

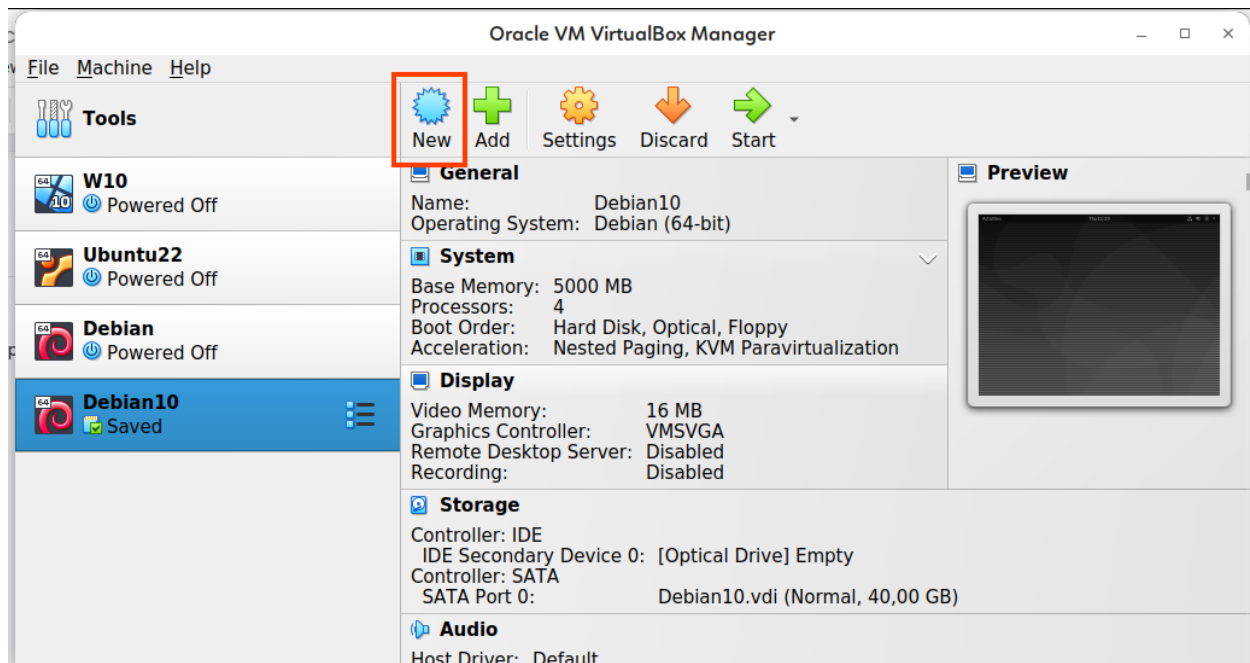
Application development for Syrus4G

The recommended OS for the development of Syrus4G is Linux Debian 10 “Buster”

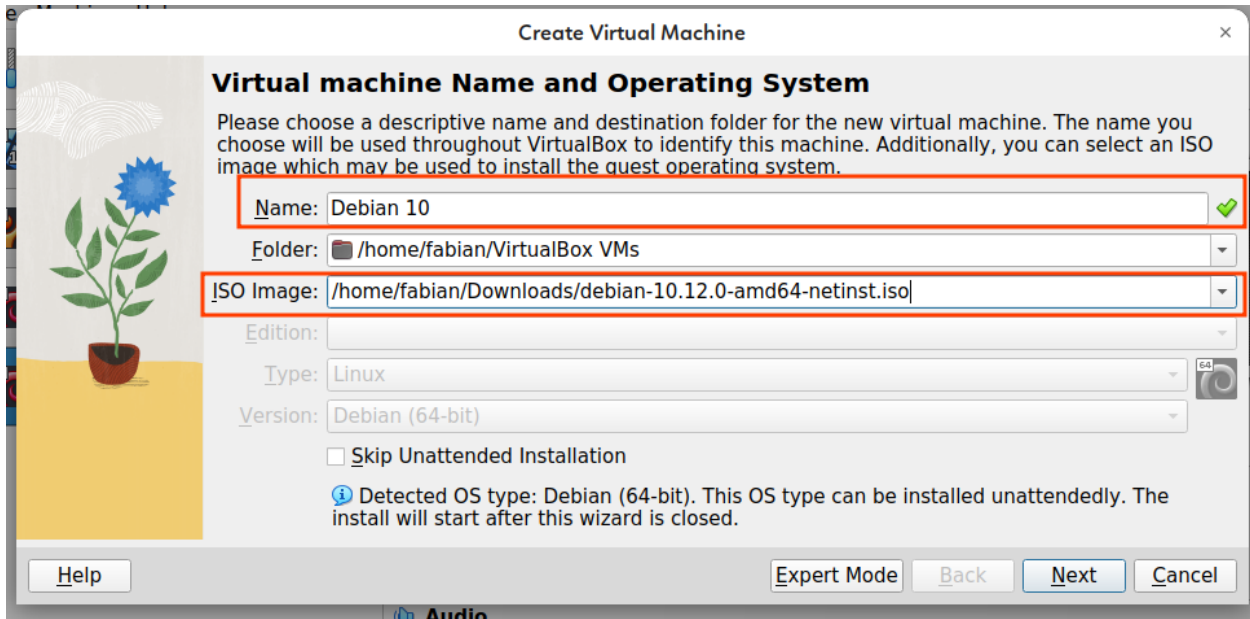
In Virtual Machine

Install the provided virtual machine files or create your own virtual machine following these steps:

1. Download the last Virtual Box from: <https://www.virtualbox.org/wiki/Downloads>
2. Download the ISO installation file of Debian 10 from: <https://cdimage.debian.org/cdimage/archive/10.12.0/amd64/iso-cd/debian-10.12.0-amd64-netinst.iso>
3. Create a new virtual machine



Choose the name and the ISO file and click next



Create Virtual Machine

Virtual machine Name and Operating System

Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.

Name: ✓

Folder:

ISO Image:

Edition:

Type: 64

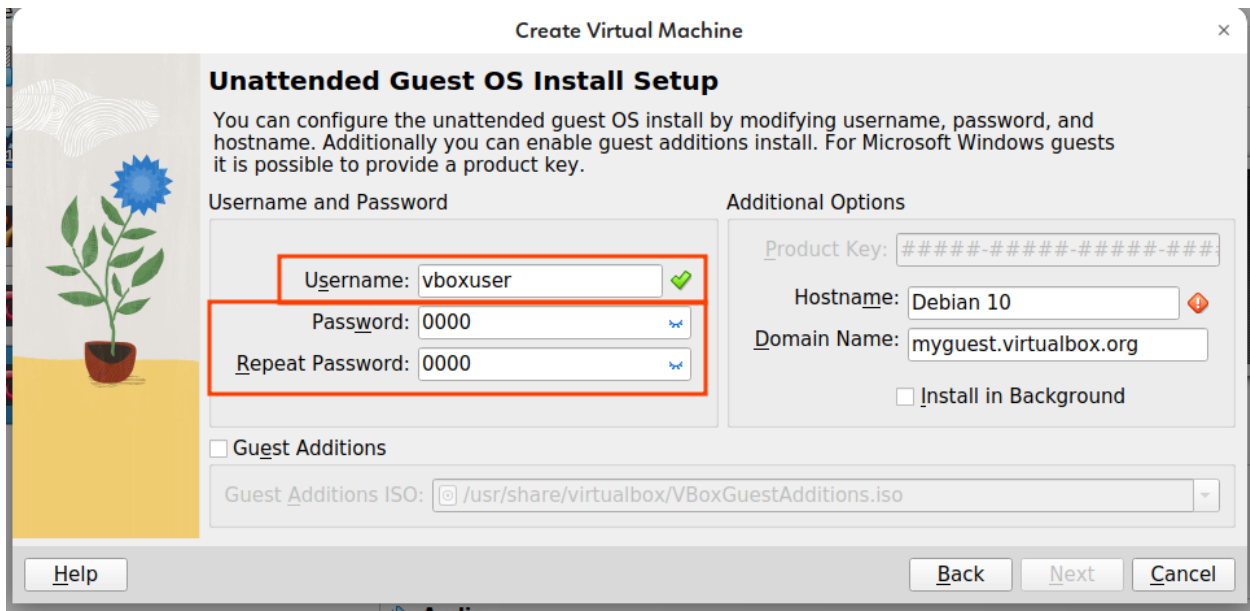
Version:

☐ Skip Unattended Installation

ⓘ Detected OS type: Debian (64-bit). This OS type can be installed unattendedly. The install will start after this wizard is closed.

[Help](#) [Expert Mode](#) [Back](#) [Next](#) [Cancel](#)

Select an username and password and click next



Create Virtual Machine

Unattended Guest OS Install Setup

You can configure the unattended guest OS install by modifying username, password, and hostname. Additionally you can enable guest additions install. For Microsoft Windows guests it is possible to provide a product key.

Username and Password

Username: ✓

Password:

Repeat Password:

☐ Guest Additions

Guest Additions ISO:

Additional Options

Product Key:

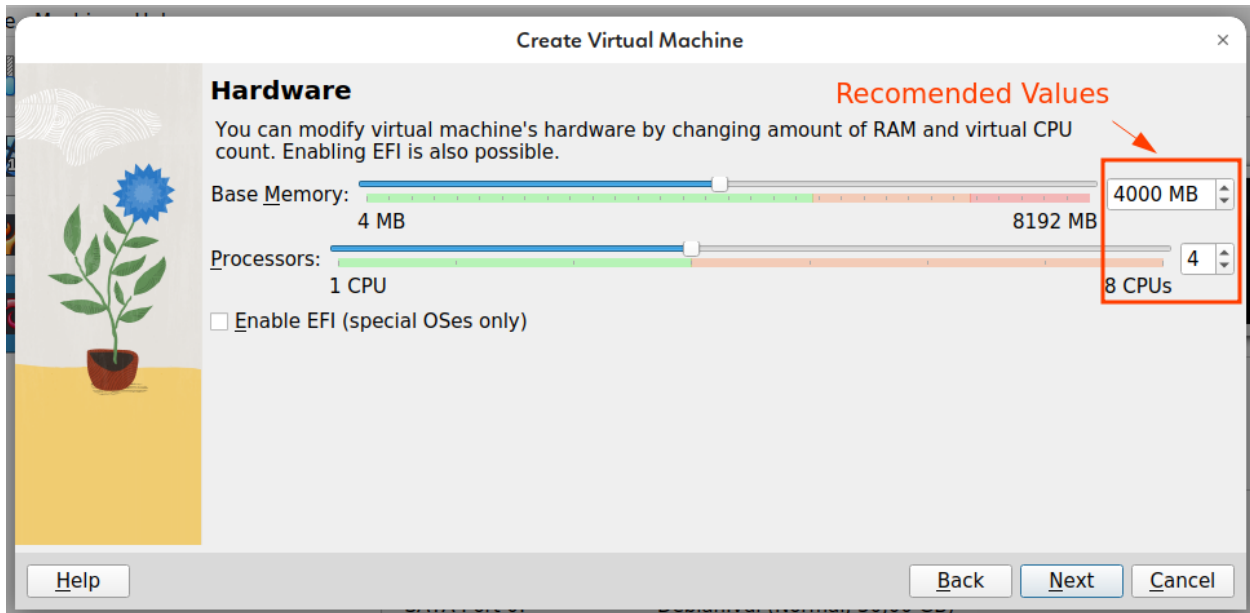
Hostname: ⓘ

Domain Name:

☐ Install in Background

[Help](#) [Back](#) [Next](#) [Cancel](#)

Set hardware values and click next



The screenshot shows the 'Create Virtual Machine' window with the 'Hardware' tab selected. The window title is 'Create Virtual Machine'. On the left is a decorative illustration of a potted plant with a blue flower. The main content area has a heading 'Hardware' and a sub-heading 'Recommended Values'. Below this, there is a text box stating: 'You can modify virtual machine's hardware by changing amount of RAM and virtual CPU count. Enabling EFI is also possible.' There are two sliders: 'Base Memory' and 'Processors'. The 'Base Memory' slider ranges from 4 MB to 8192 MB, with a value of 4000 MB selected. The 'Processors' slider ranges from 1 CPU to 8 CPUs, with a value of 4 selected. Both values are also shown in input boxes on the right. There is a checkbox for 'Enable EFI (special OSes only)' which is unchecked. At the bottom are buttons for 'Help', 'Back', 'Next', and 'Cancel'.

Create Virtual Machine

Hardware

You can modify virtual machine's hardware by changing amount of RAM and virtual CPU count. Enabling EFI is also possible.

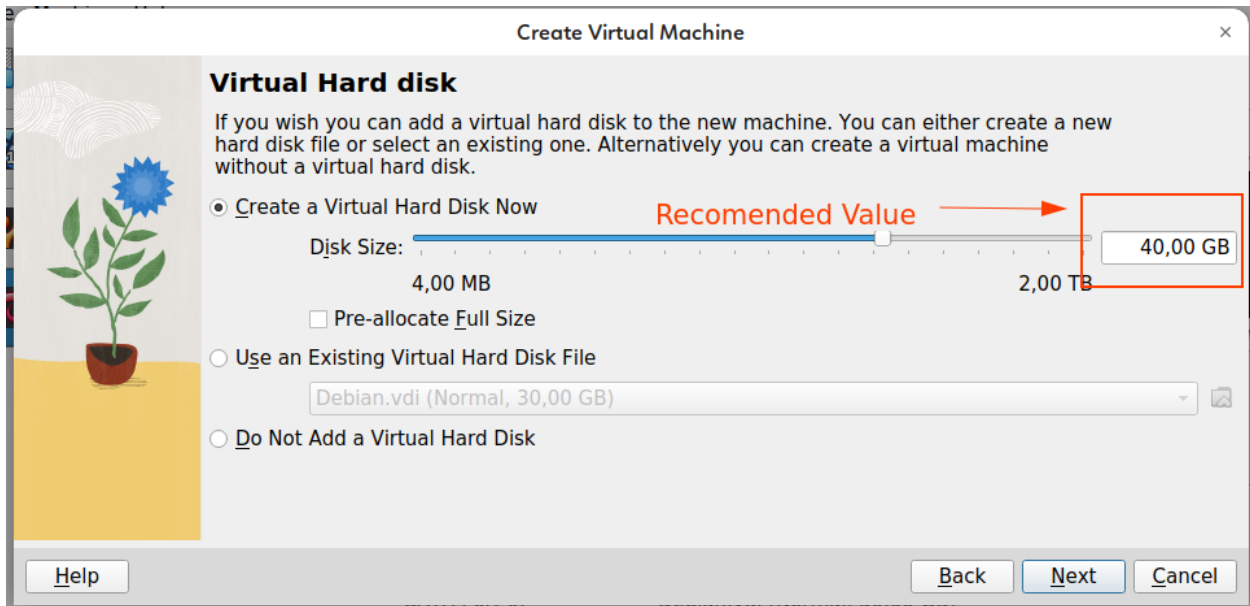
Base Memory: 4 MB 8192 MB 4000 MB

Processors: 1 CPU 8 CPUs 4

☐ Enable EFI (special OSes only)

Help Back Next Cancel

Set disk values and click next



The screenshot shows the 'Create Virtual Machine' window with the 'Virtual Hard disk' tab selected. The window title is 'Create Virtual Machine'. On the left is a decorative illustration of a potted plant with a blue flower. The main content area has a heading 'Virtual Hard disk' and a text box stating: 'If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select an existing one. Alternatively you can create a virtual machine without a virtual hard disk.' There are three radio buttons: 'Create a Virtual Hard Disk Now' (selected), 'Use an Existing Virtual Hard Disk File', and 'Do Not Add a Virtual Hard Disk'. The 'Create a Virtual Hard Disk Now' option has a sub-option 'Pre-allocate Full Size' which is unchecked. Below this is a 'Disk Size' slider ranging from 4,00 MB to 2,00 TB, with a value of 40,00 GB selected. This value is also shown in an input box on the right. Below the slider is a text box containing 'Debian.vdi (Normal, 30,00 GB)'. At the bottom are buttons for 'Help', 'Back', 'Next', and 'Cancel'.

Create Virtual Machine

Virtual Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select an existing one. Alternatively you can create a virtual machine without a virtual hard disk.

☒ Create a Virtual Hard Disk Now

Disk Size: 4,00 MB 2,00 TB 40,00 GB

☐ Pre-allocate Full Size

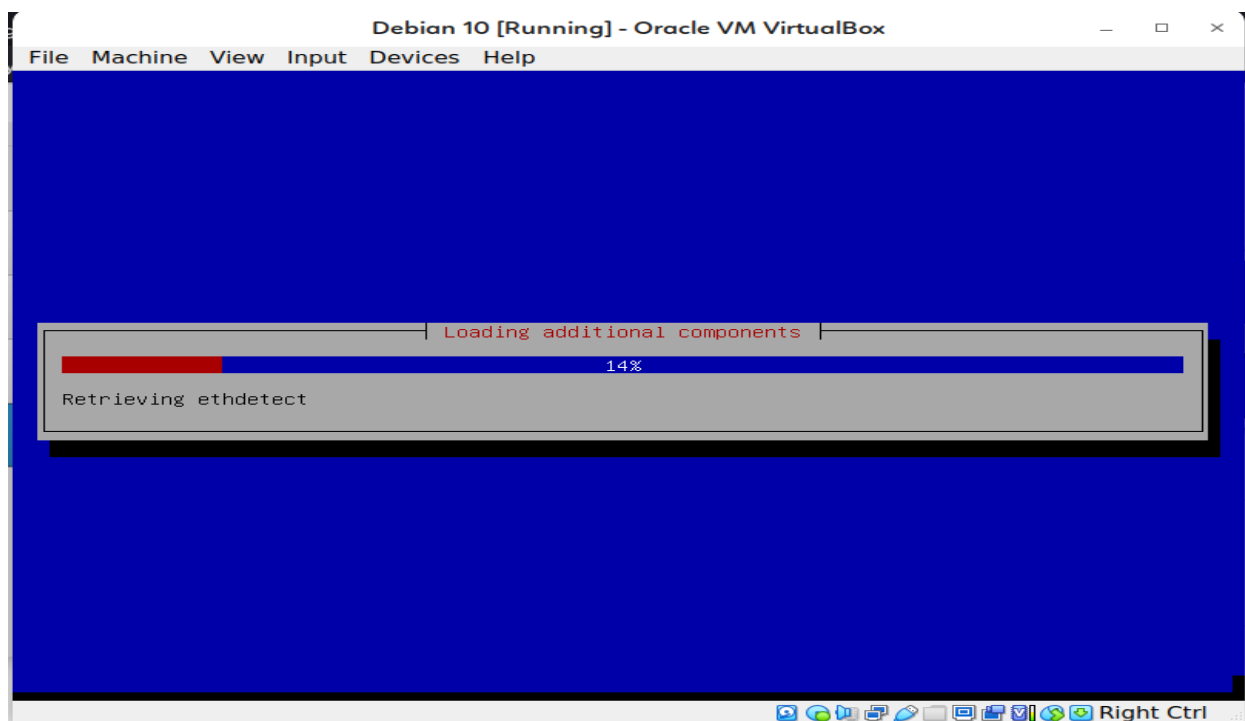
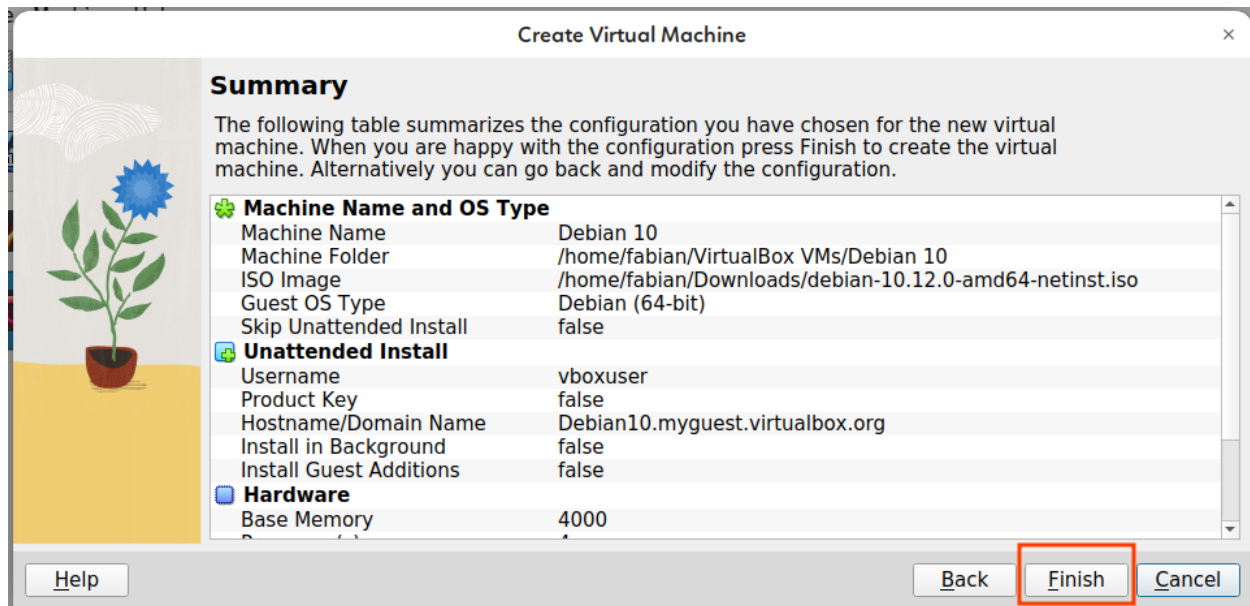
☐ Use an Existing Virtual Hard Disk File

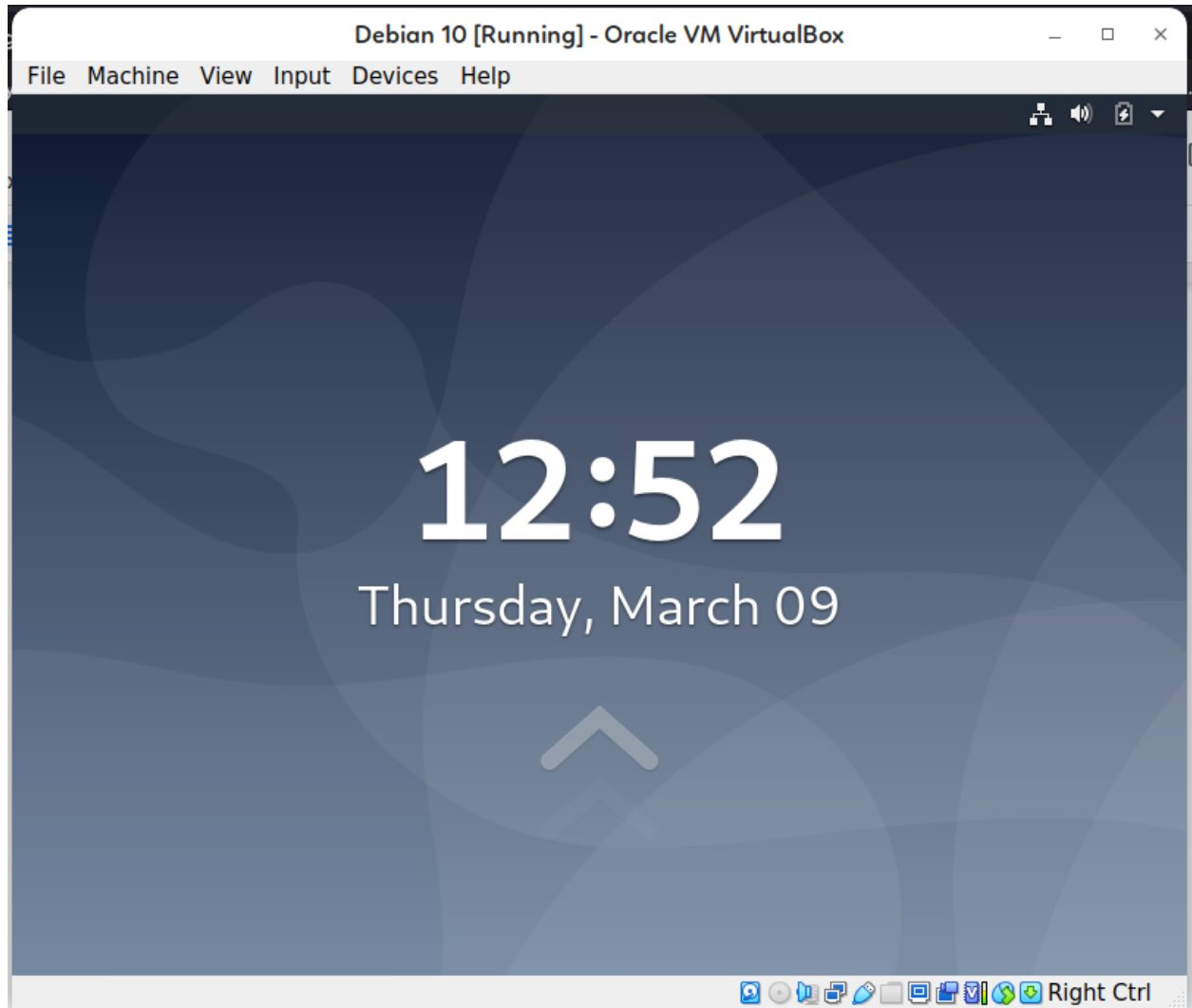
Debian.vdi (Normal, 30,00 GB)

☐ Do Not Add a Virtual Hard Disk

Help Back Next Cancel

Click Finish and wait until the installation ends...





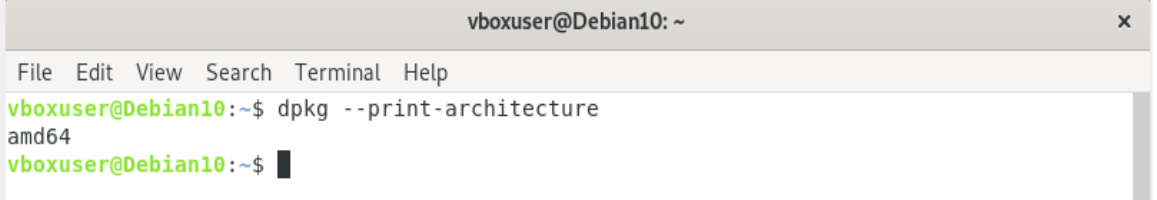
Prepare the development environment

To be compatible with the architecture of the syrus4 device (**armhf**), it is necessary to install both the **armhf** architecture and the libraries related to this architecture.

Using a linux console, follow the next steps:

1. Check the current architectures installed:

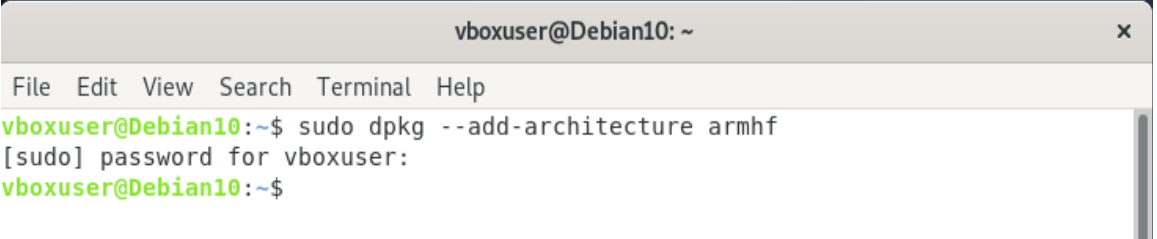
```
dpkg --print-architecture
```



A terminal window titled 'vboxuser@Debian10: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'vboxuser@Debian10:~\$'. The command 'dpkg --print-architecture' is entered, and the output 'amd64' is displayed on the next line. The prompt is then 'vboxuser@Debian10:~\$' with a cursor.

2. If **armhf** is not installed, install it:

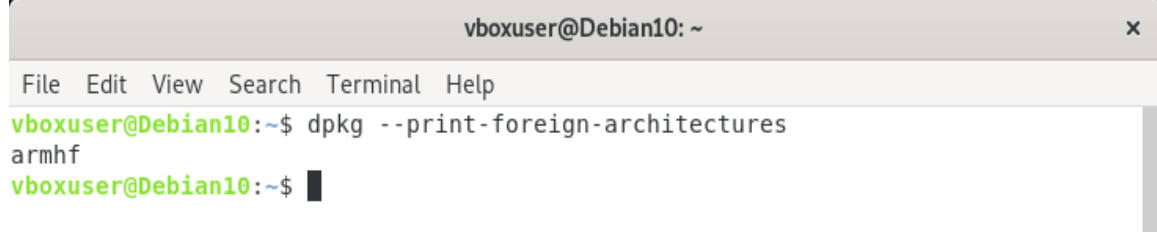
```
sudo dpkg --add-architecture armhf
```



A terminal window titled 'vboxuser@Debian10: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'vboxuser@Debian10:~\$'. The command 'sudo dpkg --add-architecture armhf' is entered. The next line shows '[sudo] password for vboxuser:'. The prompt then returns to 'vboxuser@Debian10:~\$'.

3. Check that the installation was OK

```
sudo dpkg --print-foreign-architectures
```



A terminal window titled 'vboxuser@Debian10: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'vboxuser@Debian10:~\$'. The command 'sudo dpkg --print-foreign-architectures' is entered. The next line shows 'armhf'. The prompt then returns to 'vboxuser@Debian10:~\$'.

4. Update the repositories, repeat this process each time you need to install a new library.

```
sudo apt-get update
```

```
vboxuser@Debian10:~$ sudo apt-get update
Hit:1 http://deb.debian.org/debian buster InRelease
Hit:2 http://security.debian.org/debian-security buster/updates InRelease
Hit:3 http://deb.debian.org/debian buster-updates InRelease
Get:4 http://deb.debian.org/debian buster/main armhf Packages [7,706 kB]
Get:5 http://security.debian.org/debian-security buster/updates/main armhf Packages [462 kB]
Get:6 http://deb.debian.org/debian buster-updates/main armhf Packages [8,784 B]
Fetched 8,176 kB in 3s (2,649 kB/s)
Reading package lists... Done
vboxuser@Debian10:~$
```

```
vboxuser@Debian10:~$
```

- ```
sudo apt-get install build-essential crossbuild-essential-armhf
```

```
vboxuser@Debian10:~$ sudo apt-get install build-essential crossbuild-essential-armhf
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.6).
The following additional packages will be installed:
 binutils-arm-linux-gnueabihf cpp-8-arm-linux-gnueabihf
 cpp-arm-linux-gnueabihf cross-config dpkg-cross g++-8-arm-linux-gnueabihf
```

- ```
sudo apt-get install libsystemd-dev:armhf
```

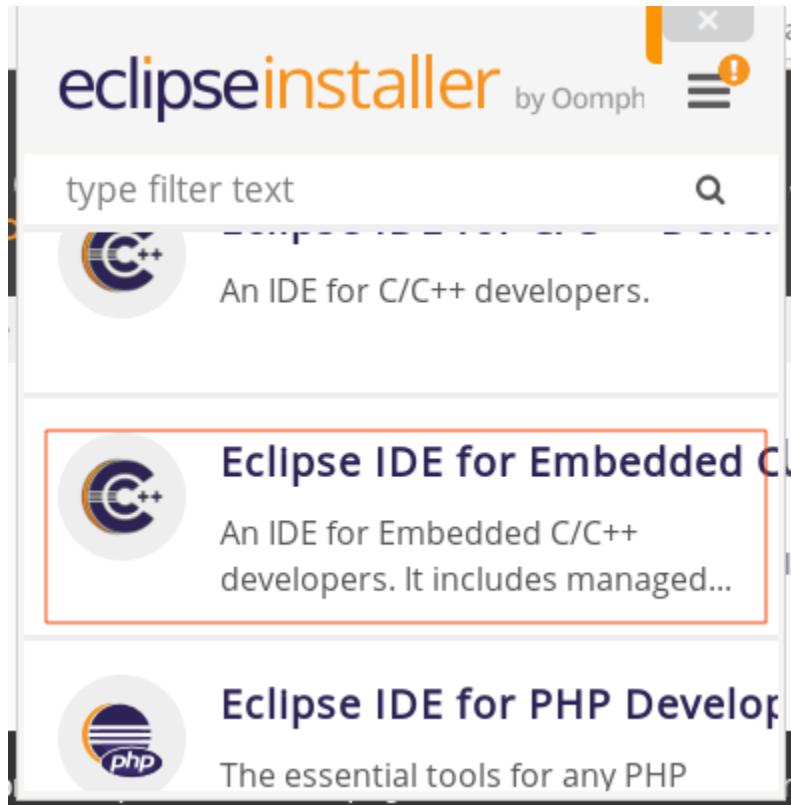
```
vboxuser@Debian10:~$ sudo apt-get install libsystemd-dev:armhf
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  gcc-8-base:armhf libc6:armhf libgcc1:armhf libgcrypt20:armhf libgpg-error-l10n
  libgpg-error0:armhf libidn2-0:armhf liblz4-1:armhf liblzma5:armhf libsystemd0:armhf
  libunistring2:armhf
Suggested packages:
```

Remember to put `":armhf"` after each library you want to install

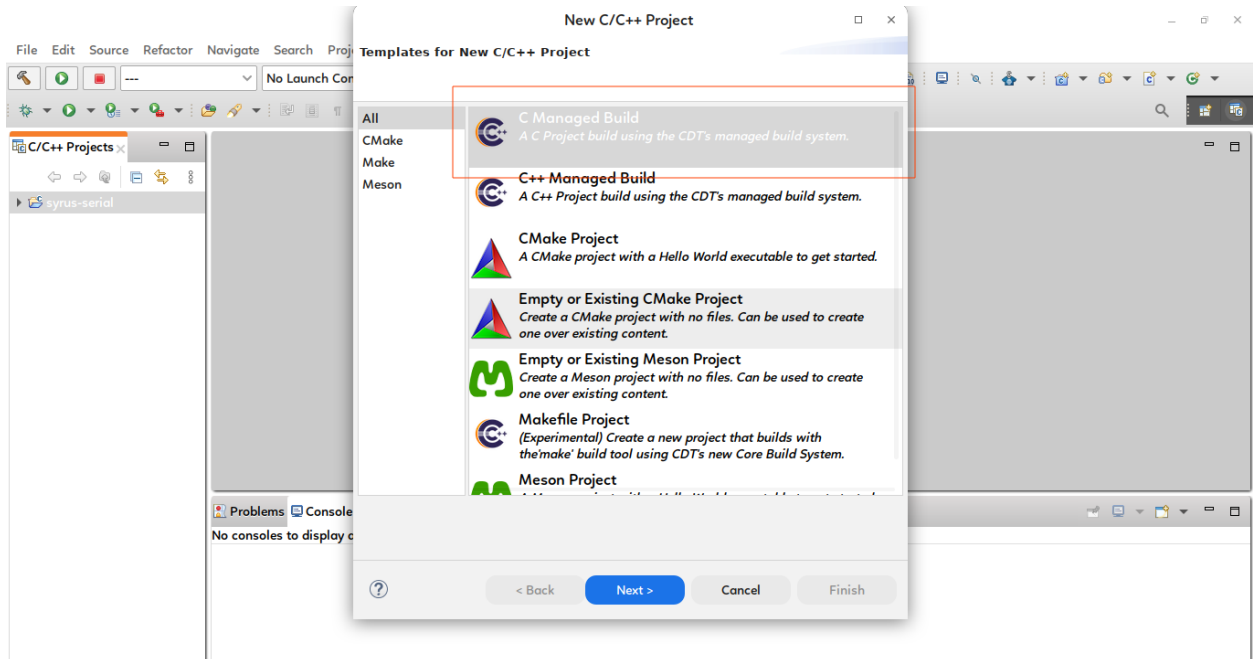
Develop Process

For easy development it is recommended to use the Eclipse IDE that allows easy library configuration and compiling with error detection.

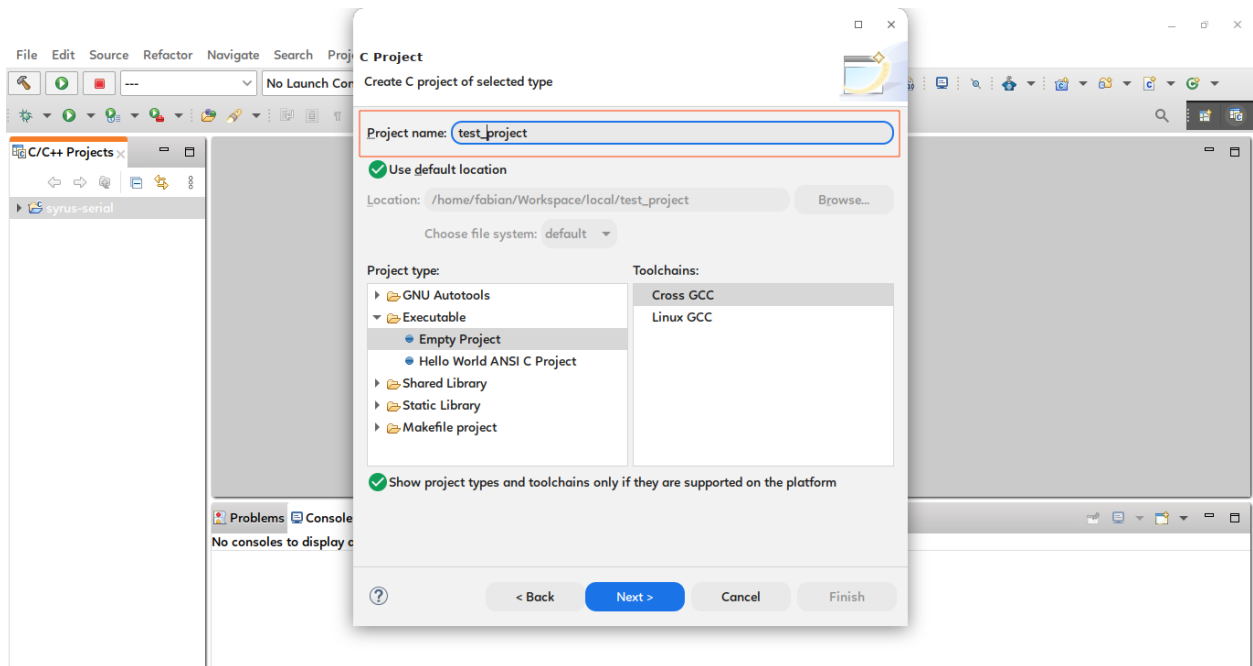
Download Eclipse from: <https://www.eclipse.org/downloads/>



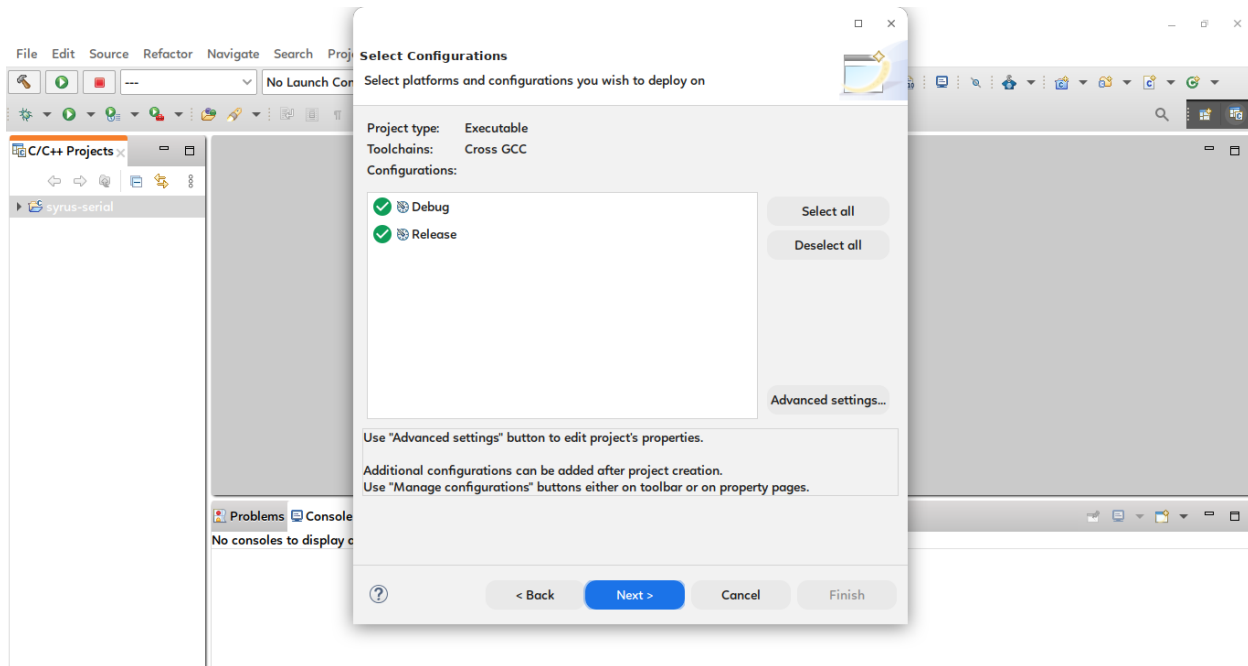
once installed open , Click on File → New → C/C++ project



Name the project and click next

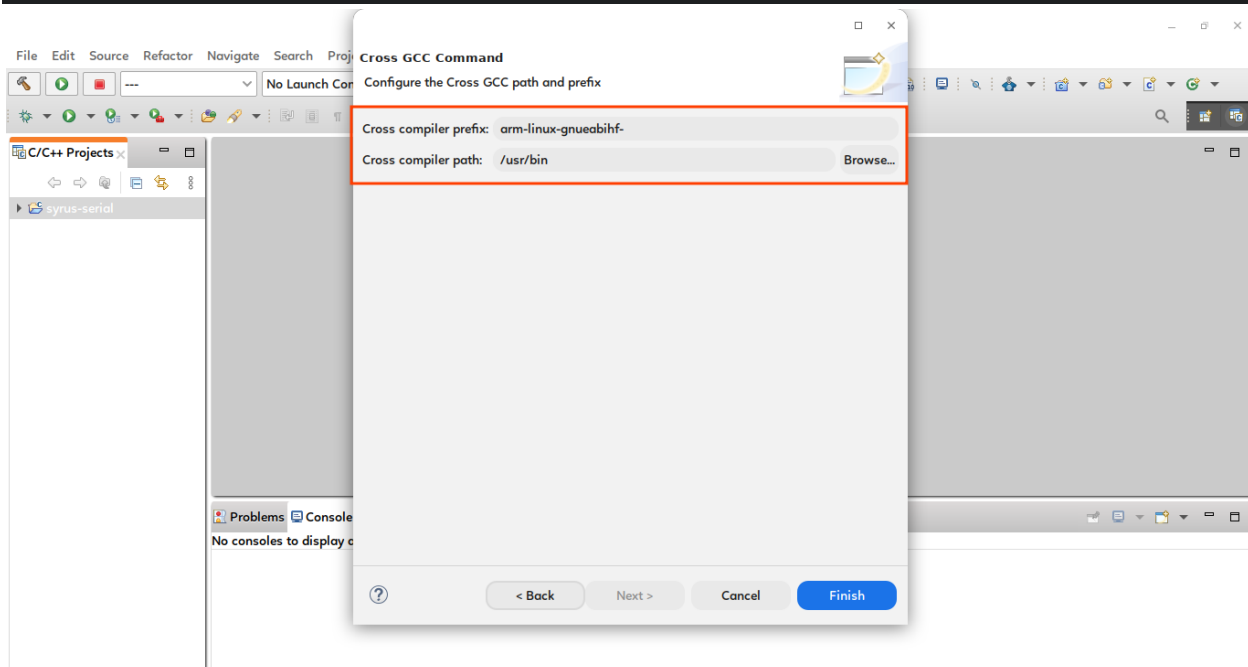


Click next



Important be sure to set these variables in this step for each project:

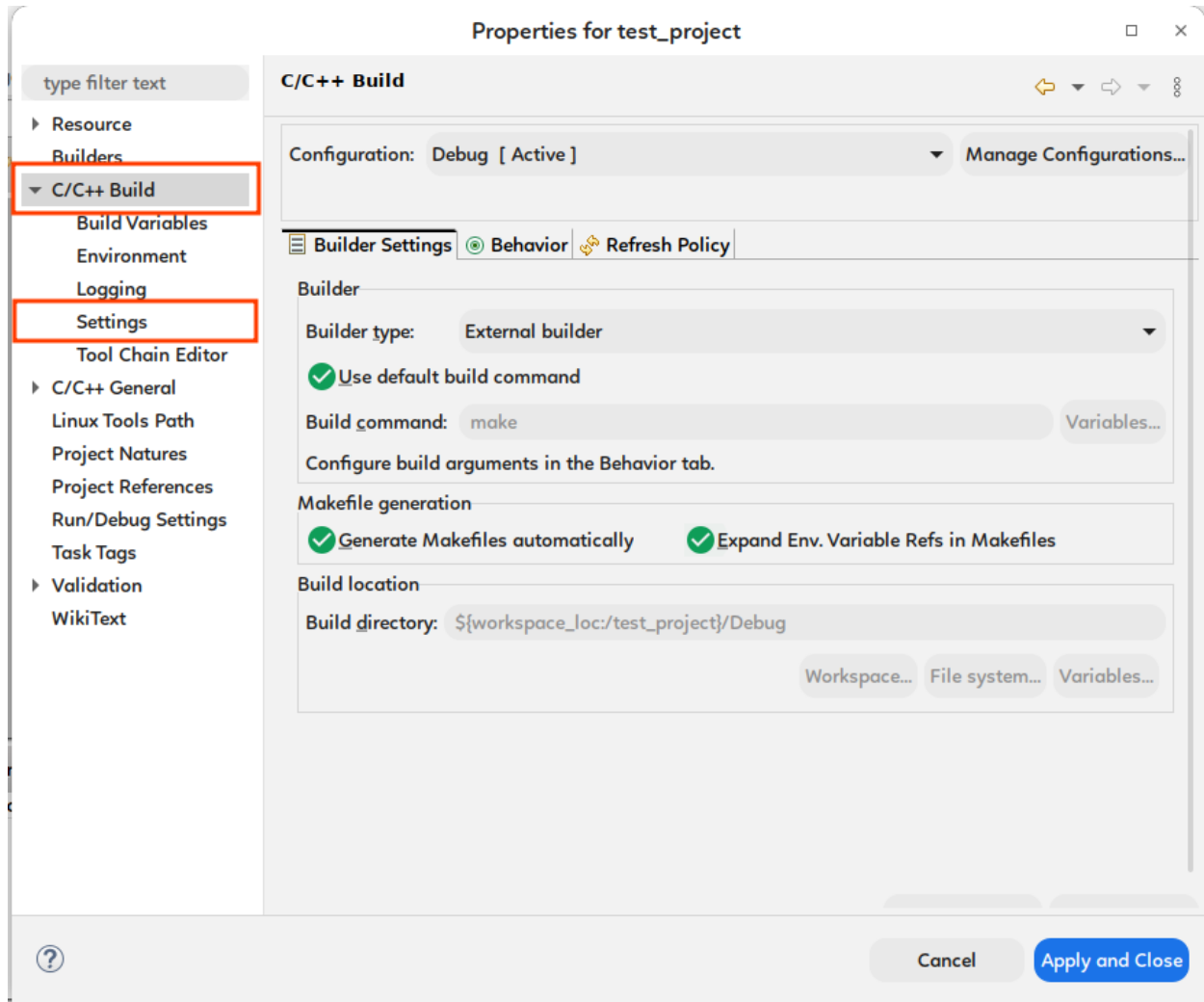
```
Cross compiler prefix: arm-linux-gnueabi-  
Cross compiler path: /usr/bin
```



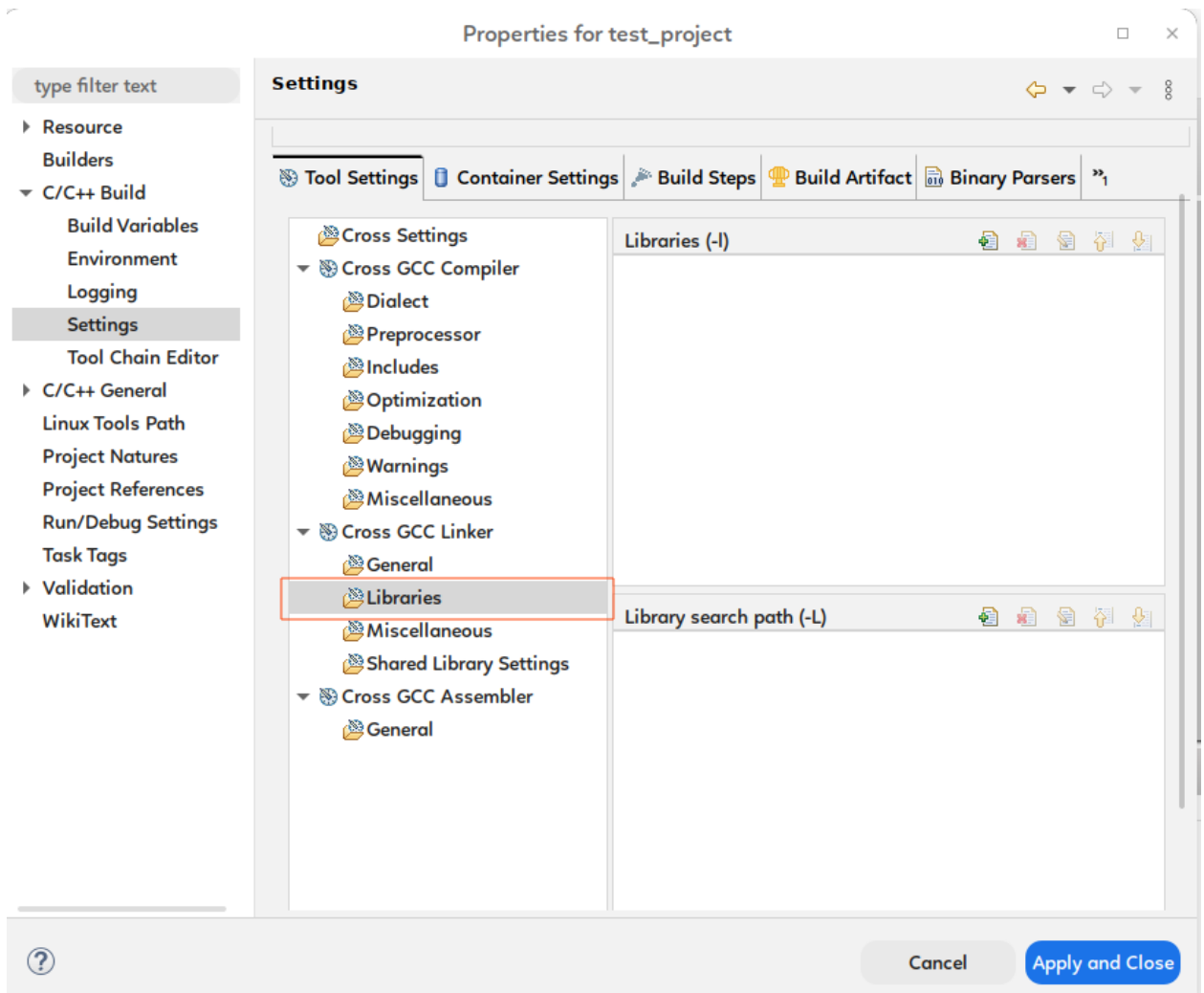
Once created, you can start to develop in C.

If you need to add a previously installed library follow this steps:

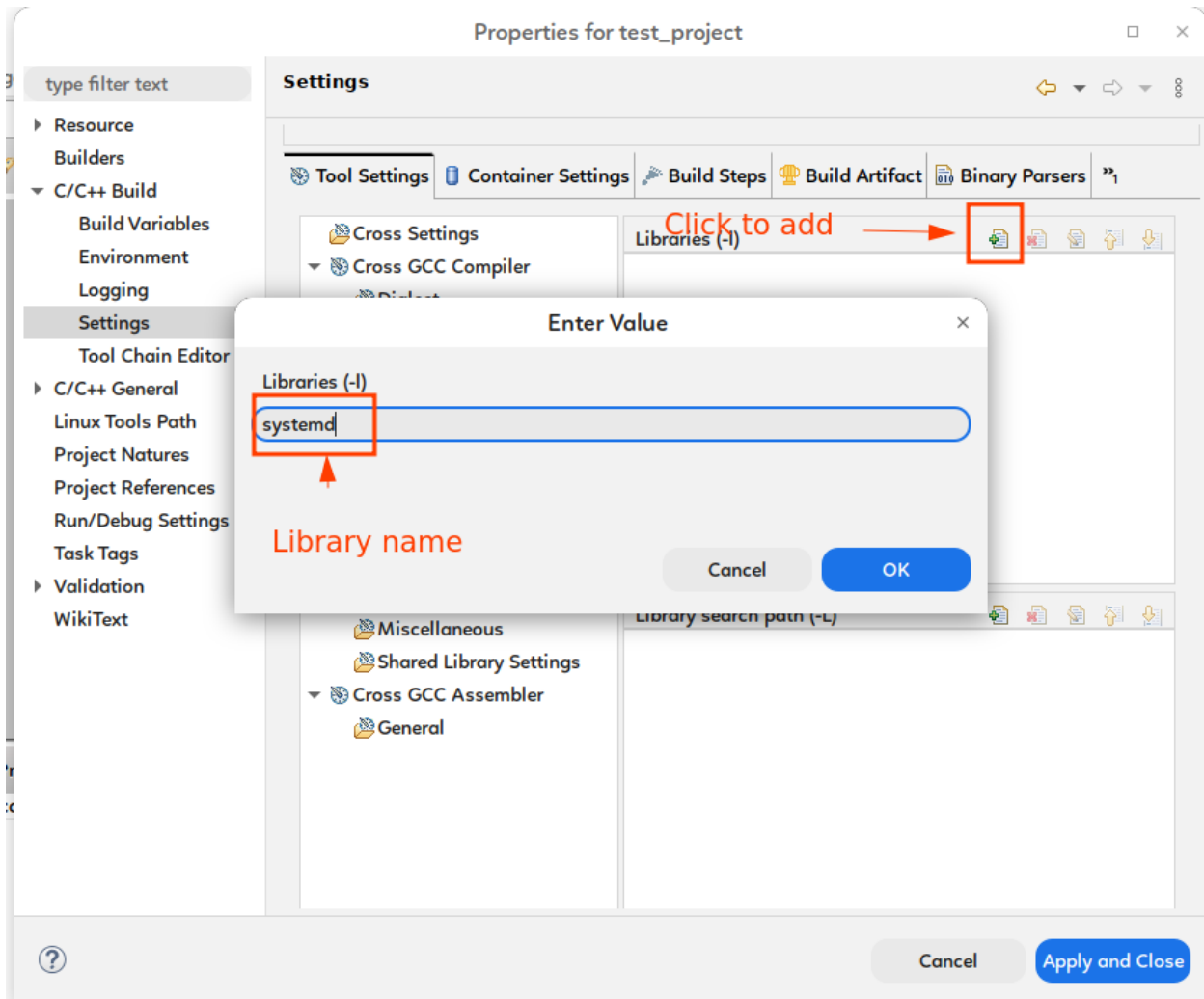
1. Right click on the project → Properties



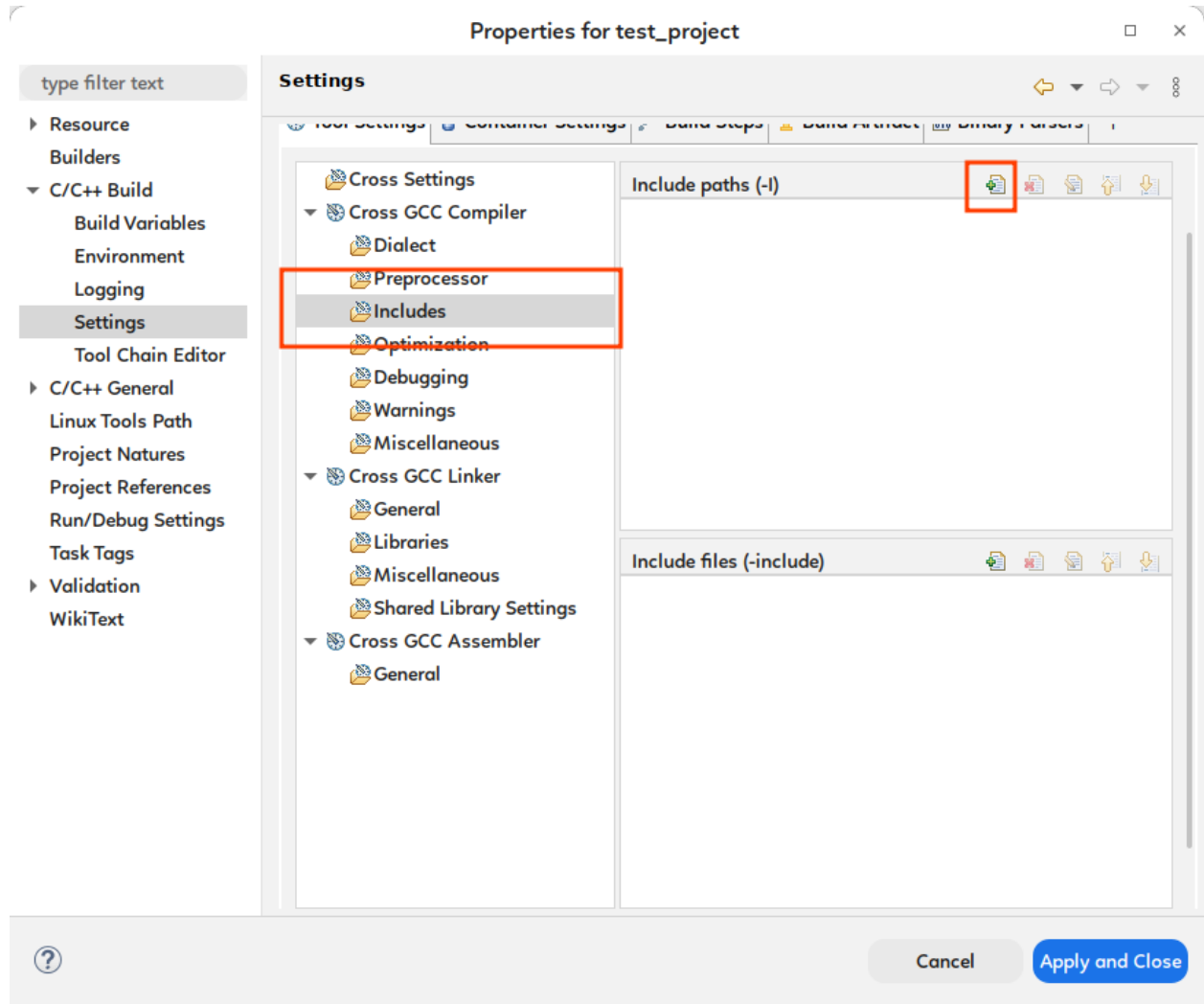
2. Click on C/C++ → settings → and go to libraries



3. Click on the closed icon and put the name of the library

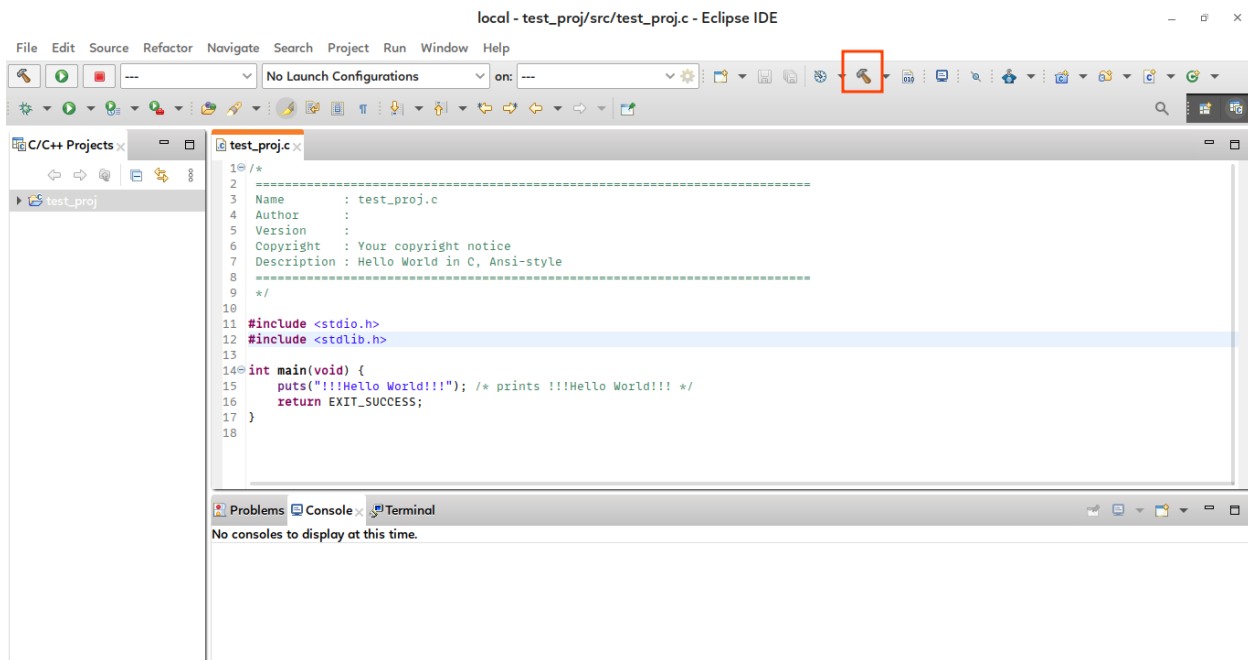


4. Click on OK → Apply and Close, Remember that if you need to add an include path in the same section go to include and repeat the previous step

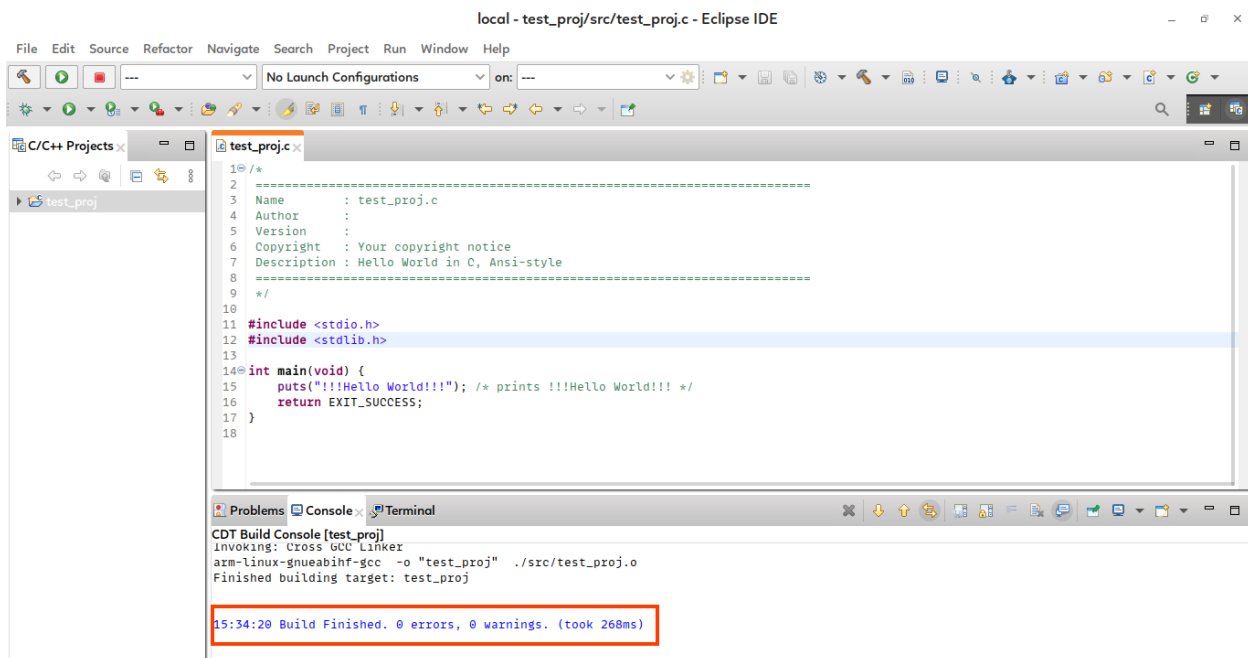


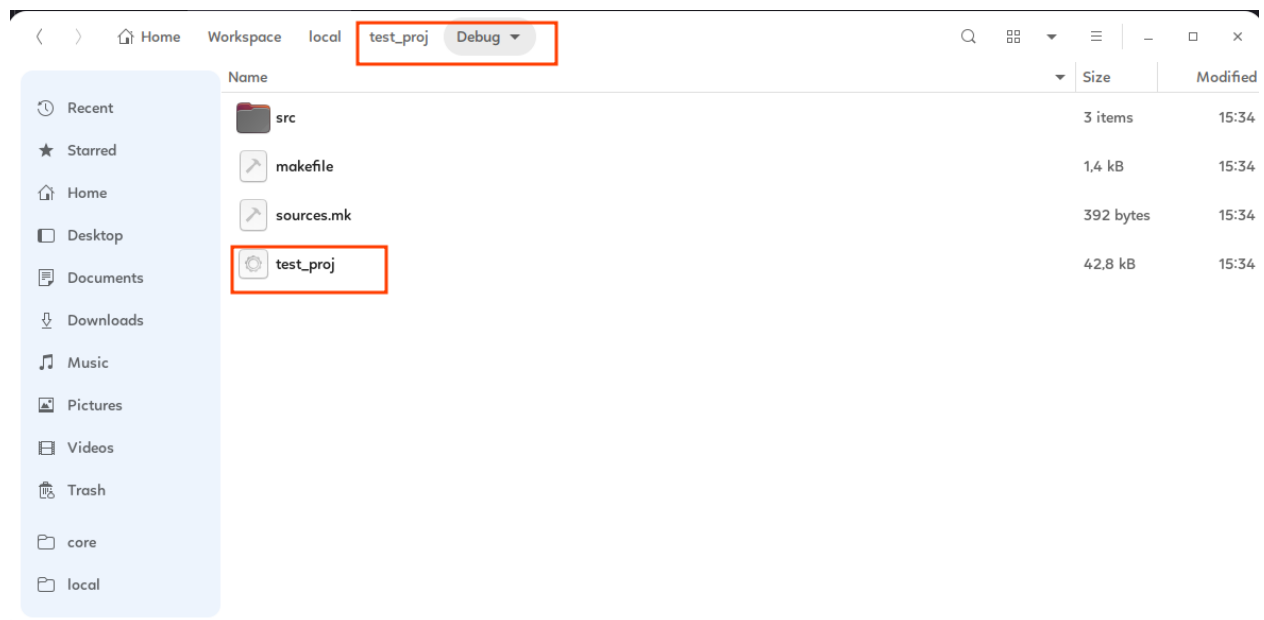
To add .c or .h files in your project, right click on the project → new → choose “header or source” file.

To compile your project press “CTRL + B”, right click on the project and select “build project” or click on the next icon



Once your application is compiled without errors, go to the project path → Debug and extract the file with the same name as the project, this will be the binary file that is ready to run in your Syrus4G device.





Now you are ready to run a C application for Syrus4g

To Run the application

Using your linux console, go to the Debug folder

```
cd ~/Workspace/local/test_proj/Debug
```

Test the connection between your syrus4 device and your local computer. **192.168.9.2** is the IP address of the syrus4 device when it is connected to the local PC using the USB cable.

Remember to configure the Virtual Machine in order your Debian can use this network connection. The easy way to validate if your virtual machine has connection with the syrus4 device is by doing a ping test.

```
ping 192.168.9.2
```

Then, by using scp, transfer the binary file from your local machine to the Syrus device:

```
scp test_proj syrus4g@192.168.9.2:~
```

Open a remote ssh console in order to execute commands inside your syrus4.

```
ssh syrus4g@192.168.9.2
```

The password is 123456

Once there, execute your binary.

```
syrus4g@syrus-867698041101547 ~
```

```
$ ./test_proj
```

```
!!! Hello World !!!
```

If you want to convert your C application to a Syrus4 application, please consult the following documentation:

<https://syrus.digitalcomtech.com/docs/app-development>