EEET2490 – Embedded System: OS and Interfacing

Lab 01 - Setup Development Environment and Toolchain

Objectives

The aim of this tutorial is to guide you to setup development environment and necessary toolchains for cross-development of ARM-based embedded applications, which is a bare metal OS with peripheral interfacing in our course.

Note: although the guide below will be described in Windows, most of the tools are also available in Linux and Mac OS.

1. Visual Studio Code

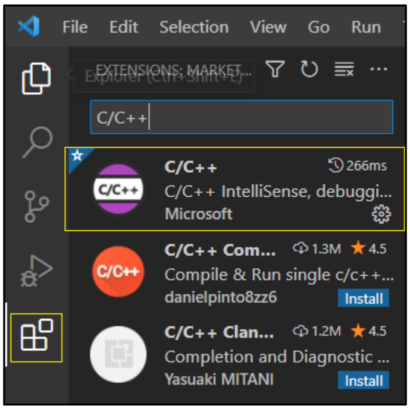
The first tool that we need is a source-code editor. You can use any tool you like, such as Eclipse CDT or Sublime Text. In our course, it is recommended to use **Visual Studio Code** - a powerful and lightweight editor.

a. Install **VS Code**

- Download the VS Code installer for Windows
- Double-click on the downloaded file to run the installer

b. Install the **C/C++ extension for VS Code**

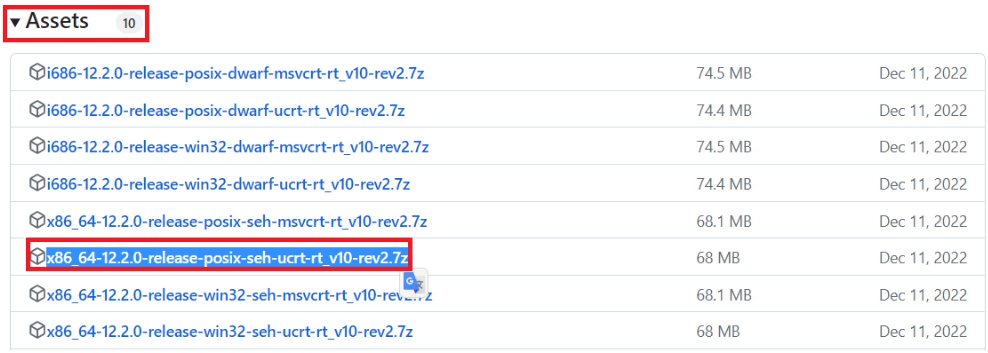
- Open VS Code
- Click at the Extensions icon in the Activity Bar on the left side of VS Code
- Type "C/C++" in the Search box to search for the C/C++ extension
- Click the **Install** button of the extension



2. Install standard GCC toolchain

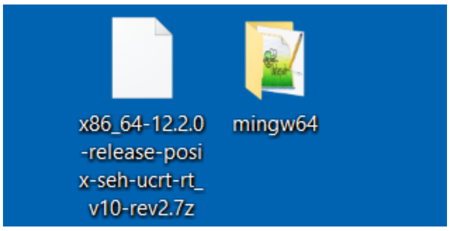
Sometimes you may like to check your code with standard console input and output. In that case, you need the standard GCC to compile the program targeting your computer (like in traditional programming activities).

a) Go to this link: MinGW-W64-builds. Download **x86_64 release-posix-seh-ucrt-rt** with latest version.

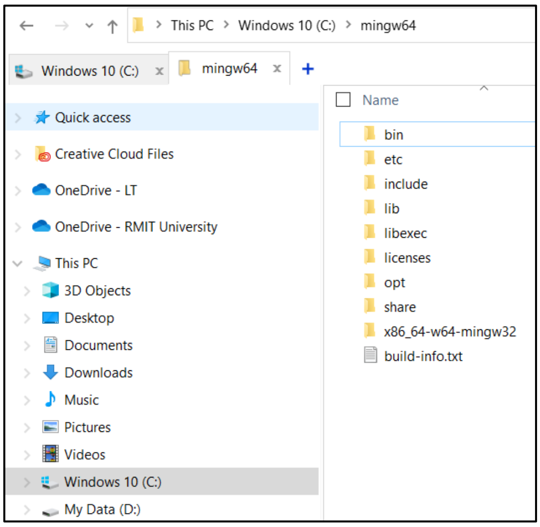


i686-12.2.0-release-posix-dwarf-msvcrt-rt_v10-rev2.7z	74.5 MB	Dec 11, 2022
i686-12.2.0-release-posix-dwarf-ucrt-rt_v10-rev2.7z	74.4 MB	Dec 11, 2022
i686-12.2.0-release-win32-dwarf-msvcrt-rt_v10-rev2.7z	74.5 MB	Dec 11, 2022
i686-12.2.0-release-win32-dwarf-ucrt-rt_v10-rev2.7z	74.4 MB	Dec 11, 2022
x86_64-12.2.0-release-posix-seh-msvcrt-rt_v10-rev2.7z	68.1 MB	Dec 11, 2022
x86_64-12.2.0-release-posix-seh-ucrt-rt_v10-rev2.7z	68 MB	Dec 11, 2022
x86_64-12.2.0-release-win32-seh-msvcrt-rt_v10-rev2.7z	68.1 MB	Dec 11, 2022
x86_64-12.2.0-release-win32-seh-ucrt-rt_v10-rev2.7z	68 MB	Dec 11, 2022

b) After you download the file, right click on it and select "**Extract Here**" with WinRAR or 7-Zip tool to extract it. You will get the extracted folder **mingw64** as below:

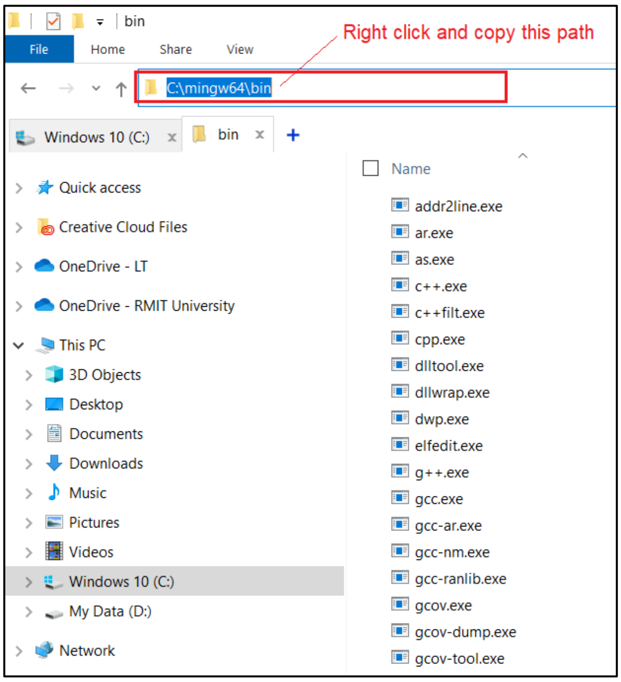


c) Now, copy that folder to your **C drive** as below.



bin
etc
include
lib
libexec
licenses
opt
share
x86_64-w64-mingw32
build-info.txt

d) Go into the **bin** folder and copy its path. For example, with the instructions above, the path will be "**C:\mingw64\bin**" (*it is fine if you copy it in another place with a different path but must have no space characters in the path*).

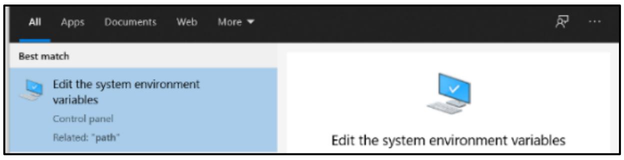


Right click and copy this path

C:\mingw64\bin

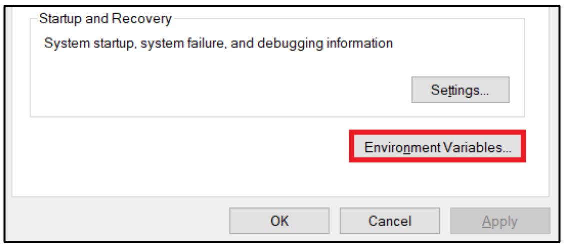
e) Add the above path to the Windows **PATH environment variable** so that **gcc** tool can run without the full path in Windows Terminal (PowerShell and Command Prompt):

- In the search bar, search for "**variables**" and select "**Edit the system environment variables**".

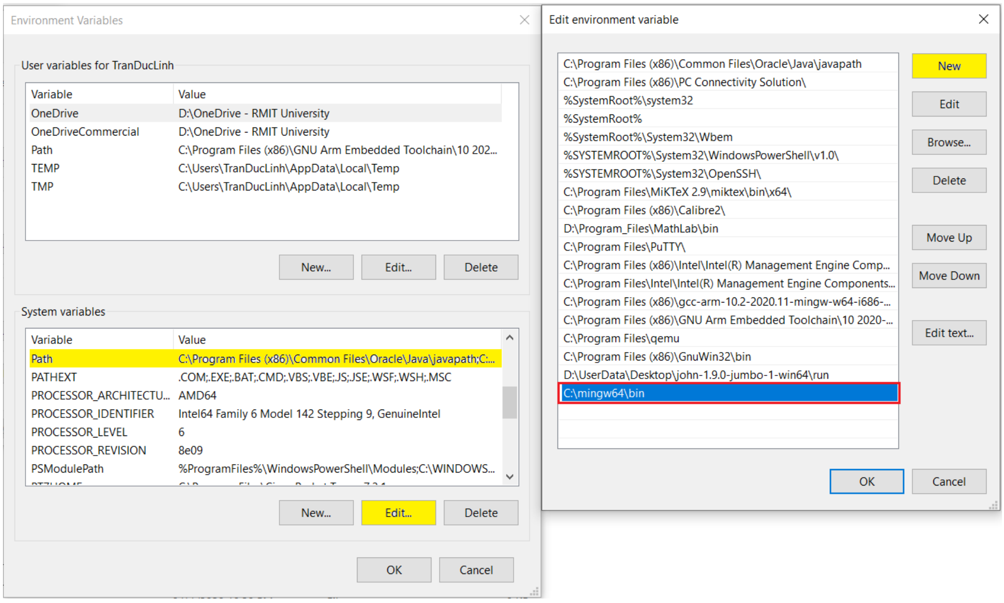


All
Apps
Documents
Web
More
Best match
Edit the system environment variables
Control panel
Related: "path"
Edit the system environment variables

- Select "**Environment Variables**"



- Under **System variables** area, select **Path** > **Edit**. In the **Edit environment variable** window, press **New** and input path to the bin directory of our installed GCC tool. After that, press **OK** in all windows to complete PATH setting.



Variable	Value
OneDrive	D:\OneDrive - RMIT University
OneDriveCommercial	D:\OneDrive - RMIT University
Path	C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 202...
TEMP	C:\Users\TranDucLinh\AppData\Local\Temp
TMP	C:\Users\TranDucLinh\AppData\Local\Temp

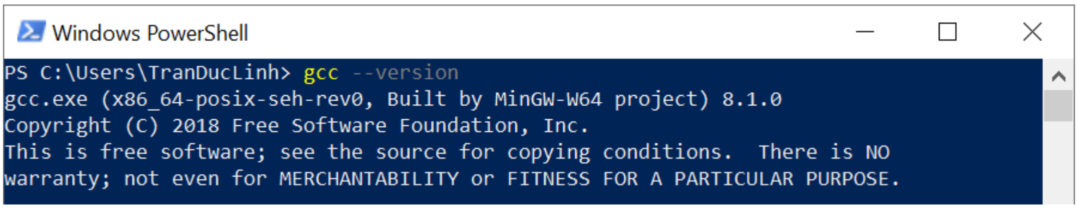
Variable	Value
Path	C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTU...	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 142 Stepping 9, GenuineIntel
PROCESSOR_LEVEL	6
PROCESSOR_REVISION	8e09
PSModulePath	%ProgramFiles%\WindowsPowerShell\Modules;C:\WINDOWS...

```
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
C:\Program Files (x86)\PC Connectivity Solution\
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\MiKTeX 2.9\miktex\bin\x64\
C:\Program Files (x86)\Calibre2\
D:\Program_Files\MathLab\bin
C:\Program Files\PuTTY\
C:\Program Files (x86)\Intel\Intel(R) Management Engine Comp...
C:\Program Files\Intel\Intel(R) Management Engine Components...
C:\Program Files (x86)\gcc-arm-10.2-2020.11-mingw-w64-i686-...
C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2020-...
C:\Program Files\qemu
C:\Program Files (x86)\GnuWin32\bin
D:\UserData\Desktop\john-1.9.0-jumbo-1-win64\run
C:\mingw64\bin
```

f) Test the **gcc** tool to make sure that it is installed properly:

- Type **powershell** in Search bar to open PowerShell.
  Note: *close and reopen powershell if you are opening it while or before doing PATH setting step above* (so that it can be updated with the PATH setting).

- Type "**gcc --version**". If it is installed correctly, you should see the result as below:



```
PS C:\Users\TranDucLinh> gcc --version
gcc.exe (x86_64-posix-seh-rev0, Built by MinGW-W64 project) 8.1.0
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

3. GNU Embedded Toolchain for ARM (gcc-arm)

To be able to compile your code, you will need a compiler. Since our target is to develop applications for a Raspberry Pi board (in which its processor is an ARM microcontroller), we will install GNU Embedded Toolchain for ARM.

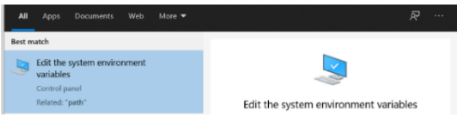
a. Download link for **32-bit gcc-arm**:
   https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads

   For Windows, select latest version of gcc-arm-none-eabi-…-win32.exe to download and install.

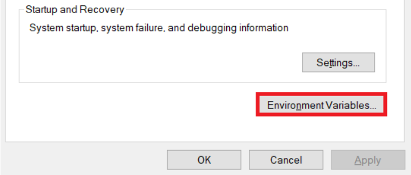
b. After installation, go to this installed location and browse to **bin** folder then copy its path. For example, in my computer, the path is below (note: *you may get newer version with different number*):
   C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2020-q4-major\bin

c. Now, we need to change the PATH environment variable of the Windows to be able to run it later.

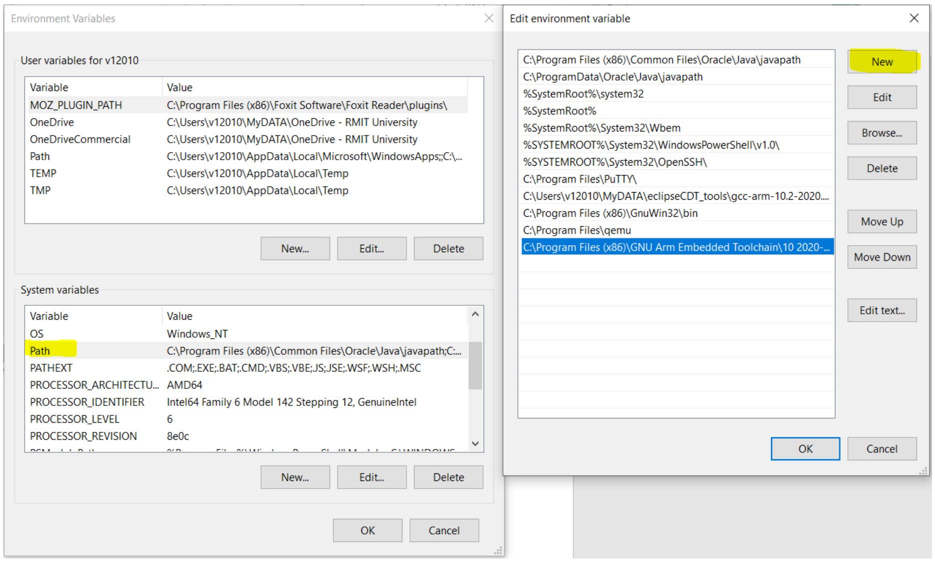
   - In the search bar, search for "**path**" and select "**Edit the system environment variables**".


All
Apps
Documents
Web
More
Best match
Edit the system environment variables
Control panel
Related: "path"
Edit the system environment variables

   - Select "**Environment Variables**"



   - Under **System variables** area, select **Path** > Edit. In the **Edit environment variable** window, press **New** and input path to the bin directory of our installed gcc-arm tool (by default is *C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2020-q4-major\bin*).



User variables for v12010

Variable	Value
MOZ_PLUGIN_PATH	C:\Program Files (x86)\Foxit Software\Foxit Reader\plugins\
OneDrive	C:\Users\v12010\MyDATA\OneDrive - RMIT University
OneDriveCommercial	C:\Users\v12010\MyDATA\OneDrive - RMIT University
Path	C:\Users\v12010\AppData\Local\Microsoft\WindowsApps;;C:\...
TEMP	C:\Users\v12010\AppData\Local\Temp
TMP	C:\Users\v12010\AppData\Local\Temp

System variables

Variable	Value
OS	Windows_NT
Path	C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTU...	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 142 Stepping 12, GenuineIntel
PROCESSOR_LEVEL	6
PROCESSOR_REVISION	8e0c

Edit environment variable

```
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
C:\ProgramData\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\PuTTY\
C:\Users\v12010\MyDATA\eclipseCDT_tools\gcc-arm-10.2-2020....
C:\Program Files (x86)\GnuWin32\bin
C:\Program Files\qemu
C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2020-...
```

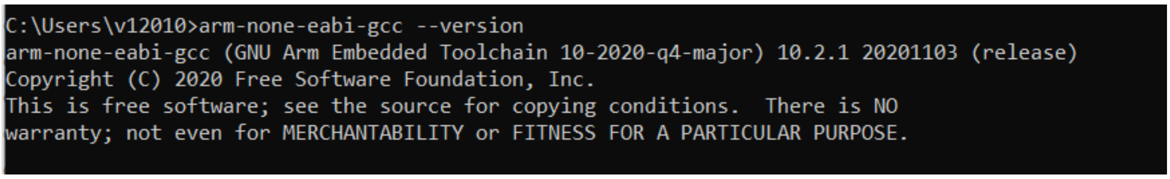
After that, press OK in all windows to complete PATH setting.

d. Test the gcc-arm tool to make sure that it is installed properly:

- Type **cmd** in Search bar to open **Command Prompt** (in latest version of Windows, there could be some problems with cmd. If so, you can use **PowerShell** instead).

  Note: Close and Reopen Command Prompt if it is running before.

- Type "**arm-none-eabi-gcc --version**". If it is installed correctly, you should see the result as below:



```
C:\Users\v12010>arm-none-eabi-gcc --version
arm-none-eabi-gcc (GNU Arm Embedded Toolchain 10-2020-q4-major) 10.2.1 20201103 (release)
Copyright (C) 2020 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

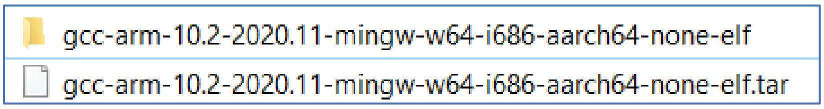
e. Similarly, install **64-bit gcc-arm tool:**

- Go to the link: https://developer.arm.com/downloads/-/gnu-a

  For Windows, refer to section **Windows hosted cross compilers.**

- Under category **AArch64 bare-metal target,** select and download file gcc-arm-10.3-2021.07-mingw-w64-i686-aarch64-none-elf.tar.xz

- After downloading, extract it. You may need 7-zip to do so.



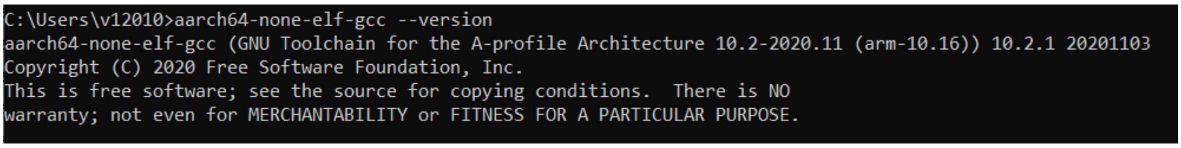
gcc-arm-10.2-2020.11-mingw-w64-i686-aarch64-none-elf

gcc-arm-10.2-2020.11-mingw-w64-i686-aarch64-none-elf.tar

- Cut and paste the extracted folder into the directory that you installed other tools above (by default is *C:\Program Files (x86)*).

- Add the path of **bin** directory of the tool (for example, ***C:\Program Files (x86)\gcc-arm-10.2-2020.11-mingw-w64-i686-aarch64-none-elf\bin***) to the path environment variable like we did above.

- Check with command prompt: **aarch64-none-elf-gcc --version.** If successful, you should see the output as below:

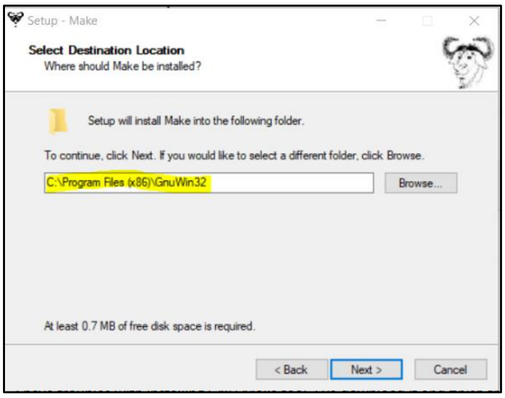


```
C:\Users\v12010>aarch64-none-elf-gcc --version
aarch64-none-elf-gcc (GNU Toolchain for the A-profile Architecture 10.2-2020.11 (arm-10.16)) 10.2.1 20201103
Copyright (C) 2020 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

4. Install Make tool for Windows

a. Go to the link http://gnuwin32.sourceforge.net/packages/make.htm

Select “Complete package, except sources” Setup to download and install it.
Leave the path by default when installing it as below:



Select Destination Location
Where should Make be installed?

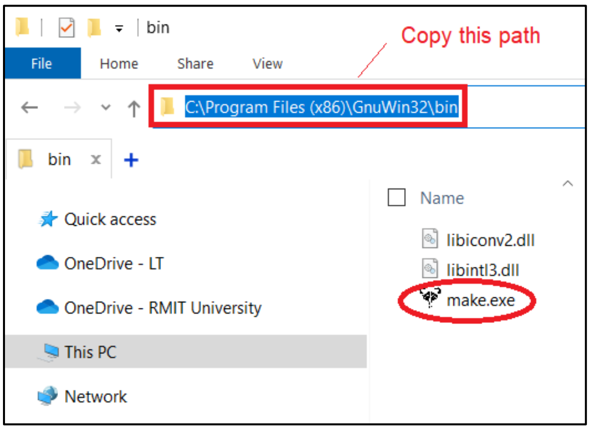
Setup will install Make into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files (x86)\GnuWin32

At least 0.7 MB of free disk space is required.

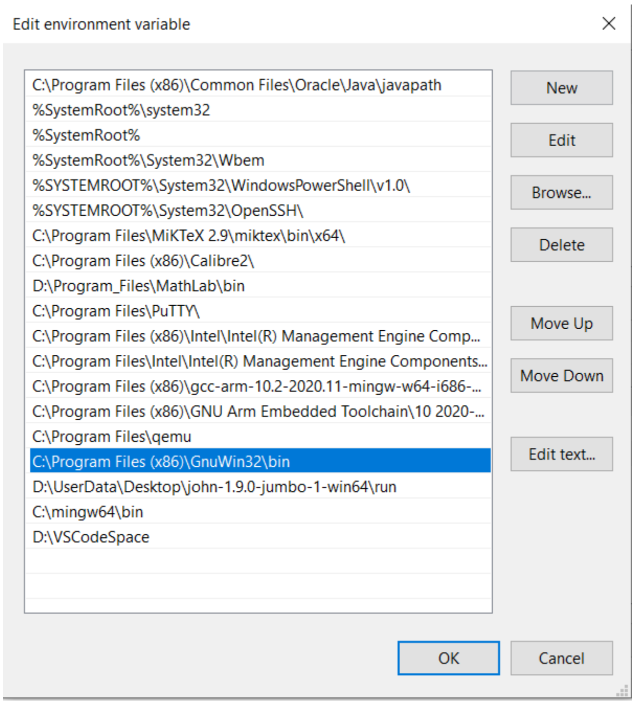
b. Then go there, go inside its **bin** folder (you should see make.exe file there), and copy the path.



C:\Program Files (x86)\GnuWin32\bin

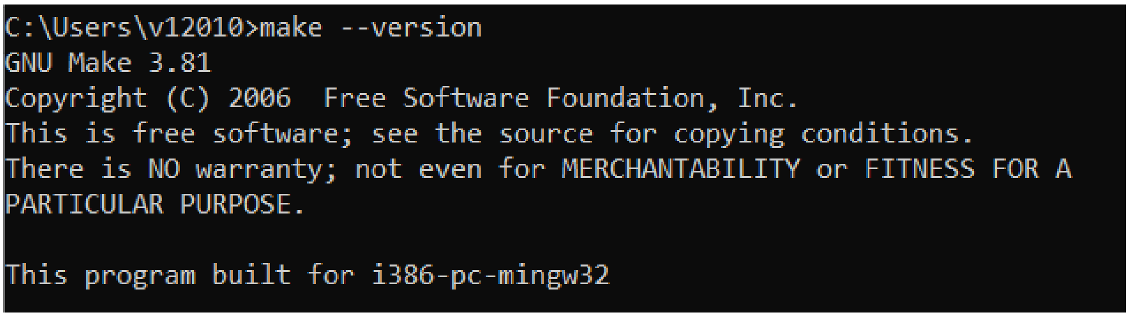
libiconv2.dll
libintl3.dll
make.exe

c. Similary, add path to its bin directory (by default is *C:\Program Files (x86)\GnuWin32\bin*) to system PATH:



```
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\MiKTeX 2.9\miktex\bin\x64\
C:\Program Files (x86)\Calibre2\
D:\Program_Files\MathLab\bin
C:\Program Files\PuTTY\
C:\Program Files (x86)\Intel\Intel(R) Management Engine Comp...
C:\Program Files\Intel\Intel(R) Management Engine Components...
C:\Program Files (x86)\gcc-arm-10.2-2020.11-mingw-w64-i686-...
C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2020-...
C:\Program Files\qemu
C:\Program Files (x86)\GnuWin32\bin
D:\UserData\Desktop\john-1.9.0-jumbo-1-win64\run
C:\mingw64\bin
D:\VSCodeSpace
```

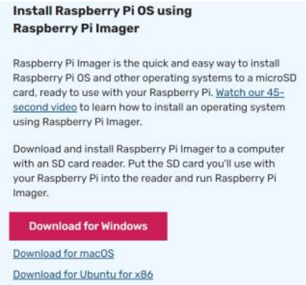
d. Next, close and re-open PowerShell to be updated if you are opening it. Finally, you should be successful with "**make --version**" command.



```
C:\Users\v12010>make --version
GNU Make 3.81
Copyright (C) 2006  Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.

This program built for i386-pc-mingw32
```

5. Raspberry Pi Imager

We will need this tool to write OS image to the SD card later
Download link: https://www.raspberrypi.com/software/



Install Raspberry Pi OS using Raspberry Pi Imager

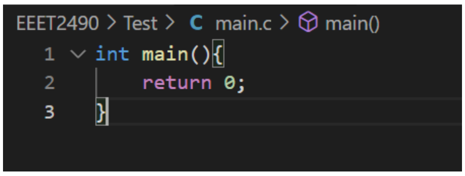
Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. Watch our 45-second video to learn how to install an operating system using Raspberry Pi Imager.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

6. Test the development environment

a. Close and re-open **VS Code** if it is opening. Click on **File** > **Add Folder to Workspace.** Browse to/ create a folder where you want to save all of your programs (e.g. D:\EEET2490), and select it.

b. Create a folder for your project, e.g. named **Test**. Create a source file named main.c with a simple main function inside like below:



```
int main(){
    return 0;
}
```

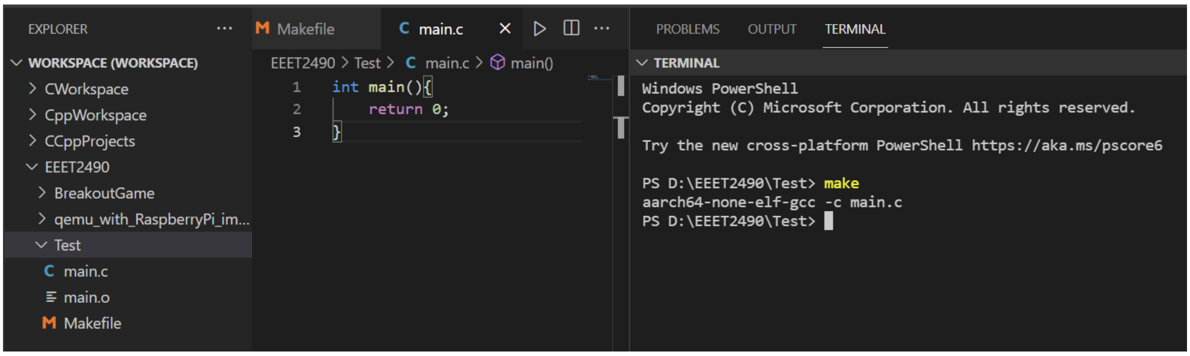
c. Now, let's create a simple **Makefile** (its name is Makefile) in the same folder. Write into the file:

```
all:
	aarch64-none-elf-gcc -c main.c
```

Note: The second line in the Makefile's content above is indented with **TAB** key.

d. Right click on the **Test** folder, select "Open in Integrated Terminal" to have the command prompt terminal opened to that folder.

e. Type "**make**" in the terminal. If all the required tools are installed correctly above, you will see the source file is compiled and generated **main.o** file as below.



```c
int main(){
    return 0;
}
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\EEET2490\Test> make
aarch64-none-elf-gcc -c main.c
PS D:\EEET2490\Test>
```

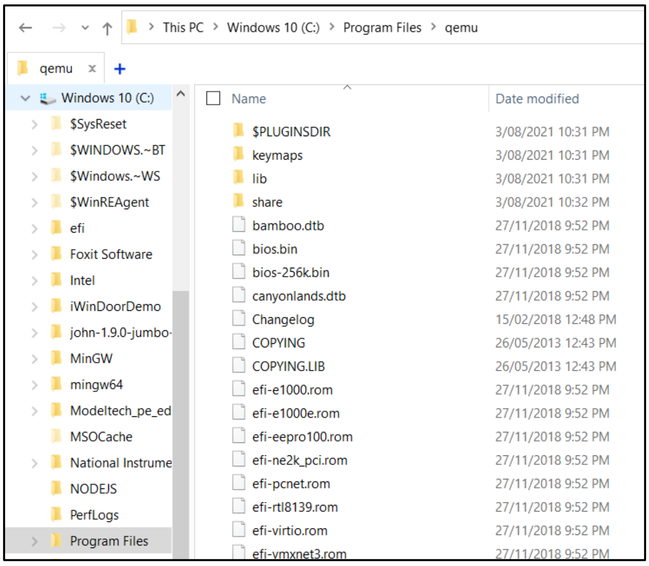
7. Raspberry Pi Emulation with QEMU

We have completed setting up our development above. Now, we will setup the QEMU tool to emulate a Raspberry Pi machine and test our development projects later.

a. Download the QEMU version 20181127 via following link: [qemu20181127.zip]()

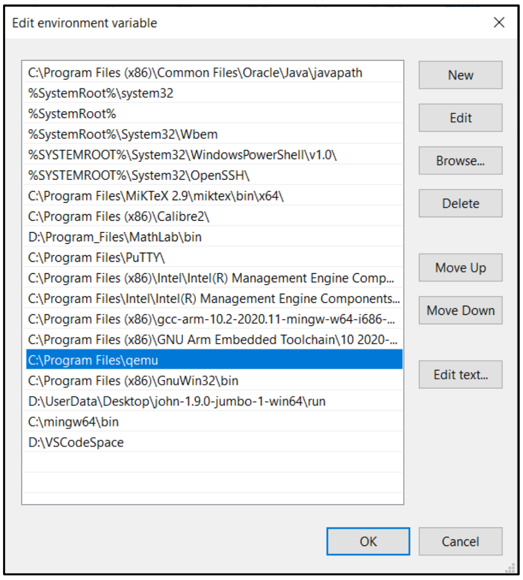
b. Create a folder namely **qemu** in the "Program Files" folder of C drive (its path will be C:\Program Files\qemu).

c. Copy the **qemu20181127.zip file there,** right click on it and select "**Extract Here**" with Winrar or 7-Zip tool to extract it. You will see it as below



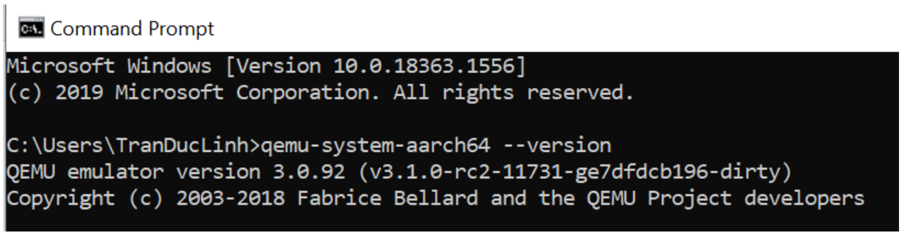
Name	Date modified
$PLUGINSDIR	3/08/2021 10:31 PM
keymaps	3/08/2021 10:31 PM
lib	3/08/2021 10:31 PM
share	3/08/2021 10:32 PM
bamboo.dtb	27/11/2018 9:52 PM
bios.bin	27/11/2018 9:52 PM
bios-256k.bin	27/11/2018 9:52 PM
canyonlands.dtb	27/11/2018 9:52 PM
Changelog	15/02/2018 12:48 PM
COPYING	26/05/2013 12:43 PM
COPYING.LIB	26/05/2013 12:43 PM
efi-e1000.rom	27/11/2018 9:52 PM
efi-e1000e.rom	27/11/2018 9:52 PM
efi-eepro100.rom	27/11/2018 9:52 PM
efi-ne2k_pci.rom	27/11/2018 9:52 PM
efi-pcnet.rom	27/11/2018 9:52 PM
efi-rtl8139.rom	27/11/2018 9:52 PM
efi-virtio.rom	27/11/2018 9:52 PM
efi-vmxnet3.rom	27/11/2018 9:52 PM

d. Similarly, add path (C:\Program Files\qemu) to system PATH environment variable

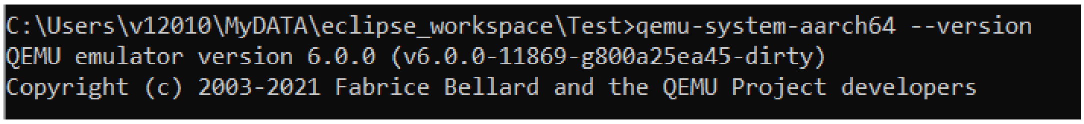


```
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\MiKTeX 2.9\miktex\bin\x64\
C:\Program Files (x86)\Calibre2\
D:\Program_Files\MathLab\bin
C:\Program Files\PuTTY\
C:\Program Files (x86)\Intel\Intel(R) Management Engine Comp...
C:\Program Files\Intel\Intel(R) Management Engine Components...
C:\Program Files (x86)\gcc-arm-10.2-2020.11-mingw-w64-i686-...
C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2020-...
C:\Program Files\qemu
C:\Program Files (x86)\GnuWin32\bin
D:\UserData\Desktop\john-1.9.0-jumbo-1-win64\run
C:\mingw64\bin
D:\VSCodeSpace
```

e. Check again with Command Prompt: **qemu-system-aarch64 --version**



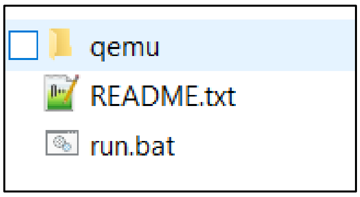
```
Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\TranDucLinh>qemu-system-aarch64 --version
QEMU emulator version 3.0.92 (v3.1.0-rc2-11731-ge7dfdcb196-dirty)
Copyright (c) 2003-2018 Fabrice Bellard and the QEMU Project developers
```



```
C:\Users\v12010\MyDATA\eclipse_workspace\Test>qemu-system-aarch64 --version
QEMU emulator version 6.0.0 (v6.0.0-11869-g800a25ea45-dirty)
Copyright (c) 2003-2021 Fabrice Bellard and the QEMU Project developers
```

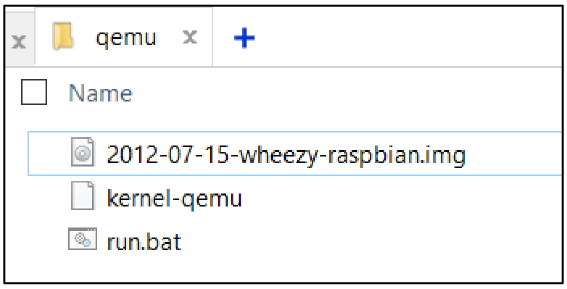
So, we have sucessfully installed the QEMU tool. Since it works as a virtualization tool (similar to VirtualBox and Vmware), we will need an OS image of the machine that is going to be emulated. For testing our QEMU tool, we will to have download Raspberry pi images here: LINK.

- After downloading and extracting, you should see those files



```
qemu
README.txt
run.bat
```

Inside the qemu folder above, it contains run.bat (.bat file contains command for qemu to run emulation), kernel-qemu (Rasberry Pi kernel) and 2012-07-15-wheezy-raspbian.img (Rasberry Pi SD card image).

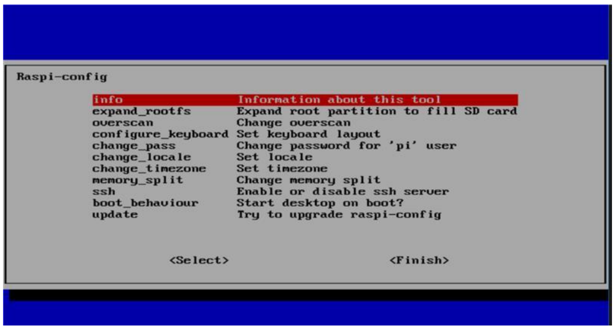


```
2012-07-15-wheezy-raspbian.img
kernel-qemu
run.bat
```

- Now, you can double click on the **run.bat** file to start the emulation. You can right click on this file, and select edit to view its content. Indeed, it will execute the command below (try copy and paste it in Command Prompt to run it manually):

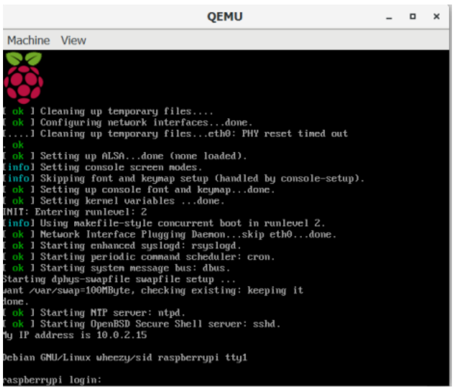
```
qemu-system-arm.exe -M versatilepb -cpu arm1176 -hda 2012-07-15-wheezy-raspbian.img -kernel kernel-qemu -m 192 -append "root=/dev/sda2"
```

- Our emulated Raspberry will start booting up. You may see the config window below. If yes, use **TAB** button to select "**Finish**".



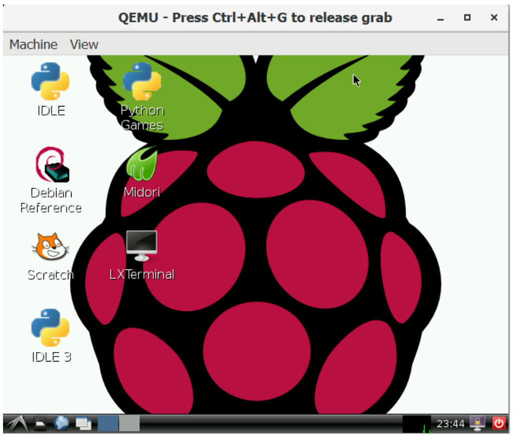
```
Raspi-config

        info                Information about this tool
        expand_rootfs       Expand root partition to fill SD card
        overscan            Change overscan
        configure_keyboard  Set keyboard layout
        change_pass         Change password for 'pi' user
        change_locale       Set locale
        change_timezone     Set timezone
        memory_split        Change memory split
        ssh                 Enable or disable ssh server
        boot_behaviour      Start desktop on boot?
        update              Try to upgrade raspi-config


              <Select>                    <Finish>
```

- Then, it will continue booting up to the command prompt. The username and password of the default account in that OS image is:
  - Username: pi
  - Password: raspberry



```
[ ok ] Cleaning up temporary files....
[ ok ] Configuring network interfaces...done.
[....] Cleaning up temporary files...eth0: PHY reset timed out
. ok
[ ok ] Setting up ALSA...done (none loaded).
[info] Setting console screen modes.
[info] Skipping font and keymap setup (handled by console-setup).
[ ok ] Setting up console font and keymap...done.
[ ok ] Setting kernel variables ...done.
INIT: Entering runlevel: 2
[info] Using makefile-style concurrent boot in runlevel 2.
[ ok ] Network Interface Plugging Daemon...skip eth0...done.
[ ok ] Starting enhanced syslogd: rsyslogd.
[ ok ] Starting periodic command scheduler: cron.
[ ok ] Starting system message bus: dbus.
Starting dphys-swapfile swapfile setup ...
want /var/swap=100MByte, checking existing: keeping it
done.
[ ok ] Starting NTP server: ntpd.
[ ok ] Starting OpenBSD Secure Shell server: sshd.
My IP address is 10.0.2.15

Debian GNU/Linux wheezy/sid raspberrypi tty1

raspberrypi login:
```

- After logging in, type "startx" to start the GUI. You will see the following screen of Raspbian OS.



Up to here, we have sucesfully installed and tested our necessary tools for development and verification. In following tutorials, we will explore the architecture of the ARM based Raspberry Pi SoC and start the development.