CAOS-PRJ-WIN-QEMU

Luca Ponzo

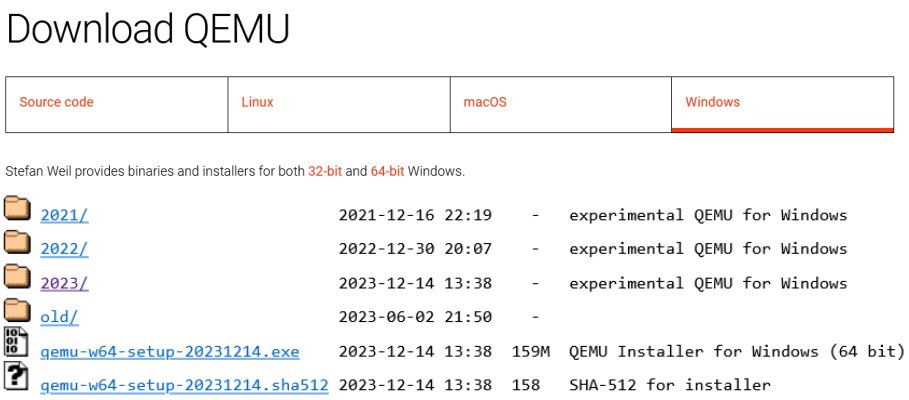
December 2023

1 Introduction

In the upcoming steps, this guide will walk you through setting up **FreeRTOS** and **QEMU** on Windows. You'll also run a simple demo to make sure everything is working smoothly. To proceed with the installation, keep in mind that there are different choices for an **SoC** to emulate; in this guide it is used a **mps-an385**, meaning that it will be explained how to set up an ARM debugger. If you want to install FreeRTOS and QEMU but decide to go with another architecture, please refer to the official FreeRTOS documentation.

2 QEMU

To download QEMU on Windows, refer to the official QEMU documentation and download the appropriate executable for your OS architecture, either 64 bit or 32bit. After that, open the executable. If it opens just skip chapter **2.1** (Command prompt execution) and go to **2.2** (Standard execution).

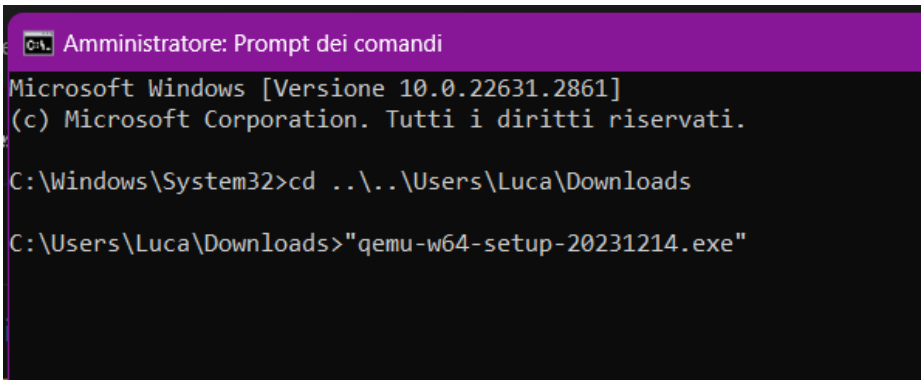


2.1 Command prompt execution

1. Open the command prompt as an administrator
2. Go to the directory where you saved the QEMU installer

3. Type the name of the QEMU installer in the command prompt; it should open now

4. Go forward with the **Standard installation**.



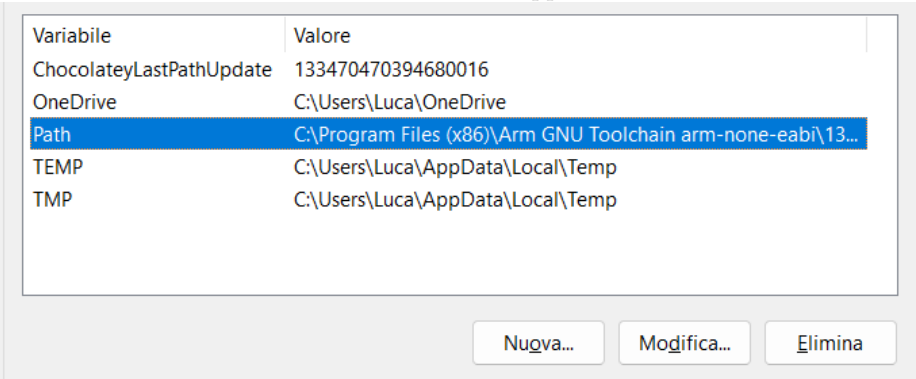
2.2 Standard execution

Just press forward when it asks and keep in mind the location in which you decide to install QEMU; it will become handy soon.

2.3 Add to environment path

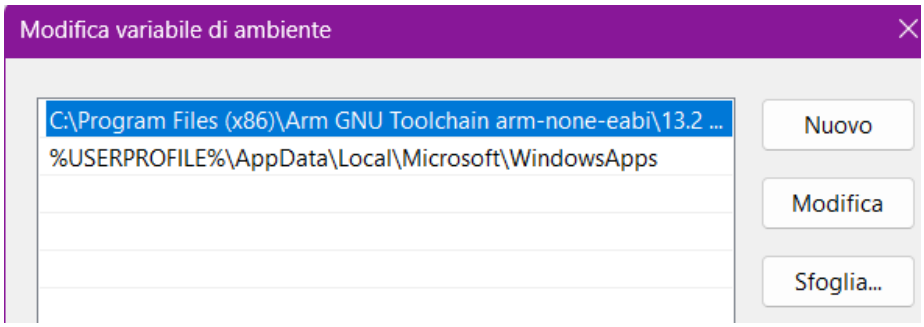
After the installation it's needed to add QEMU to the system PATH; you can do it by searching "environment variables" in the Windows Search bar and clicking on the first result. When the window opens:

1. Search the item with the text PATH in the upper section of the window



2. Click on edit

3. Add a new entry and put the path where QEMU is installed (default is **C:\Program Files\qemu**)



4. Click OK on all the windows opened.

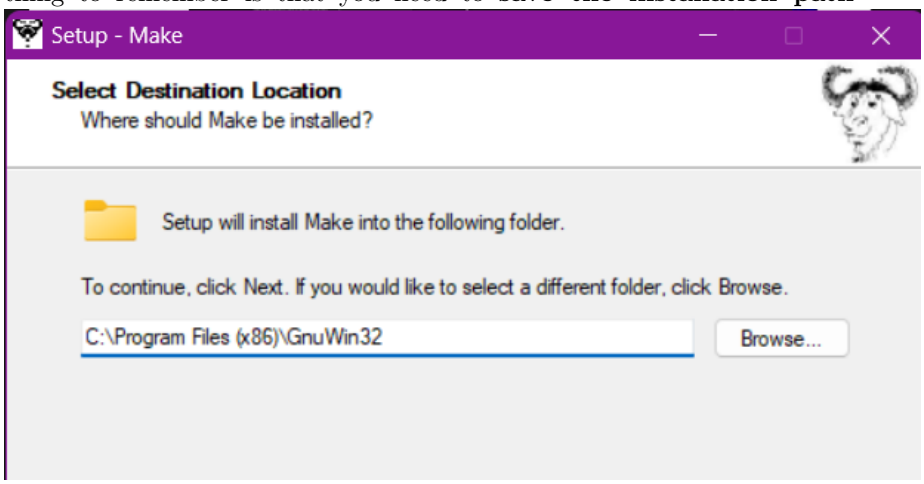
3 MAKE

Following the installation of QEMU, additional tools are required to build our FreeRTOS instance. This guide utilizes the **make** tool, which can be obtained from the provided source Upon selecting the appropriate operating system, you may proceed to initiate the download.

Description	Download	Size	Last change	Md5sum
• Complete package, except sources	Setup	3384653	25 November 2006	8ae51379d1f3eef8360df4e674f17d6d
• Sources	Setup	1252948	25 November 2006	b896c02e3d581040ba1ad65024bbf2cd

3.1 Installation

1. For the installation process you can just keep pressing forward; the only thing to remember is that you need to **save the installation path**



2. After the installation is concluded, add make to the system PATH. Just follow point **2.3** but change the directory to add from *C:\Program Files\qemu*

to **make-dir**\bin. As an example, if you install make in the same path as the image above, you'll need to add to the environment PATH the directory **C:\Program Files(x86)\GnuWin32\bin**

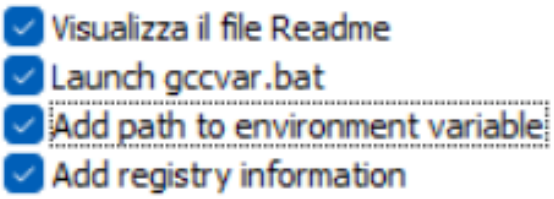
3. Check that make is properly installed by typing **make -v** in a command prompt window **freshly opened**. If the prompt gives an error, make sure the PATH you set is correct and then try again.

4 ARM Toolchain

Since we are using an ARM SoC, we'll need a matching compiler and debugger. We can download them from the official website.

4.1 Installation

As for the ARM toolchain, life is easier: just make sure to check the box **Add path to environment variable** in the end. If you didn't, you'll need to refer again to **2.3** and add *arm-toolchain-dir*\bin to the PATH manually.



5 FreeRTOS

Having successfully installed all requisite tools, the next step involves downloading and installing FreeRTOS. As a preliminary measure, it is imperative to initiate the download of the system; in this instance, the process is marginally more intricate than previously described.

5.1 Download

1. Open a command prompt and navigate to the directory where you intend to store the FreeRTOS files. From now on we'll refer to that directory by **FreeRTOS-dir**

2. Enable git's submodules by typing

   ```
   git config --global core.symlinks true
   ```

3. Clone the repository by typing

   ```
   git clone https://github.com/FreeRTOS/FreeRTOS.git --recurse-submodules
   ```

5.2 Installation

1. Go to ***FreeRTOS-dir*\FreeRTOS\Demo\CORTEX_MPS2_QEMU_IAR_GCC\build\gcc**
2. Type **make**
3. If all the steps above went fine, you should have the file **.\output\RTOSDemo.out**
4. If you want to run the blink demo, edit **.\output\main.o** and make sure the line **mainCREATE_SIMPLE_BLINKY_DEMO_ONLY** is defined to **1**.

```
If mainCREATE_SIMPLE_BLINKY_DEMO_ONLY is not 1 then the comprehensive test and
demo application will be built.  The comprehensive test and demo application is
implemented and described in main_full.c. */
#define mainCREATE_SIMPLE_BLINKY_DEMO_ONLY      1
```

5.3 Starting FreeRTOS

If the installation procedure is concluded, you should be able to run FreeRTOS in QEMU by just typing one single command:

```
qemu-system-arm -machine mps2-an385 -cpu cortex-m3 -kernel
<FreeRTOS-dir>\FreeRTOS\Demo\CORTEX_MPS2_QEMU_IAR_GCC\build\gcc\output\RTOSDemo.out
-monitor stdio -s -S
```

5.4 Debugger utilization

1. Open a new command prompt and go to the directory of RTOSDemo.out
2. Type the command

   ```
   arm-none-eabi-gdb RTOSDemo.out
   ```

3. Connect to the QEMU instance; it should be opened in the default port: **1234**. You can do that by typing

```
target remote localhost:1234
```

4. If you type **continue**, the blinky demo will start.