Run **RT-THREAD** using **QEMU**

RT-THREAD Document Center

WWW.RT-THREAD.ORG

Friday 28th September, 2018

Table of contents

This application note describes how to use QEMU to run

Run the RT-Thread qemu-vexpress-a9 BSP project.

1 Purpose and structure of this paper

1.1 Purpose and Background of this Paper

Embedded software development requires a development board. If a physical development board isn't available, you can use a virtual machine like QEMU to simulate it. QEMU is a cross-platform virtualization virtual machine that can simulate many development boards. To facilitate the experience of RT-Thread without a development board, RT-Thread provides a board support package (BSP) for the ARM vExpress A9 development board, which is emulated by QEMU. This article describes how to run the RT-Thread qemu-vExpress-a9 BSP project on Windows and Ubuntu platforms using QEMU, and explains how to connect QEMU to a network using a virtual network card.

1.2 Structure of this paper

This article first introduces the usage details of qemu-vexpress-a9 BSP on the Windows platform, and then introduces the usage details on the Ubuntu platform.

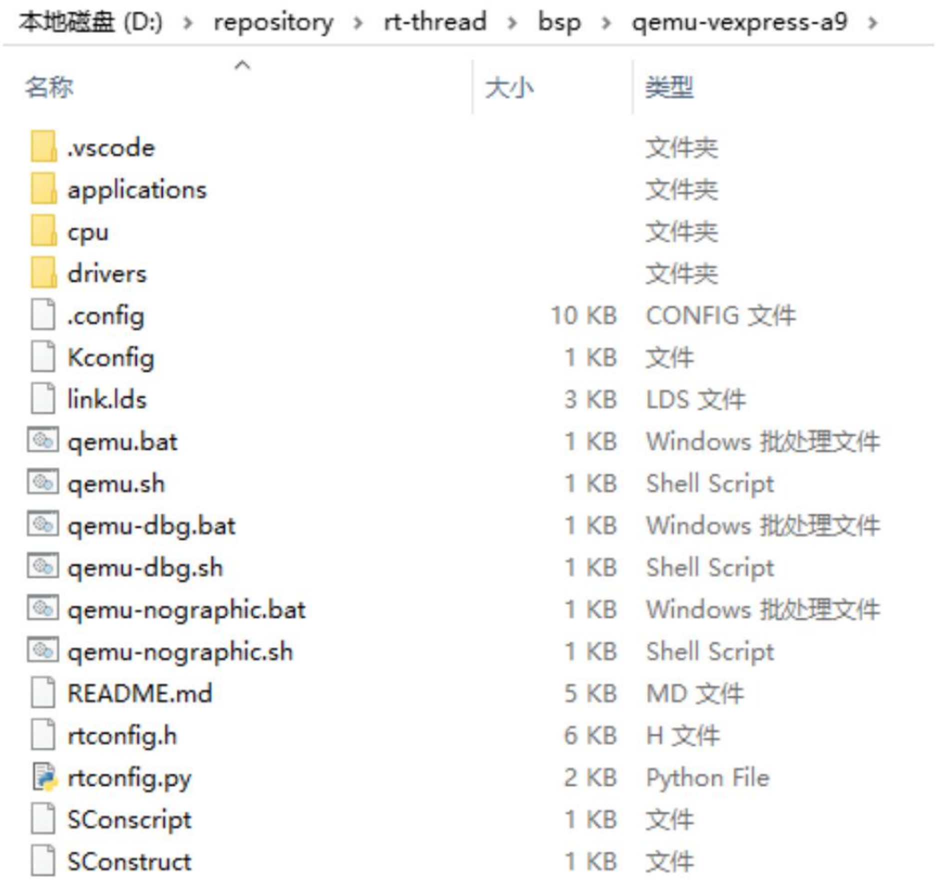
2. Run **RT-Thread** using **QEMU** on **Windows**

2.1 Preparation

- Download RT-Thread Source code, it is recommended to download version 3.1.0 or above.

- Download RT-Thread Env Tools, it is recommended to download version 1.0.0 or above.

The board support package (BSP) for the ARM vexpress A9 development board simulated by QEMU provided by RT-Thread is located in the qemu-vexpress-a9 folder under the RT-Thread source code BSP directory. This BSP implements related drivers such as LCD, keyboard, mouse, SD card, Ethernet card, and serial port. The folder contents are shown in the figure below.



本地磁盘 (D:) › repository › rt-thread › bsp › qemu-vexpress-a9 ›

名称	大小	类型
.vscode		文件夹
applications		文件夹
cpu		文件夹
drivers		文件夹
.config	10 KB	CONFIG 文件
Kconfig	1 KB	文件
link.lds	3 KB	LDS 文件
qemu.bat	1 KB	Windows 批处理文件
qemu.sh	1 KB	Shell Script
qemu-dbg.bat	1 KB	Windows 批处理文件
qemu-dbg.sh	1 KB	Shell Script
qemu-nographic.bat	1 KB	Windows 批处理文件
qemu-nographic.sh	1 KB	Shell Script
README.md	5 KB	MD 文件
rtconfig.h	6 KB	H 文件
rtconfig.py	2 KB	Python File
SConscript	1 KB	文件
SConstruct	1 KB	文件

Figure **1:** *qemu-vexpress-a9* Folder

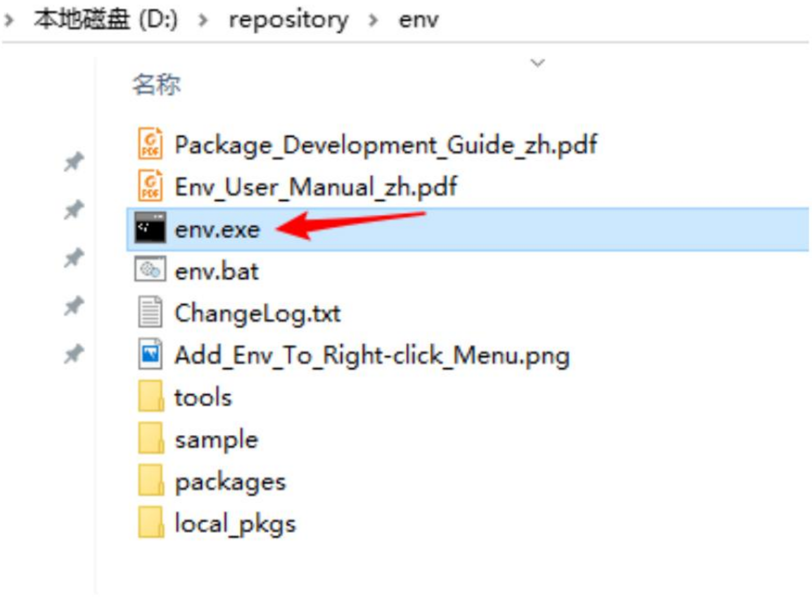
The main files and directories of qemu-vexpress-a9 BSP are described as follows:

Files/Directories	describe
.vscode	vscode configuration files
applications	User application code directory
cpu	Chip related
drivers	The underlying driver provided by RT-Thread
qemu.bat	Running script files on Windows platforms
qemu.sh	Running script files on Linux platforms
qemu-dbg.bat	Windows platform debugging script file
qemu-dbg.sh	Linux platform debugging script file
README.md	BSP documentation
rtconfig.h	BSP configuration header file

2.2 Compile and run RT-Thread

2.2.1. Step 1: Use the **scons** command to compile the project

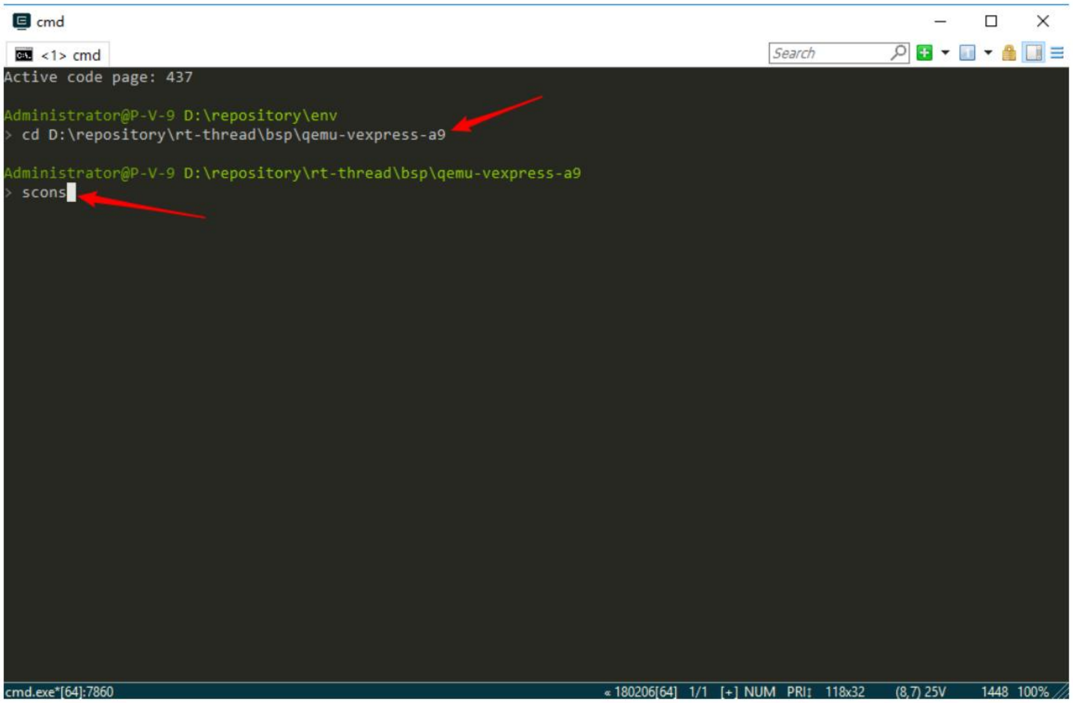
Open the Env folder and double-click the env.exe file to open the Env console:



Package_Development_Guide_zh.pdf
Env_User_Manual_zh.pdf
env.exe
env.bat
ChangeLog.txt
Add_Env_To_Right-click_Menu.png
tools
sample
packages
local_pkgs

Figure **2:** *Env* Folder

Switch directories in the Env console and enter the command cd D:\repository\rt-thread\bsp\qemu-vexpress -a9 Switch to the qemu-vexpress-a9 BSP root directory under the RT-Thread source code folder, and then enter the scons command Compile the project. If the compilation is correct, the rtthread.elf target running under QEMU will be generated in the BSP directory. document.

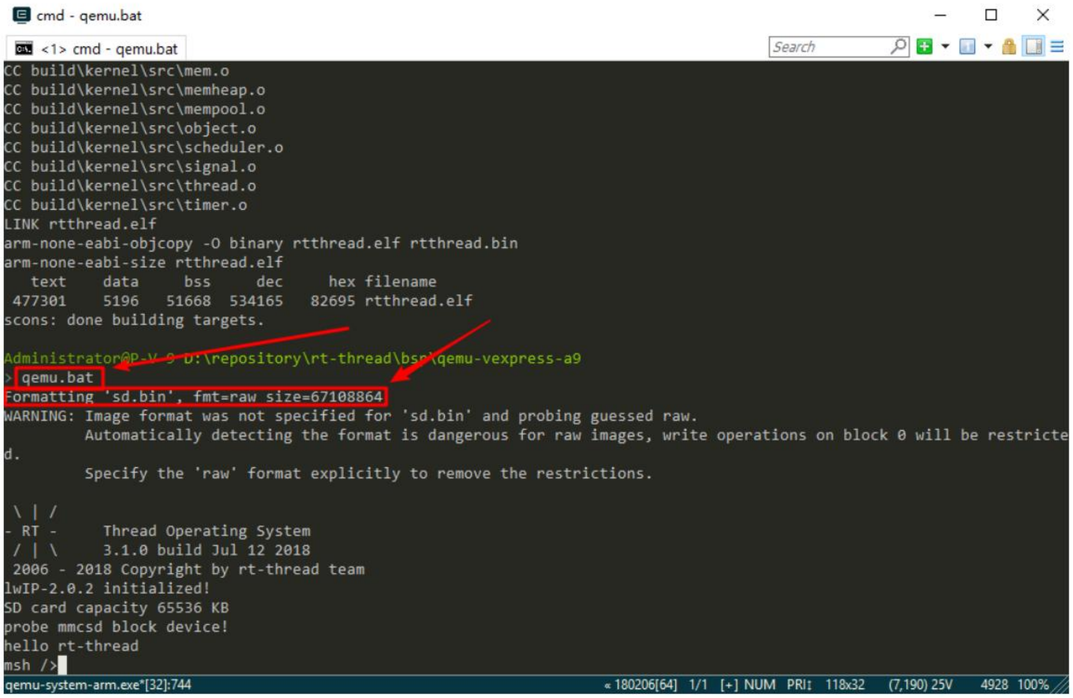


```
Active code page: 437

Administrator@P-V-9 D:\repository\env
> cd D:\repository\rt-thread\bsp\qemu-vexpress-a9

Administrator@P-V-9 D:\repository\rt-thread\bsp\qemu-vexpress-a9
> scons
```

Figure 3: Compile project

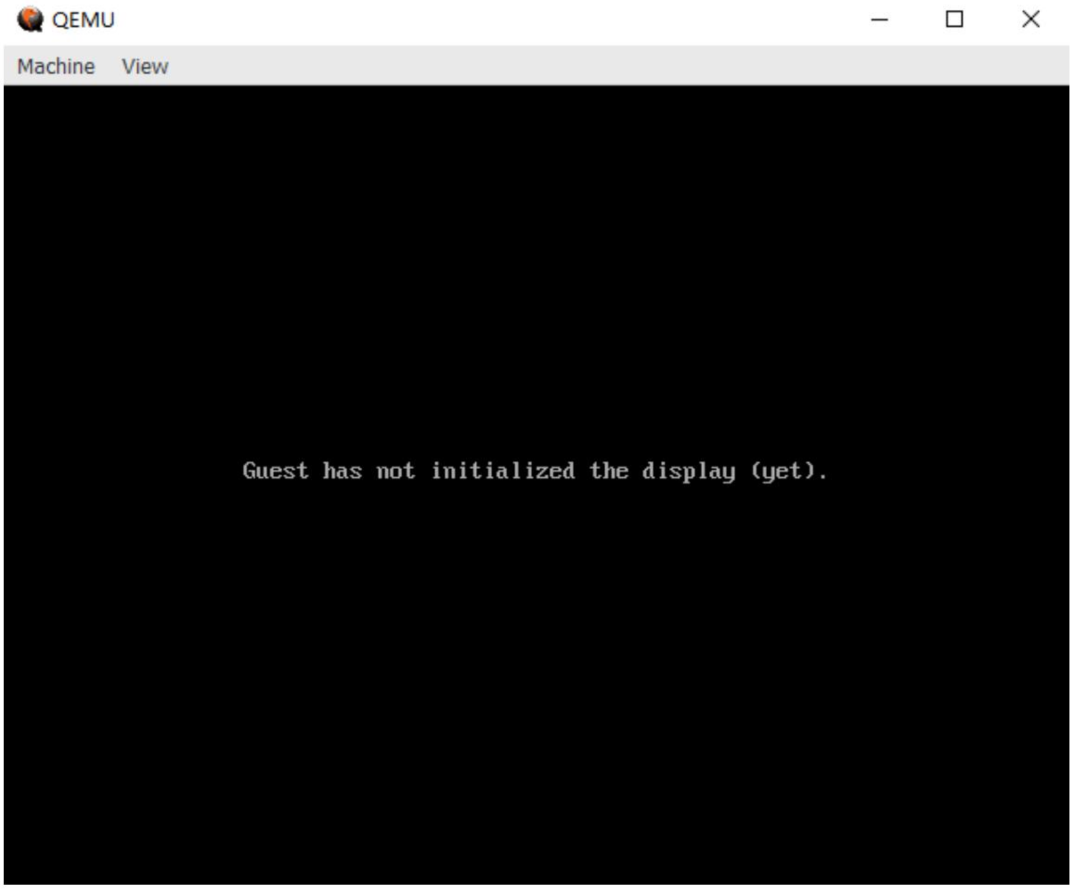
2.2.2. Step 2: Run the project using the **qemu.bat** command

After the compilation is complete, enter qemu.bat to start the virtual machine and BSP project. qemu.bat is a Window batch file. This file is located in the BSP folder and mainly includes QEMU execution instructions. The first time you run the project, it will be in the BSP folder. Create a blank sd.bin file in the folder. This is a virtual SD card with a size of 64M. The Env command line interface displays The initialization information and version number information printed during the startup of the RT-Thread system are displayed, and the qemu virtual machine is also running. As shown in the following picture:


cmd - qemu.bat
<1> cmd - qemu.bat
Search
qemu-system-arm.exe*[32]:744
« 180206[64] 1/1 [+] NUM PRI: 118x32 (7,190) 25V 4928 100%

```
CC build\kernel\src\mem.o
CC build\kernel\src\memheap.o
CC build\kernel\src\mempool.o
CC build\kernel\src\object.o
CC build\kernel\src\scheduler.o
CC build\kernel\src\signal.o
CC build\kernel\src\thread.o
CC build\kernel\src\timer.o
LINK rtthread.elf
arm-none-eabi-objcopy -O binary rtthread.elf rtthread.bin
arm-none-eabi-size rtthread.elf
   text    data     bss     dec     hex filename
 477301    5196   51668  534165   82695 rtthread.elf
scons: done building targets.

Administrator@P-V-9 D:\repository\rt-thread\bsp\qemu-vexpress-a9
> qemu.bat
Formatting 'sd.bin', fmt=raw size=67108864
WARNING: Image format was not specified for 'sd.bin' and probing guessed raw.
         Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricte
d.
         Specify the 'raw' format explicitly to remove the restrictions.

 \ | /
- RT -     Thread Operating System
 / | \     3.1.0 build Jul 12 2018
 2006 - 2018 Copyright by rt-thread team
lwIP-2.0.2 initialized!
SD card capacity 65536 KB
probe mmcsd block device!
hello rt-thread
msh />
```

Figure 4: Run the project



Guest has not initialized the display (yet).

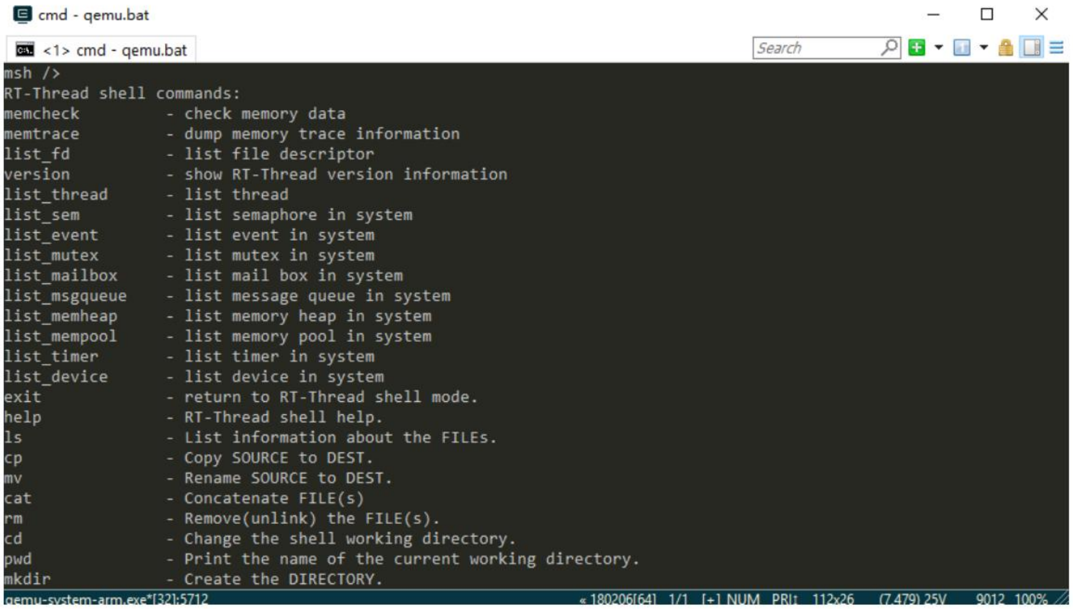
Figure 5: Virtual Machine

Note: If 360 Security Guard is installed on your computer, there will be a warning. Please click to allow the program to run.

2.3 Run **RT-Thread Finsh** Console

RT-Thread supports Finsh, and users can use command operations in command line mode. Enter help or press the tab key

You can view all supported commands. As shown in the figure below, the left side shows the command and the right side shows the command description.



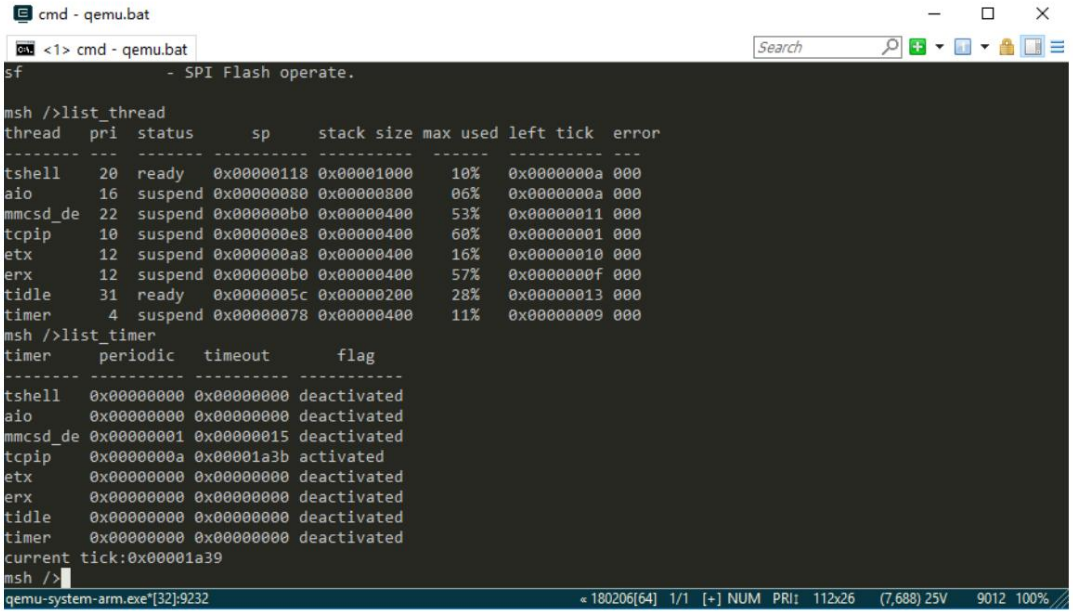
```
msh />
RT-Thread shell commands:
memcheck         - check memory data
memtrace         - dump memory trace information
list_fd          - list file descriptor
version          - show RT-Thread version information
list_thread      - list thread
list_sem         - list semaphore in system
list_event       - list event in system
list_mutex       - list mutex in system
list_mailbox     - list mail box in system
list_msgqueue    - list message queue in system
list_memheap     - list memory heap in system
list_mempool     - list memory pool in system
list_timer       - list timer in system
list_device      - list device in system
exit             - return to RT-Thread shell mode.
help             - RT-Thread shell help.
ls               - List information about the FILEs.
cp               - Copy SOURCE to DEST.
mv               - Rename SOURCE to DEST.
cat              - Concatenate FILE(s)
rm               - Remove(unlink) the FILE(s).
cd               - Change the shell working directory.
pwd              - Print the name of the current working directory.
mkdir            - Create the DIRECTORY.
```

qemu-system-arm.exe*[32]:5712
« 180206[64] 1/1 [+] NUM PRI↕ 112x26 (7,479) 25V 9012 100%

Figure 6: Check *Finsh* Order

As shown in the figure below, for example, you can enter the list_thread command to view the currently running threads, as well as the thread status and stack

Stack size, etc. Enter list_timer to view the status of the timer.



```
sf               - SPI Flash operate.

msh />list_thread
thread   pri  status      sp     stack size max used left tick  error
-------- ---  ------- ---------- ----------  ------  ---------- ---
tshell    20  ready   0x00000118 0x00001000    10%   0x0000000a 000
aio       16  suspend 0x00000080 0x00000800    06%   0x0000000a 000
mmcsd_de  22  suspend 0x000000b0 0x00000400    53%   0x00000011 000
tcpip     10  suspend 0x000000e8 0x00000400    60%   0x00000001 000
etx       12  suspend 0x000000a8 0x00000400    16%   0x00000010 000
erx       12  suspend 0x000000b0 0x00000400    57%   0x0000000f 000
tidle     31  ready   0x0000005c 0x00000200    28%   0x00000013 000
timer      4  suspend 0x00000078 0x00000400    11%   0x00000009 000
msh />list_timer
timer     periodic   timeout       flag
-------- ---------- ---------- -----------
tshell   0x00000000 0x00000000 deactivated
aio      0x00000000 0x00000000 deactivated
mmcsd_de 0x00000001 0x00000015 deactivated
tcpip    0x0000000a 0x00001a3b activated
etx      0x00000000 0x00000000 deactivated
erx      0x00000000 0x00000000 deactivated
tidle    0x00000000 0x00000000 deactivated
timer    0x00000000 0x00000000 deactivated
current tick:0x00001a39
msh />
```

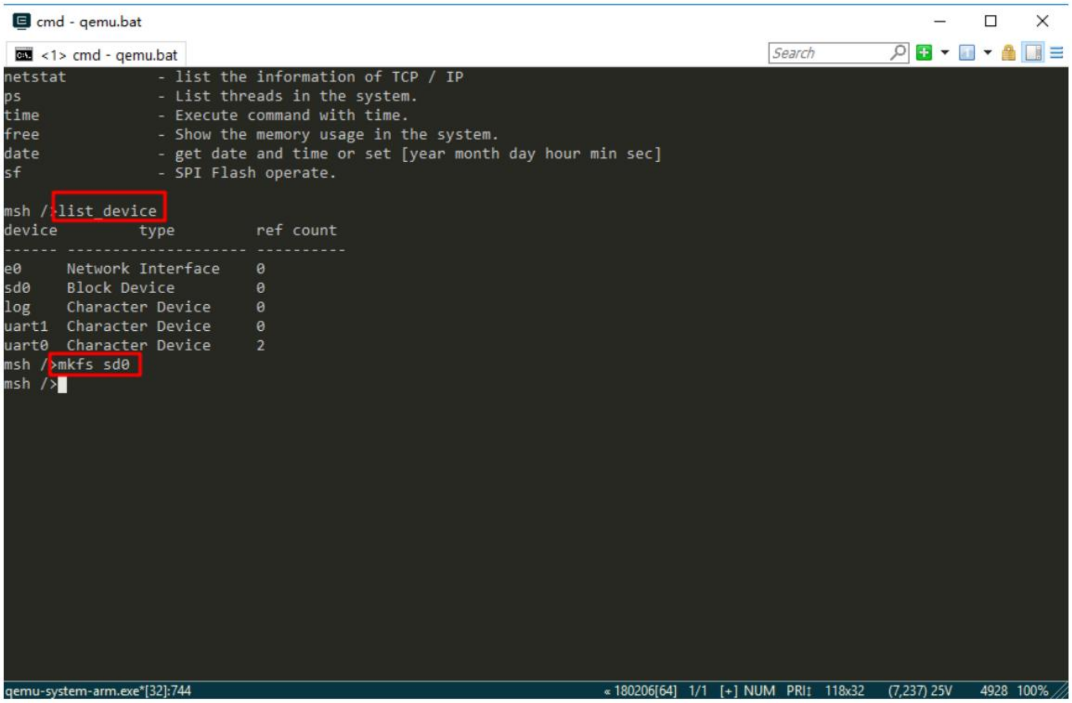
qemu-system-arm.exe*[32]:9232
« 180206[64] 1/1 [+] NUM PRI↕ 112x26 (7,688) 25V 9012 100%

Figure 7: Check system thread status

2.4 Running the RT-Thread File System

Enter list_device to view all devices registered to the system. As shown in the following picture, you can see the virtual sd card "sd0" device, then we can use the mkfs sd0 command to format the sd card, executing this command will format the sd card Format it into FatFS file system. FatFs is a file system developed for small embedded devices that is compatible with Microsoft FAT. The system is written in ANSI C, uses an abstract hardware I/O layer and provides continuous maintenance, so it has good hardware Software independence and portability.

For more information about FatFS , please click on the link: http://elm-chan.org/fsw/ff/00index_e.html


cmd - qemu.bat
<1> cmd - qemu.bat
Search
qemu-system-arm.exe*[32]:744
« 180206[64] 1/1 [+] NUM PRI↕ 118x32 (7,237) 25V 4928 100%

```
netstat          - list the information of TCP / IP
ps               - List threads in the system.
time             - Execute command with time.
free             - Show the memory usage in the system.
date             - get date and time or set [year month day hour min sec]
sf               - SPI Flash operate.

msh />list_device
device         type         ref count
------ -------------------- ----------
e0     Network Interface    0
sd0    Block Device         0
log    Character Device     0
uart1  Character Device     0
uart0  Character Device     2
msh />mkfs sd0
msh />
```

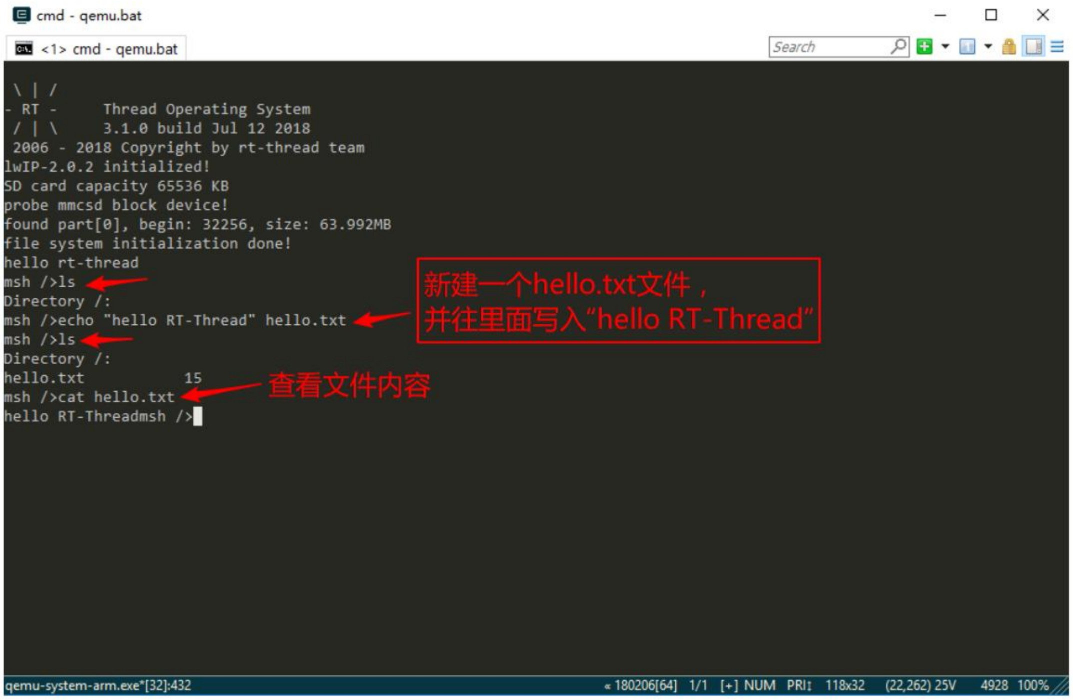
Figure 8: format sd Card

After formatting the SD card for the first time, the file system will not be loaded immediately. It will be loaded correctly after the second boot.

Exit the virtual machine, then enter qemu.bat in the Env command line interface to restart the virtual machine and project, and enter the ls command to

You can see that the new Directory directory has been added and the file system has been mounted. Then you can use other functions provided by RT-Thread.

Command experience file system


cmd - qemu.bat
<1> cmd - qemu.bat
Search
新建一个hello.txt文件，
并往里面写入“hello RT-Thread”
查看文件内容
qemu-system-arm.exe*[32]:432
« 180206[64] 1/1 [+] NUM PRI↕ 118x32 (22,262) 25V 4928 100%

```
 \ | /
- RT -     Thread Operating System
 / | \     3.1.0 build Jul 12 2018
 2006 - 2018 Copyright by rt-thread team
lwIP-2.0.2 initialized!
SD card capacity 65536 KB
probe mmcsd block device!
found part[0], begin: 32256, size: 63.992MB
file system initialization done!
hello rt-thread
msh />ls
Directory /:
msh />echo "hello RT-Thread" hello.txt
msh />ls
Directory /:
hello.txt           15
msh />cat hello.txt
hello RT-Threadmsh />
```

Figure 9: Other file system commands

2.5 Running the RT-Thread Network

2.5.1. Step 1: Install and configure the TAP network card

1. Download the TAP network card tap-windows-9.21.2.exe. After downloading, double-click the installer and follow the default installation procedure.

2. Open the Network and Sharing Center to change the adapter settings and rename the installed virtual network card to tap, as shown below:

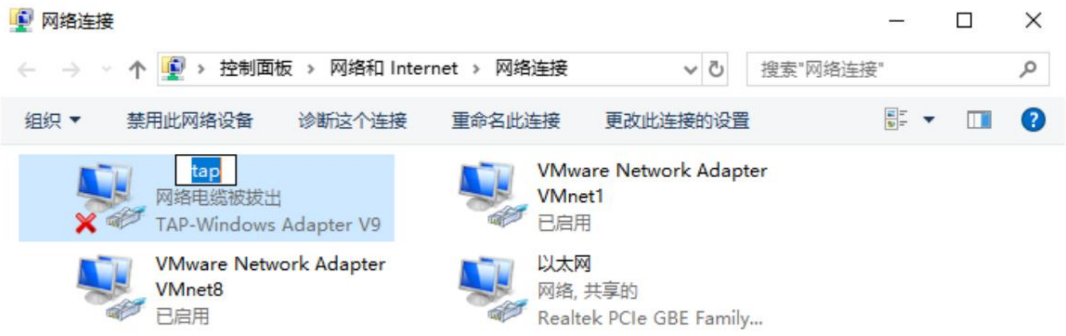
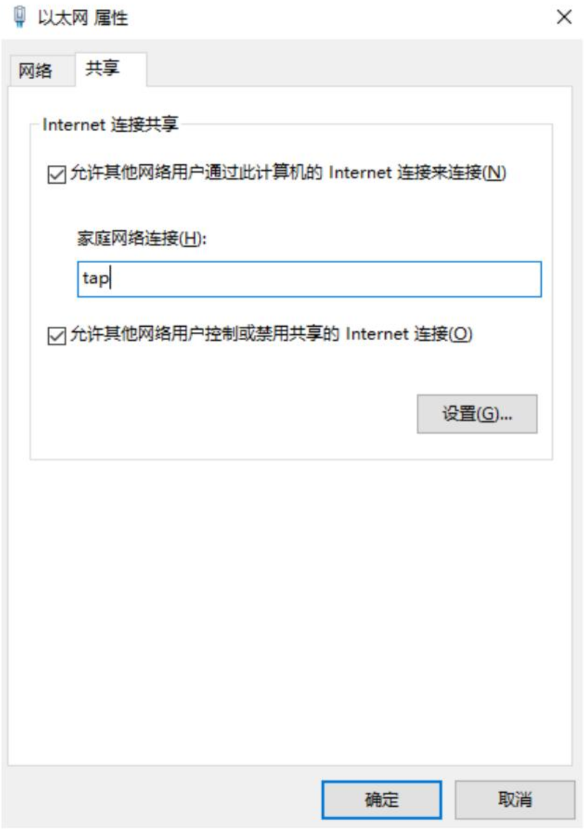


Figure **10:** *tap_rename*

Method **A:** Right-click the network connection that can access the Internet (this article uses Ethernet), open Properties -> Sharing, and select Home

The network connection is tap, click OK to complete the setting, as shown below:



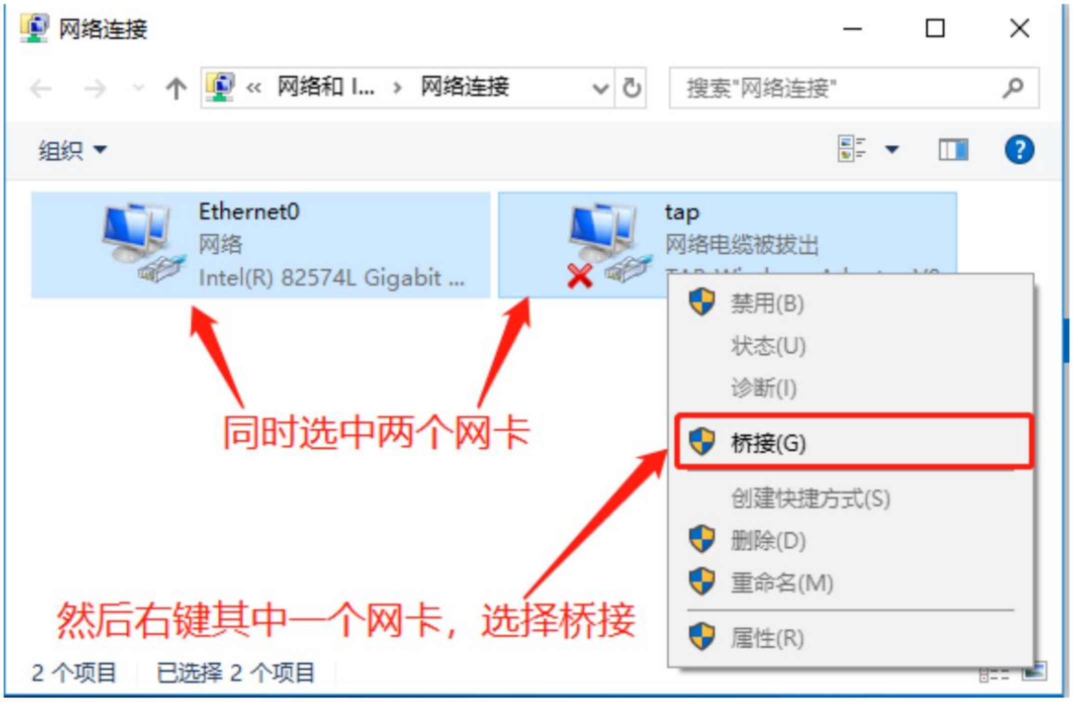
☑ 允许其他网络用户通过此计算机的 Internet 连接来连接(N)

家庭网络连接(H): tap

☑ 允许其他网络用户控制或禁用共享的 Internet 连接(O)

Figure **11:** *tap_share_internet*

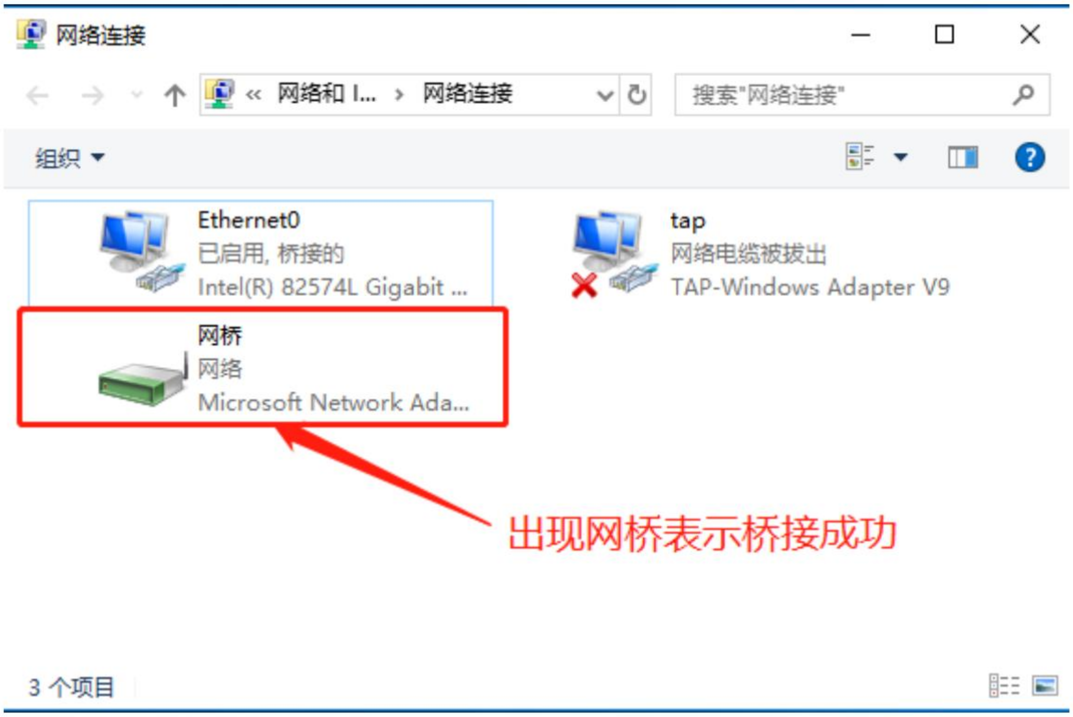
Method B: You can also bridge a physical network card that can connect to the network normally with the tap. After the bridge is successfully connected, A network bridge appears. As shown in the following picture:



同时选中两个网卡

然后右键其中一个网卡，选择桥接

Figure 12: *tap_* bridging



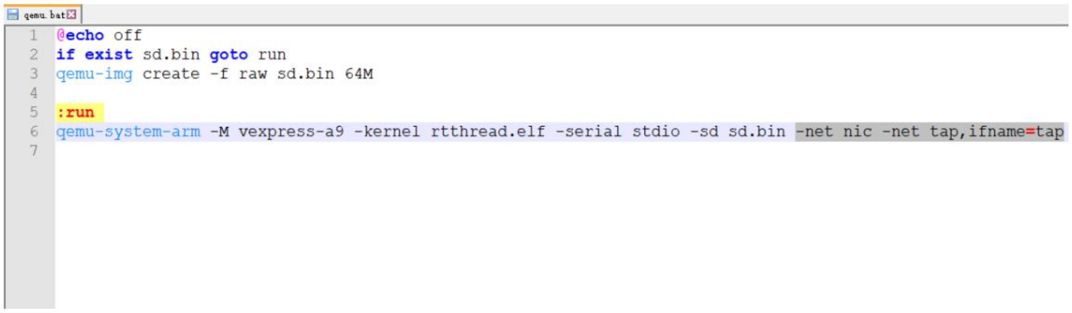
出现网桥表示桥接成功

Figure 13: *tap_* Bridge successful

Note: The tap network card and VMware's virtual network card may conflict. If network sharing cannot be enabled, Or if you cannot ping the network, please delete the VMware virtual network card and try again.

2.5.2. Step 2: Modify the **qemu.bat** script file

Open the qemu.bat file in the qemu-vexpress-a9 BSP directory and add -net nic in the location shown below -net tap,ifname=tap configuration. Ifname=tap means the name of the network card is tap.

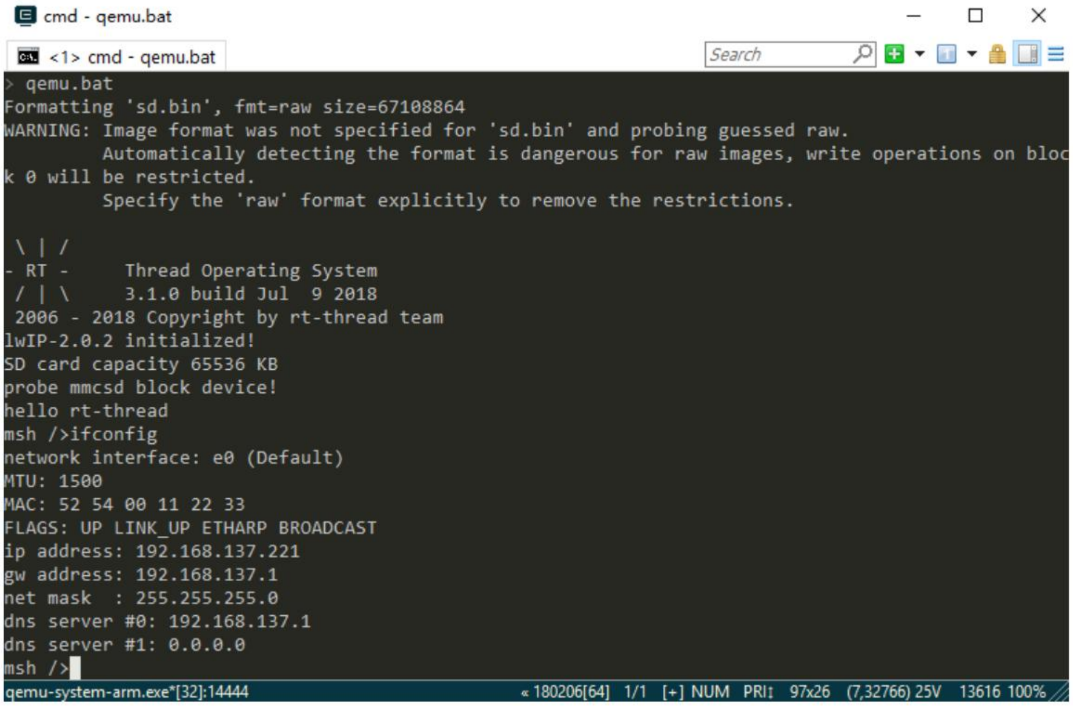


```
@echo off
if exist sd.bin goto run
qemu-img create -f raw sd.bin 64M

:run
qemu-system-arm -M vexpress-a9 -kernel rtthread.elf -serial stdio -sd sd.bin -net nic -net tap,ifname=tap
```

Figure **14:** *qemu_modify*

2.5.3. Step 3 Check **the IP** address

Enter the qemu.bat command to run the project, enter the ifconfig command in the shell to check the network status, and obtain it normally If the IP is received, it means the network driver is normal and the configuration is completed. The effect is as shown in the figure below:


cmd - qemu.bat
<1> cmd - qemu.bat
Search
qemu-system-arm.exe*[32]:14444
« 180206[64] 1/1 [+] NUM PRI↕ 97x26 (7,32766) 25V 13616 100%

```
> qemu.bat
Formatting 'sd.bin', fmt=raw size=67108864
WARNING: Image format was not specified for 'sd.bin' and probing guessed raw.
         Automatically detecting the format is dangerous for raw images, write operations on bloc
k 0 will be restricted.
         Specify the 'raw' format explicitly to remove the restrictions.

 \ | /
- RT -     Thread Operating System
 / | \     3.1.0 build Jul  9 2018
 2006 - 2018 Copyright by rt-thread team
lwIP-2.0.2 initialized!
SD card capacity 65536 KB
probe mmcsd block device!
hello rt-thread
msh />ifconfig
network interface: e0 (Default)
MTU: 1500
MAC: 52 54 00 11 22 33
FLAGS: UP LINK_UP ETHARP BROADCAST
ip address: 192.168.137.221
gw address: 192.168.137.1
net mask  : 255.255.255.0
dns server #0: 192.168.137.1
dns server #1: 0.0.0.0
msh />
```

Figure **15:** *ifconfig*

2.5.4. Notes

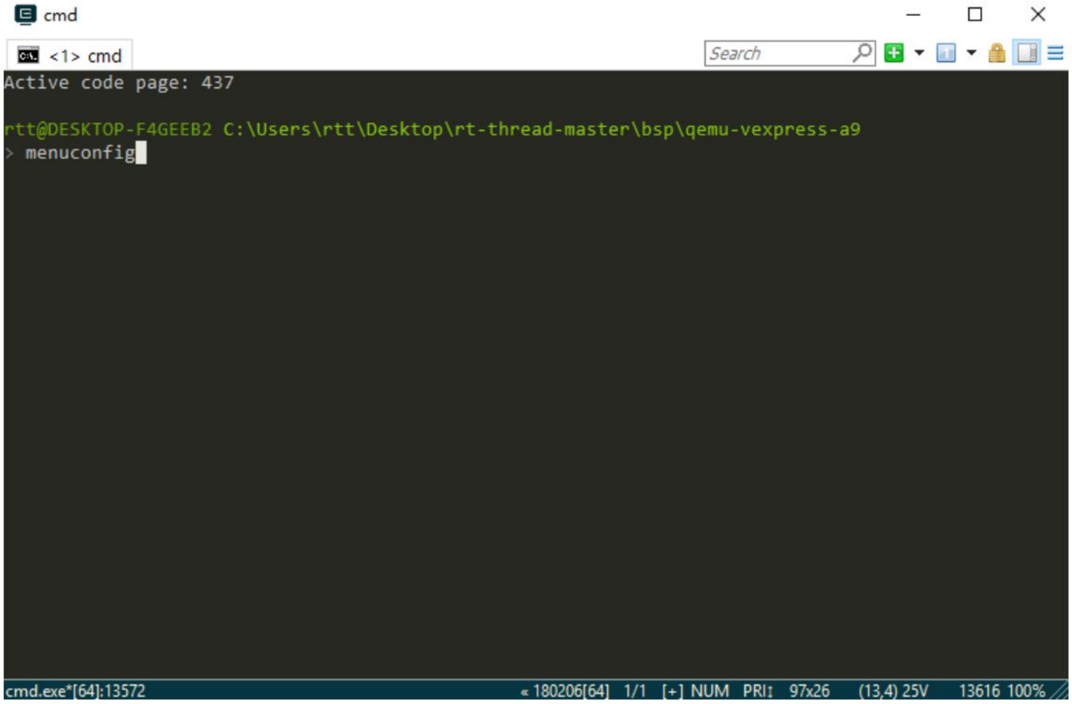
- If you cannot obtain an IP address, first turn off Ethernet sharing and then turn it on again.

- If the IP obtained is 10.0.x,x, it is because the startup parameters -net nic -net are not added to QEMU tap,ifname=tap .
- When the virtual machine starts running, it will not get the IP address immediately. Sometimes it takes several seconds to get it. to IP.
- To shut down the virtual machine, press Ctrl + 'C' to end the program.

2.6 Run the RT-Thread Ping Tool

2.6.1. Step 1: Download the network tool software package

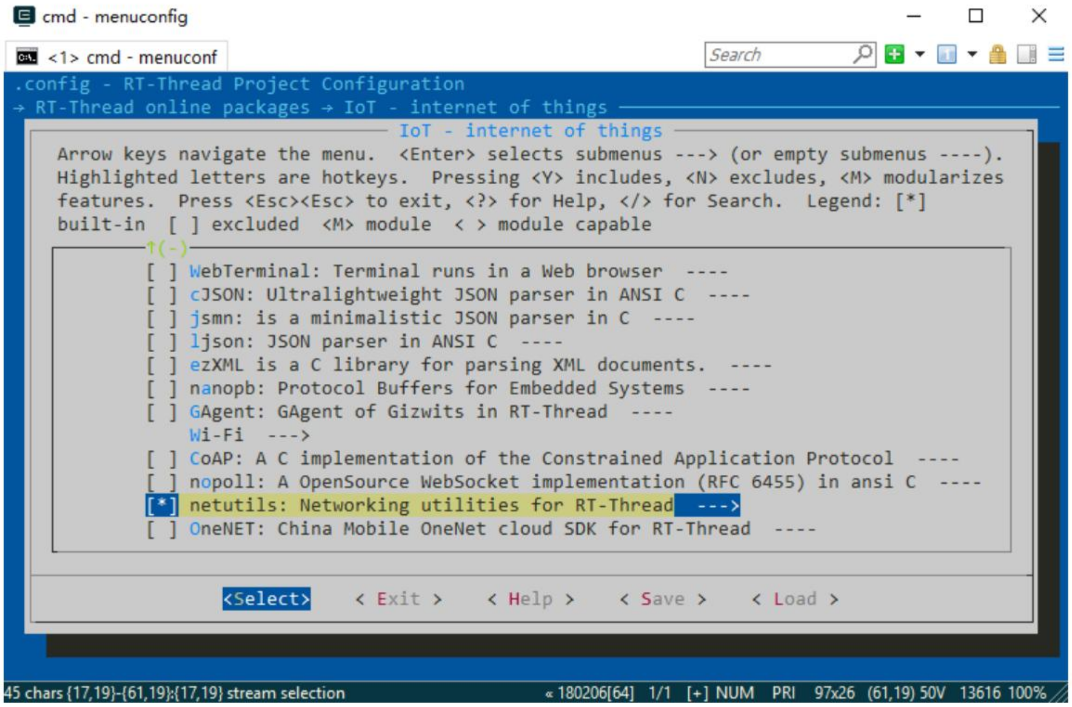
1. Open the Env tool in the path bsp\qemu-vexpress-a9 and execute menuconfig, as shown below:



```
Active code page: 437

rtt@DESKTOP-F4GEEB2 C:\Users\rtt\Desktop\rt-thread-master\bsp\qemu-vexpress-a9
> menuconfig
```

Figure 16: *Env_menuconfig*

2. Open the netutils: Networking utilities for RT-Thread function on the RT-Thread online packages->IoT - internet of things page, as shown below:



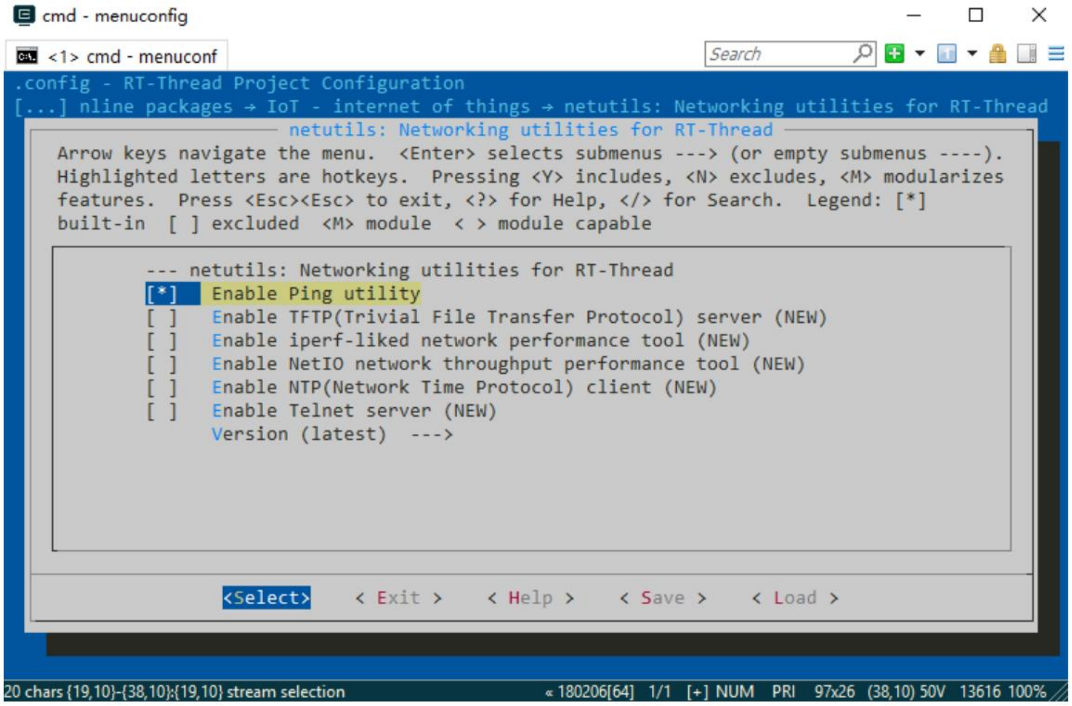
cmd - menuconfig
<1> cmd - menuconf
Search
45 chars {17,19}-{61,19}:{17,19} stream selection
« 180206[64] 1/1 [+] NUM PRI 97x26 (61,19) 50V 13616 100%

```
.config - RT-Thread Project Configuration
→ RT-Thread online packages → IoT - internet of things ──────────────────────
 ─────────────────────── IoT - internet of things ───────────────────────
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty 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 capable
 ┌───────↑(-)──────────────────────────────────────────────────────────────────┐
 │       [ ] WebTerminal: Terminal runs in a Web browser  ----                  │
 │       [ ] cJSON: Ultralightweight JSON parser in ANSI C  ----                │
 │       [ ] jsmn: is a minimalistic JSON parser in C  ----                     │
 │       [ ] ljson: JSON parser in ANSI C  ----                                 │
 │       [ ] ezXML is a C library for parsing XML documents.  ----              │
 │       [ ] nanopb: Protocol Buffers for Embedded Systems  ----                │
 │       [ ] GAgent: GAgent of Gizwits in RT-Thread  ----                       │
 │           Wi-Fi  --->                                                        │
 │       [ ] CoAP: A C implementation of the Constrained Application Protocol  ---- │
 │       [ ] nopoll: A OpenSource WebSocket implementation (RFC 6455) in ansi C  ---- │
 │       [*] netutils: Networking utilities for RT-Thread  --->                 │
 │       [ ] OneNET: China Mobile OneNet cloud SDK for RT-Thread  ----          │
 └──────────────────────────────────────────────────────────────────────────────┘

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

Figure **17:** *online_packages*

3. Go to netutils: Networking utilities for RT-Thread page and turn on Enable Ping utility.

Yes, as shown below:



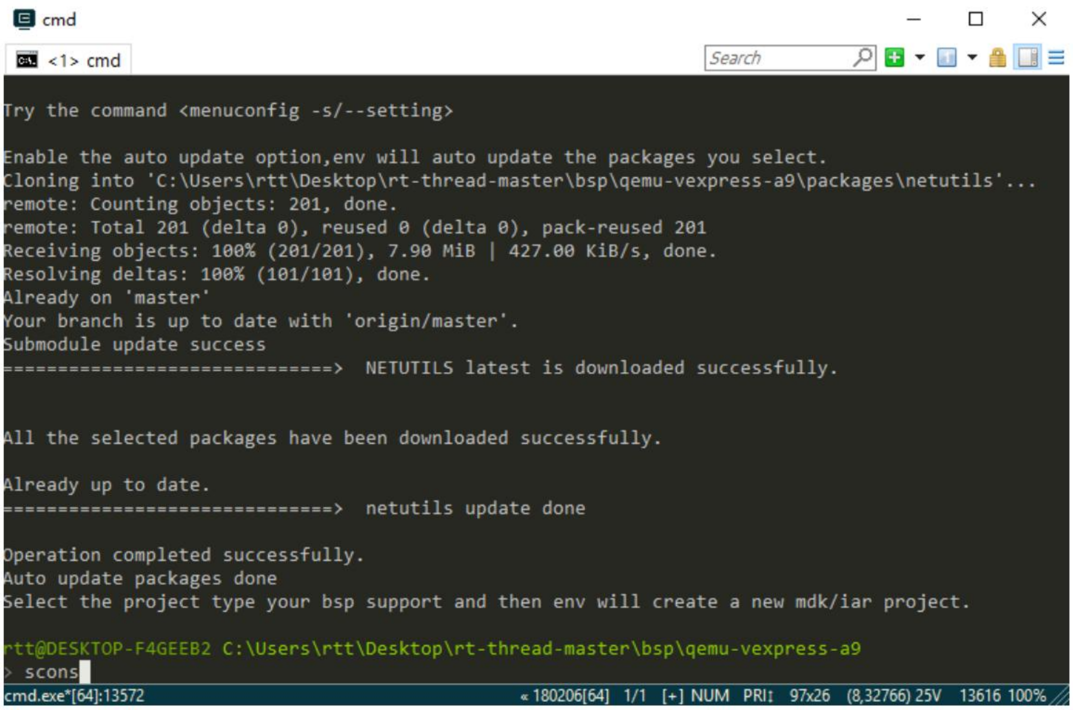
cmd - menuconfig
<1> cmd - menuconf
Search
20 chars {19,10}-{38,10}:{19,10} stream selection
« 180206[64] 1/1 [+] NUM PRI 97x26 (38,10) 50V 13616 100%

```
.config - RT-Thread Project Configuration
[...] nline packages → IoT - internet of things → netutils: Networking utilities for RT-Thread
 ─────────────── netutils: Networking utilities for RT-Thread ───────────────
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty 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 capable
 ┌──────────────────────────────────────────────────────────────────────────────┐
 │       --- netutils: Networking utilities for RT-Thread                       │
 │       [*]   Enable Ping utility                                              │
 │       [ ]   Enable TFTP(Trivial File Transfer Protocol) server (NEW)         │
 │       [ ]   Enable iperf-liked network performance tool (NEW)                │
 │       [ ]   Enable NetIO network throughput performance tool (NEW)           │
 │       [ ]   Enable NTP(Network Time Protocol) client (NEW)                   │
 │       [ ]   Enable Telnet server (NEW)                                       │
 │             Version (latest)  --->                                           │
 │                                                                              │
 └──────────────────────────────────────────────────────────────────────────────┘

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

Figure **18:** *enable_ping*

4. Save and exit the configuration interface. If the Env automatic update package function is not enabled, you need to enter

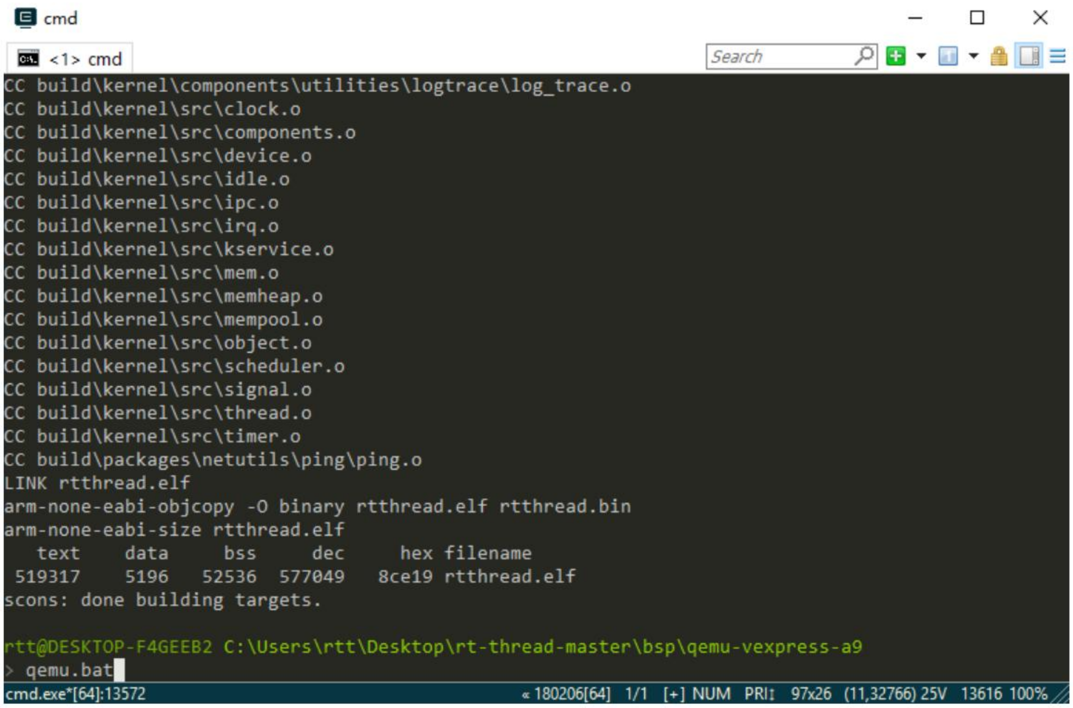
pkgs --update updates the software package configuration. After the update is complete, use the scons command to recompile the project, as shown below:


cmd
<1> cmd
Search
cmd.exe*[64]:13572
« 180206[64] 1/1 [+] NUM PRI↕ 97x26 (8,32766) 25V 13616 100%

```
Try the command <menuconfig -s/--setting>

Enable the auto update option,env will auto update the packages you select.
Cloning into 'C:\Users\rtt\Desktop\rt-thread-master\bsp\qemu-vexpress-a9\packages\netutils'...
remote: Counting objects: 201, done.
remote: Total 201 (delta 0), reused 0 (delta 0), pack-reused 201
Receiving objects: 100% (201/201), 7.90 MiB | 427.00 KiB/s, done.
Resolving deltas: 100% (101/101), done.
Already on 'master'
Your branch is up to date with 'origin/master'.
Submodule update success
==============================>  NETUTILS latest is downloaded successfully.


All the selected packages have been downloaded successfully.

Already up to date.
==============================>  netutils update done

Operation completed successfully.
Auto update packages done
Select the project type your bsp support and then env will create a new mdk/iar project.

rtt@DESKTOP-F4GEEB2 C:\Users\rtt\Desktop\rt-thread-master\bsp\qemu-vexpress-a9
> scons
```

Figure **19:** *scons*

5. After compilation is complete, run the qemu.bat file, as shown below:

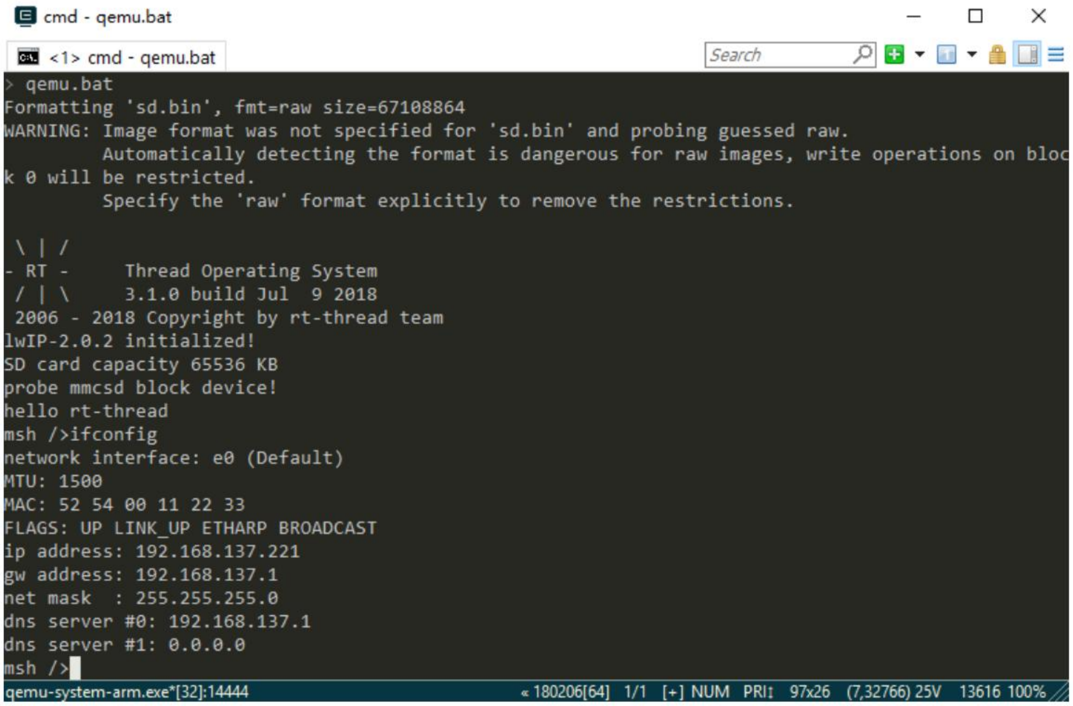

cmd
<1> cmd
Search
cmd.exe*[64]:13572
« 180206[64] 1/1 [+] NUM PRI↕ 97x26 (11,32766) 25V 13616 100%

```
CC build\kernel\components\utilities\logtrace\log_trace.o
CC build\kernel\src\clock.o
CC build\kernel\src\components.o
CC build\kernel\src\device.o
CC build\kernel\src\idle.o
CC build\kernel\src\ipc.o
CC build\kernel\src\irq.o
CC build\kernel\src\kservice.o
CC build\kernel\src\mem.o
CC build\kernel\src\memheap.o
CC build\kernel\src\mempool.o
CC build\kernel\src\object.o
CC build\kernel\src\scheduler.o
CC build\kernel\src\signal.o
CC build\kernel\src\thread.o
CC build\kernel\src\timer.o
CC build\packages\netutils\ping\ping.o
LINK rtthread.elf
arm-none-eabi-objcopy -O binary rtthread.elf rtthread.bin
arm-none-eabi-size rtthread.elf
   text    data     bss     dec     hex filename
 519317    5196   52536  577049   8ce19 rtthread.elf
scons: done building targets.

rtt@DESKTOP-F4GEEB2 C:\Users\rtt\Desktop\rt-thread-master\bsp\qemu-vexpress-a9
> qemu.bat
```

Figure **20:** *qemu_bat*

2.6.2. Step 2: Run the **ping** tool

Enter the ifconfig command in the shell to check the network status. If the IP address is obtained normally, it means that the network driver is normal:

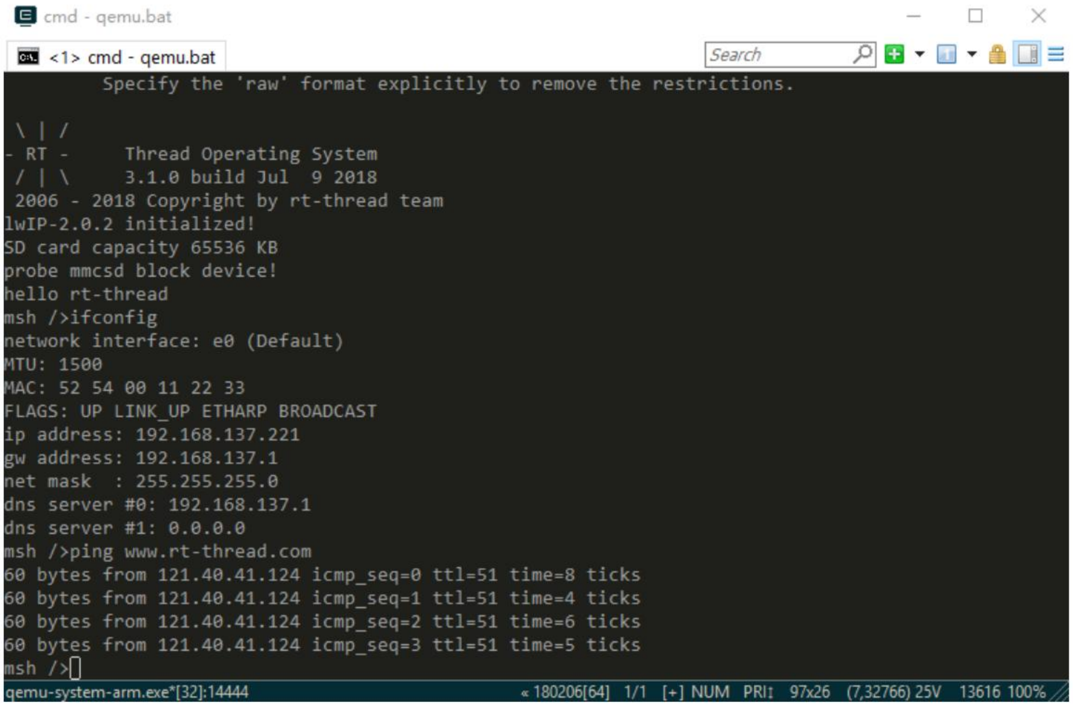


```
> qemu.bat
Formatting 'sd.bin', fmt=raw size=67108864
WARNING: Image format was not specified for 'sd.bin' and probing guessed raw.
         Automatically detecting the format is dangerous for raw images, write operations on bloc
k 0 will be restricted.
         Specify the 'raw' format explicitly to remove the restrictions.

 \ | /
- RT -     Thread Operating System
 / | \     3.1.0 build Jul  9 2018
 2006 - 2018 Copyright by rt-thread team
lwIP-2.0.2 initialized!
SD card capacity 65536 KB
probe mmcsd block device!
hello rt-thread
msh />ifconfig
network interface: e0 (Default)
MTU: 1500
MAC: 52 54 00 11 22 33
FLAGS: UP LINK_UP ETHARP BROADCAST
ip address: 192.168.137.221
gw address: 192.168.137.1
net mask  : 255.255.255.0
dns server #0: 192.168.137.1
dns server #1: 0.0.0.0
msh />
```

qemu-system-arm.exe*[32]:14444
« 180206[64] 1/1 [+] NUM PRI↕ 97x26 (7,32766) 25V 13616 100%

Figure **21:** *ifconfig*

Enter ping www.rt-thread.com in the shell and you can see the ping result, indicating that the network configuration is Success, ping is possible, as shown below:


cmd - qemu.bat
<1> cmd - qemu.bat
Search
qemu-system-arm.exe*[32]:14444
« 180206[64] 1/1 [+] NUM PRI↕ 97x26 (7,32766) 25V 13616 100%

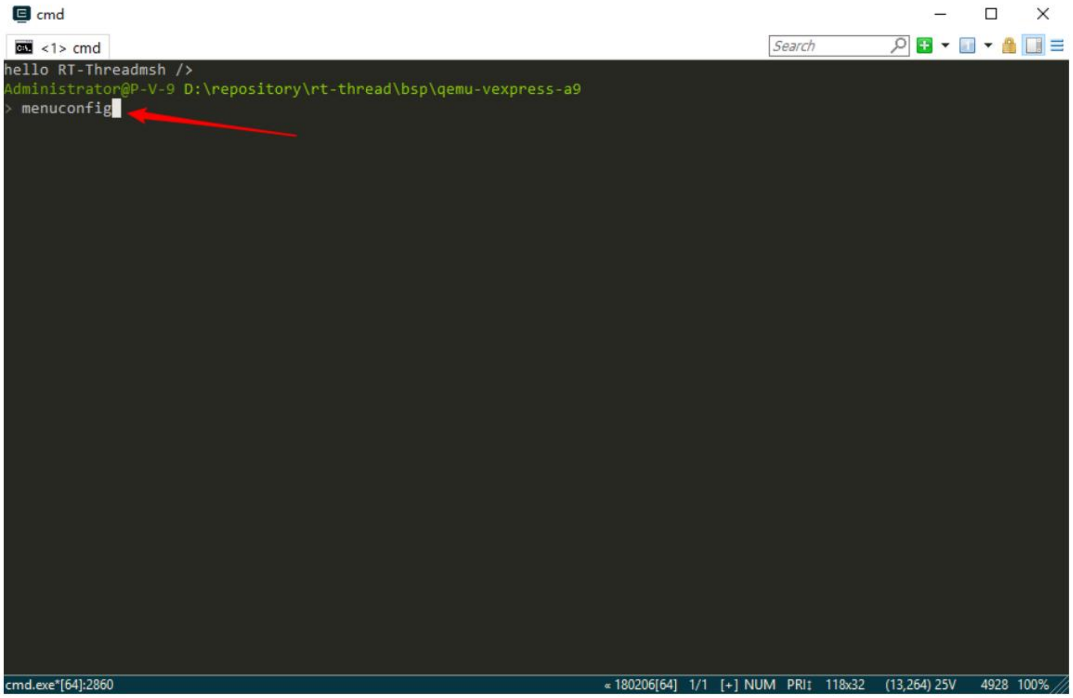
```
         Specify the 'raw' format explicitly to remove the restrictions.

 \ | /
- RT -     Thread Operating System
 / | \     3.1.0 build Jul  9 2018
 2006 - 2018 Copyright by rt-thread team
lwIP-2.0.2 initialized!
SD card capacity 65536 KB
probe mmcsd block device!
hello rt-thread
msh />ifconfig
network interface: e0 (Default)
MTU: 1500
MAC: 52 54 00 11 22 33
FLAGS: UP LINK_UP ETHARP BROADCAST
ip address: 192.168.137.221
gw address: 192.168.137.1
net mask  : 255.255.255.0
dns server #0: 192.168.137.1
dns server #1: 0.0.0.0
msh />ping www.rt-thread.com
60 bytes from 121.40.41.124 icmp_seq=0 ttl=51 time=8 ticks
60 bytes from 121.40.41.124 icmp_seq=1 ttl=51 time=4 ticks
60 bytes from 121.40.41.124 icmp_seq=2 ttl=51 time=6 ticks
60 bytes from 121.40.41.124 icmp_seq=3 ttl=51 time=5 ticks
msh />
```

Figure **22:** *ping*

2.7 Running the **RT-Thread GUI** Engine

2.7.1. Step 1: Download the **GUI** engine software package

Shut down the QEMU virtual machine, return to the ENV console, and enter the menuconfig command to enter the configuration interface:



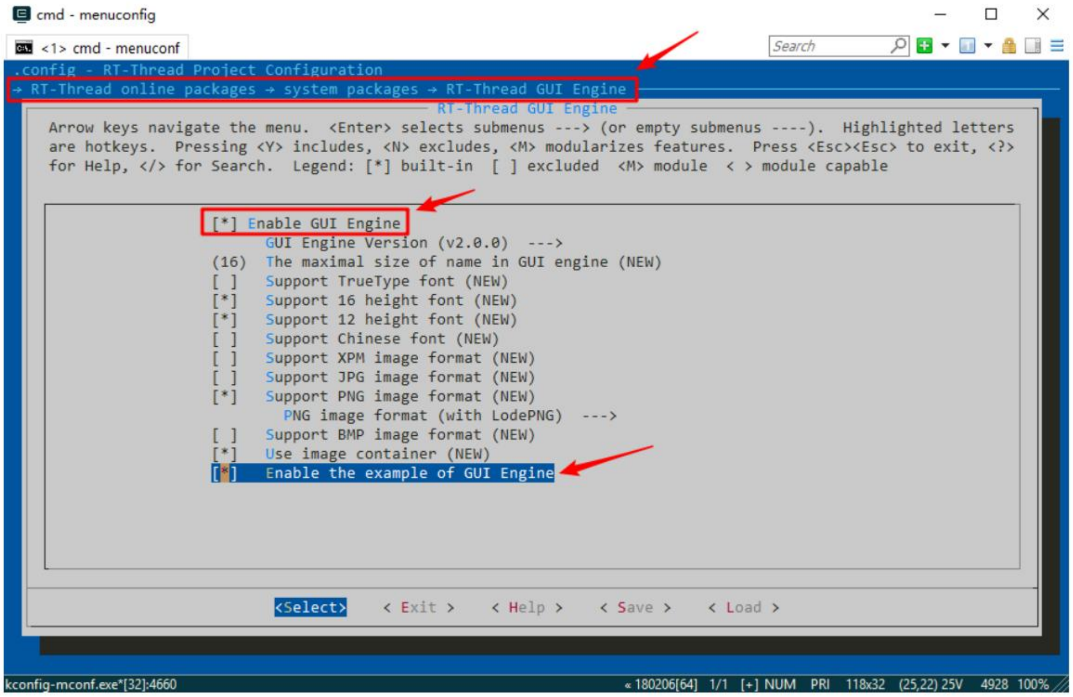
```
hello RT-Threadmsh />
Administrator@P-V-9 D:\repository\rt-thread\bsp\qemu-vexpress-a9
> menuconfig
```

```
cmd.exe*[64]:2860          « 180206[64]  1/1  [+] NUM  PRI↕  118x32  (13,264) 25V  4928 100%
```

Figure **23:** *menuconfig*

Enter "RT-Thread oneline packages" ÿ "system packages" ÿ "RT-Thread GUI Engine"

In the submenu, select “Enable GUI Engine” and “Enable the example of GUI Engine”:

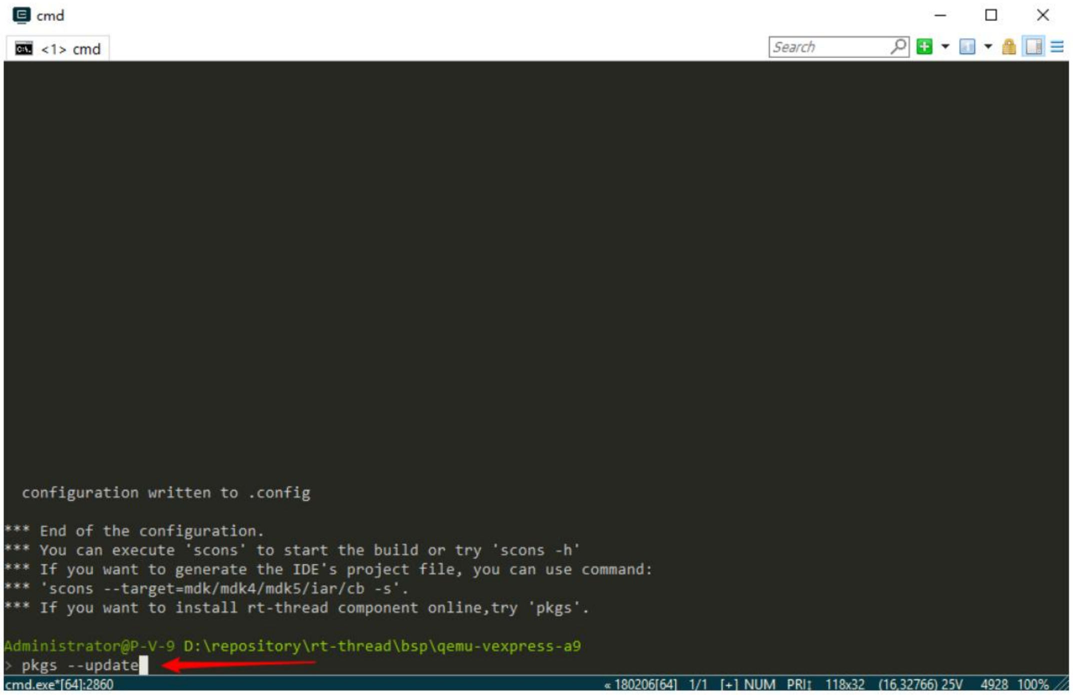


```
.config - RT-Thread Project Configuration
→ RT-Thread online packages → system packages → RT-Thread GUI Engine
                              RT-Thread GUI Engine
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters
  are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?>
  for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                 [*] Enable GUI Engine
                        GUI Engine Version (v2.0.0)  --->
                 (16)   The maximal size of name in GUI engine (NEW)
                 [ ]    Support TrueType font (NEW)
                 [*]    Support 16 height font (NEW)
                 [*]    Support 12 height font (NEW)
                 [ ]    Support Chinese font (NEW)
                 [ ]    Support XPM image format (NEW)
                 [ ]    Support JPG image format (NEW)
                 [*]    Support PNG image format (NEW)
                          PNG image format (with LodePNG)  --->
                 [ ]    Support BMP image format (NEW)
                 [*]    Use image container (NEW)
                 [*]    Enable the example of GUI Engine

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

```
kconfig-mconf.exe*[32]:4660          « 180206[64]  1/1  [+] NUM  PRI  118x32  (25,22) 25V  4928 100%
```

Figure **24:** choose *GUI* Example

Press the 'ÿ' key to select "save" to save the configuration, and press the "Exit" key to exit the configuration interface and return to the command line interface. Enter pkgs --update downloads the GUI software package and sample code:

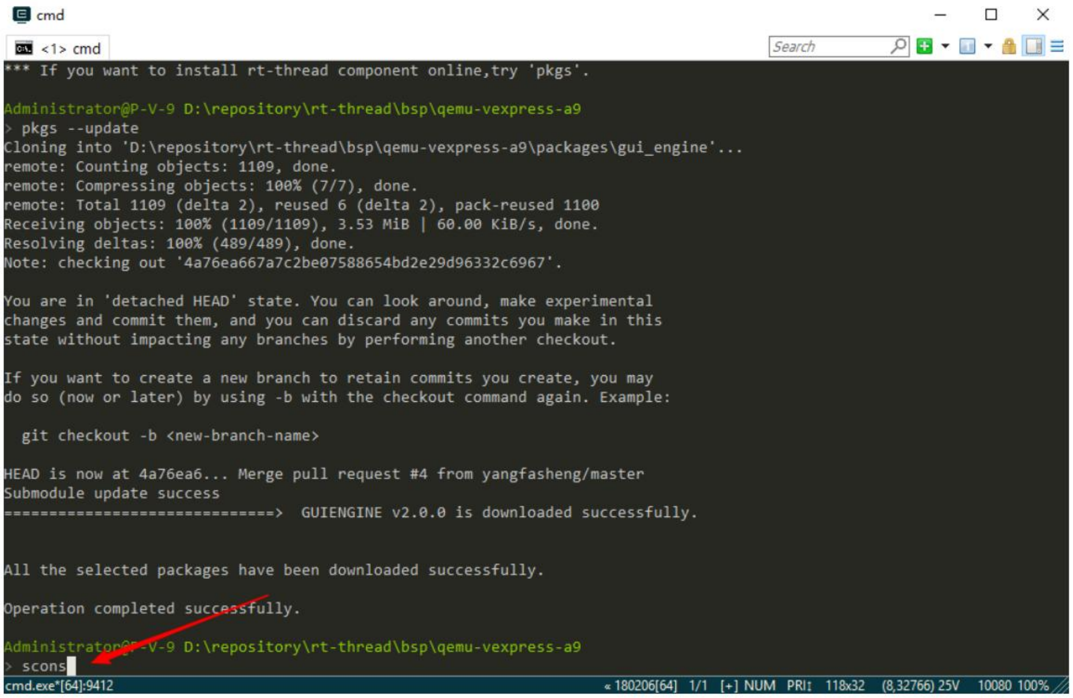

cmd
<1> cmd
Search
cmd.exe*[64]:2860
« 180206[64] 1/1 [+] NUM PRI↕ 118x32 (16,32766) 25V 4928 100%

```
  configuration written to .config

*** End of the configuration.
*** You can execute 'scons' to start the build or try 'scons -h'
*** If you want to generate the IDE's project file, you can use command:
*** 'scons --target=mdk/mdk4/mdk5/iar/cb -s'.
*** If you want to install rt-thread component online,try 'pkgs'.

Administrator@P-V-9 D:\repository\rt-thread\bsp\qemu-vexpress-a9
> pkgs --update
```

Figure 25: download *GUI* Software Package

2.7.2. Step 2: Run the GUI engine

After the software package is downloaded, enter scons to recompile the project:


cmd
<1> cmd
Search
cmd.exe*[64]:9412
« 180206[64] 1/1 [+] NUM PRI↕ 118x32 (8,32766) 25V 10080 100%

```
*** If you want to install rt-thread component online,try 'pkgs'.

Administrator@P-V-9 D:\repository\rt-thread\bsp\qemu-vexpress-a9
> pkgs --update
Cloning into 'D:\repository\rt-thread\bsp\qemu-vexpress-a9\packages\gui_engine'...
remote: Counting objects: 1109, done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 1109 (delta 2), reused 6 (delta 2), pack-reused 1100
Receiving objects: 100% (1109/1109), 3.53 MiB | 60.00 KiB/s, done.
Resolving deltas: 100% (489/489), done.
Note: checking out '4a76ea667a7c2be07588654bd2e29d96332c6967'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b <new-branch-name>

HEAD is now at 4a76ea6... Merge pull request #4 from yangfasheng/master
Submodule update success
==============================>  GUIENGINE v2.0.0 is downloaded successfully.


All the selected packages have been downloaded successfully.

Operation completed successfully.

Administrator@P-V-9 D:\repository\rt-thread\bsp\qemu-vexpress-a9
> scons
```

Figure 26: *scons* Command compilation project

After the compilation is complete, enter the qemu.bat command to start the QEMU virtual machine and project. You can see the QEMU virtual display.

The screen displays the pictures, text, and graphic information displayed by the sample code:

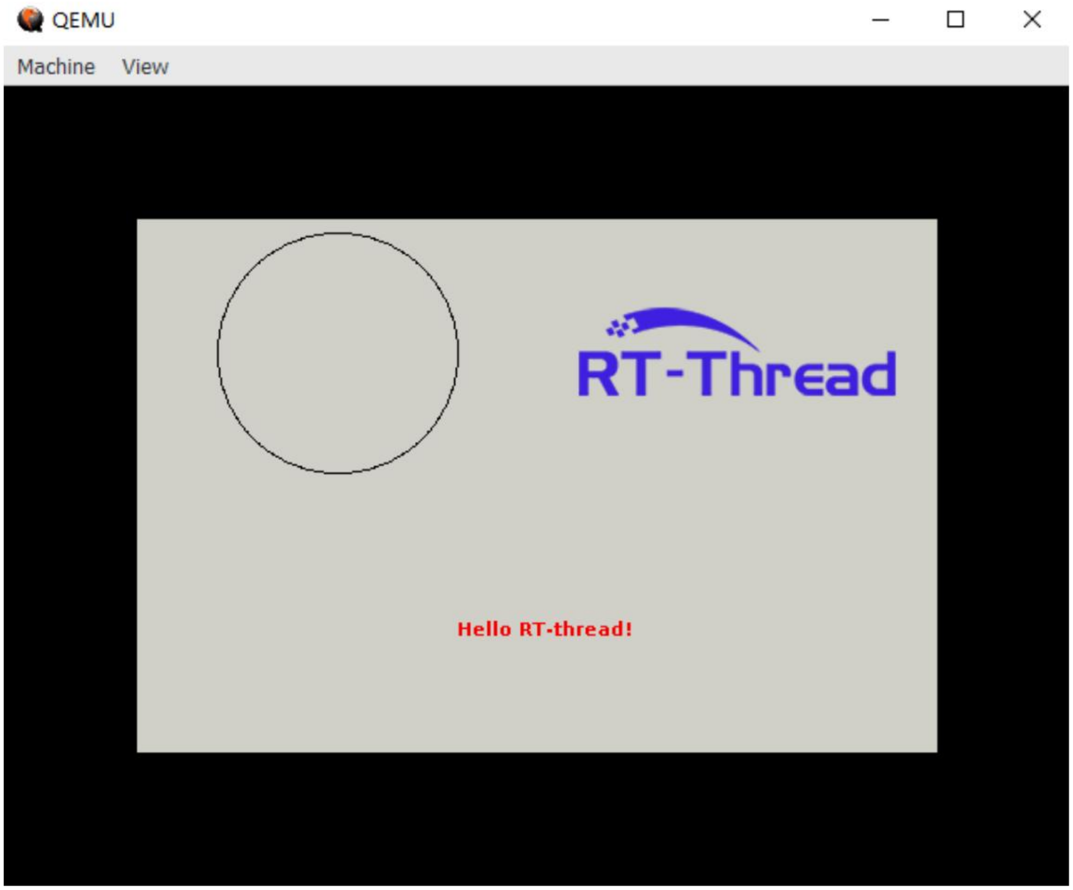


Figure **27:** Virtual Screen *GUI* Example display

3. Run **RT-Thread** using **QEMU** on **Ubuntu**

3.1 Preparation

- Download RT-Thread source code, use the command: git clone https://github.com/RT-Thread/rt-thread.git
- Install QEMU using the command: sudo apt-get install qemu
- Install Scons using the command: sudo apt-get install scons
- Install the compiler. If the compiler version installed using apt-get is too old, compilation errors will occur. You can use the following command in sequence:

  Run the following command to download and install the new version. The download link and decompression folder name vary depending on the downloaded version:

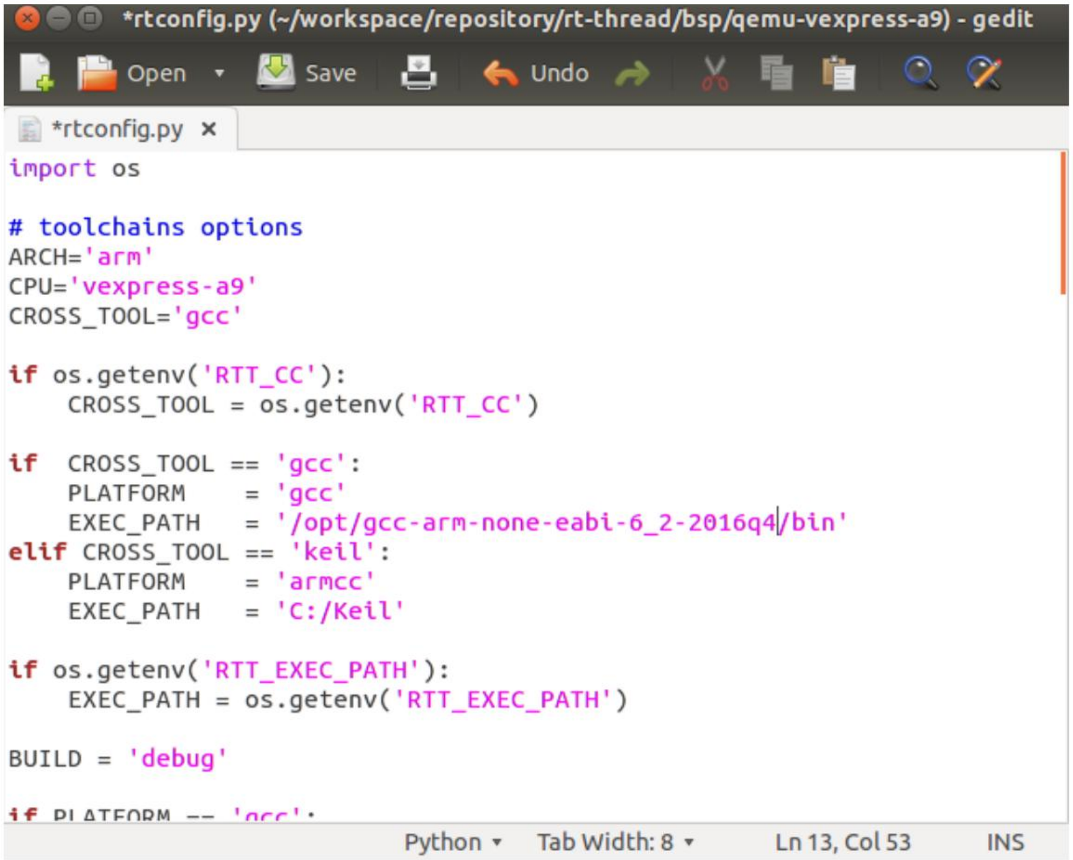
1. wget https://armkeil.blob.core.windows.net/developer/Files/downloads/gnu-rm /6-2016q4/gcc-arm-none-eabi-6_2-2016q4-20161216-linux.tar.bz2

2. cd /opt

3. sudo tar xf ~/Downloads/ gcc-arm-none-eabi-6_2-2016q4-20161216-linux.tar.
bz2

- Install the ncurses library using the command: sudo apt-get install libncurses5-dev

After the compiler is installed, you need to modify the rtconfig.py file under the qemu-vexpress-a9 BSP and change the corresponding path to the bin directory corresponding to the compiler extracted to the opt directory. Refer to the figure below. The directory name varies depending on the downloaded compiler version:

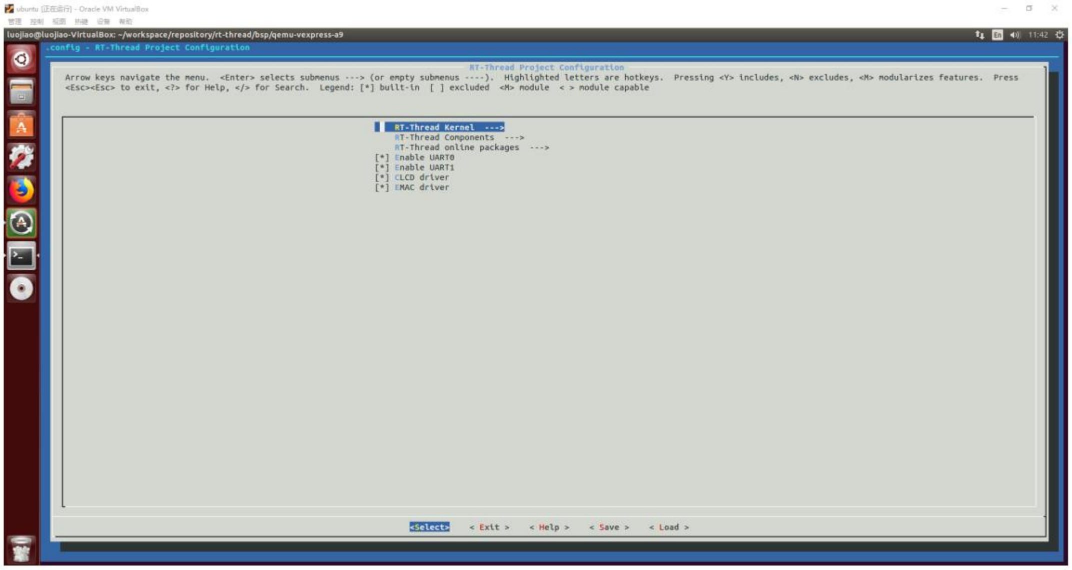

*rtconfig.py (~/workspace/repository/rt-thread/bsp/qemu-vexpress-a9) - gedit
Open
Save
Undo
*rtconfig.py
Python
Tab Width: 8
Ln 13, Col 53
INS

```
import os

# toolchains options
ARCH='arm'
CPU='vexpress-a9'
CROSS_TOOL='gcc'

if os.getenv('RTT_CC'):
    CROSS_TOOL = os.getenv('RTT_CC')

if  CROSS_TOOL == 'gcc':
    PLATFORM    = 'gcc'
    EXEC_PATH   = '/opt/gcc-arm-none-eabi-6_2-2016q4/bin'
elif CROSS_TOOL == 'keil':
    PLATFORM    = 'armcc'
    EXEC_PATH   = 'C:/Keil'

if os.getenv('RTT_EXEC_PATH'):
    EXEC_PATH = os.getenv('RTT_EXEC_PATH')

BUILD = 'debug'

if PLATFORM == 'gcc':
```

Figure 28: Compiler path modification

3.2 Use **menuconfig** to configure the project

1. Enter scons --menuconfig in the root directory of qemu-vexpress-a9 BSP to open the configuration interface and configure the operation The operation is the same as that on Windows platform:



```
luojiao@luojiao-VirtualBox: ~/workspace/repository/rt-thread/bsp/qemu-vexpress-a9
.config - RT-Thread Project Configuration
                         RT-Thread Project Configuration
 Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty 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 capable

                    RT-Thread Kernel  --->
                    RT-Thread Components  --->
                    RT-Thread online packages  --->
                [*] Enable UART0
                [*] Enable UART1
                [*] CLCD driver
                [*] EMAC driver

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

Figure 29: *menuconfig* Configuration interface

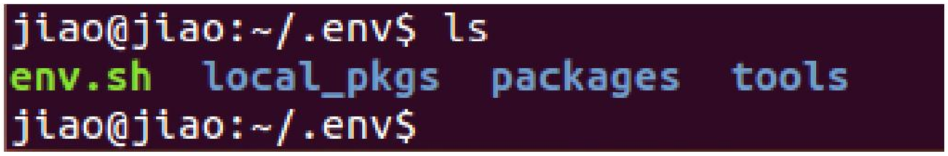
2. After using the scons --menuconfig command, the Env tool will be installed and initialized, and under the home directory

Generate a ".env" folder, which is a hidden folder. Switch to the home directory and use the la command to view all

Directories and files. The env.sh file is the file that needs to be executed. It will configure the environment variables so that we can use pkgs

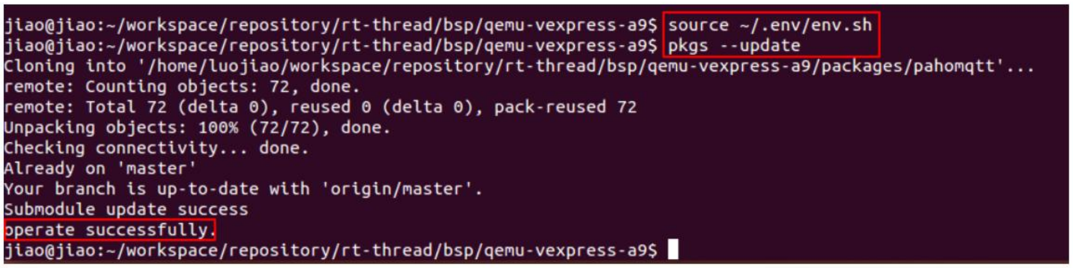
To update the software package, enter the source ~/.env/env.sh command to execute. If you have selected the online software package,

You can use the pkgs --update command to download the software package to the packages folder under the BSP directory:



```
jiao@jiao:~/.env$ ls
env.sh  local_pkgs  packages  tools
jiao@jiao:~/.env$
```

Figure 30: **.env** Folder

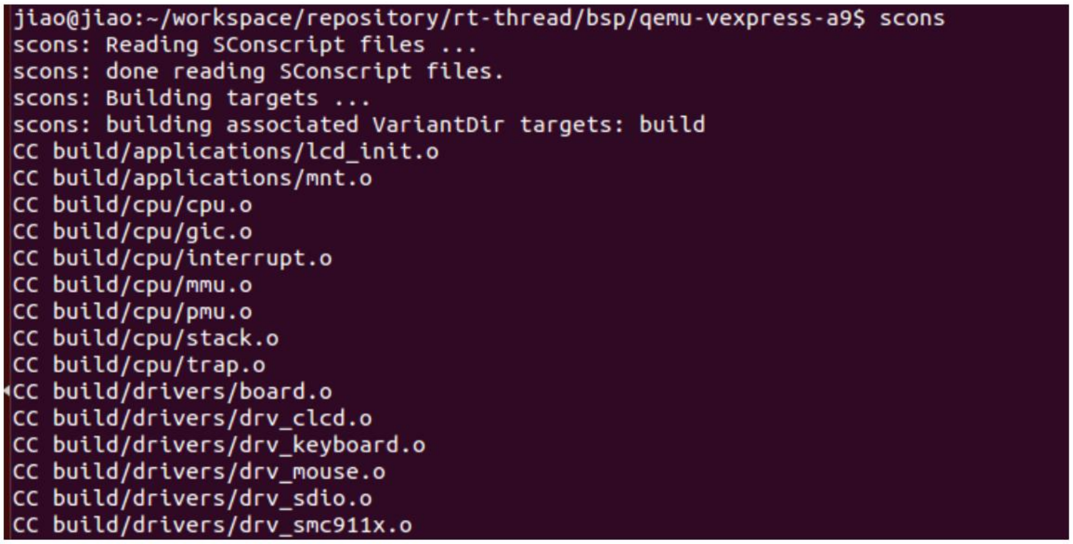


```
jiao@jiao:~/workspace/repository/rt-thread/bsp/qemu-vexpress-a9$ source ~/.env/env.sh
jiao@jiao:~/workspace/repository/rt-thread/bsp/qemu-vexpress-a9$ pkgs --update
Cloning into '/home/luojiao/workspace/repository/rt-thread/bsp/qemu-vexpress-a9/packages/pahomqtt'...
remote: Counting objects: 72, done.
remote: Total 72 (delta 0), reused 0 (delta 0), pack-reused 72
Unpacking objects: 100% (72/72), done.
Checking connectivity... done.
Already on 'master'
Your branch is up-to-date with 'origin/master'.
Submodule update success
operate successfully.
jiao@jiao:~/workspace/repository/rt-thread/bsp/qemu-vexpress-a9$
```

Figure 31: *pkgs –update* Command to download the software package

3.3 Compile and run **RT-Thread**

1. Enter the scons command in the qemu-vexpress-a9 BSP directory to compile the project:

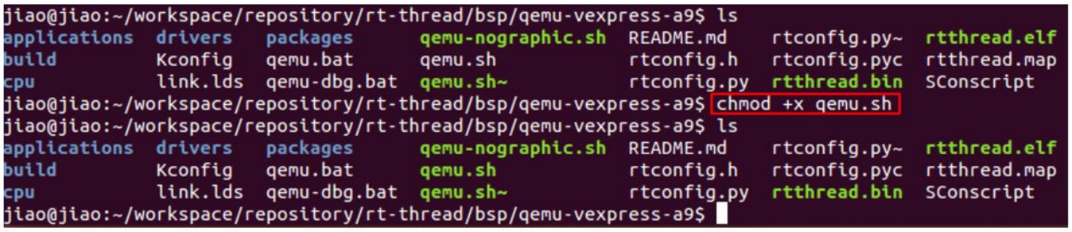


```
jiao@jiao:~/workspace/repository/rt-thread/bsp/qemu-vexpress-a9$ scons
scons: Reading SConscript files ...
scons: done reading SConscript files.
scons: Building targets ...
scons: building associated VariantDir targets: build
CC build/applications/lcd_init.o
CC build/applications/mnt.o
CC build/cpu/cpu.o
CC build/cpu/gic.o
CC build/cpu/interrupt.o
CC build/cpu/mmu.o
CC build/cpu/pmu.o
CC build/cpu/stack.o
CC build/cpu/trap.o
CC build/drivers/board.o
CC build/drivers/drv_clcd.o
CC build/drivers/drv_keyboard.o
CC build/drivers/drv_mouse.o
CC build/drivers/drv_sdio.o
CC build/drivers/drv_smc911x.o
```

Figure 32: *scons* Command compilation project

2. Enter the ls command to view the file details under BSP. The files displayed in green are files with execution permissions.

We need to add execution permissions to the qemu.sh file and enter the chmod +x qemu.sh command:

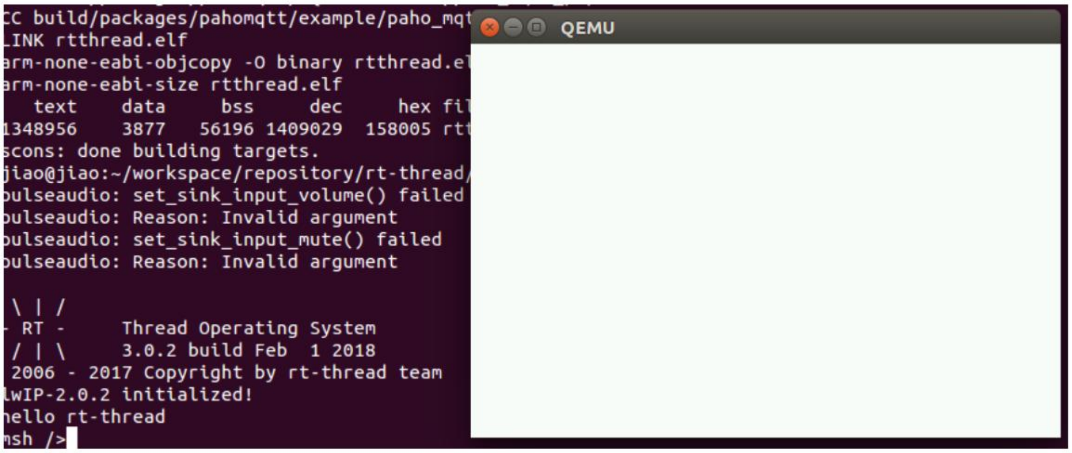


```
jiao@jiao:~/workspace/repository/rt-thread/bsp/qemu-vexpress-a9$ ls
applications  drivers   packages      qemu-nographic.sh  README.md     rtconfig.py~   rtthread.elf
build         Kconfig   qemu.bat      qemu.sh            rtconfig.h    rtconfig.pyc   rtthread.map
cpu           link.lds  qemu-dbg.bat  qemu.sh~           rtconfig.py   rtthread.bin   SConscript
jiao@jiao:~/workspace/repository/rt-thread/bsp/qemu-vexpress-a9$ chmod +x qemu.sh
jiao@jiao:~/workspace/repository/rt-thread/bsp/qemu-vexpress-a9$ ls
applications  drivers   packages      qemu-nographic.sh  README.md     rtconfig.py~   rtthread.elf
build         Kconfig   qemu.bat      qemu.sh            rtconfig.h    rtconfig.pyc   rtthread.map
cpu           link.lds  qemu-dbg.bat  qemu.sh~           rtconfig.py   rtthread.bin   SConscript
jiao@jiao:~/workspace/repository/rt-thread/bsp/qemu-vexpress-a9$
```

Figure 33: *qemu.sh* Add execute permissions to the file

3. Enter the ./qemu.sh command to execute the script file. At this time, the virtual machine will run, as shown in the figure below.

The lines show the relevant information printed during the RT-Thread operating system startup process, and the white window is a virtual LCD screen.



```
CC build/packages/pahomqtt/example/paho_mqt
LINK rtthread.elf
arm-none-eabi-objcopy -O binary rtthread.el
arm-none-eabi-size rtthread.elf
   text    data     bss     dec     hex fil
1348956    3877   56196 1409029  158005 rtt
scons: done building targets.
jiao@jiao:~/workspace/repository/rt-thread/
pulseaudio: set_sink_input_volume() failed
pulseaudio: Reason: Invalid argument
pulseaudio: set_sink_input_mute() failed
pulseaudio: Reason: Invalid argument

 \ | /
- RT -     Thread Operating System
 / | \     3.0.2 build Feb  1 2018
 2006 - 2017 Copyright by rt-thread team
lwIP-2.0.2 initialized!
hello rt-thread
msh />
```

Figure 34: Virtual machine running interface

4References

- Env tool user manual

- File System Application Notes

5 Frequently Asked Questions

- For issues related to the Env tool, please refer to the Common Resources section of the Env Tool User

Manual. • When compiling a project, the error message "fatal error: rtgui/driver.h: No such file or

directory" appears. • Solution: After enabling "Enable GUI Engine" using menuconfig, you need to use the command pkgs --update

Download the GUI package.