

---

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

---

**RT-THREAD** Documentation Center

Copyright ©2019 Shanghai Ruiside Electronic Technology Co., Ltd.



[WWW.RT-THREAD.ORG](http://WWW.RT-THREAD.ORG)

Friday 28th September, 2018

Table of contents

Table of contents	i
1 Purpose and structure of this paper . . . . .	1
1.1 Purpose and background of this paper . . . . .	1
1.2 Structure of this paper . . . . .	1
2 Use QEMU to run RT-Thread on Windows platform. . . . .	1
2.1 Preparation . . . . .	1
2.2 Compile and run RT-Thread. . . . .	3
2.2.1. Step 1 Use the scons command to compile the project. . . . .	3
2.2.2. Step 2: Use the qemu.bat command to run the project. . . . .	4
2.3 Run RT-Thread Finsh console. . . . .	6
2.4 Run RT-Thread file system. . . . .	7
2.5 Run RT-Thread network. . . . .	9
2.5.1. Step 1: Install and configure the TAP network card. . . . .	9
2.5.2. Step 2: Modify the qemu.bat script file. . . . .	12
2.5.3. Step 3 Check the IP address. . . . .	12
2.5.4. Notes . . . . .	12
2.6 Run the RT-Thread Ping tool. . . . .	13
2.6.1. Step 1: Download the network tool software package. . . . .	13
2.6.2. Step 2: Run the ping tool. . . . .	16
2.7 Run RT-Thread GUI engine. . . . .	17
2.7.1. Step 1: Download the GUI engine package. . . . .	17
2.7.2. Step 2: Run the GUI engine. . . . .	19
3 Run RT-Thread using QEMU on Ubuntu platform. . . . .	20

- 3.1 Preparation . . . . . 20
- 3.2 Use menuconfig to configure the project. . . . . twenty one
- 3.3 Compile and run RT-Thread. . . . . twenty two
- 4 References. . . . . twenty four
- 5 Frequently Asked Questions. . . . . twenty four

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 is inseparable from the development board. In the absence of a physical development board, you can use a virtual machine such as QEMU to simulate the development board. QEMU is a virtual machine that supports cross-platform virtualization. It can virtualize many development boards. In order to facilitate everyone to experience RT-Thread without a development board, RT-Thread provides a board support package (BSP) for the ARM vexpress A9 development board simulated by QEMU. This article mainly introduces how to use QEMU to run the RT-Thread qemu-vexpress-a9 BSP project on the Window platform and Ubuntu platform, and introduces how to use a virtual network card to connect QEMU to the network.

### 1.2 Structure of this paper

This article first introduces the usage details of qemu-vexpress-a9 BSP on the Window 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, serial port, etc. 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:

File Directory	describe
.vscode	vscode configuration file
applications	User application code directory
cpu	Chip related
drivers	The underlying driver provided by RT-Thread
qemu.bat	Running script files on Windows platform
qemu.sh	Running script files on Linux platform
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 **scons** command to compile the project

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

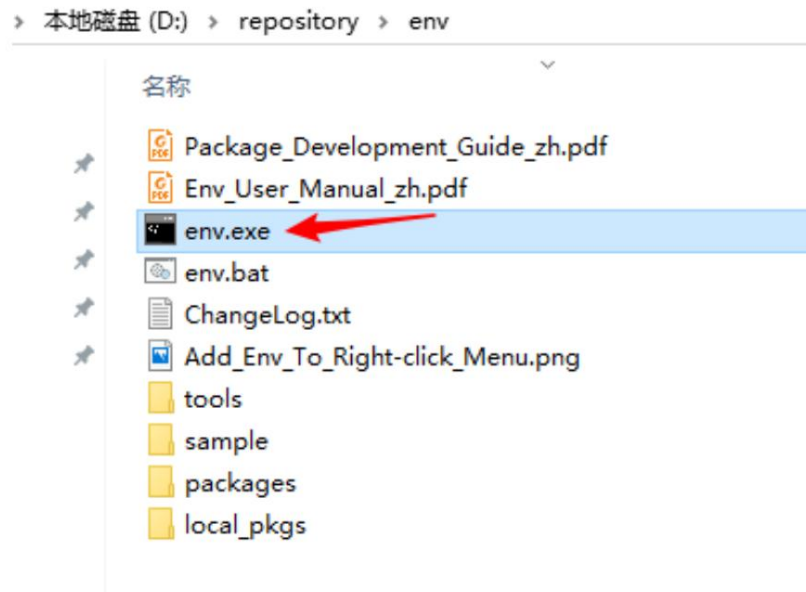


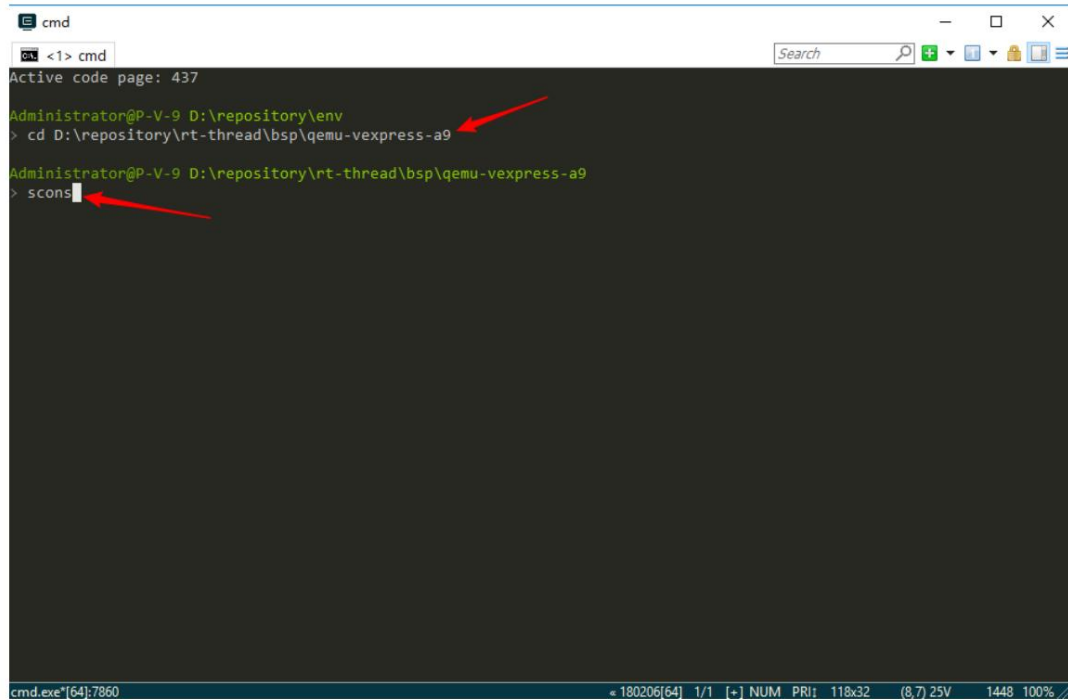
Figure 2: Env folder

Switch directory in Env console and enter command `cd D:\repository\rt-thread\bsp\qemu-vexpress`

-a9 Switch to the qemu-vexpress-a9 BSP root directory under the RT-Thread source folder, and then enter the `scons` command

Run the command to compile the project. If the compilation is correct, the `rtthread.elf` target running under QEMU will be generated in the BSP directory.

document.



```
cmd
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
```

The screenshot shows a Windows command prompt window titled 'cmd'. The active code page is 437. The user is an Administrator on a machine named P-V-9. The current directory is D:\repository\env. The user has navigated to D:\repository\rt-thread\bsp\qemu-vexpress-a9 and entered the 'scons' command to compile the project. Two red arrows point to the 'cd' and 'scons' commands. The taskbar at the bottom shows the command prompt window is running, with system information like CPU usage (180206/64), memory (1/1), and other metrics.

image 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 shown, and the **qemu** virtual machine is also running

As shown in the following picture:

```

cmd - qemu.bat
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@PP-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 restricted.
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 mmcblk0 block device!
hello rt-thread
msh />
qemu-system-arm.exe[32]:744
+ 180206[64] 1/1 [+] NUM PRI: 118x32 (7,190) 25V 4928 100%

```

Figure 4: Run the project

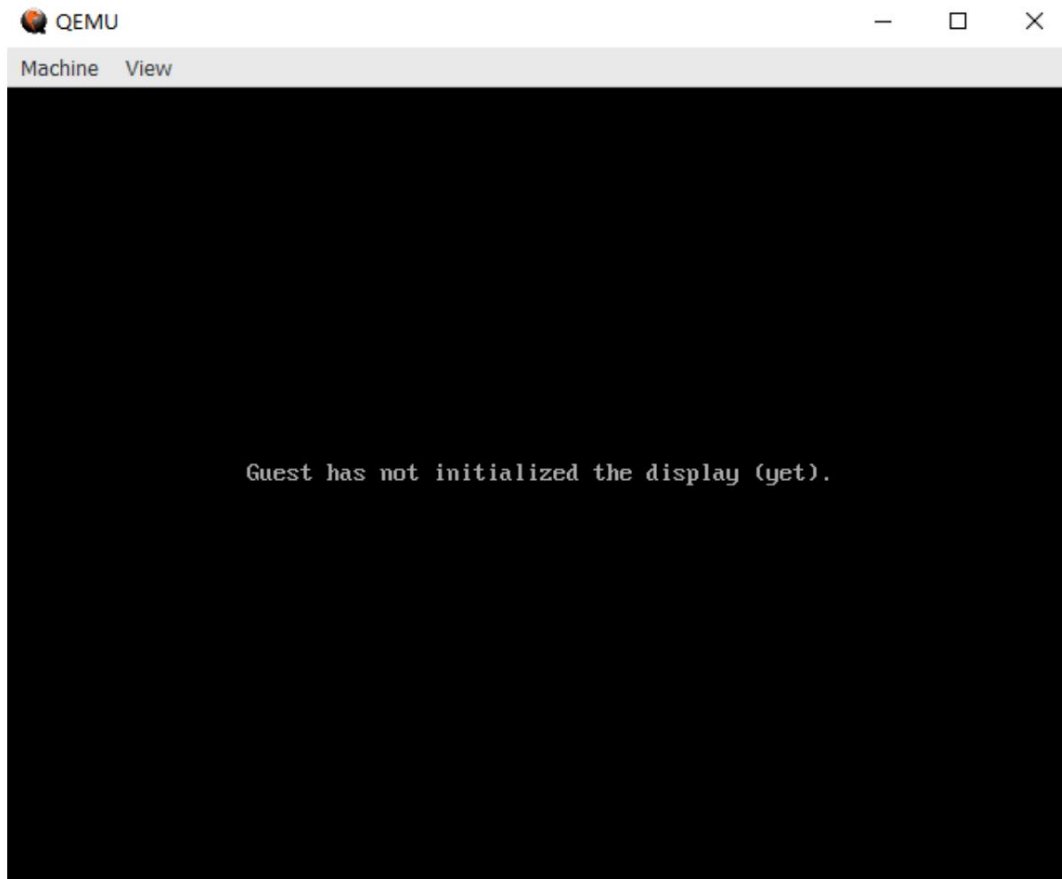


Figure 5: virtual machine

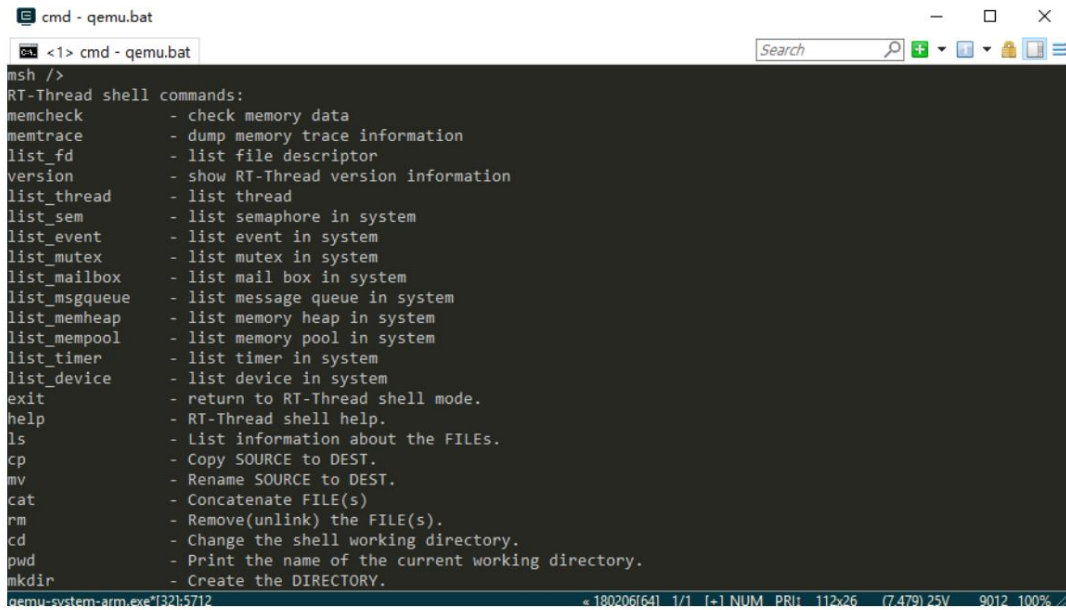


Note: If 360 Security Guard is installed on your computer, there will be a warning. Please click to allow the program to run.

## 2.3 Run RT-Thread Finsh console

RT-Thread supports Finsh, and users can use commands 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 is the command and the right side is the command description.



```

cmd - qemu.bat
<1> cmd - qemu.bat
Search
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*1321:5712  1802061641 1/1 [+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.

```

cmd - qemu.bat
<1> cmd - qemu.bat
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 Run 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 Formatted into FatFS file system. FatFS is a file system compatible with Microsoft FAT developed specifically for small embedded devices. 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](http://elm-chan.org/fsw/ff/00index_e.html)

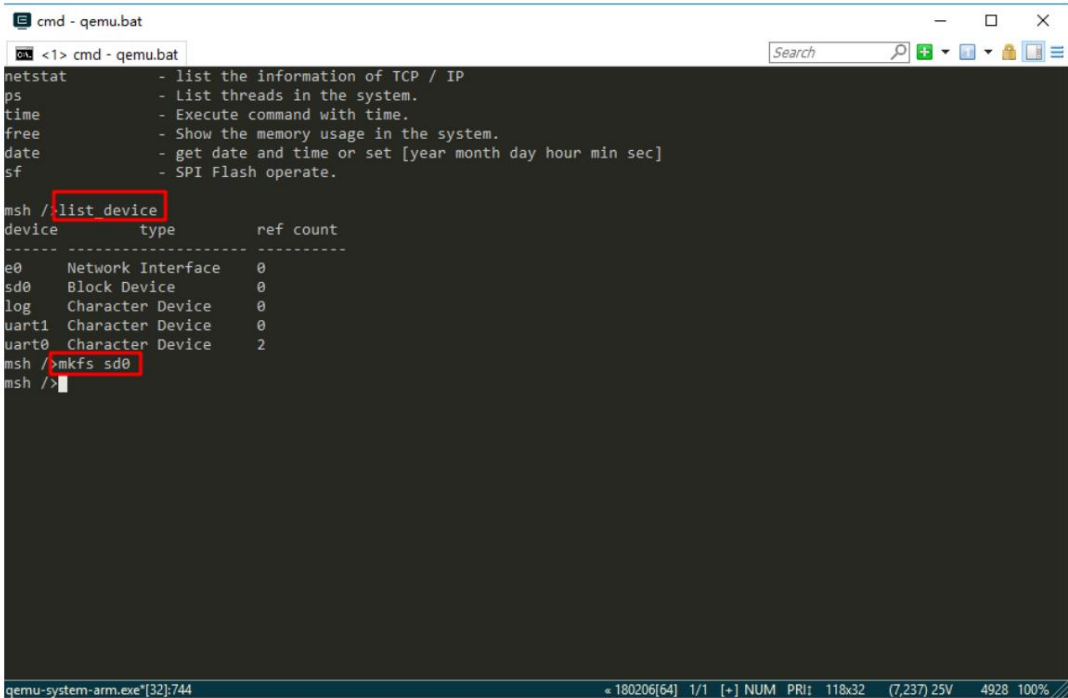


Figure 8: format sd Card

After formatting the SD card for the first time, the file system will not be loaded immediately, and it will be loaded correctly only 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. Enter the ls command to You can see that the 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

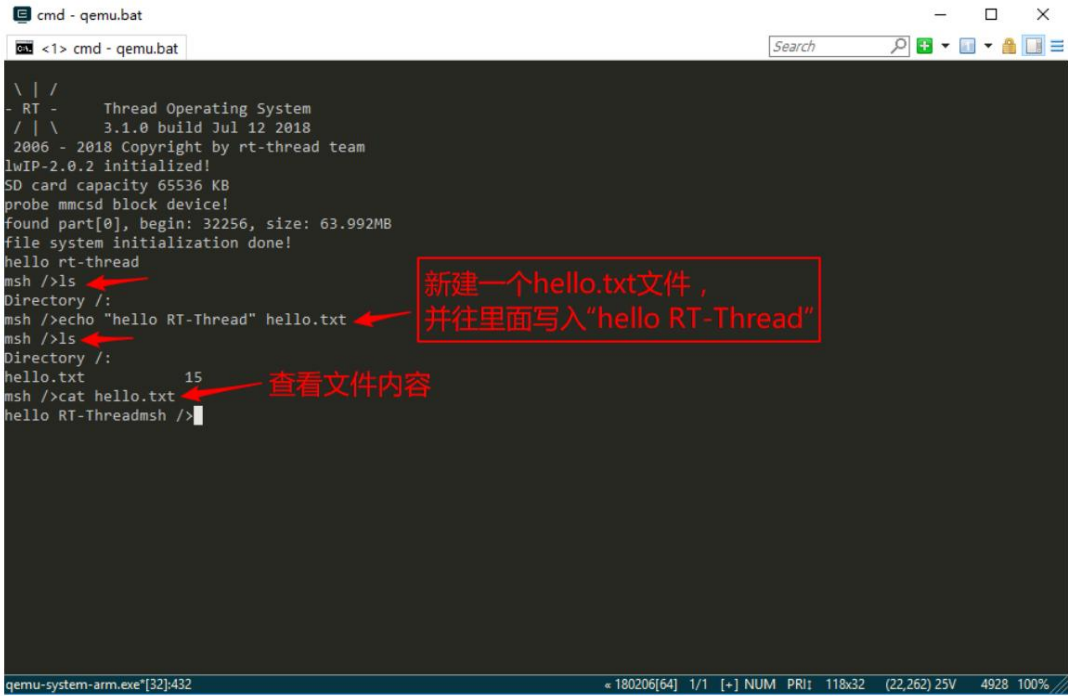


Figure 9: Other file system commands

## 2.5 Run RT-Thread Network

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

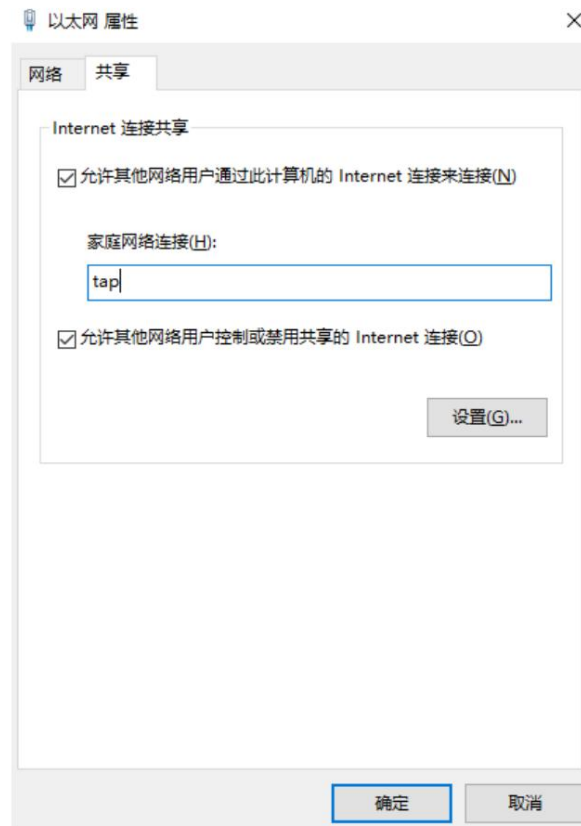
1. Download the TAP network card [tap-windows-9.21.2.exe](#). After downloading, double-click the installer and follow the default installation procedure.
2. Open the Network and Sharing Center to change the adapter settings and rename the installed virtual network card to tap, as shown in the following figure:



Figure 10: *tap\_rename*

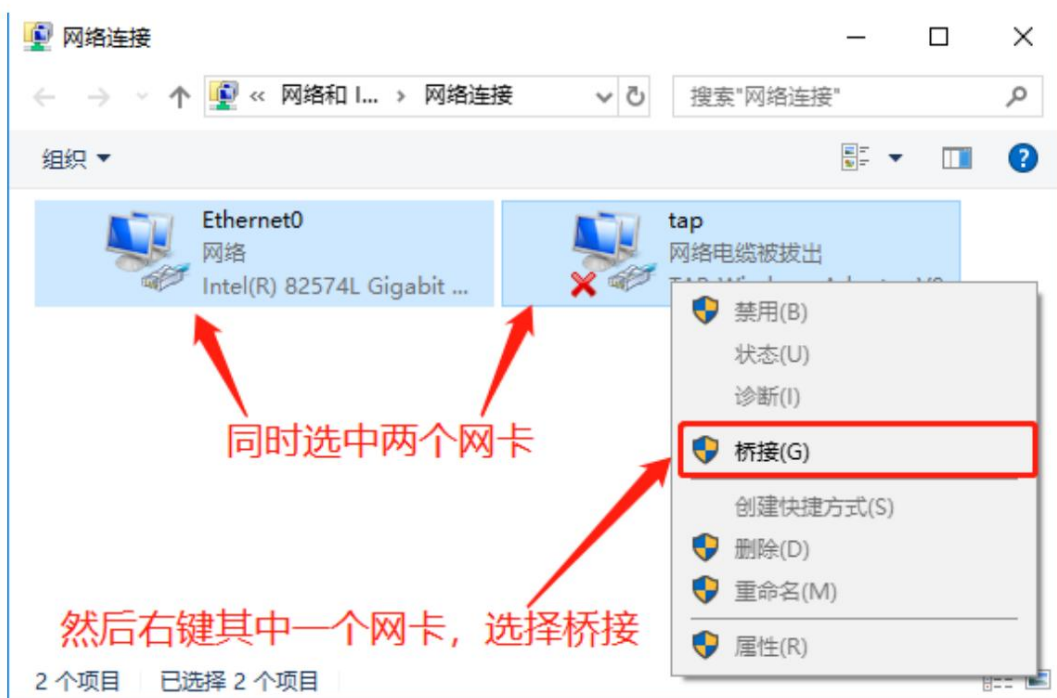
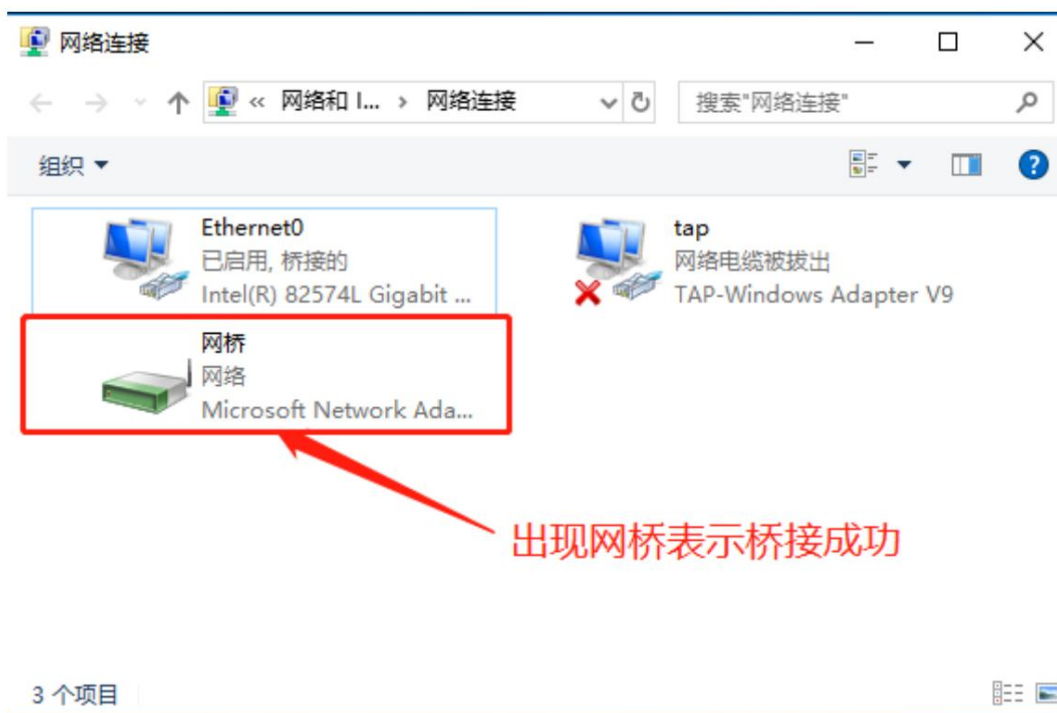
Method A: Right-click the network connection that can access the Internet (this article uses Ethernet), open Properties -> Sharing, and select Home

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

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,

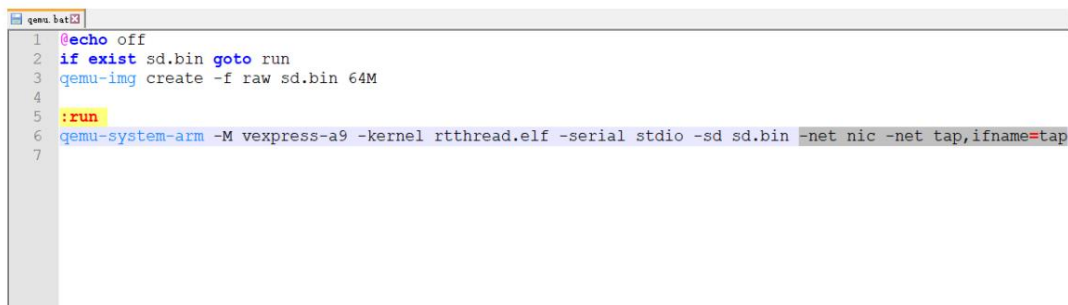
A network bridge appears. As shown in the following picture:

Figure 12: *tap\_* bridgingFigure 13: *tap\_* Bridge Success

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** at the location shown in the figure below  
**-net tap,ifname=tap** configuration. **ifname=tap** means the name of the network card is tap.



```

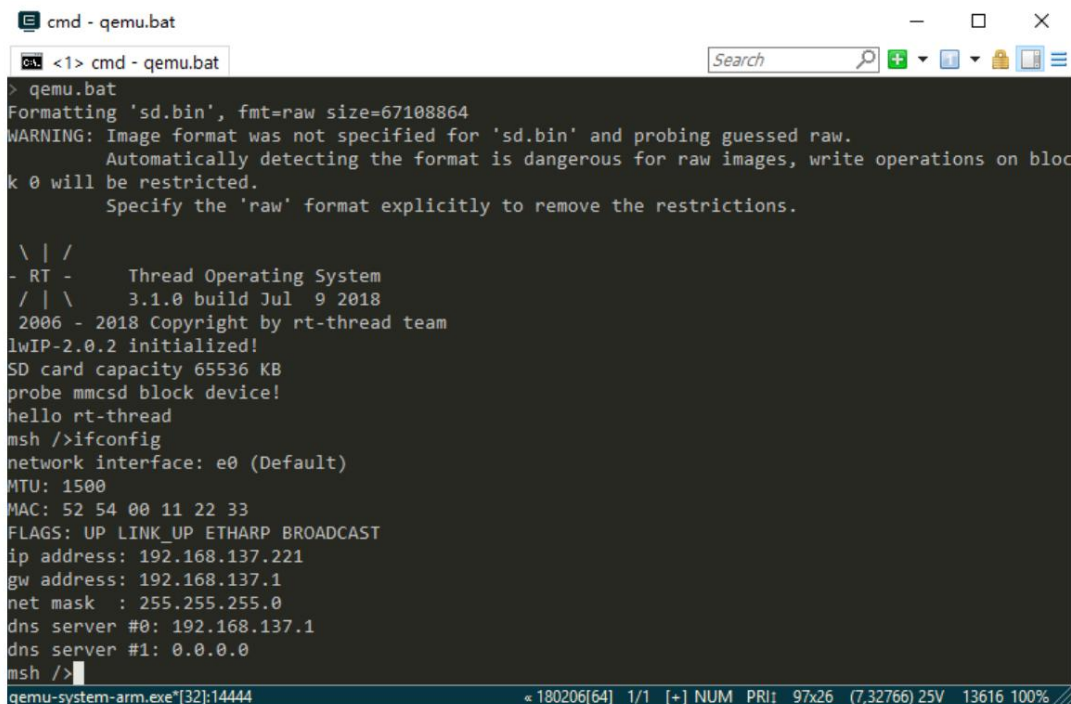
1 @echo off
2 if exist sd.bin goto run
3 qemu-img create -f raw sd.bin 64M
4
5 :run
6 qemu-system-arm -M vexpress-a9 -kernel rtthread.elf -serial stdio -sd sd.bin -net nic -net tap,ifname=tap
7

```

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 get it normally  
 If the IP is reached, it means the network driver is normal and the configuration is complete. The effect is as shown in the figure below:



```

cmd - qemu.bat
> 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
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 mmcsc 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

```

Figure 15: *ifconfig*

### 2.5.4. Notes

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

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

## 2.6 Run 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 in the following figure:

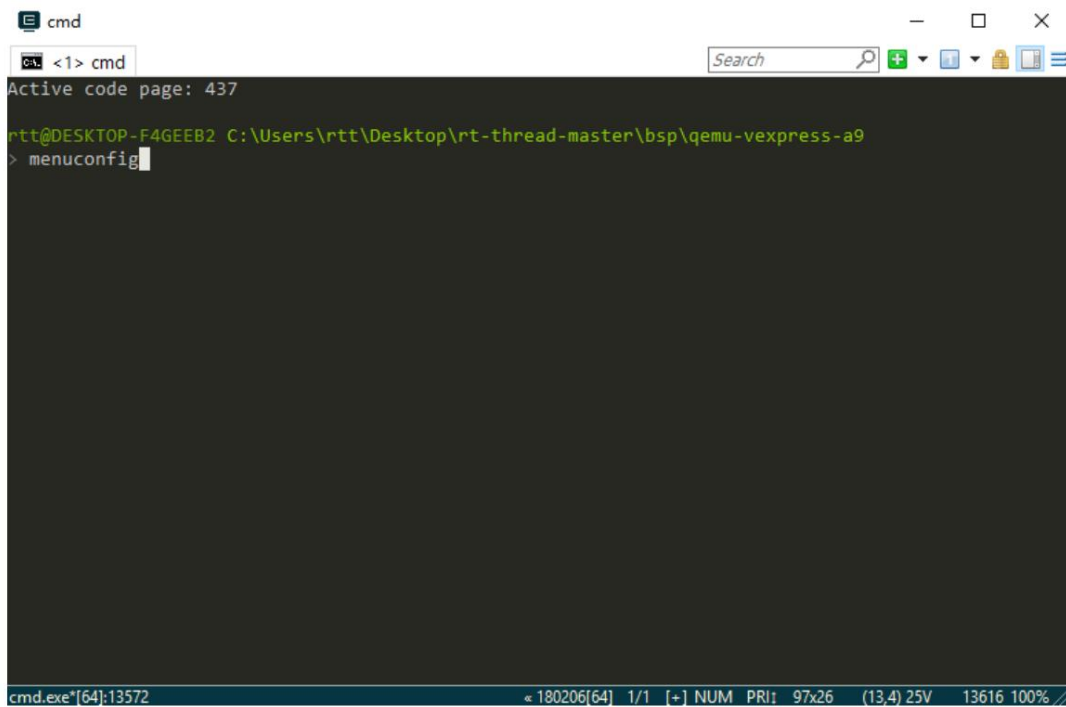
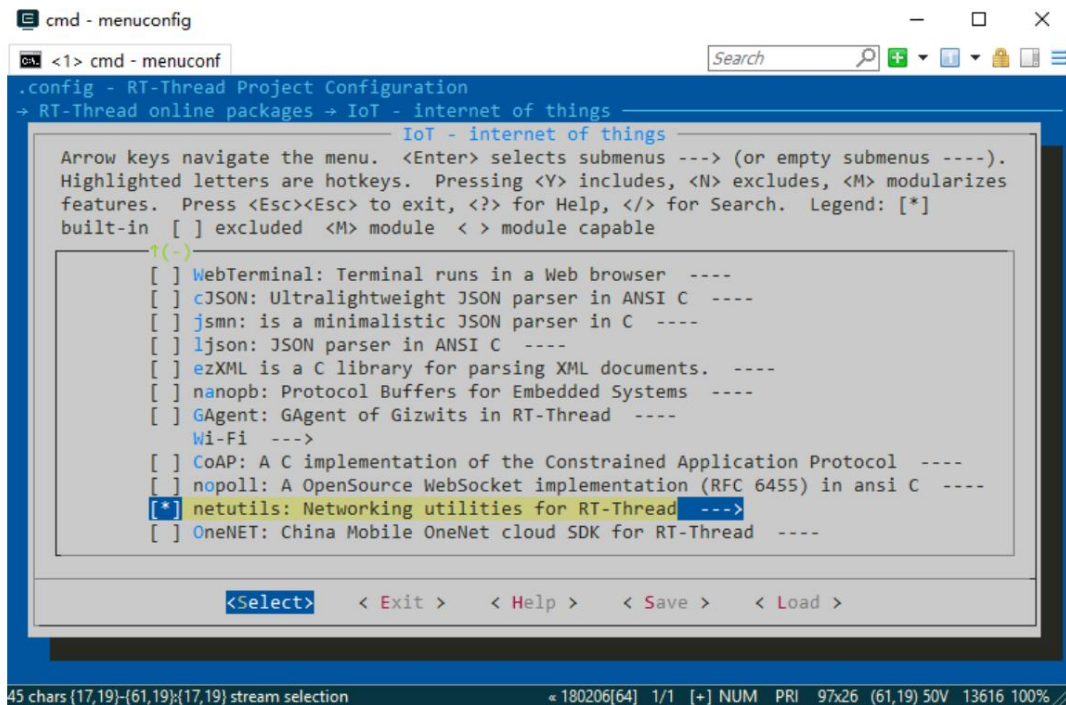


Figure 16: *Env\_menuconfig*

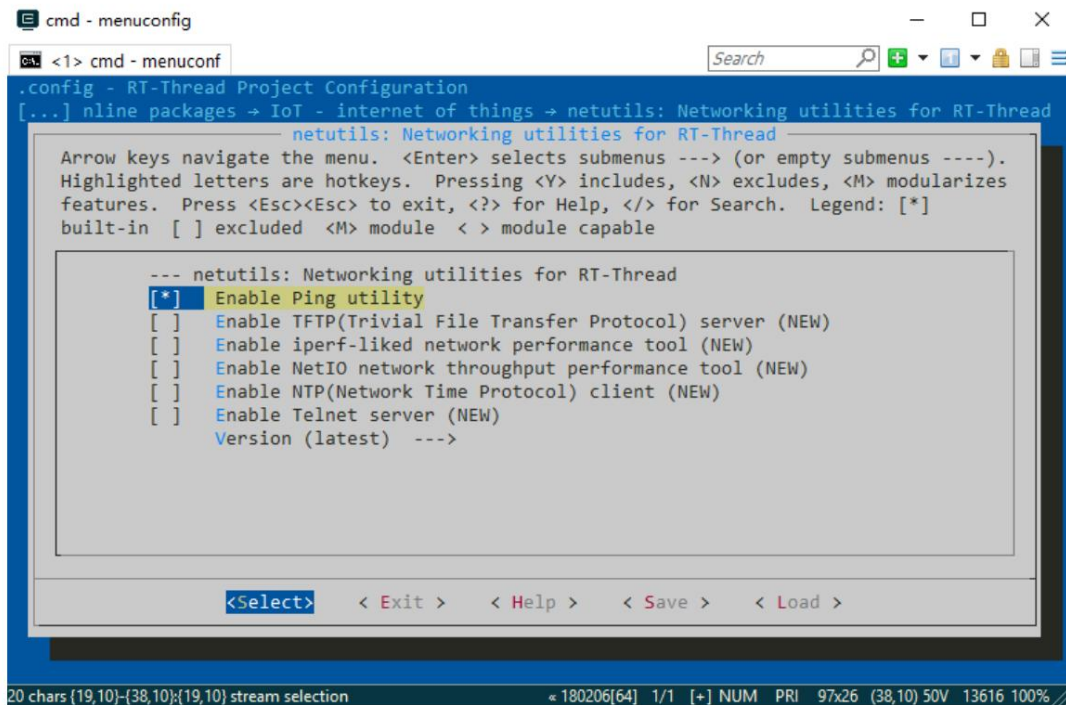
2. Open the netutils: Networking utilities for RT-Thread function in the RT-Thread online packages->IoT - internet of things page, as shown in the following figure:



Figure 17: *online\_packages*

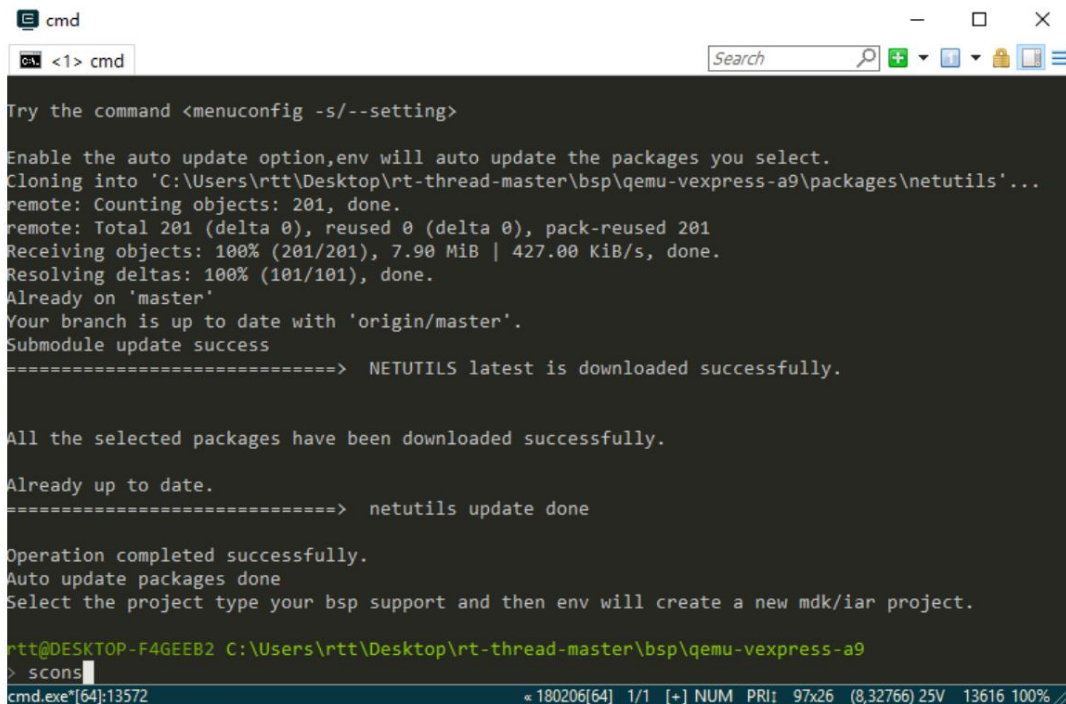
3. Go to netutils: Networking utilities for RT-Thread page and turn on Enable Ping utility.

Yes, as shown in the figure below:

Figure 18: *enable\_ping*

4. Save and exit the configuration interface. If the Env automatic package update 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
C:\> cmd

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

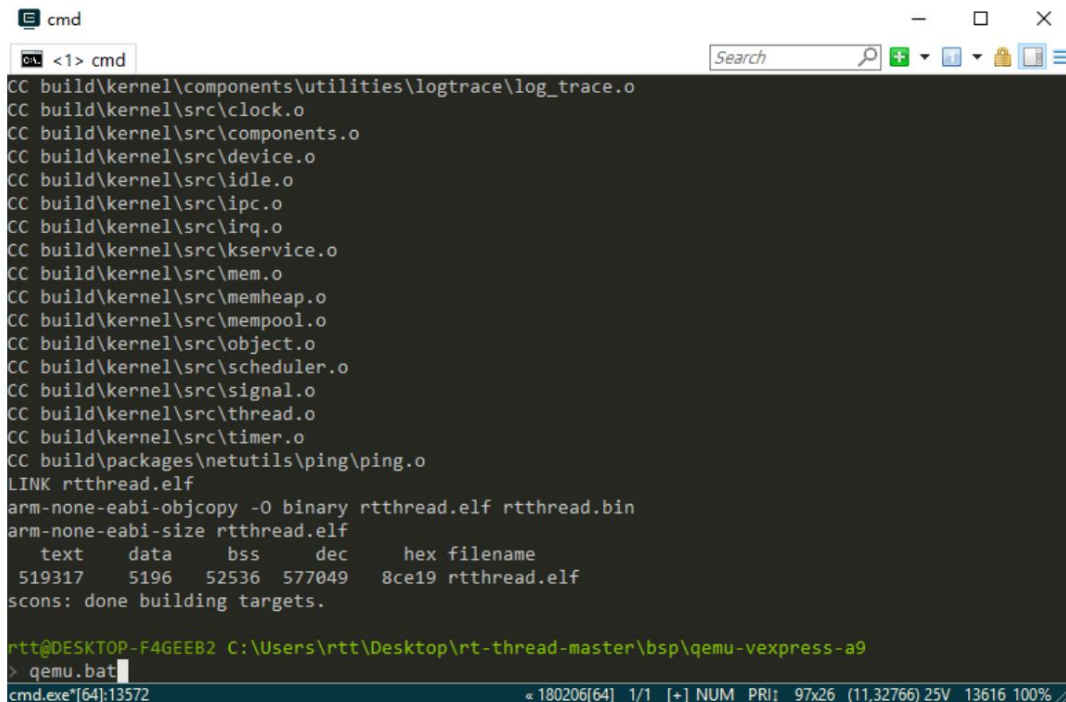
Operation completed successfully.
Auto update packages done
Select the project type your bsp support and then env will create a new mdk/iar project.

rtt@DESKTOP-F4GEEB2 C:\Users\rtt\Desktop\rt-thread-master\bsp\qemu-vexpress-a9
> scons
cmd.exe*[64]:13572

```

Figure 19: `scons`

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



```

cmd
C:\> cmd

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.

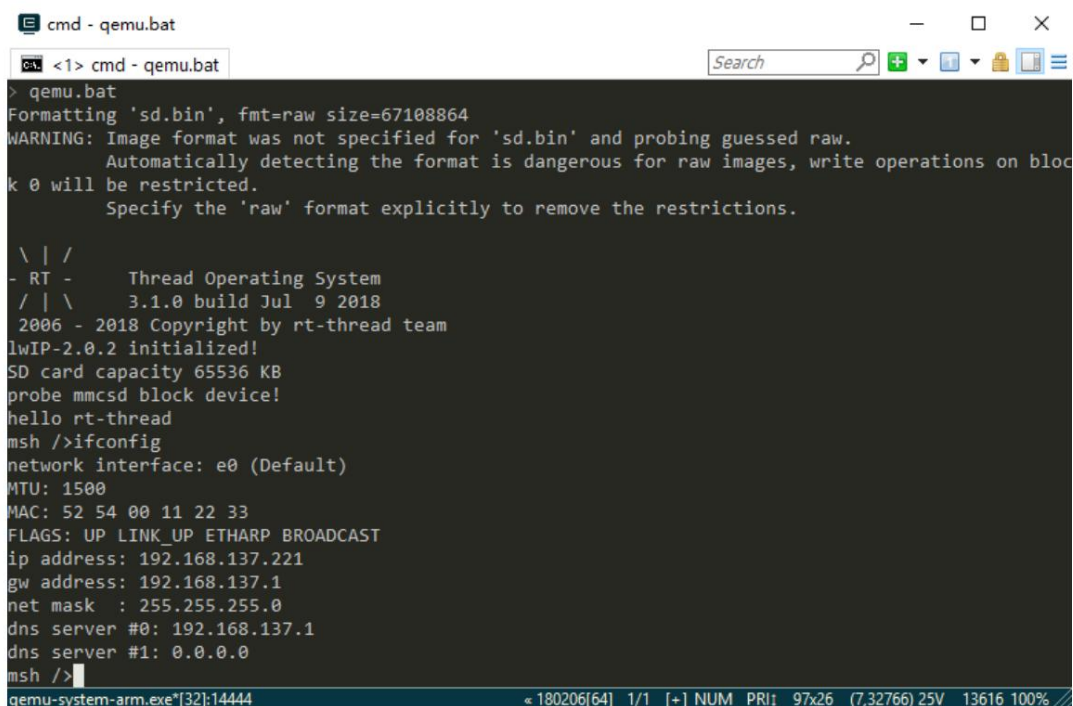
rtt@DESKTOP-F4GEEB2 C:\Users\rtt\Desktop\rt-thread-master\bsp\qemu-vexpress-a9
> qemu.bat
cmd.exe*[64]:13572

```

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 is obtained normally, it means that the network driver is normal:



```

cmd - qemu.bat
> 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
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 mmcblk0 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

```

Figure 21: *ifconfig*

Enter ping [www.rt-thread.com](http://www.rt-thread.com) in the shell and you can see the ping result, indicating that the network configuration is

Success, ping is successful, as shown in the following figure:

```

cmd - qemu.bat

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 mmcblk0 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 />

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

```

Figure 22: ping

## 2.7 Run 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:

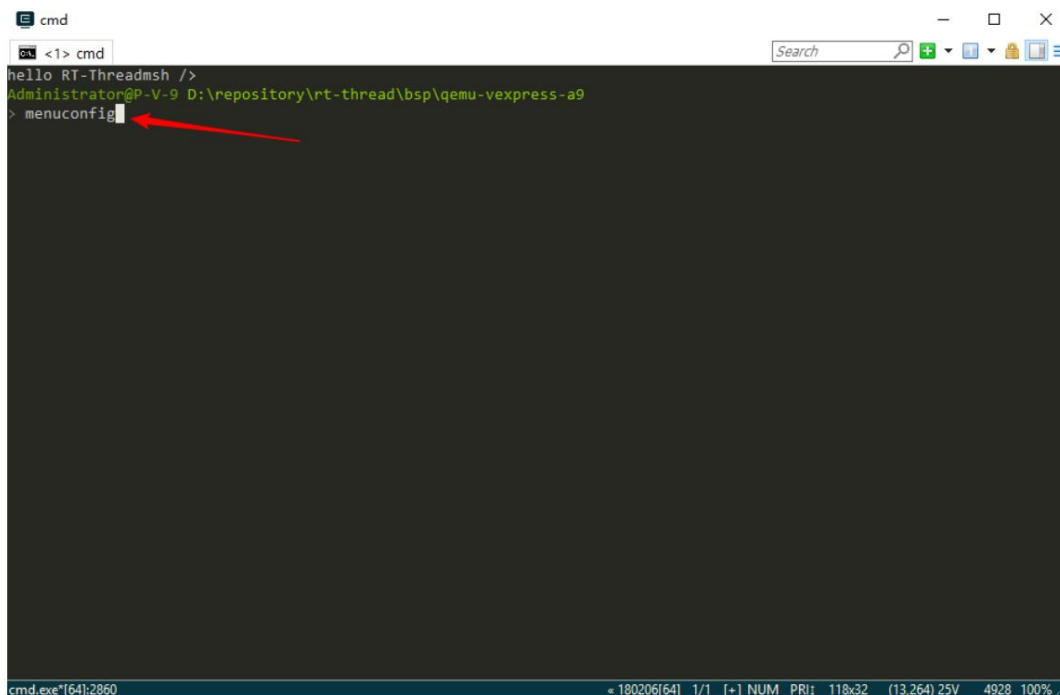


Figure 23: menuconfig

Enter "RT-Thread online packages" ÿ "system packages" ÿ "RT-Thread GUI Engine"  
Submenu, select "Enable GUI Engine" and "Enable the example of GUI Engine":

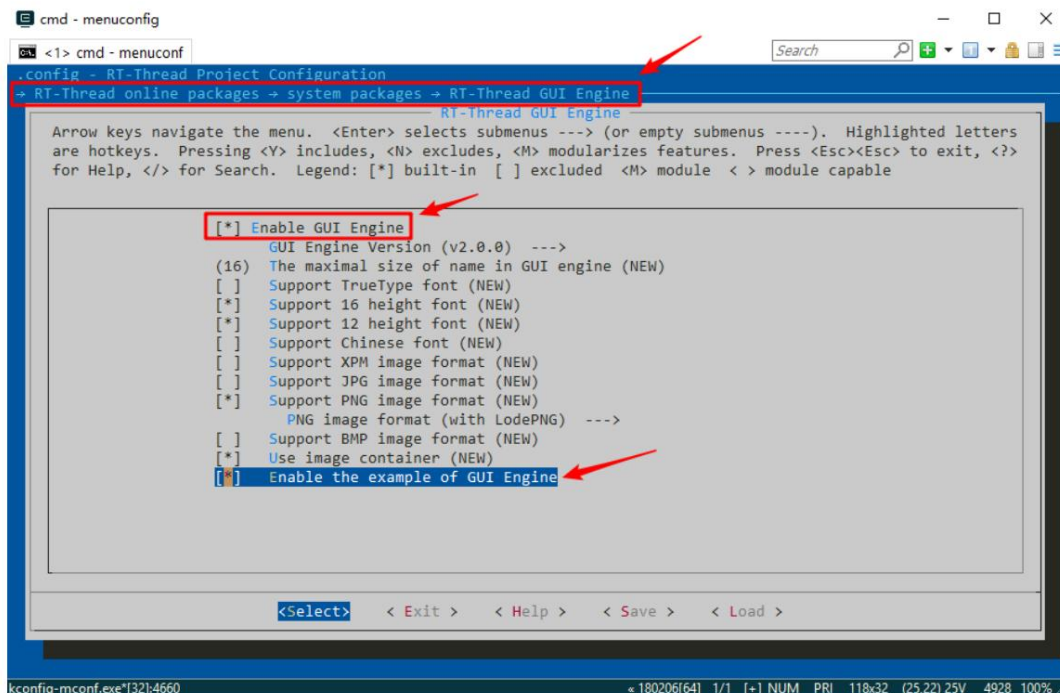


Figure 24: choose GUI Example

Press the 'y' 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

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
cmd.exe [64]:2860

```

Figure 25: download GUI Software Packages

## 2.7.2. Step 2: Run the GUI engine

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

```

cmd
<1> cmd

*** 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
cmd.exe [64]:9412

```

Figure 26: `scons` Command to compile the 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:

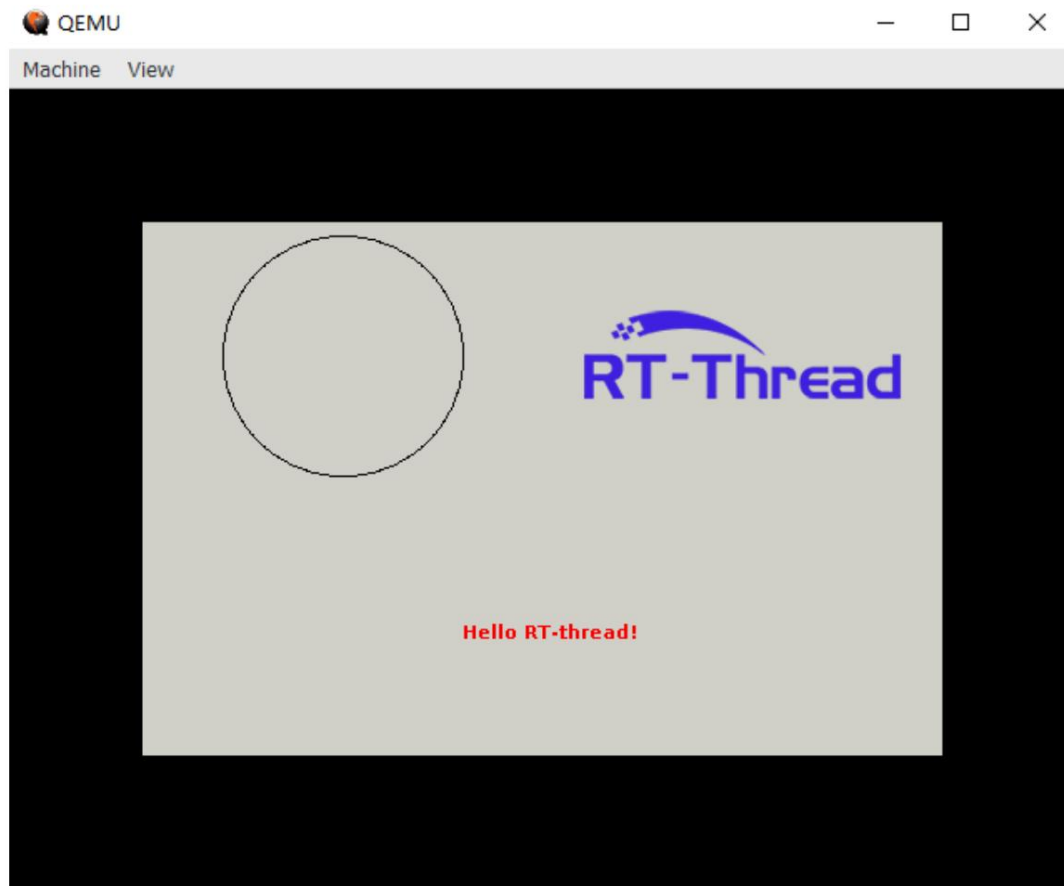


Figure 27: Virtual screen *GUI* Example display

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

### 3.1 Preparation

- Download RT-Thread source code, use command: `git clone https://github.com/RT-Thread/rt-thread.git`
- Install QEMU, use 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 the `apt-get` command is too old, compilation errors will occur. You can use the following steps in sequence:

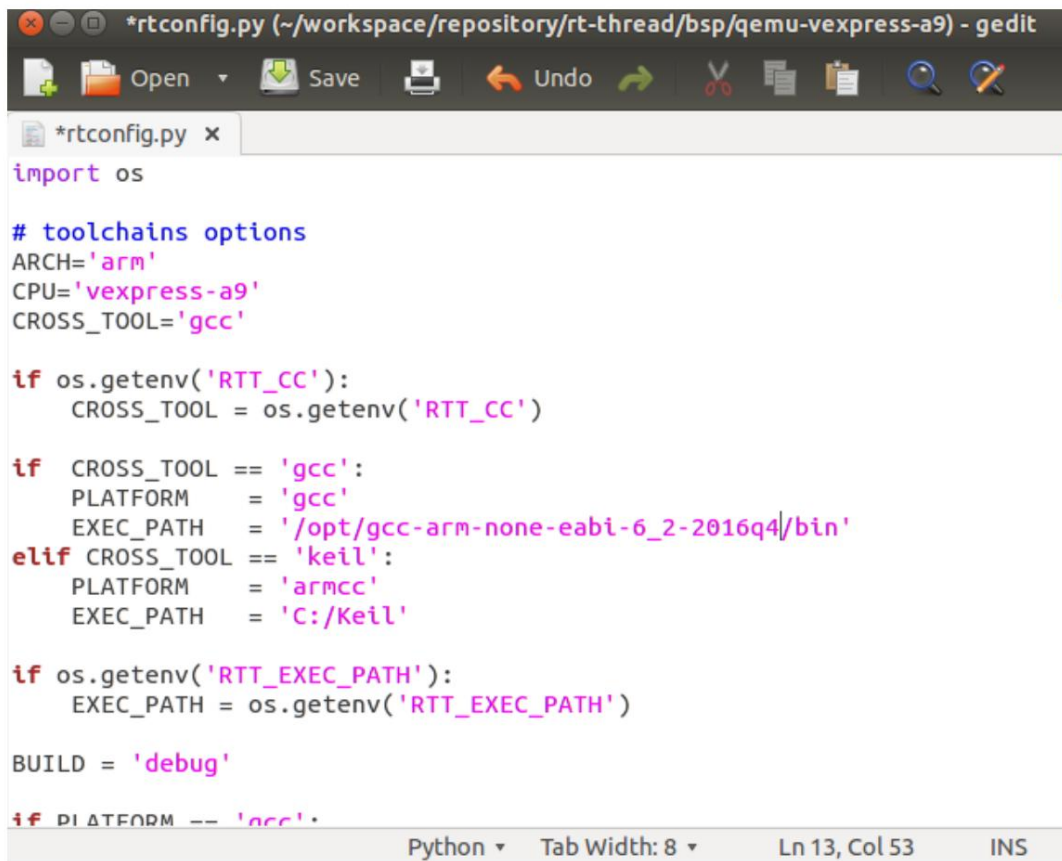
Run the command to download and install the new version. The download link and decompression folder name vary depending on the downloaded version:

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 of the compiler unzipped 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

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':
```

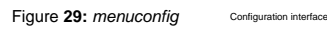
Figure 28: Compiler path modification

### 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 on Windows platform:





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

Figure 30: .env folder

Figure 31: *pkgs -update* Command to download the software packages

### Keywords

```
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 to compile the project

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

We need to add execution permissions to the `qemu.sh` file, 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 execution permissions to the file

3. Enter the `./qemu.sh` command to execute the script file. The virtual machine will start running, as shown in the figure below.

The lines show the relevant information printed during the startup process of the RT-Thread operating system, 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.elf
arm-none-eabi-size rtthread.elf
text data bss dec hex file
1348956 3877 56196 1409029 158005 rt
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 operation interface

#### 4 References

- [Env tool user manual](#)
- [File System Application Notes](#)

#### 5 Frequently Asked Questions

- For Env tool related issues, please refer to the Common Documentation Links section of the Env tool user manual.
- When compiling a project, it prompts fatal error: rtgui/driver.h: No such file or directory
- Solution: After using menuconfig to enable "Enable GUI Engine", you need to use the command `pkgs --update` Download the GUI package.