Debugging with **ECLIPSE + QEMU**

RT-THREAD

RT-THREAD Document Center

WWW.RT-THREAD.ORG

Friday 28th September, 2018

Table of contents

This application note describes how to use Eclipse to debug RT-Thread on Windows.

qemu-vexpress-a9 BSP project.

1 Purpose and structure of this paper

1.1 Purpose and Background of this Paper

Eclipse is a free, cross-platform integrated development environment (IDE). Originally designed for Java development, Eclipse can support a variety of computer languages, such as C++ and Python, by installing various plug-ins. While Eclipse itself is just a framework platform, the numerous plug-ins it supports give it a level of flexibility rarely found in other, more fixed-function IDEs.

This article mainly introduces how to use Eclipse to debug RT-Thread qemu-vexpress-a9 BSP on Windows platform.

Procedure.

1.2 Structure of this paper

This article mainly introduces the configuration of Eclipse debugging options and how to debug the project.

2. 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.
- Download Eclipse

3 Running and Debugging **RT-Thread**

3.1 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:

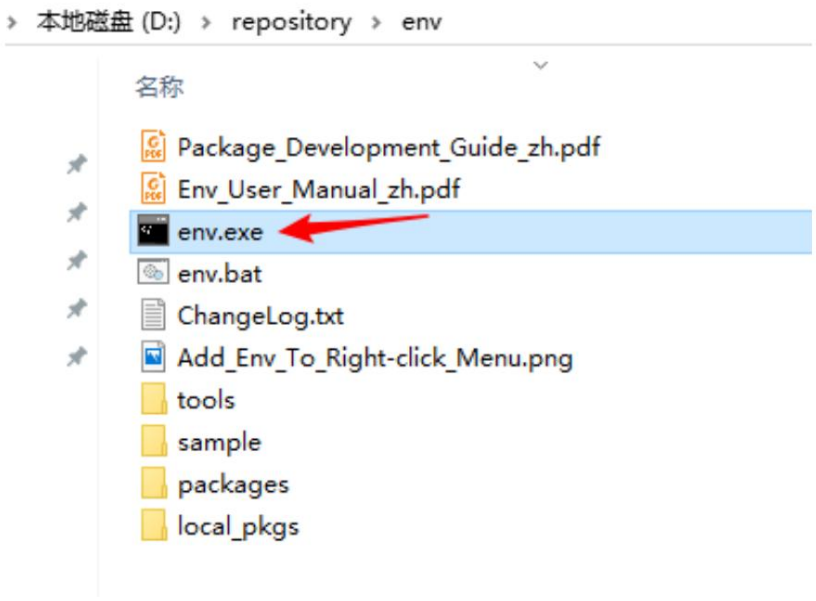

Figure **1:** *Env* Folder

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. File, this file is required for debugging the project.

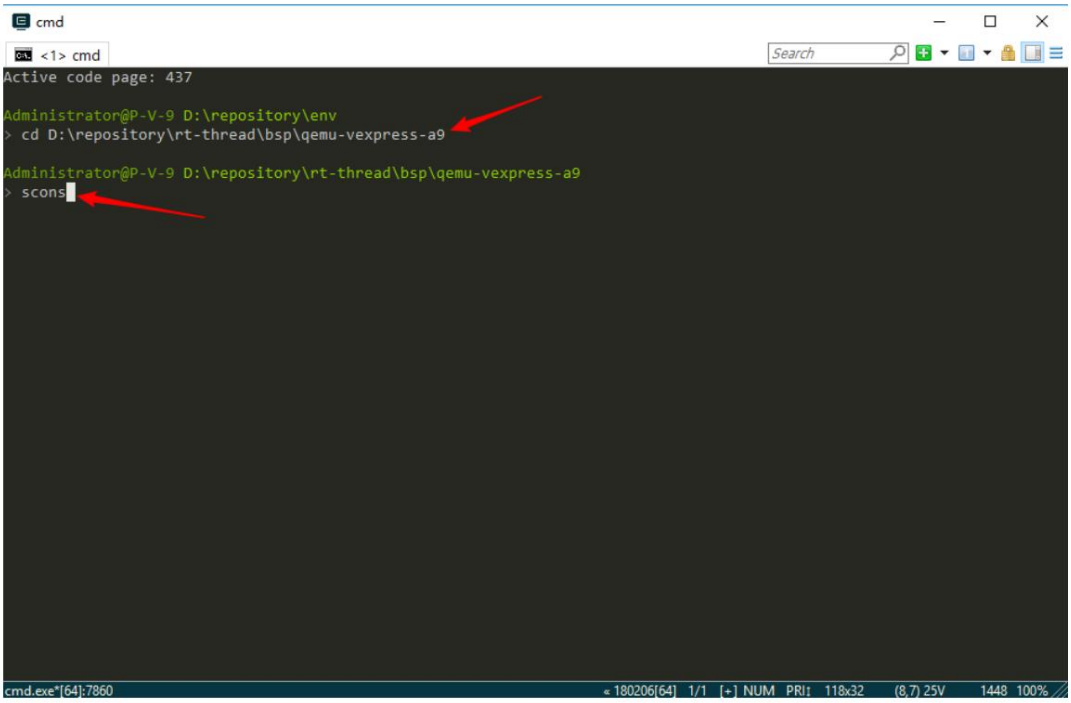

Figure **2:** Compile project

3.2 Step 2 Install the debugging plug-in

Download and install the debugging plugin that supports QEMU from the Eclipse Marketplace:

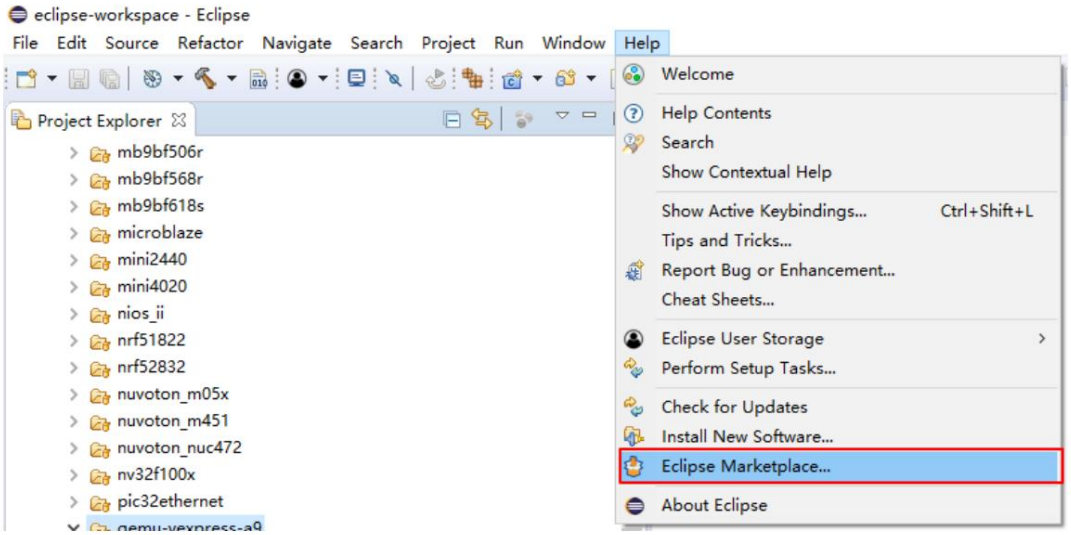

Figure **3:** *Eclipse Marketplace*

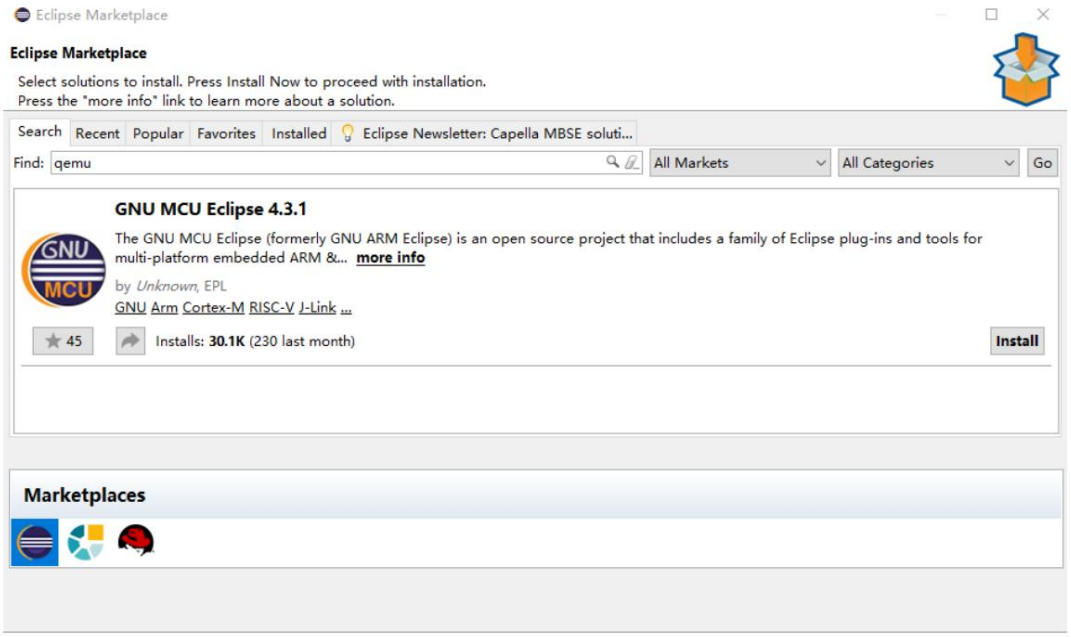

Figure **4:** Debugging tool installation

3.3 Step 3 Create a new **Eclipse** project

Follow the steps below to add the RT-Thread source code to Eclipse:

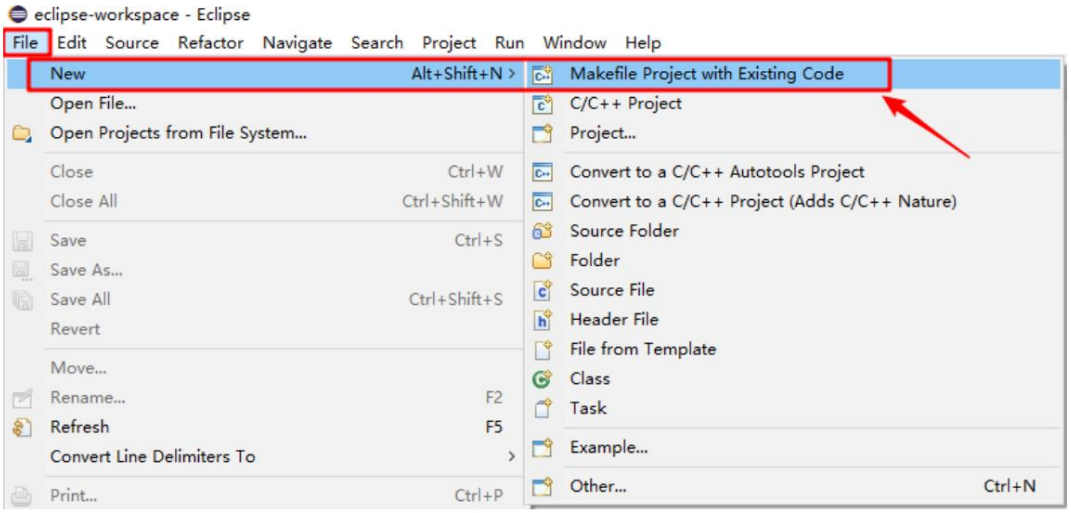

Figure **5:** New *eclipse* project

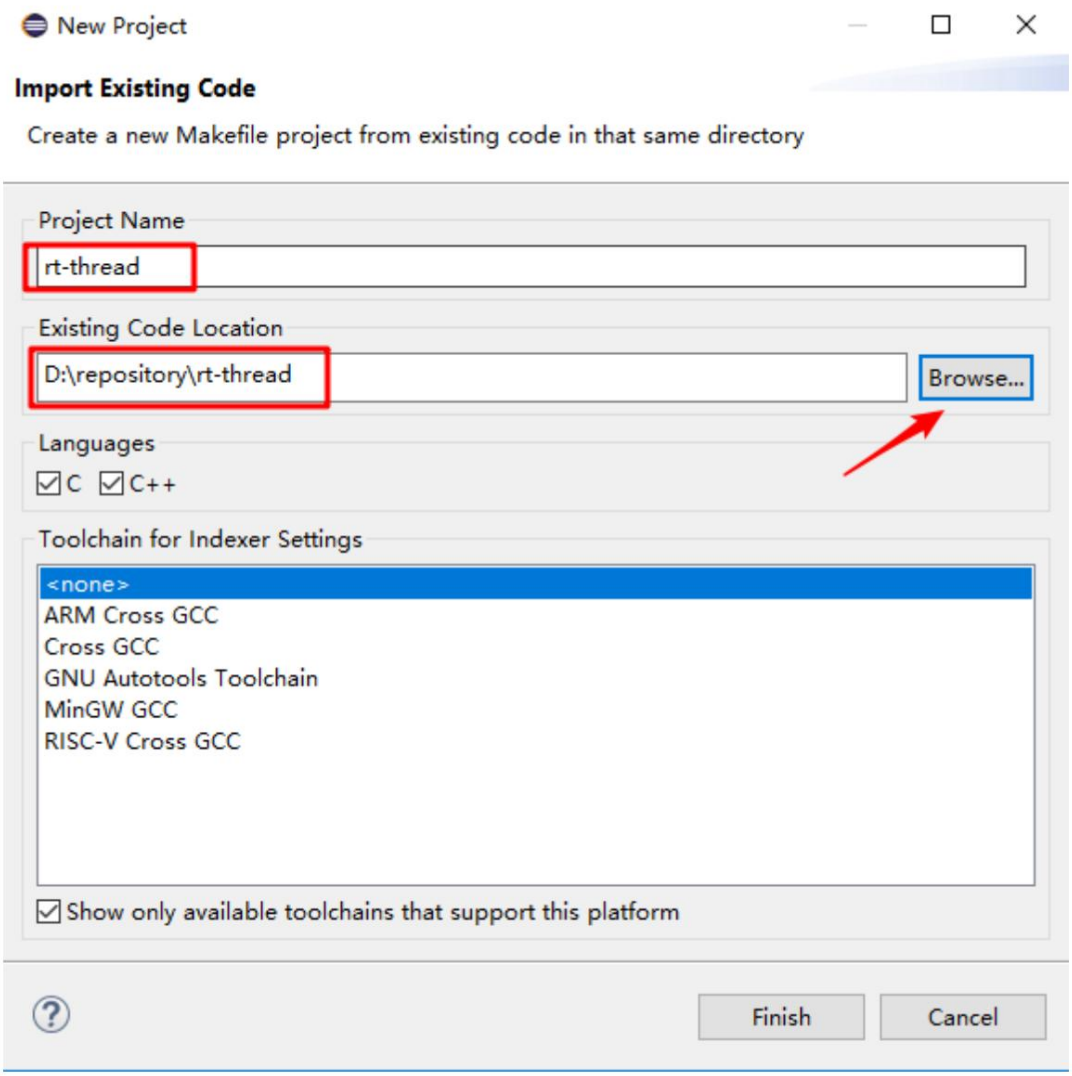

Figure **6:** Add to *RT-Thread* source code

3.4 Step 4 Create a new debugging project

Create a new debugging project and configure debugging parameters as shown in the following figure:

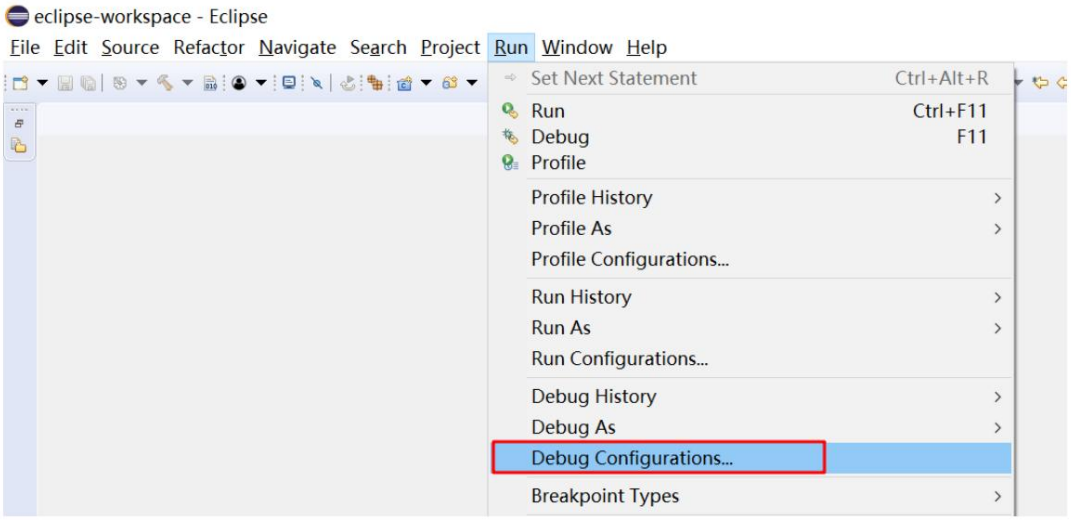

Figure 7: *Eclipse* Debug Configuration Options

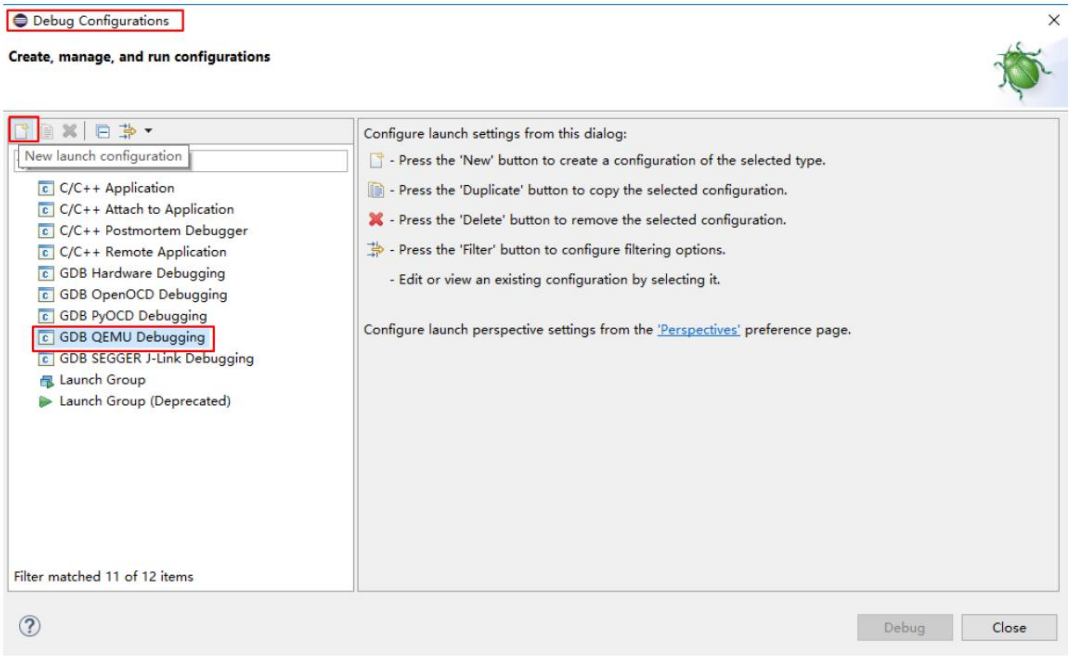

Figure 8: Create a new debugging project

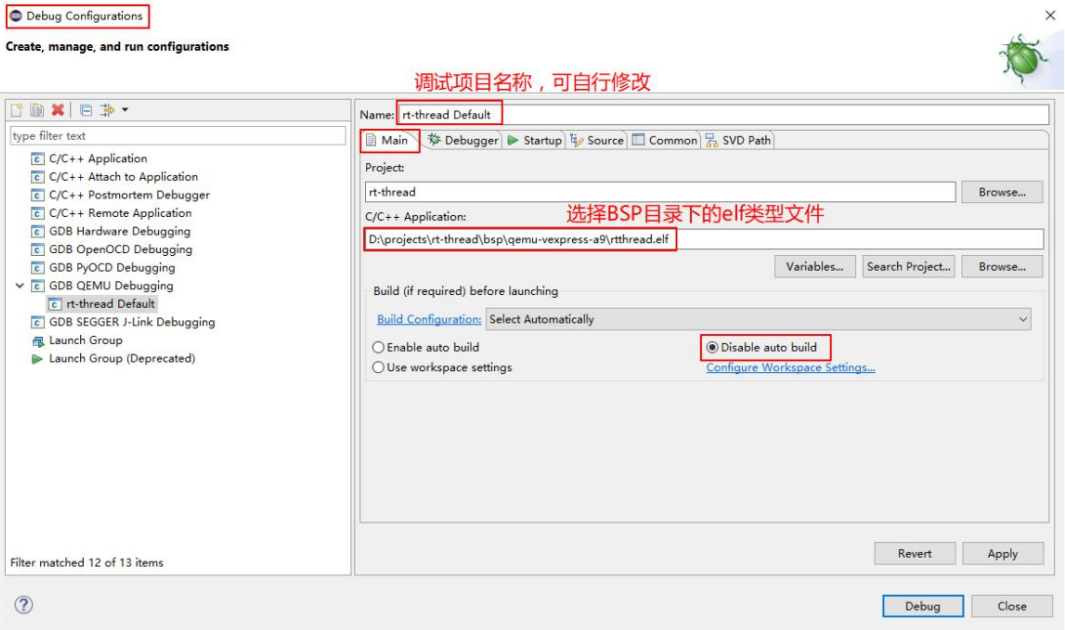

Figure **9:** Select debug file

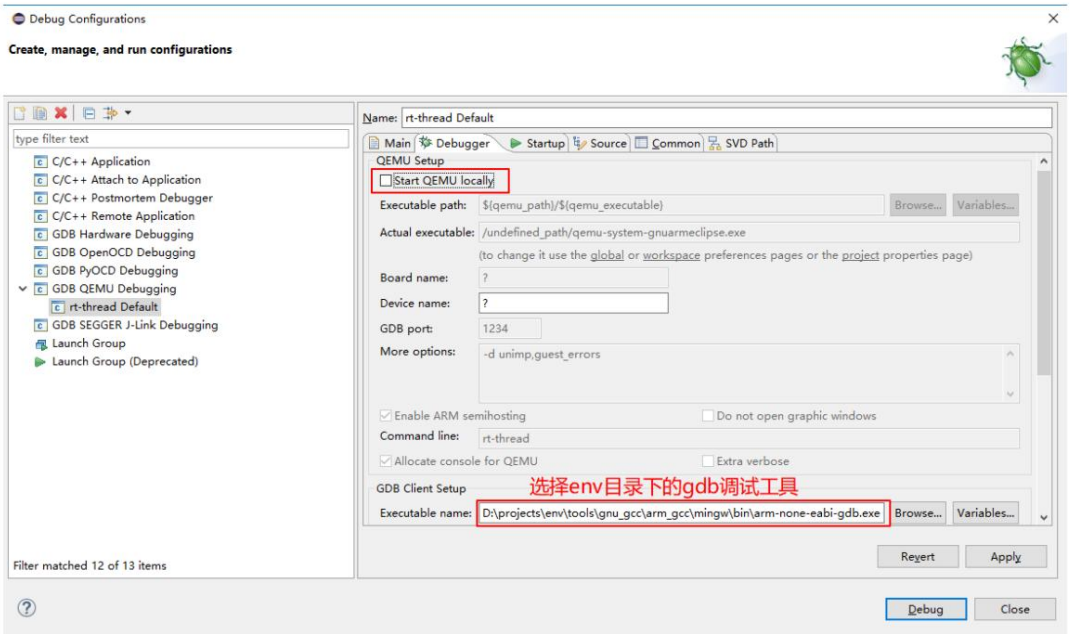

Figure **10:** Select debugging tools

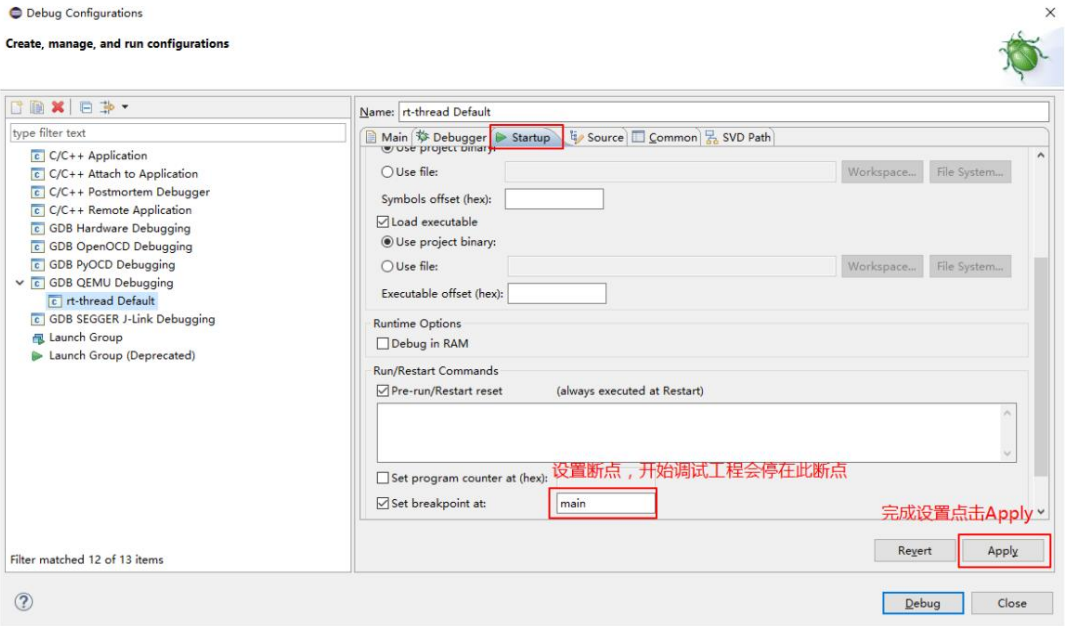

Figure **11:** Select a location

3.5 Step 5 Debugging the Project

1. After configuring the debugging related parameters, you can start debugging. Return to the Env command line interface and enter qemu-dbg.bat

Enable debug mode:

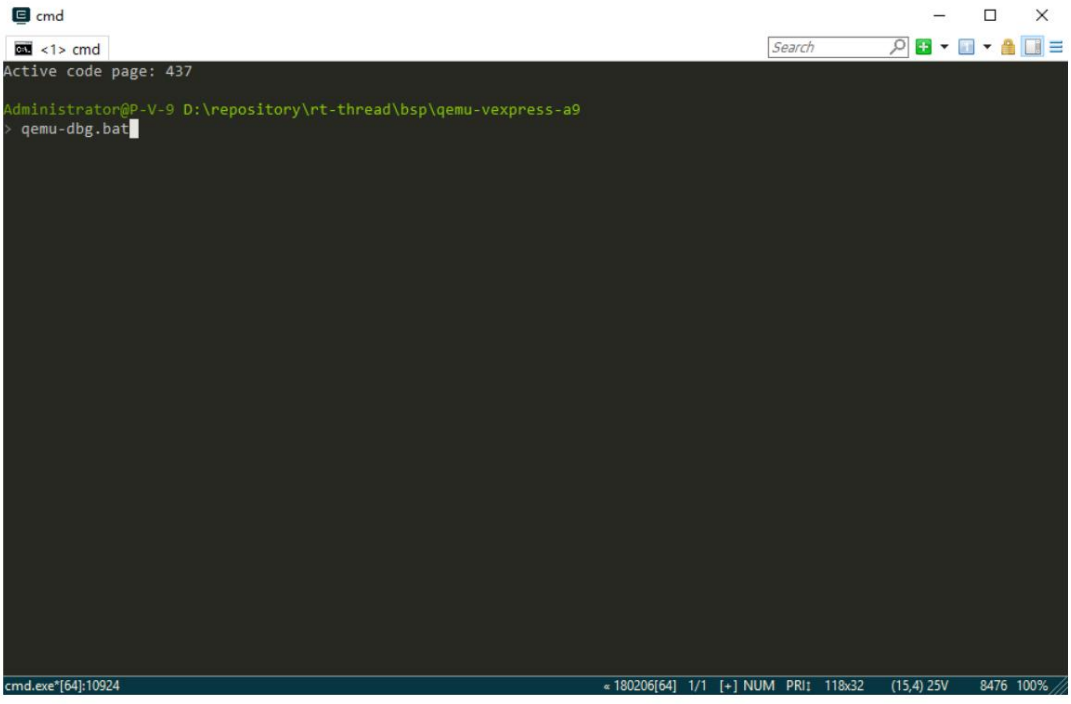

Figure **12:** Start debugging

At this time, the started QEMU virtual machine is in a paused state, waiting for connection debugging:

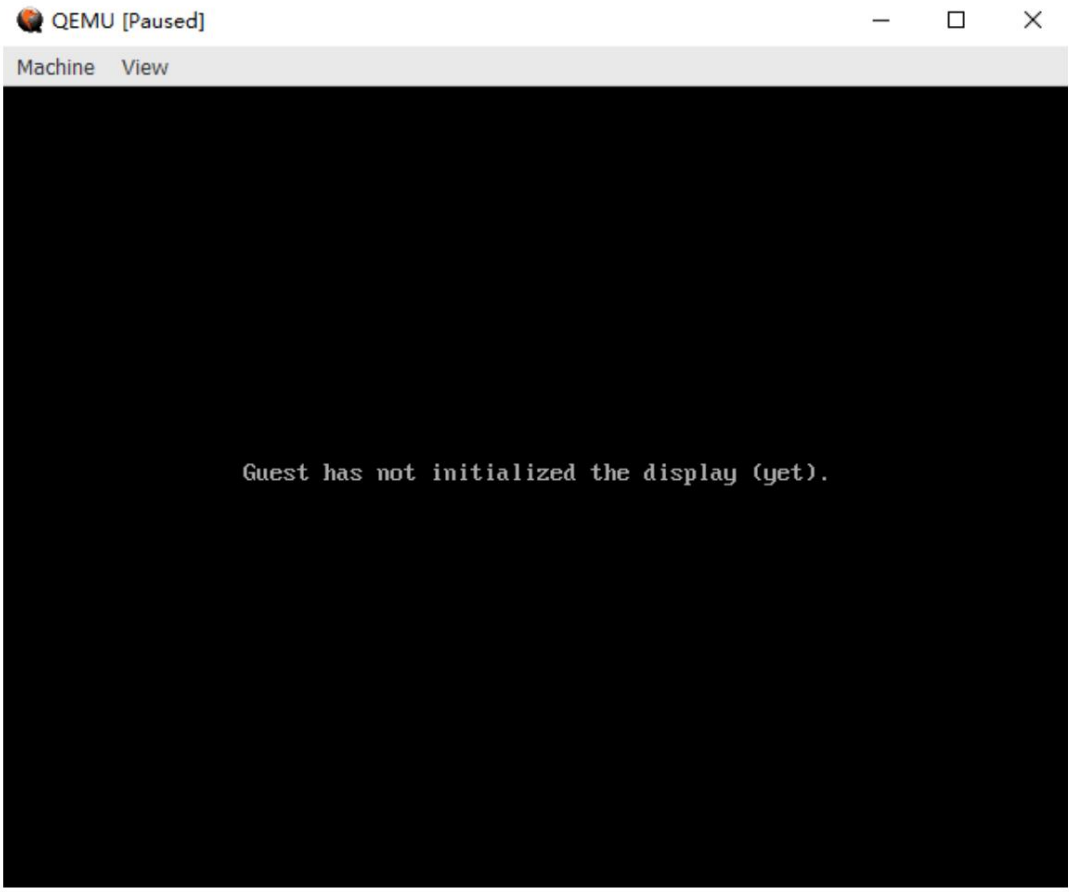

Figure **13:** Virtual commissioning initial interface

2. Click the "Debug" button in the eclipse debug configuration interface, or click the debug project name to open eclipse

Debug interface, you can now debug the project:

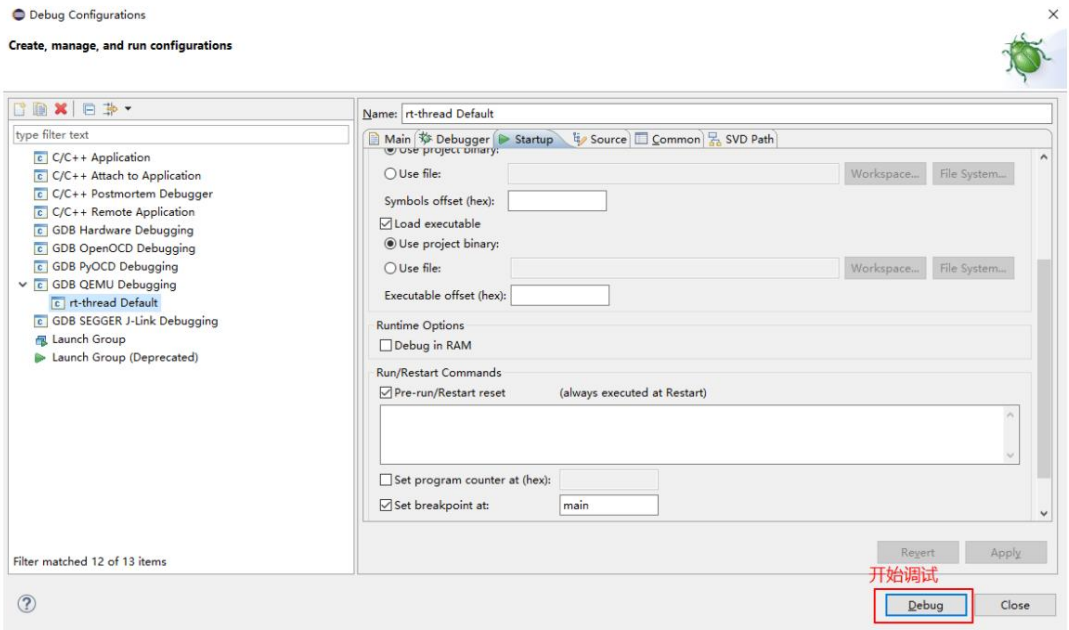

Figure **14:** Start debugging

3. An introduction to the main debugging options of Eclipse is as follows:

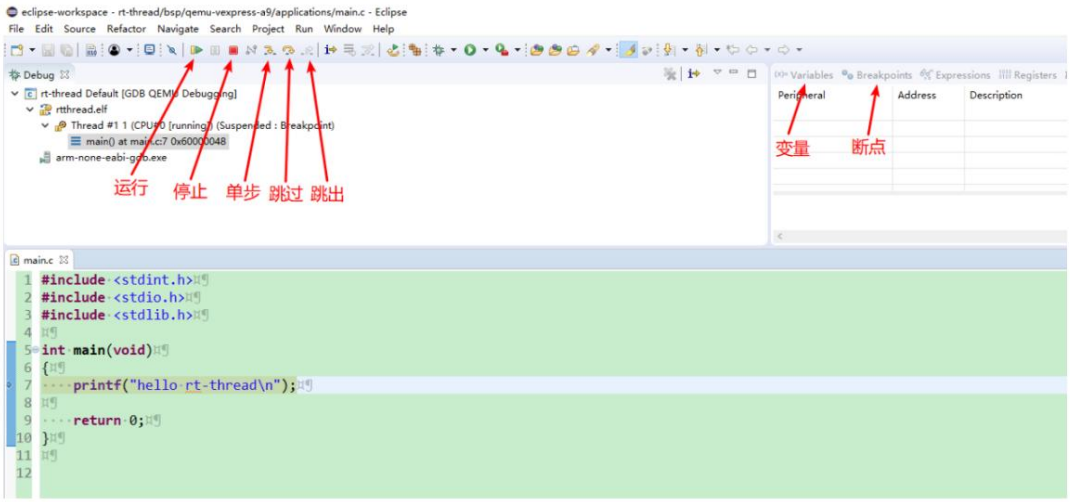

Figure **15:** *Eclipse* Debug options

You can use the shortcut key Ctrl+Shift+r to view the contents of other source files.

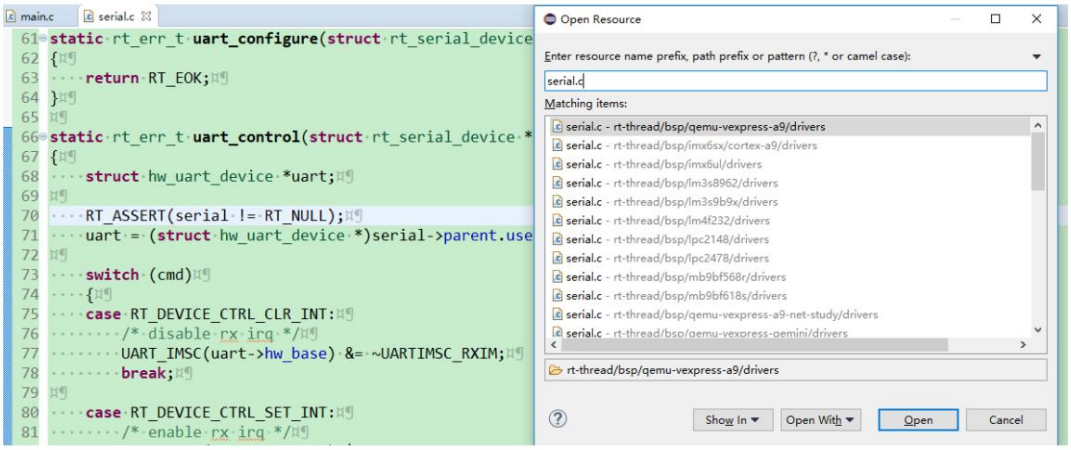

Figure **16:** *Eclipse* Debug options

4References

- Env tool user manual

5 Frequently Asked Questions

- For questions related to the Env tool, please refer to the Common Resource Links section of the Env Tool User Manual.