ] Reached target Timers.
[ OK ] Reached target Basic System.
  Starting Regular background program processing daemon...
[ OK ] Started Regular background program processing daemon.
  Starting getty on tty2-tty6 if dbus and logind are not available...
  Starting System Logging Service...
[ OK ] Started getty on tty2-tty6 if dbus and logind are not available.
[ OK ] Started System Logging Service.
[ TIME ] Timed out waiting for device dev-ttyS2.device.
[DEPEND] Dependency failed for Serial Getty on ttyS2.
[ TIME ] Timed out waiting for device sys-subsystem-net-devices-eth0.device.
[DEPEND] Dependency failed for ifup for eth0.
[ OK ] Reached target Network.
  Starting /etc/rc.local Compatibility...
  Starting Permit User Sessions...
[ OK ] Started /etc/rc.local Compatibility.
[ OK ] Started Permit User Sessions.
  Starting Getty on tty6...
[ OK ] Started Getty on tty6.
  Starting Getty on tty5...
[ OK ] Started Getty on tty5.
  Starting Getty on tty4...
[ OK ] Started Getty on tty4.
  Starting Getty on tty3...
[ OK ] Started Getty on tty3.
  Starting Getty on tty2...
[ OK ] Started Getty on tty2.
  Starting Getty on ttym...
[ OK ] Started Getty on ttym.
[ OK ] Reached target Login Prompts.
[ OK ] Reached target Multi-User System.
[ OK ] Reached target Graphical Interface.
  Starting Update UTMP about System Runlevel Changes...
[ OK ] Started Update UTMP about System Runlevel Changes.
```

一牛网
16RD.com

The new ifup timeout problem is very similar to the previous getty timeout. After analysis, it is judged that the problem also comes out in roots.

Next, the roots of Debian 8 will be repackaged in order to fix related problems. The operation steps are as follows:

1. Repeat the previous packaging steps, but do not use exit to exit the mkdebianfs.sh script after the packaging is complete, as shown in the following figure:

jdi@jdi-virtual-machine: /Ingenic

```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

I: Configuring systemd...
I: Base system installed successfully.

Configuring target system...

Please enter a password for the root user:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Generating locales (this might take a while)...
  zh_CN.GB2312... done
  zh_CN.GB18030... done
  zh_CN.GBK... done
  zh_CN.UTF-8... done
Generation complete.

Current default time zone: 'Asia/Chongqing'
Local time is now:      Thu Jun 28 21:37:12 CST 2018.
Universal Time is now:  Thu Jun 28 13:37:12 UTC 2018.

Entering target system for additional configuration. Type 'exit' when done.
root@jdi-virtual-machine:/#
```

一牛网
16RD.com

2. Open a new terminal (hereinafter referred to as terminal 2) and enter the directory where the mkdebianfs.sh script is located, and execute the following command to mount the corresponding directory.

01. sudo mount -t proc proc debian8-rootfs/proc
 02. sudo mount --bind /dev/pts/ debian8-rootfs/dev/pts/
- [Copy code](#)

After executing the above two commands, if there is no error prompt, the mounting is successful, otherwise you can re-mount after troubleshooting the problem according to the prompts.

3. Switch back to the previously packaged terminal (hereinafter referred to as terminal 1), and enter the following commands in turn to install the required software packages to the packaged rootfs.

01. apt-get update
02. apt-get upgrade
03. apt-get install dbus vim apt-
04. get clean

[Copy code](#)

4. Create the ttyS2 configuration file required by getty to solve the problem of getty timeout and unable to log in from the serial port, execute the following command in terminal 1:

01. cp /lib/systemd/system/serial-getty@.service /etc/systemd/system/getty.target.wants/serial-getty@ttyS2.service

[Copy code](#)

After executing the above command, switch to Terminal 2 and execute the following command:

01. sudo gedit debian8-rootfs/etc/systemd/system/getty.target.wants/serial-getty@ttyS2.service

[Copy code](#)

Modify the relevant configuration in the pop-up window, the following is the original code:

01. [Unit]
02. Description=Serial Getty on %l Documentation=man:agetty(8) man:systemd-getty-generator(8) Documentation=http://0pointer.de/blog/projects/serial-console.html BindsTo=dev-%i.device
- 03.
- 04.
- 05.
06. After=dev-%i.device systemd-user-sessions.service plymouth-quit-wait.service After=rc-local.service
- 07.

[Copy code](#)

The modified code is as follows:

01. [Unit]
02. Description=Serial Getty on %l Documentation=man:agetty(8) man:systemd-getty-generator(8) Documentation=http://0pointer.de/blog/projects/serial-console.html After=systemd-user-sessions.service plymouth-quit-wait.service
- 03.
- 04.
05. After=rc-local.service
- 06.

[Copy code](#)

After the modification is completed and saved, the window can be closed. At this point, the creation and editing of the ttyS2 configuration file required by getty has been completed.

5. Create the eth0 configuration file required by ifup to solve the ifup timeout problem, execute the following commands in Terminal 1:

01. mkdir /etc/systemd/system/network.target.wants
02. cp /lib/systemd/system/ifup@.service /etc/systemd/system/network.target.wants/ifup@eth0.service

[Copy code](#)

After executing the above command, switch to Terminal 2 and execute the following command:

01. sudo gedit debian8-rootfs/etc/systemd/system/network.target.wants/ifup@eth0.service

[Copy code](#)

Modify the relevant configuration in the pop-up window, the following is the original code:

01. [Unit]

```
02. Description=ifup for %l
03. After=local-fs.target network-pre.target networking.service systemd-sysctl.service
04. Before=network.target
05. BindsTo=sys-subsystem-net-devices-%i.device
06. After=sys-subsystem-net-devices-%i.device
07. ConditionPathIsDirectory=/run/network
08. DefaultDependencies=no
```

[Copy code](#)

The modified code is as follows:

```
01. [Unit]
02. Description=ifup for %l
03. After=local-fs.target network-pre.target networking.service systemd-sysctl.service
04. Before=network.target
05. ConditionPathIsDirectory=/run/network
06. DefaultDependencies=no
```

[Copy code](#)

After the modification is completed and saved, the window can be closed. At this point, the creation and editing of the eth0 configuration file required by ifup has been completed.

6. Use exit to exit the script in Terminal 1, then switch to Terminal 2, and enter the following commands in turn to uninstall the previously mounted directory.

```
01. sudo umount debian8-rootfs/proc sudo
02. umount debian8-rootfs/dev/pts/
```

[Copy code](#)

If an error similar to the following occurs when uninstalling debian8-rootfs/proc:

```
01. umount: /Ingenic/debian8-rootfs/proc: target is busy
02.           (In some cases useful info about processes that use the
03.             device is found by lsof(8) or fuser(1).)
```

[Copy code](#)

First, make sure that the folder or file related to the directory to be uninstalled is not opened, and then use the following command to force uninstall:

```
01. sudo umount -lf debian8-rootfs/proc
```

[Copy code](#)

After the uninstallation is complete, turn off Terminal 2, follow the previous steps for making an ext4 format image to recreate the image, and then burn the new image to the target board, and perform a new round of boot attempts.

- NSFourSecond-ratetastety:

From the print log, you can see that the previous problems have been solved, and the login prompt of Debian 8 has appeared, and you can successfully log in to the system by using the root account and the password set during packaging.

```
COM5 - PuTTY
Starting Update UTMP about System Boot/Shutdown...
Starting LSB: Raise network interfaces...
Starting Copy rules generated while the root was ro...
[ OK ] Started Copy rules generated while the root was ro.
[ OK ] Started Update UTMP about System Boot/Shutdown.
[ 6.374511] Bus Mode Reg after reset: 0x00020101, cnt=0
[ 8.529501] libphy: 0:1 - Link is Up - 100/Full
[ OK ] Started LSB: Raise network interfaces..
Starting ifup for eth0...
[ OK ] Started ifup for eth0.
[ OK ] Reached target Network.
[ OK ] Reached target System Initialization.
[ OK ] Listening on D-Bus System Message Bus Socket.
[ OK ] Reached target Sockets.
[ OK ] Reached target Timers.
[ OK ] Reached target Basic System.
Starting Regular background program processing daemon...
[ OK ] Started Regular background program processing daemon.
Starting /etc/rc.local Compatibility...
Starting Login Service...
Starting D-Bus System Message Bus...
[ OK ] Started D-Bus System Message Bus.
Starting System Logging Service...
Starting Permit User Sessions...
[ OK ] Started /etc/rc.local Compatibility.
[ OK ] Started Permit User Sessions.
[ OK ] Started System Logging Service.
[ OK ] Started Login Service.
Starting Getty on ttym...
[ OK ] Started Getty on ttym.
Starting Serial Getty on ttys2...
[ OK ] Started Serial Getty on ttys2.
[ OK ] Reached target Login Prompts.
[ OK ] Reached target Multi-User System.
[ OK ] Reached target Graphical Interface.
Starting Update UTMP about System Runlevel Changes...
[ OK ] Started Update UTMP about System Runlevel Changes.

Debian GNU/Linux 8 debian ttyS2

debian login: root
密码:
Linux debian 3.10.14 #3 PREEMPT Fri Jun 29 23:31:36 CST 2018 mips

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian:~#
```

一牛网
16RD.com

After logging in to the system, you can use apt-get to install the software, where screenfetch and htop are installed.

```
COM5 - PuTTY
[ OK ] Started Regular background program processing daemon.
Starting /etc/rc.local Compatibility...
Starting Login Service...
Starting D-Bus System Message Bus...
[ OK ] Started D-Bus System Message Bus.
Starting System Logging Service...
Starting Permit User Sessions...
[ OK ] Started /etc/rc.local Compatibility.
[ OK ] Started Permit User Sessions.
[ OK ] Started System Logging Service.
[ OK ] Started Login Service.
Starting Getty on ttym...
[ OK ] Started Getty on ttym.
Starting Serial Getty on ttys2...
[ OK ] Started Serial Getty on ttys2.
[ OK ] Reached target Login Prompts.
[ OK ] Reached target Multi-User System.
[ OK ] Reached target Graphical Interface.
Starting Update UTMP about System Runlevel Changes...
[ OK ] Started Update UTMP about System Runlevel Changes.

Debian GNU/Linux 8 debian ttyS2

debian login: root
密码:
Linux debian 3.10.14 #3 PREEMPT Fri Jun 29 23:31:36 CST 2018 mips

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian:~# apt-get install screenfetch htop
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
将会安装下列额外的软件包:
glib2 libfreetype6 libgif4 libid3tag0 libimlib2 libjbig0 libjpeg62-turbo
libpng12-0 libtiff5 libxll-6 libxll-data libxau6 libxcb libxdmcp6 libxext6
scrot
建议安装的软件包:
strace ltrace
下列【新】软件包将被安装:
glib2 libfreetype6 libgif4 libid3tag0 libimlib2 libjbig0
libjpeg62-turbo libpng12-0 libtiff5 libxll-6 libxll-data libxau6 libxcb
libxdmcp6 libxext6 screenfetch scrot
升级了 0 个软件包，新安装了 18 个软件包，要卸载 0 个软件包，有 0 个软件包未被升级。
需要下载 2,256 kB 的软件包。
解压缩后会消耗掉 7,078 kB 的额外空间。
您希望继续执行吗? [Y/n]
```

一牛网
16RD.com

Run screenfetch after installation, you can identify various system information. What's more interesting is that the processor is recognized as IBM's PowerPC.

,\$\$P' `\$\$\$. Kernel: mips Linux 3.10.14
'\$\$P ,ggs. '\$\$b: Packages: 188
'\$\$` ,,\$P' . '\$\$ Shell: bash 4.3.30
\$\$P d\$\$' , '\$\$P CPU: IBM PowerPC G3 Ingenic Xburst V4.15 FPU V0.0
\$\$: \$\$` - ,d\$\$' RAM: 19MB / 57MB
\$\$\; Y\$b. ,d\$\$'
Y\$\$.:"Y\$\$\$\$\$P"
'\$\$b "-.
`Y\$\$.
`\$b.
`Y\$b.
`"Y\$b.
`"m"m"
root@debian:~#

A watermark for '16RD.com' is visible in the bottom right corner of the terminal window.

Run htop, you can view the CPU occupancy rate, memory occupancy rate, SWAP swap file occupancy rate, current running process and other information.

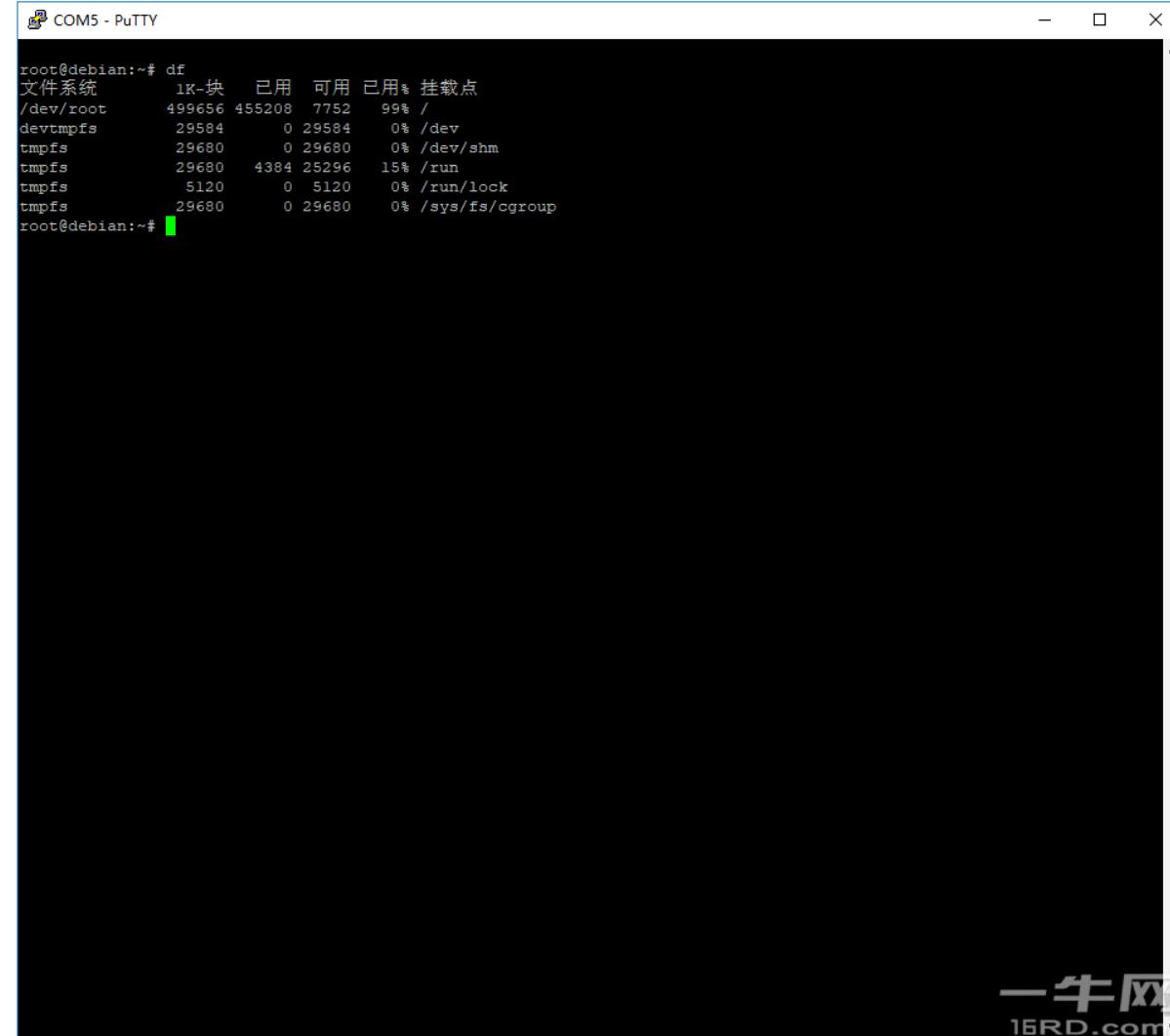
```
CPU[*]      0.9%]    Tasks: 12, 3 thr; 1 running
Mem[|||||*****17/57MB] Load average: 1.04 1.03 0.96
Swp[          0/0MB]   Uptime: 00:42:56
```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
744	root	20	0	7724	1716	1348	R	0.5	2.9	0:00.17	htop
1	root	20	0	5824	1824	940	S	0.0	3.1	0:02.33	/sbin/init
82	root	20	0	10880	1188	788	S	0.0	2.0	0:00.30	/lib/systemd/syst
101	root	20	0	8616	1420	1184	S	0.0	2.4	0:00.40	/lib/systemd/syst
165	root	20	0	9192	4840	136	S	0.0	8.2	0:00.00	dhclient -v -pf /
189	root	20	0	7684	1080	876	S	0.0	1.8	0:00.03	/usr/sbin/cron -f
191	root	20	0	3736	980	768	S	0.0	1.7	0:00.06	/lib/systemd/syst
193	messagebu	20	0	6056	1172	852	S	0.0	2.0	0:00.09	/usr/bin/dbus-dae
200	root	20	0	31948	1552	816	S	0.0	2.6	0:00.01	/usr/sbin/rsyslog
201	root	20	0	31948	1552	816	S	0.0	2.6	0:00.00	/usr/sbin/rsyslog
202	root	20	0	31948	1552	816	S	0.0	2.6	0:00.02	/usr/sbin/rsyslog
196	root	20	0	31948	1552	816	S	0.0	2.6	0:00.09	/usr/sbin/rsyslog
207	root	20	0	4740	816	696	S	0.0	1.4	0:00.01	/sbin/agetty --no
208	root	20	0	9388	1388	968	S	0.0	2.3	0:00.25	/bin/login --
226	root	20	0	7788	1508	1172	S	0.0	2.5	0:00.06	-bash

```
F1Help F2Setup F3SearchF4FilterF5Tree F6SortByF7Nice -F8Nice +F9Kill F10Quit
```



Since X1000E only has 64M memory, it is very important to set a larger SWAP swap area. But the packaged rootfs has no SWAP swap area by default, which can also be confirmed by SWAP being 0 in htop. Therefore, we will try to add a SWAP partition to the system. First, use the df command to view the currently available storage space.



```
root@debian:~# df
文件系统      1K-块  已用  可用  已用% 挂载点
/dev/root    499656 455208  7752   99% /
devtmpfs     29584      0 29584   0% /dev
tmpfs        29680      0 29680   0% /dev/shm
tmpfs        29680  4384 25296  15% /run
tmpfs        5120      0 5120   0% /run/lock
tmpfs        29680      0 29680   0% /sys/fs/cgroup
root@debian:~#
```

一牛网
16RD.com

From the returned information, we can see that the rootfs space has reached 99%, leaving only about 7MB of remaining space. Therefore, a larger rootfs image needs to be packaged. Considering that the capacity of the TF card used is large enough, it is decided to expand the image size from 512MB to 4GB, and execute the following commands in sequence:

01. dd if=/dev/zero of=debian8-rootfs.ext4 bs=1M count=4096 mkfs.ext4
02. debian8-rootfs.ext4
03. sudo mount -o loop debian8-rootfs.ext4 /mnt sudo cp
04. -a -fr debian8-rootfs/* /mnt sync
- 05.
06. sudo umount /mnt
07. sudo chmod 777 debian8-rootfs.ext4

[Copy code](#)

Burn the newly made ext4 image to the target board, but unfortunately, a kernel panic error occurs:

COM5 - PuTTY

```
[ 0.723489] icdc-d3 icdc-d3: codec icdc-d3 probe enter
[ 0.728809] write extend : sreg: 3 [0 - 4], creg: 10 sdata: 216
[ 0.736290] gpio speaker enable 0
[ 0.740087] dma dma0chan4: Channel 4 have been requested.(phy id 3,type 0x06 desc a3ecd000)
[ 0.748914] dma dma0chan5: Channel 5 have been requested.(phy id 2,type 0x06 desc a3ece000)
[ 0.757807] ingenic-alsa ingenic-alsa.0: icdc-d3-hifi <-> jz-asoc-aic-i2s mapping ok
[ 0.766457] dma dma0chan0: Channel 0 have been requested.(phy id 7,type 0x20 desc a3ee0000)
[ 0.775350] dma dma0chan1: Channel 1 have been requested.(phy id 6,type 0x20 desc a3ee1000)
[ 0.784243] ingenic-alsa ingenic-alsa.0: pcm dump dai <-> jz-asoc-pcm mapping ok
[ 0.792505] dma dma0chan7: Channel 7 have been requested.(phy id 0,type 0x04 desc a3ee2000)
[ 0.801359] ingenic-alsa ingenic-alsa.0: dmic dump dai <-> jz-asoc-dmic mapping ok
[ 0.811241] input: RatCharm Headset Jack as /devices/platform/ingenic-alsa.0/sound/card0/input0
[ 0.820979] Netfilter messages via NETLINK v0.30.
[ 0.825844] nfnl_acct: registering with nfnetlink.
[ 0.830999] ip_set: protocol 6
[ 0.834329] ip_tables: (C) 2000-2006 Netfilter Core Team
[ 0.840012] TCP: cubic registered
[ 0.843431] NET: Registered protocol family 17
[ 0.848145] Bridge firewalling registered
[ 0.853657] file system registered
[ 0.859043] using random self ethernet address
[ 0.863756] using random host ethernet address
[ 0.868671] android_usb_gadget: Mass Storage Function, version: 2009/09/11
[ 0.875831] android_usb_gadget: Number of LUNs=1
[ 0.880605] lun0: LUN: removable file: (no medium)
[ 0.886052] android_usb_gadget: android_usb ready
[ 0.891398] input: gpio-keys as /devices/platform/gpio-keys/input/input1
[ 0.898623] jz-rtc jz-rtc.0: setting system clock to 2018-07-02 15:22:27 UTC (1530544947)
[ 0.907214] ALSA device list:
[ 0.910308] #0: RatCharm
[ 0.913469] Waiting 1sec before mounting root device...
[ 1.649081] mmc0: new high speed SDHC card at address 59b4
[ 1.655271] mmcblk0: mmc0:59b4 RUNSD 29.5 GiB
[ 1.663074] Alternate GPT is invalid, using primary GPT.
[ 1.668626] mmcblk0: pl p1 p2 p3 p4 p5 p6 p7 p8
[ 1.921772] List of all partitions:
[ 1.925392] lf00      256 mtdblock0  (driver?)
[ 1.930764] lf01      3072 mtdblock1  (driver?)
[ 1.935974] lf02      12928 mtdblock2  (driver?)
[ 1.941238] b300    30981120 mmcblk0  driver: mmcblk
[ 1.946716]   b301      8192 mmcblk0p1  #E87 .E8t>%
[ 1.952681]   b302      16384 mmcblk0p2  B-E8e@_w-
[ 1.958602]   b303      16384 mmcblk0p3  N1PC@8P(E26
[ 1.964555]   b304      53248 mmcblk0p4  5-E1Mrde-<F
[ 1.970498]   b305      4096 mmcblk0p5  M-g_r3F-
[ 1.976423]   b306      102400 mmcblk0p6  *-v-S-xp
[ 1.982364]   b307      716800 mmcblk0p7  M-b@/d:-
[ 1.988286] 103:00000  2097152 mmcblk0p8  Uik-std(
[ 1.994223] No filesystem could mount root, tried: ext4
[ 1.999724] Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(179,7)
[ 2.008422] Rebooting in 3 seconds...
```

一牛网
16RD.com

According to the kernel panic's prompt, after analysis, the 4GB rootfs size should exceed the allowable size of the partition table. In order to solve this problem, it is necessary to modify the files related to the partition configuration in u-boot. Modify board/ingenic/RatCharm/partitions.tab in the u-boot directory, the following is the original code:

```
01. property:
02.     disk_size = 4096m
03.     gpt_header_lba = 512
04.     custom_signature = 0
05.
06. partition:
07.     # name      = start,      size, fstype
08.     xboot      = 0m,        3m,
09.     boot       = 3m,        8m, EMPTY
10.     recovery   = 12m,       16m, EMPTY
11.     pretest    = 28m,       16m, EMPTY
12.     reserved   = 44m,       52m, EMPTY
13.     misc       = 96m,       4m, EMPTY
14.     cache      = 100m,      100m, LINUX_FS
15.     system     = 200m,      700m, LINUX_FS
16.     data       = 900m,      2048m, LINUX_FS
17.
18. # fstype could be: LINUX_FS, FAT_FS, EMPTY
Copy code
```

The modified code is as follows:

```
01. property:
02.     disk_size = 8192m
03.     gpt_header_lba = 512
```

```
04.     custom_signature=0
05.
06. partition:
07.     # name      = start,    size, fstype
08.     xboot      = 0m,      3m,
09.     boot       = 3m,      8m, EMPTY
10.     recovery   = 12m,     16m, EMPTY
11.     pretest    = 28m,     16m, EMPTY
12.     reserved   = 44m,     52m, EMPTY
13.     misc       = 96m,     4m, EMPTY
14.     cache      = 100m,    100m, LINUX_FS
15.     system     = 200m,    5000m, LINUX_FS
16.     data       = 5200m,   2048m, LINUX_FS
17.
18. # fstype could be: LINUX_FS, FAT_FS, EMPTY
```

[Copy code](#)

After modification, recompile u-boot (be sure to clean up with make distclean before compiling) and download it to the target board, and then try to boot the system again.

● NSfiveSecond-ratetastety:

After logging in to the system, use the df command to view the disk usage. You can see that the current usage rate is only 12%, and the remaining space is about 3GB.

```
Starting ifup for eth0...
[ OK ] Started ifup for eth0.
[ OK ] Reached target Network.
[ OK ] Reached target System Initialization.
[ OK ] Listening on D-Bus System Message Bus Socket.
[ OK ] Reached target Sockets.
[ OK ] Reached target Timers.
[ OK ] Reached target Basic System.
Starting Regular background program processing daemon...
[ OK ] Started Regular background program processing daemon.
Starting /etc/rc.local Compatibility...
Starting Login Service...
Starting D-Bus System Message Bus...
[ OK ] Started D-Bus System Message Bus.
Starting System Logging Service...
Starting Permit User Sessions...
[ OK ] Started /etc/rc.local Compatibility.
[ OK ] Started Permit User Sessions.
[ OK ] Started System Logging Service.
[ OK ] Started Login Service.
Starting Getty on tty1...
[ OK ] Started Getty on tty1.
Starting Serial Getty on ttyS2...
[ OK ] Started Serial Getty on ttyS2.
[ OK ] Reached target Login Prompts.
[ OK ] Reached target Multi-User System.
[ OK ] Reached target Graphical Interface.
Starting Update UTMP about System Runlevel Changes...
[ OK ] Started Update UTMP about System Runlevel Changes.

Debian GNU/Linux 8 debian tty2

debian login: root
密码:
Linux debian 3.10.14 #4 PREEMPT Wed Jul 4 00:31:47 CST 2018 mips

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian:~# df
文件系统      1K-块  已用  可用  已用% 挂载点
/dev/root    3997376 417572 3353708  12% /
devtmpfs      29552    0  29552   0% /dev
tmpfs        29672    0  29672   0% /dev/shm
tmpfs        29672   4372  25300  15% /run
tmpfs        5120     0   5120   0% /run/lock
tmpfs        29672    0  29672   0% /sys/fs/cgroup
root@debian:~#
```

一牛网
16RD.com

Next, prepare to set up the SWAP exchange area, enter the following commands in sequence:

01. mkdir /opt/swap
02. dd if=/dev/zero of=/opt/swap/swapfile bs=1M count=256
03. mkswap /opt/swap/swapfile
04. chmod 0600 /opt/swap/swapfile

[Copy code](#)

After configuring the swapfile file, we also need to add the corresponding command in /etc/fstab, enter the following command:

01. vim /etc/fstab

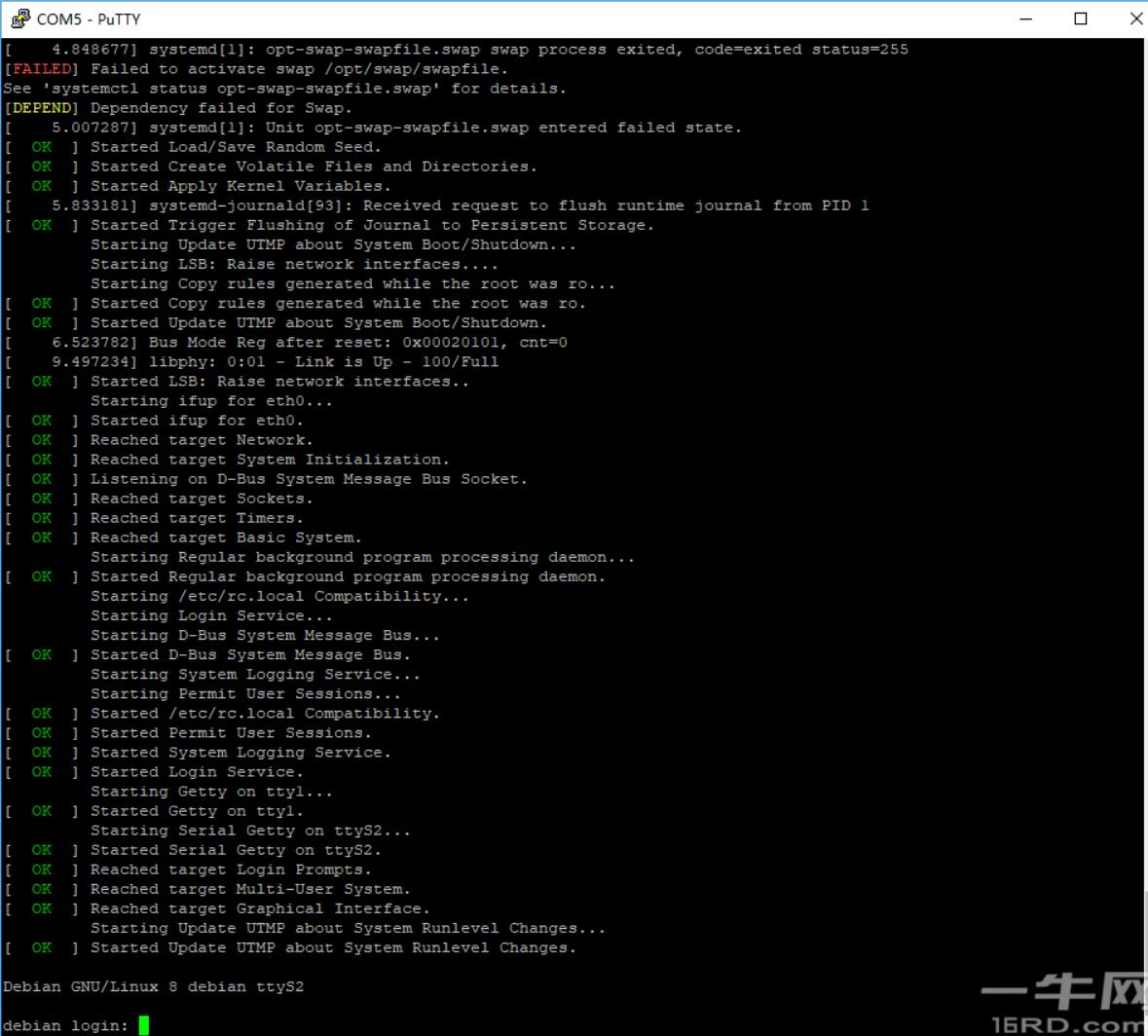
[Copy code](#)

Add the following command at the end of the edit box, then save and exit and restart the target board:

01. /opt/swap/swapfile swap swap defaults 0 0

[Copy code](#)

After restarting the target board, the following error occurred during the startup process:



```
[ 4.848677] systemd[1]: opt-swap-swapfile.swap swap process exited, code=exited status=255
[FAILED] Failed to activate swap /opt/swap/swapfile.
See 'systemctl status opt-swap-swapfile.swap' for details.
[DEPEND] Dependency failed for Swap.
[ 5.007287] systemd[1]: Unit opt-swap-swapfile.swap entered failed state.
[ OK ] Started Load/Save Random Seed.
[ OK ] Started Create Volatile Files and Directories.
[ OK ] Started Apply Kernel Variables.
[ 5.833181] systemd-journald[93]: Received request to flush runtime journal from PID 1
[ OK ] Started Trigger Flushing of Journal to Persistent Storage.
    Starting Update UTMP about System Boot/Shutdown...
    Starting LSB: Raise network interfaces....
    Starting Copy rules generated while the root was ro...
[ OK ] Started Copy rules generated while the root was ro.
[ OK ] Started Update UTMP about System Boot/Shutdown.
[ 6.523782] Bus Mode Reg after reset: 0x00020101, cnt=0
[ 9.497234] libphy: 0:0l - Link is Up - 100/Full
[ OK ] Started LSB: Raise network interfaces..
    Starting ifup for eth0...
[ OK ] Started ifup for eth0.
[ OK ] Reached target Network.
[ OK ] Reached target System Initialization.
[ OK ] Listening on D-Bus System Message Bus Socket.
[ OK ] Reached target Sockets.
[ OK ] Reached target Timers.
[ OK ] Reached target Basic System.
    Starting Regular background program processing daemon...
[ OK ] Started Regular background program processing daemon.
    Starting /etc/rc.local Compatibility...
    Starting Login Service...
    Starting D-Bus System Message Bus...
[ OK ] Started D-Bus System Message Bus.
    Starting System Logging Service...
    Starting Permit User Sessions...
[ OK ] Started /etc/rc.local Compatibility.
[ OK ] Started Permit User Sessions.
[ OK ] Started System Logging Service.
[ OK ] Started Login Service.
    Starting Getty on ttym...
[ OK ] Started Getty on ttym.
    Starting Serial Getty on ttys2...
[ OK ] Started Serial Getty on ttys2.
[ OK ] Reached target Login Prompts.
[ OK ] Reached target Multi-User System.
[ OK ] Reached target Graphical Interface.
    Starting Update UTMP about System Runlevel Changes...
[ OK ] Started Update UTMP about System Runlevel Changes.

Debian GNU/Linux 8 debian ttyS2
debian login: [REDACTED]
```

一牛网
16RD.com

After logging in to the system and using the htop command, you can see that the size of the SWAP swap area is still 0.

COMS - PuTTY

```
CPU[*]          0.9%]      Tasks: 12, 3 thr; 1 running
Mem[|||||||##*****15/57MB] Load average: 0.91 0.37 0.14
Swap[           0/0MB]     Uptime: 00:01:53

PID USER      PRI  NI   VIRT   RES   SHR S CPU% MEM%   TIME+  Command
236 root      20   0  7724 1716 1348 R  0.5  2.9  0:00.17 htop
  1 root      20   0  5792 2948 2064 S  0.0  5.0  0:02.16 /sbin/init
  79 root     20   0 10892 1488 1108 S  0.0  2.5  0:00.30 /lib/systemd/syst
  93 root     20   0  8616 1632 1396 S  0.0  2.7  0:00.37 /lib/systemd/syst
 169 root     20   0  9192 5000  300 S  0.0  8.4  0:00.00 dhclient -v -pf /
 193 root     20   0  7684 1168  964 S  0.0  2.0  0:00.02 /usr/sbin/cron -f
 195 root     20   0  3736 1352 1140 S  0.0  2.3  0:00.04 /lib/systemd/syst
 197 messagebu 20   0  6056 1900 1588 S  0.0  3.2  0:00.08 /usr/bin/dbus-dae
 204 root     20   0 31964 2328 1604 S  0.0  3.9  0:00.01 /usr/sbin/rsyslog
 205 root     20   0 31964 2328 1604 S  0.0  3.9  0:00.00 /usr/sbin/rsyslog
 206 root     20   0 31964 2328 1604 S  0.0  3.9  0:00.02 /usr/sbin/rsyslog
 200 root     20   0 31964 2328 1604 S  0.0  3.9  0:00.08 /usr/sbin/rsyslog
 213 root     20   0  4740  884  760 S  0.0  1.5  0:00.01 /sbin/agetty --no
 214 root     20   0  9388 1868 1444 S  0.0  3.1  0:00.22 /bin/login --
 232 root     20   0  7788 1996 1668 S  0.0  3.4  0:00.04 -bash

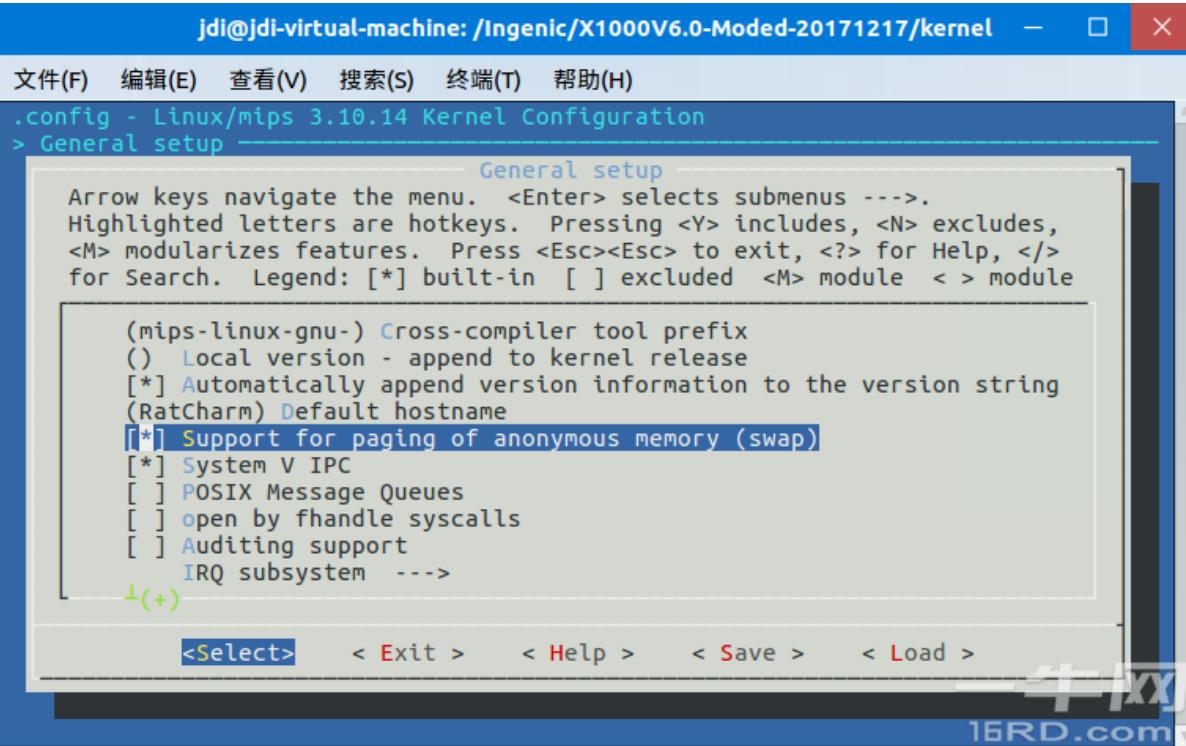
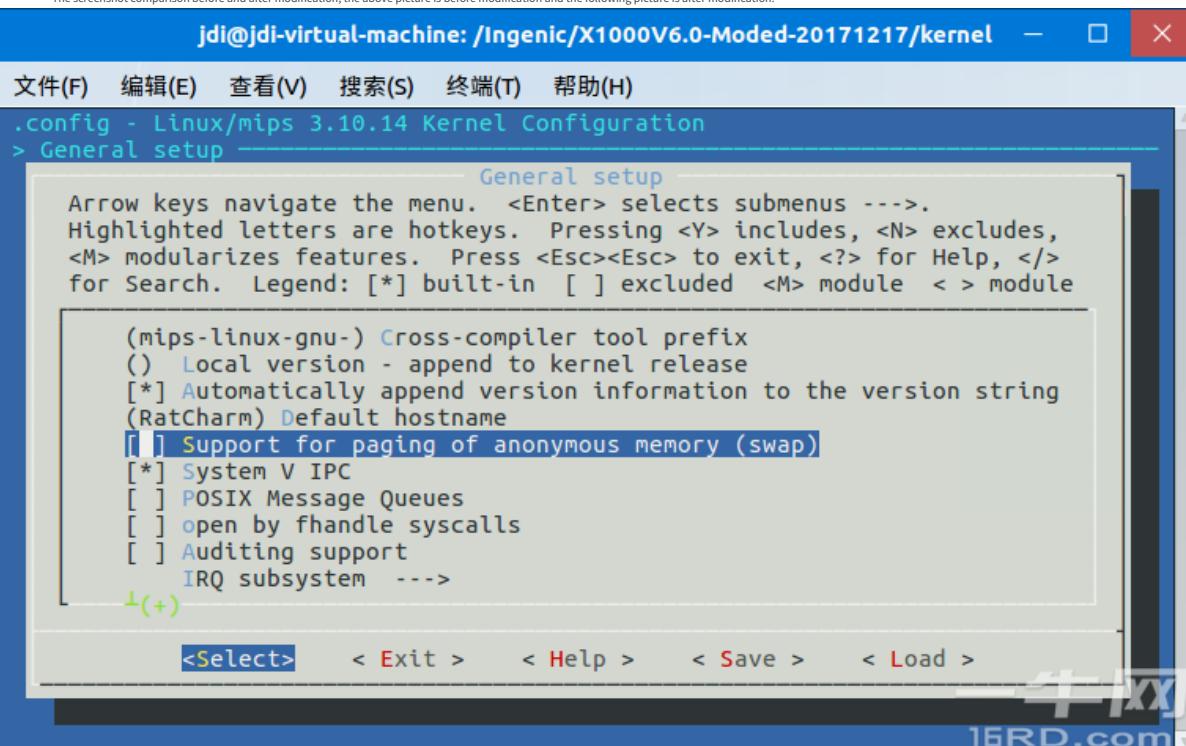
F1Help F2Setup F3SearchF4FilterF5Tree F6SortByF7Nice -F8Nice +F9Kill F10Quit
```

一牛网
16RD.com

After analyzing the cause, it is determined that there is no support for swap in the kernel, and the following configuration is required:

01. | Symbol: SWAP [=y]
02. | Type: boolean
03. | Prompt: Support for paging of anonymous memory ([swap](#)) |
04. Location:
05. | -> General setup | Defined at init/
06. Kconfig:182 | Depends on: MMU [=y] &&
07. BLOCK [=y]

[Copy code](#)



Recompile the kernel and program it to the target board, ready to boot again.

● NSsixSecond-ratetastetry:

The problem of mounting errors in the SWAP exchange area has been resolved. After logging in to the system, use the htop command. It can be seen that the SWAP has been mounted normally, with a total capacity of 255MB and a current usage of 0.

The screenshot shows a PuTTY session titled "COMS - PuTTY". The terminal displays system status information at the top, including CPU usage (0.9%), memory usage (15/57MB), and swap usage (0/255MB). Below this is a detailed tasklist from the htop command, showing processes like /sbin/init, dhclient, cron, and various daemons. At the bottom of the terminal, there is a menu bar with options F1 through F10.

一牛网
16RD.com

The above operation of setting SWAP partition can also be carried out in the link of packaging roots. Before using exit to exit the packaging script, use the same method to complete the previous operations one by one, and then use exit to exit the script and make an ext4 mirror.

So far, the basic functions of Debian 8 have been able to run normally, and the rest is some fragmentary aftermath work.

mostRear,returnHaveonesomesomemiscellaneoussevenmiscellaneousEightofgoodRearworkdo,ListNSoneindivualclearone:

- adddaddgeneralPassuseHousehold:Many operations are not suitable for the root account, so you need to create an ordinary account.
- AssumeSetsudoMutuallyshutmatchSet:Many operations under ordinary users require sudo to temporarily elevate permissions, so relevant configurations are required.
- installPackoftenuseofsoftPiecesBag:Install some software that is often used in daily use.

Next, let's gradually improve the above content. First, add ordinary users, log in to the root account using the serial port, and then execute the following commands:

01. adduser rat

[Copy code](#)

Then enter the login password, personal information, etc. (optional) according to the prompts. After successful completion, an ordinary account named rat will be established. Then execute the following command:

01. su rat

02. nano /home/rat/.bashrc

[Copy code](#)

Then add the following at the end of the edit box:

01. export PATH=\$PATH:/sbin

[Copy code](#)

Save and exit. At this time, we have completed all the configuration of the rat account, and the rat account will be available when you log in next time.

After configuring the ordinary account of rat, it is natural to configure sudo, a command that is often used in ordinary accounts. The configuration process still needs to be performed under the root account. After logging in to the root account, execute the following commands in sequence:

01. apt-get install sudo

02. chown root:root /usr/bin/sudo

03. chmod 4755 /usr/bin/sudo

[Copy code](#)

Next, you need to modify the two files /etc/sudoers and /etc/hosts. First modify /etc/sudoers, execute the following command:

01. vim /etc/sudoers

[Copy code](#)

Then make changes in the edit box, the following is the original code:

01. # User privilege specification

02. root ALL=(ALL:ALL) ALL

[Copy code](#)

The modified code is as follows:

01. # User privilege specification

02. root ALL=(ALL:ALL) ALL

03. rat ALL=(ALL:ALL) ALL

[Copy code](#)

Save the file after completing the above modification. It should be noted that because it is a read-only file, you need to use wq when saving! The command is forced to save. Then come to modify /etc/hosts, execute the following command:

01. vim /etc/hosts

[Copy code](#)

Modify in the edit box, the following is the original code

01. 127.0.0.1 localhost

02. ::1 localhost ip6-localhost ip6-loopback ip6-

03. ff02::1 allnodes

04. ff02::2 ip6-allrouters

[Copy code](#)

The modified code is as follows:

01. 127.0.0.1 localhost debian

02. ::1 localhost ip6-localhost ip6-loopback ip6-

03. ff02::1 allnodes

04. ff02::2 ip6-allrouters

[Copy code](#)

Save and exit the edit box after finishing the modification. At this point, all configurations of sudo have been completed.

Now that Debian 8 is running, it is natural to make good use of its software ecology. You can use the following commands to install common software packages under the root account:

01. apt-get install ssh build-essential gfortran openjdk-7-jdk

[Copy code](#)

After completing the installation, you can locally compile some simple programs on the target board (complex programs can also be compiled, but it will take a long time), and you can log in to the target board through the network using ssh to operate or use scp to transfer files. Increased flexibility. All the above aftermath work can also be carried out in the link of packaging rootfs. Before using exit to exit the packaging script, use the same method to complete the above operations one by one, and then use exit to exit the script and make an ext4 mirror. In this way, what you get is the configuration. Mirror files of common users, sudo, and common software packages are included.

● AttachedsuperiorbuildchangegoodofbeatBagrootfsPlaceneedoffootBook,byandmostendoverbecomeofkernelmirrorpicturewithrootfsmirrorpicture:

The modified script used to package Debian8 rootfs:



[mkdebianrfs.zip \(2.65 KB, download times: 1\)](#)

The final version of u-boot, kernel and rootfs mirrors:

 u-boot-with-spl-mbr-gpt.zip (95.18 KB, download times: 2)

 ulimage.zip (2.58 MB, download times: 2)

 debian8-rootfs.zip (309.78 MB, download times: 2)