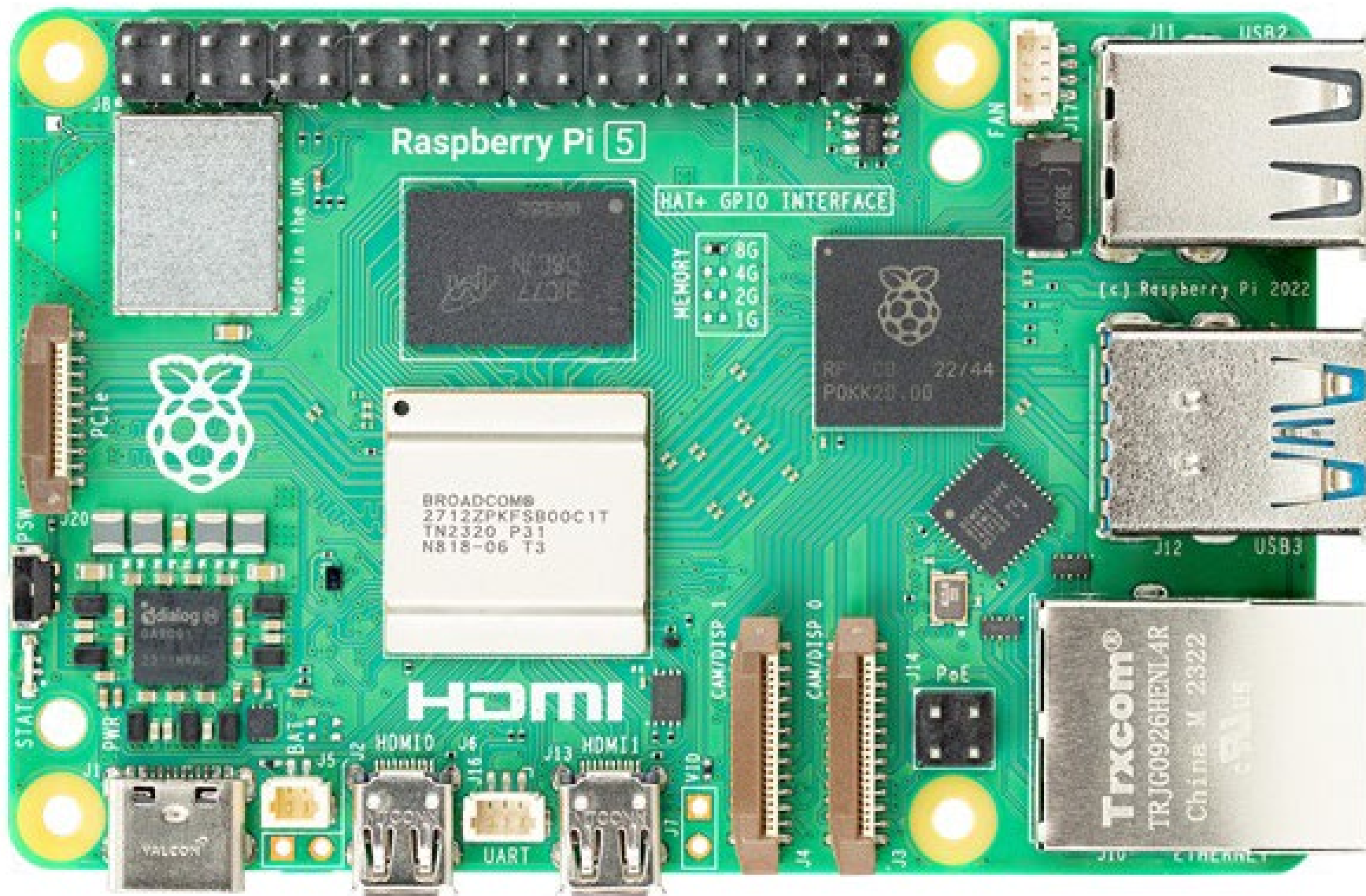
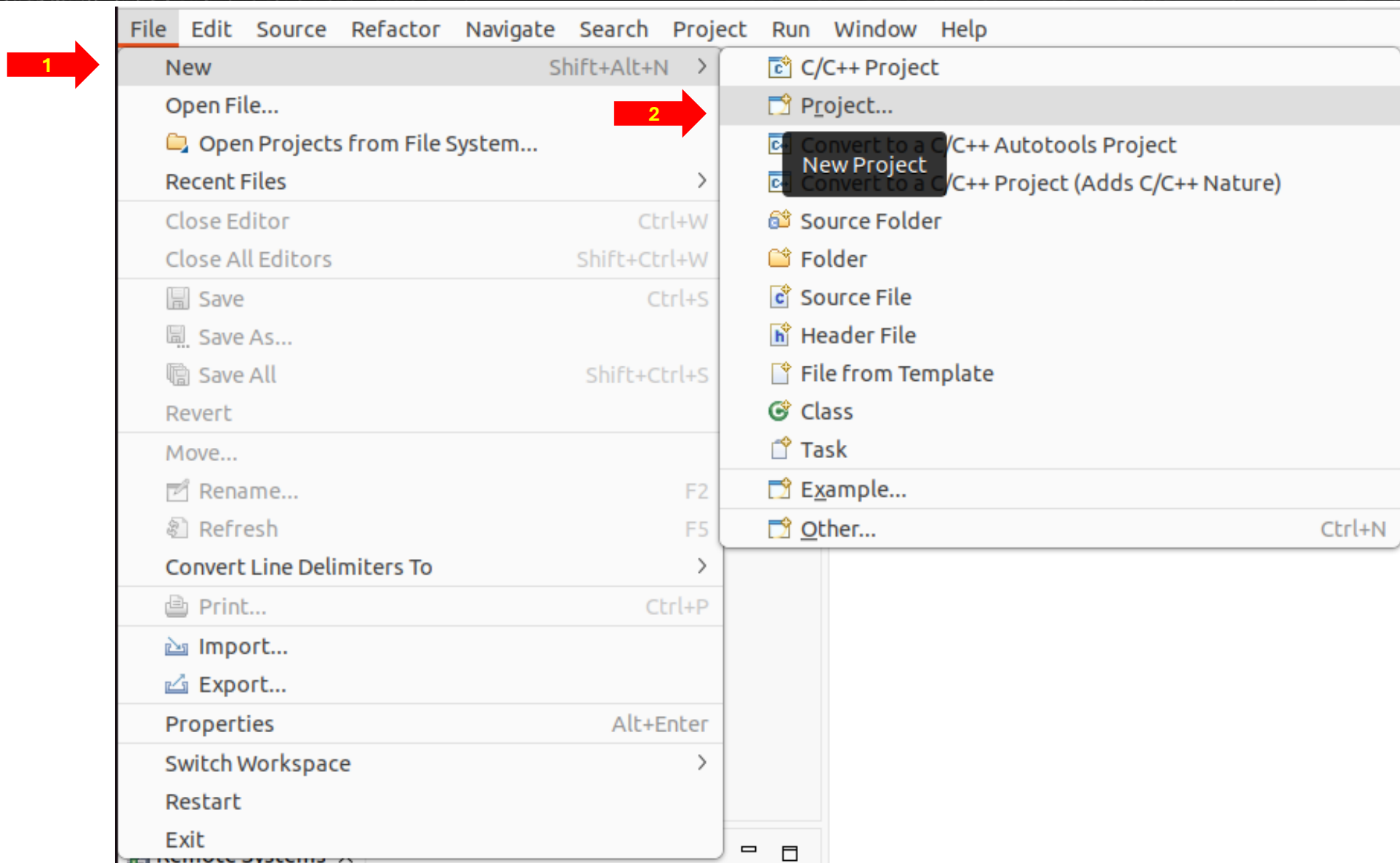


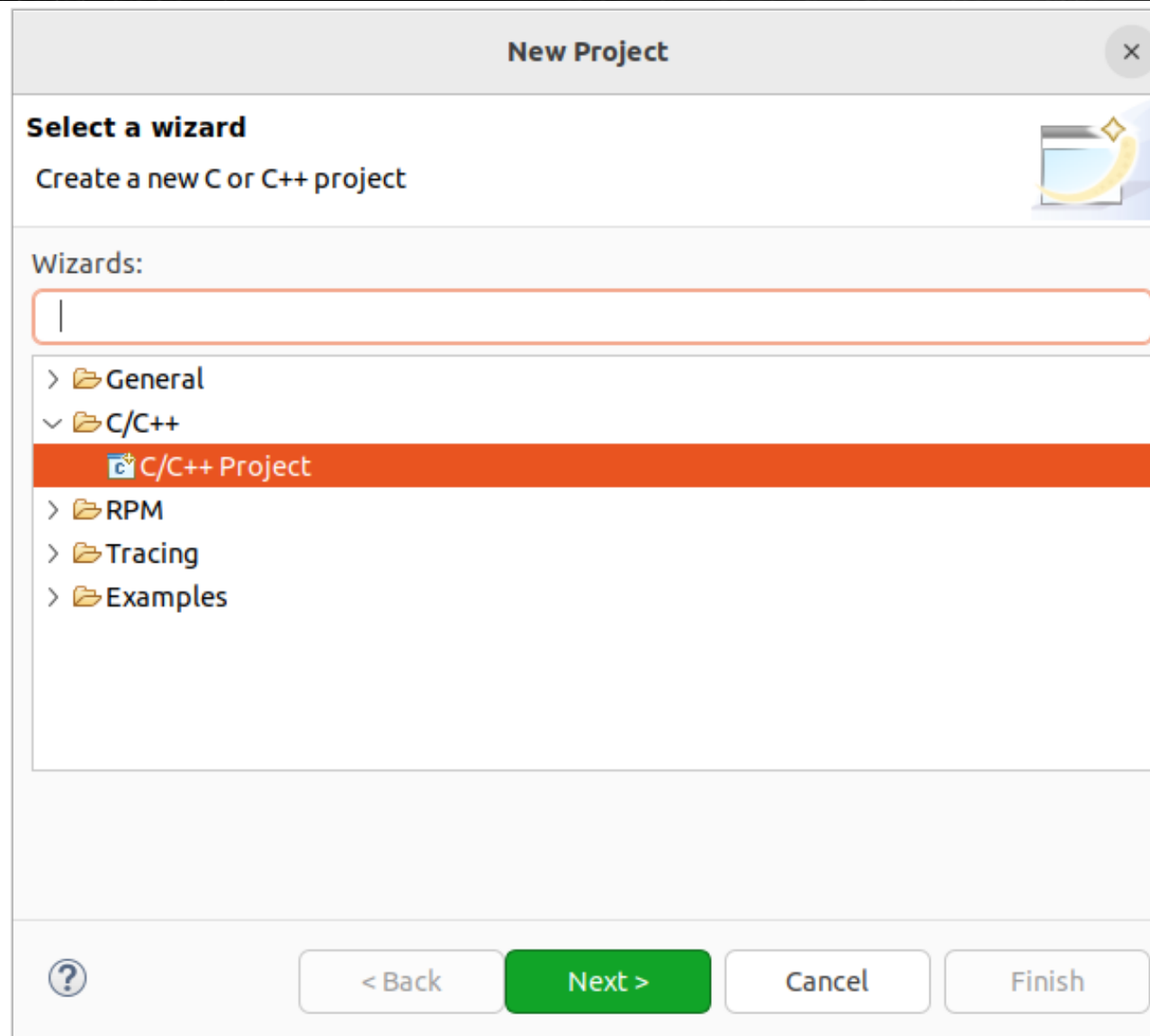
## Eclipse Yocto Application Development



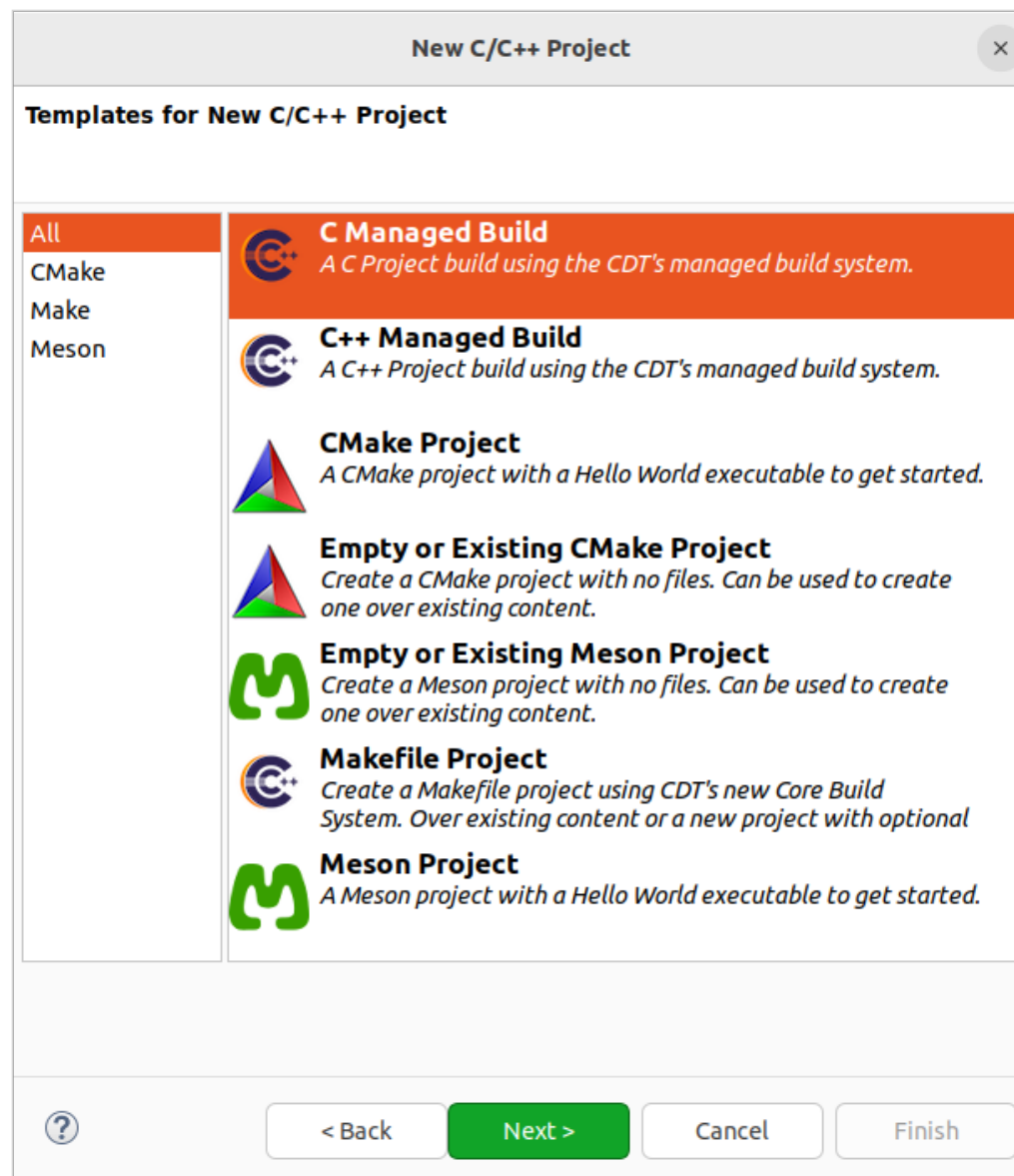
## Create a New C Project



## Create a New C Project



## Choose C Managed Build

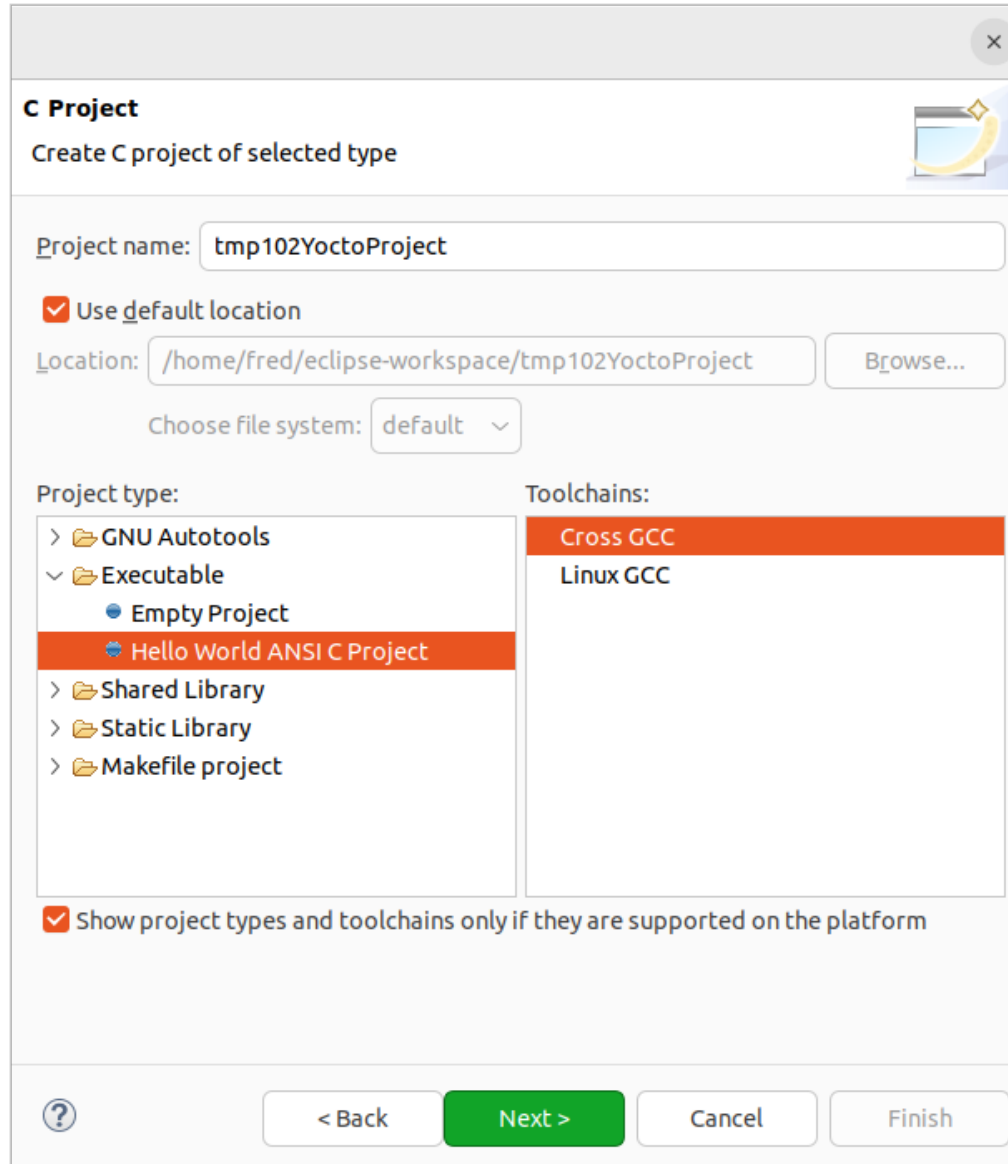


Choose this one



## Choose Hello World and Cross GCC

Select a Project Name



**C Project**  
Create C project of selected type

Project name: tmp102YoctoProject

☒ Use default location  
Location: /home/fred/eclipse-workspace/tmp102YoctoProject Browse...

Choose file system: default ▾

Project type:

- > GNU Autotools
- ▼ Executable
  - Empty Project
  - Hello World ANSI C Project**
- > Shared Library
- > Static Library
- > Makefile project

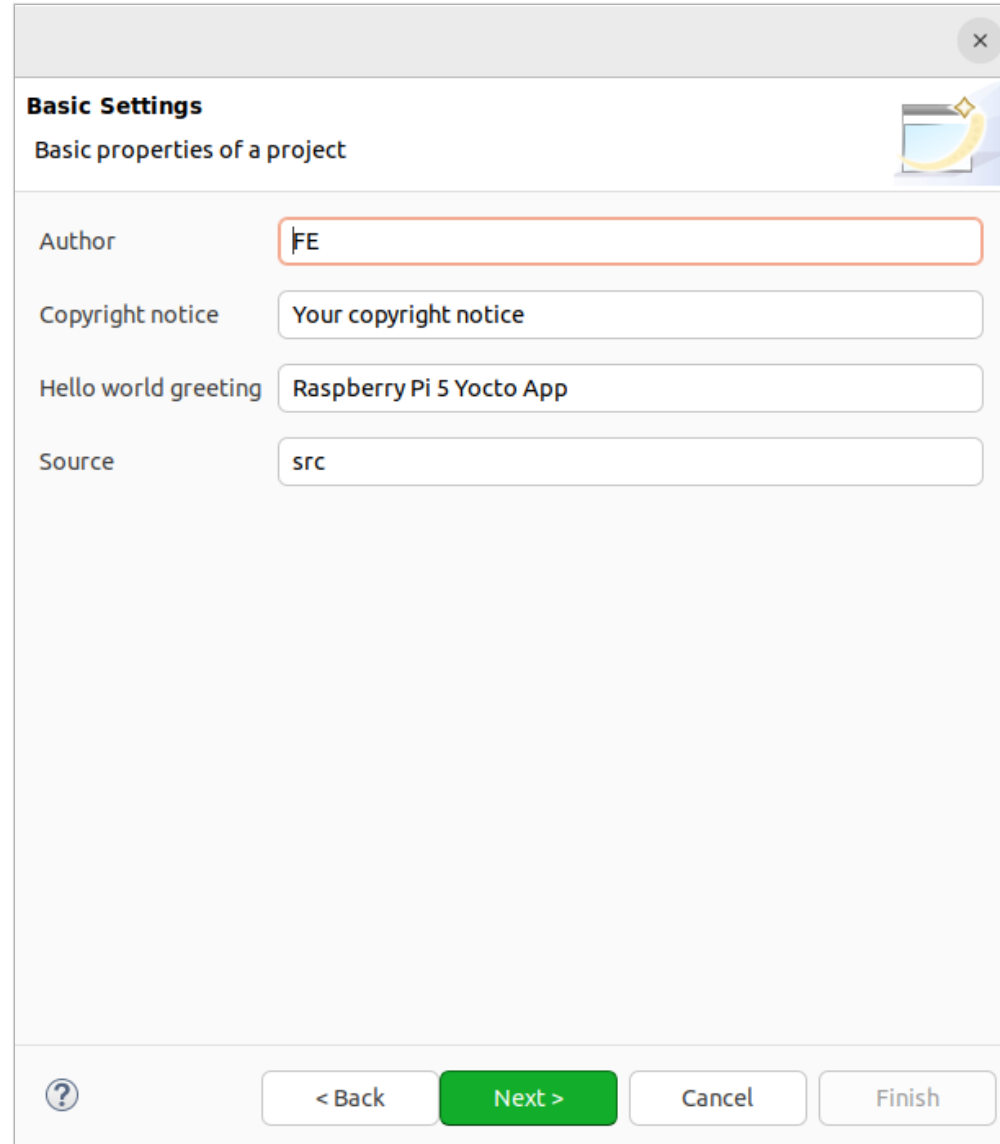
Toolchains:

- Cross GCC**
- Linux GCC

☒ Show project types and toolchains only if they are supported on the platform

? < Back Next > Cancel Finish

## Choose Hello World Greeting (Optional)



**Basic Settings**  
Basic properties of a project

Author: FE

Copyright notice: Your copyright notice

Hello world greeting: Raspberry Pi 5 Yocto App

Source: src

< Back Next > Cancel Finish

## Choose Debug and Release

×


Select Configurations


Select platforms and configurations you wish to deploy on

Project type: Executable

Toolchains: Cross GCC

Configurations:

☒  Debug

☒  Release

Select all

Deselect all

Advanced settings...

Use "Advanced settings" button to edit project's properties.

Additional configurations can be added after project creation.

Use "Manage configurations" buttons either on toolbar or on property pages.

?

< Back

Next >

Cancel

Finish

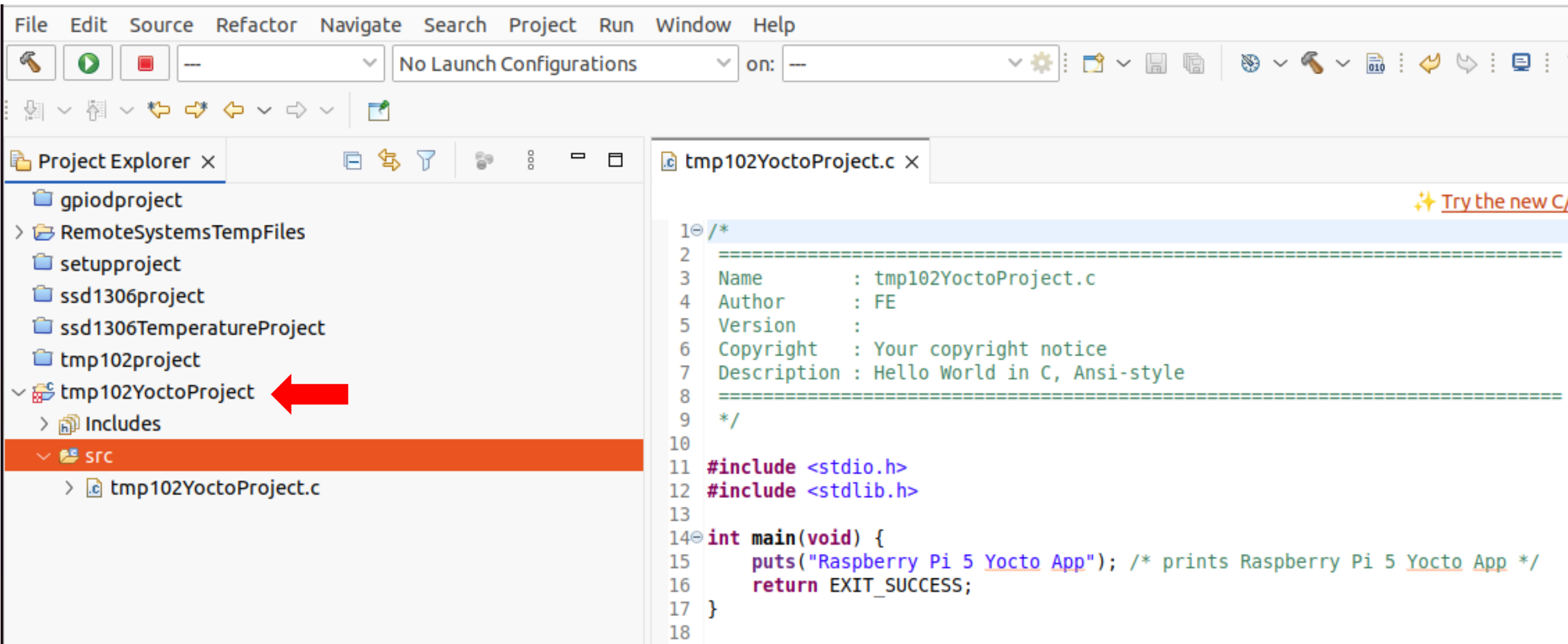
7

## Specify the Cross Compiler Prefix and Path

Cross compiler prefix: aarch64-linux-gnu-  
Cross compiler path: /usr/bin Browse...  
  
Make sure to add the "-".  
  
< Back Next > Cancel Finish

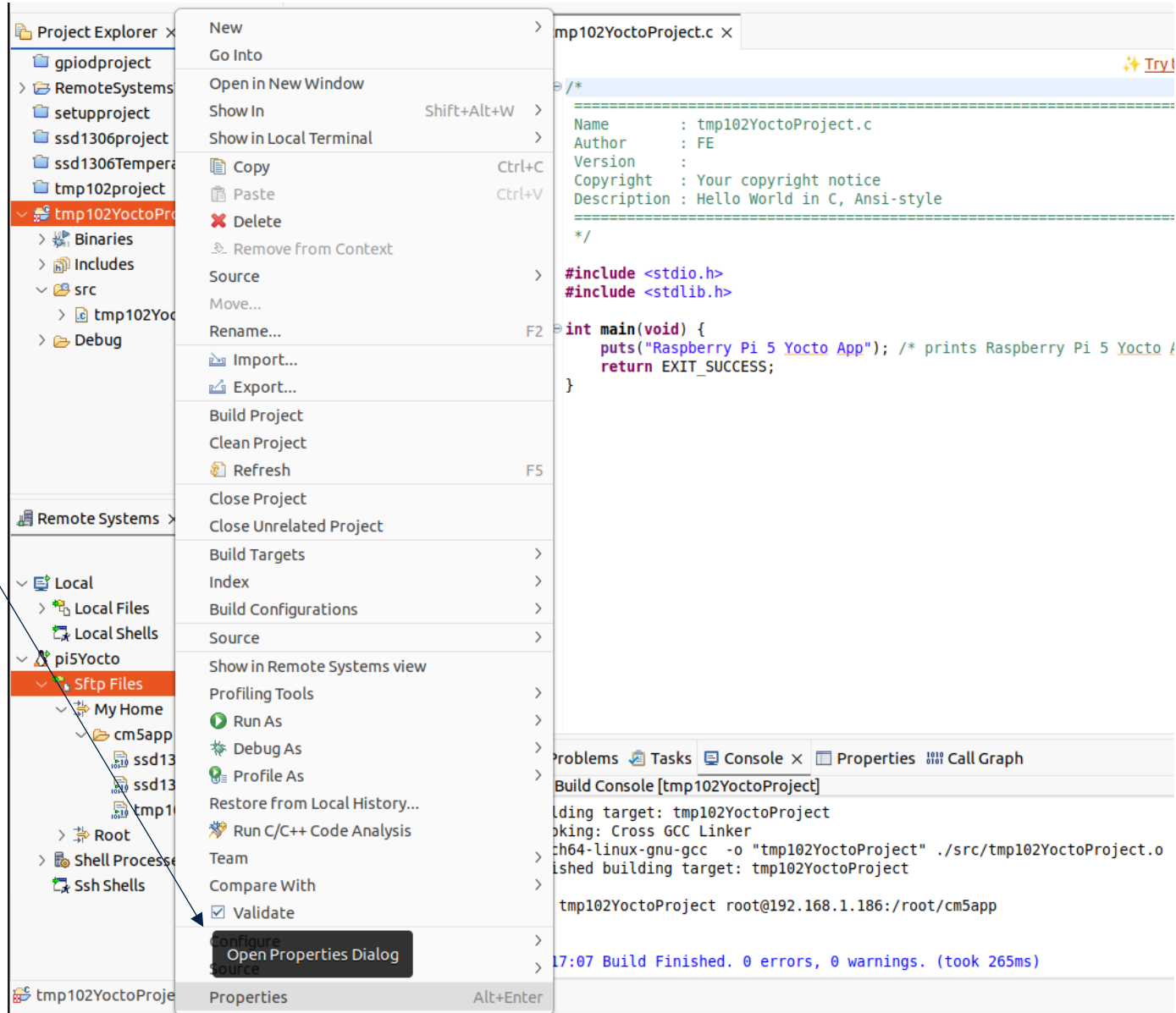


## tmp102yoctoProject Created



## How to Open the Properties Dialog

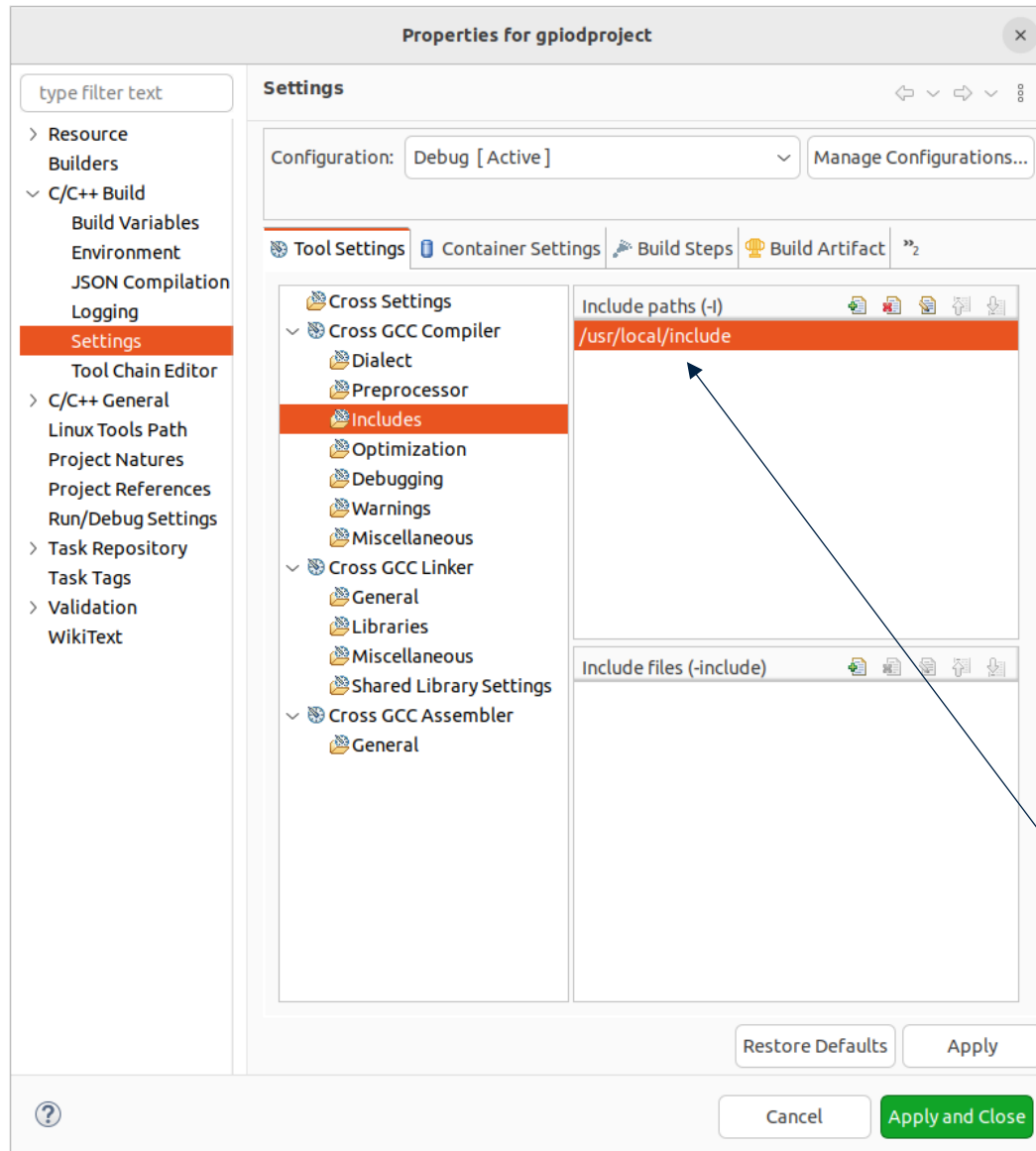
Mouse Right Click



The screenshot shows the Eclipse IDE interface. In the Project Explorer on the left, the 'tmp102YoctoProject' is selected. A right-click context menu is open, and the 'Open Properties Dialog' option is highlighted. The main editor shows the source code for 'tmp102YoctoProject.c', which includes standard C headers and a main function that prints 'Raspberry Pi 5 Yocto App'. The bottom status bar indicates the build is finished with 0 errors and 0 warnings.

```
/*  
 * Name      : tmp102YoctoProject.c  
 * Author    : FE  
 * Version   :  
 * Copyright : Your copyright notice  
 * Description: Hello World in C, Ansi-style  
 */  
  
#include <stdio.h>  
#include <stdlib.h>  
  
int main(void) {  
    puts("Raspberry Pi 5 Yocto App"); /* prints Raspberry Pi 5 Yocto /  
    return EXIT_SUCCESS;  
}
```

## Properties Settings When Using libgpod



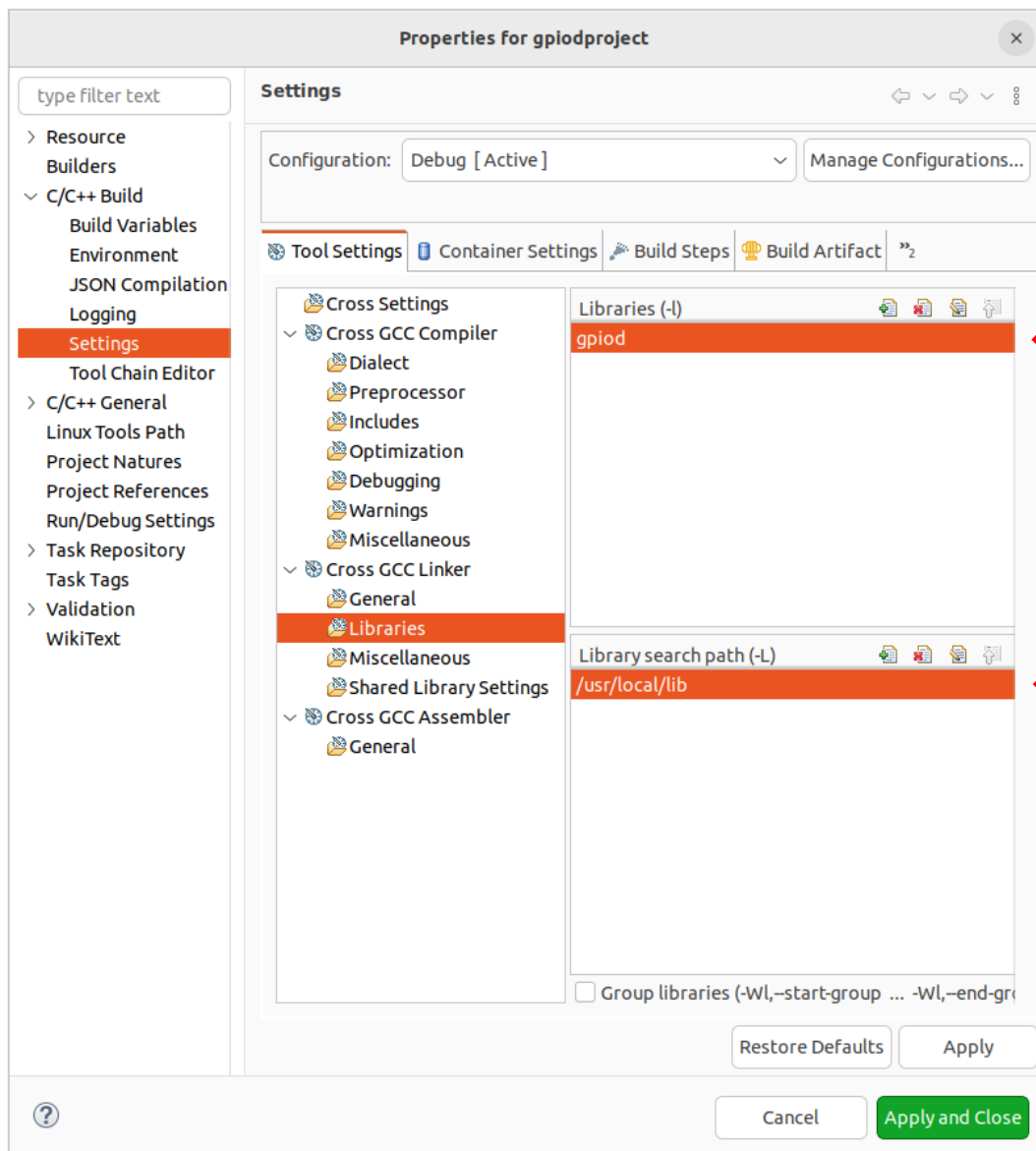
Pi 5 libgpod library files are located in the Raspberry Pi 5 directory `/usr/lib`.  
Copy Pi 5 libgpod library files to Ubuntu Host directory `/usr/local/lib`.

- **libgpod.so.3.1.1**
- **libgpodcxx.so.2.1.0**
- **libgpod.so -> libgpod.so.3.1.1**
- **libgpod.so.3 -> libgpod.so.3.1.1**
- **libgpodcxx.so -> libgpodcxx.so.2.1.0**
- **libgpodcxx.so.2 -> libgpodcxx.so.2.1.0**

Pi 5 `gpod.h` is located in the Raspberry Pi 5 directory `/usr/include`.  
Copy Pi 5 version of `gpod.h` to Ubuntu Host directory `/usr/local/include`.

Add `/usr/local/include` to Include paths (-I)

## Properties Settings When Using libgpiod

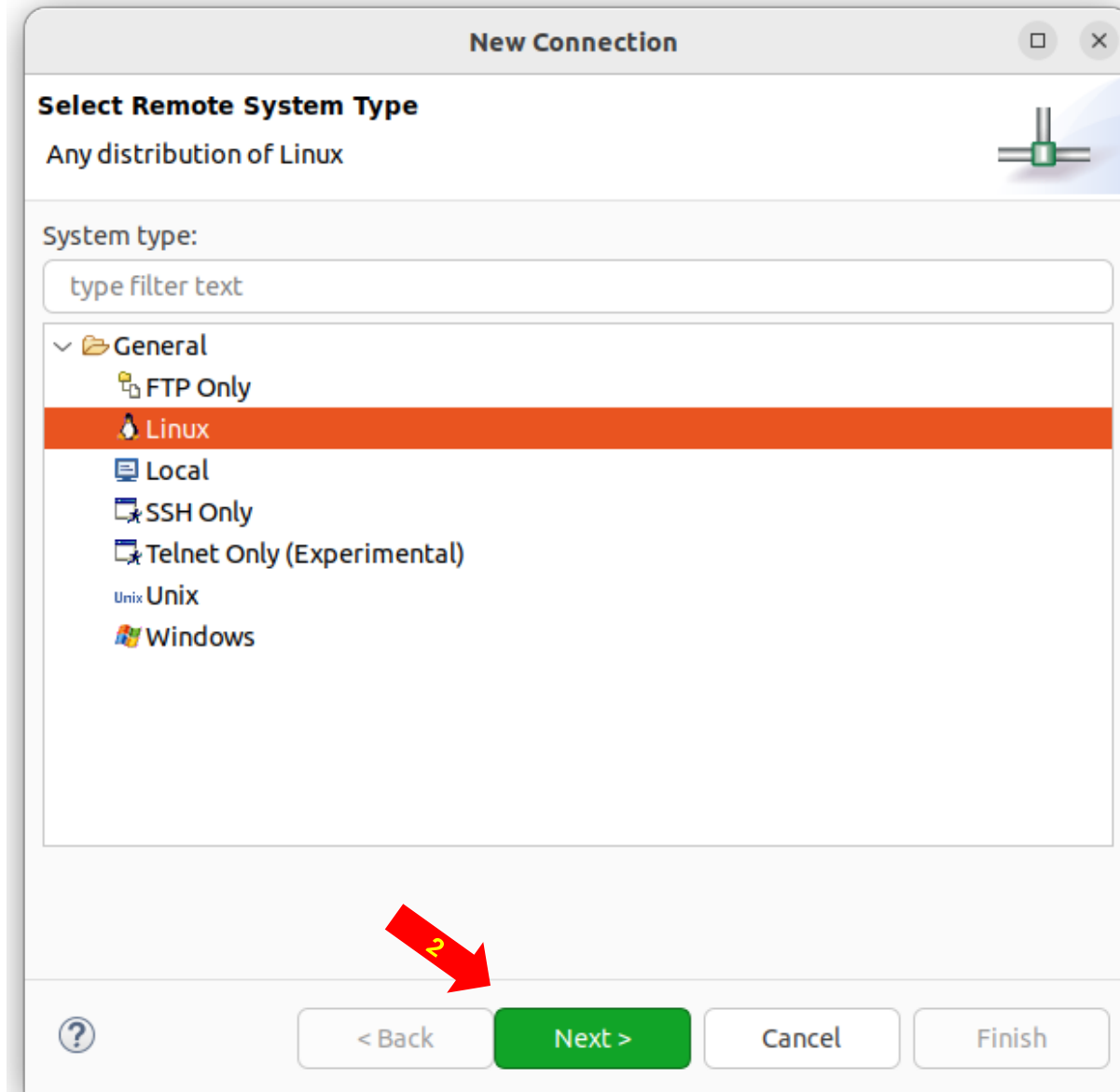


Add gpiod (-l) to linker libs

Add /usr/local/lib to (-L) linker libs search path

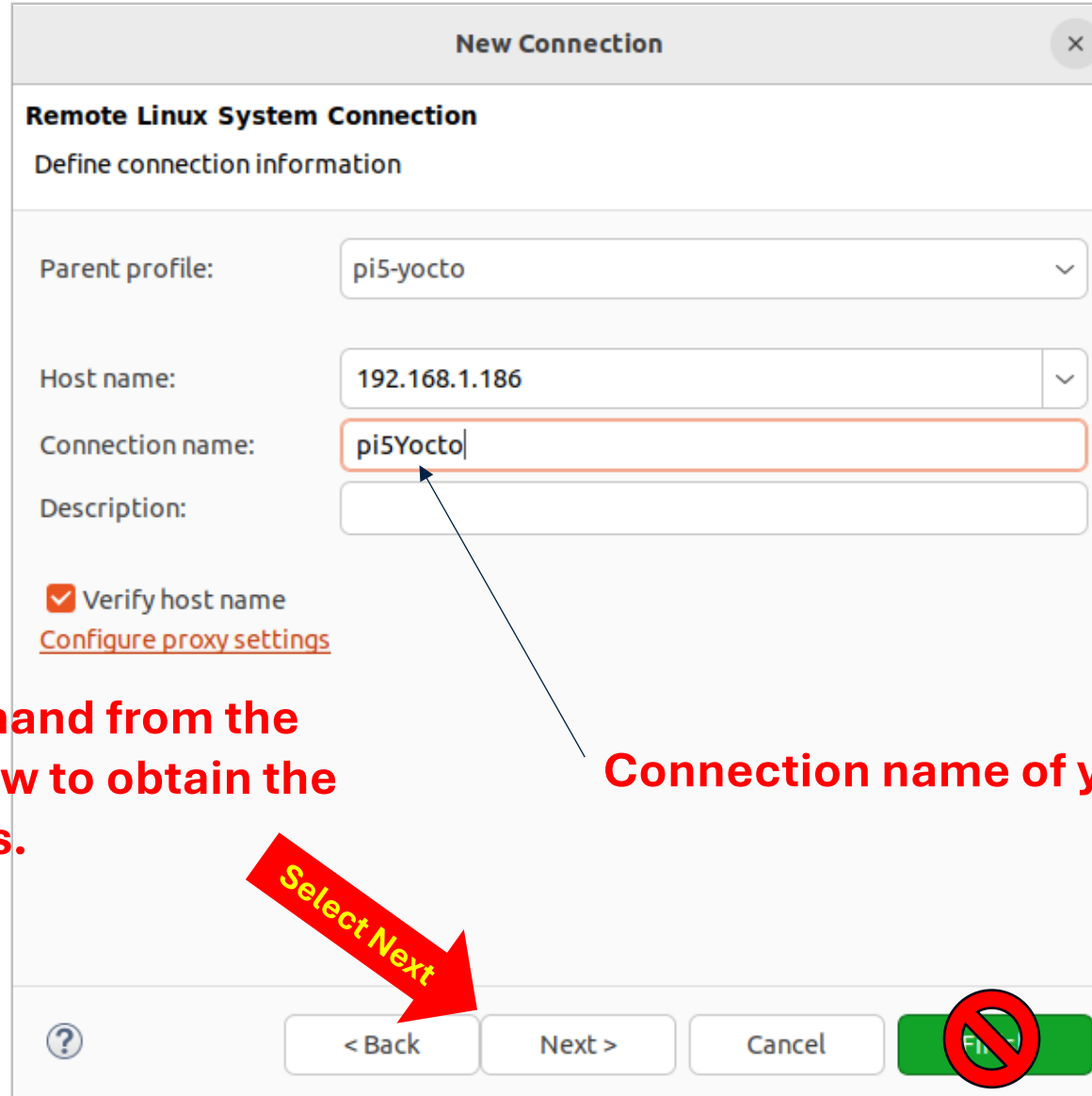


## Define and Make a Remote System Connection





## Define and Make a Remote System Connection



New Connection

Remote Linux System Connection

Define connection information



Parent profile: pi5-yocto

Host name: 192.168.1.186

Connection name: pi5Yocto

Description:

☒ Verify host name  
[Configure proxy settings](#)

 < Back Next > Cancel 

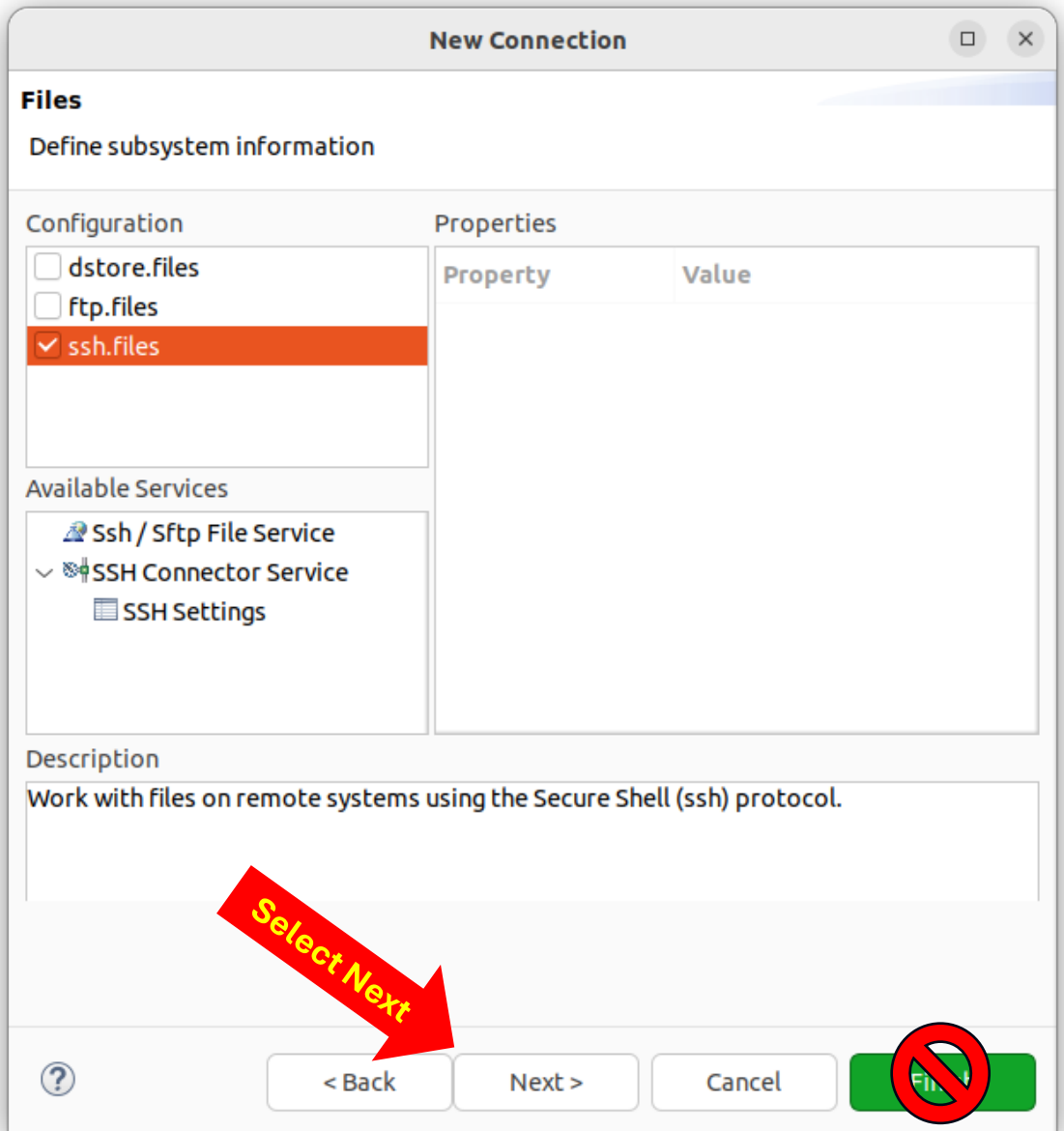
Raspberry Pi 5 IP Address

Invoke the *ifconfig* command from the *picocom* terminal window to obtain the Raspberry Pi 5 IP address.

Connection name of your choice.

Select Next

## Define and Make a Remote System Connection



**New Connection**

**Files**  
Define subsystem information

**Configuration**

- ☐ dstore.files
- ☐ ftp.files
- ☒ ssh.files

**Available Services**

- Ssh / Sftp File Service
- SSH Connector Service
  - SSH Settings

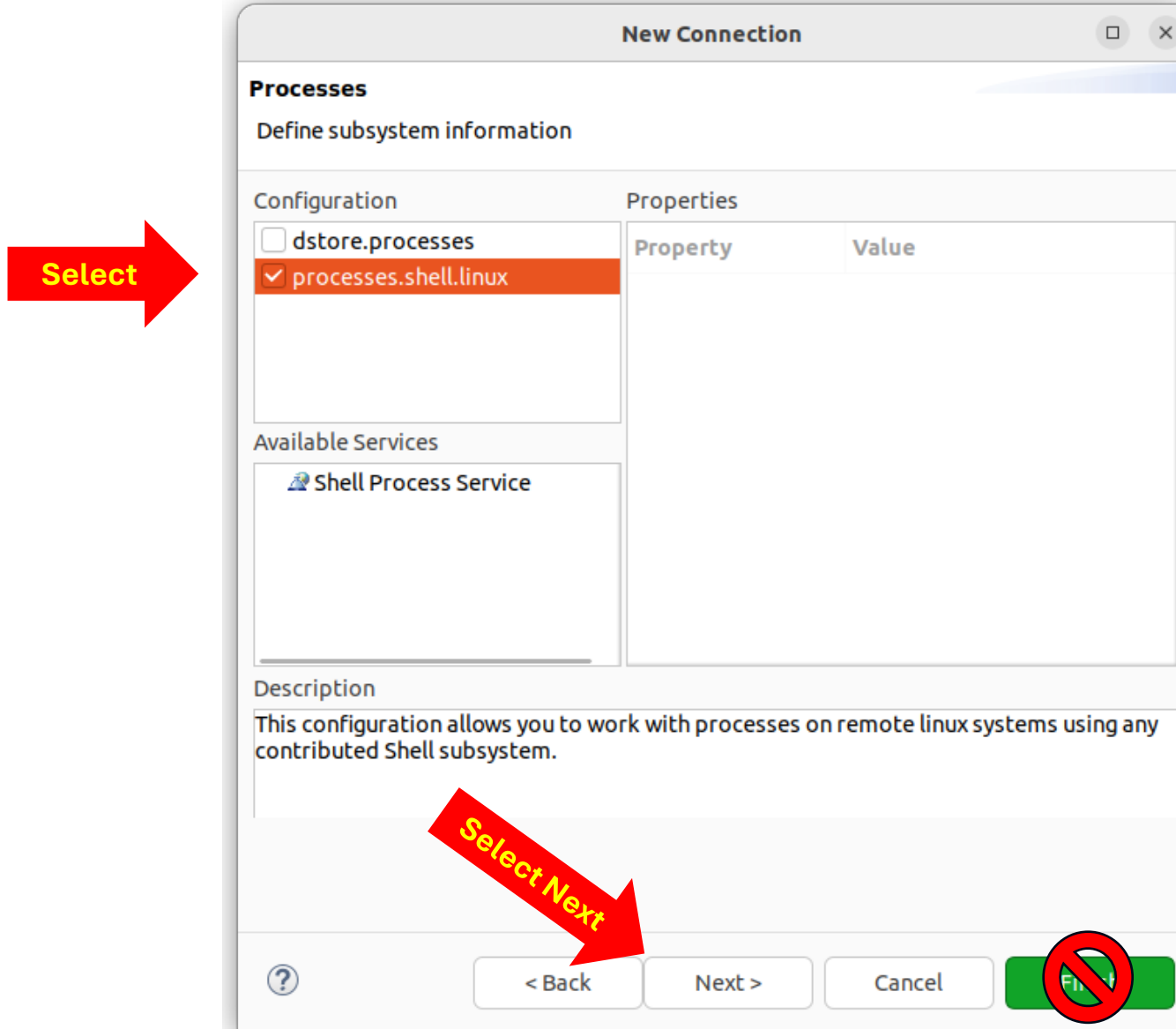
**Properties**

Property	Value
----------	-------

**Description**  
Work with files on remote systems using the Secure Shell (ssh) protocol.

**Buttons:** ? < Back Next > Cancel Finish (disabled)

## Define and Make a Remote System Connection



**New Connection**

**Processes**  
Define subsystem information

**Configuration**

- ☐ dstore.processes
- ☒ processes.shell.linux

**Available Services**

- Shell Process Service

**Properties**

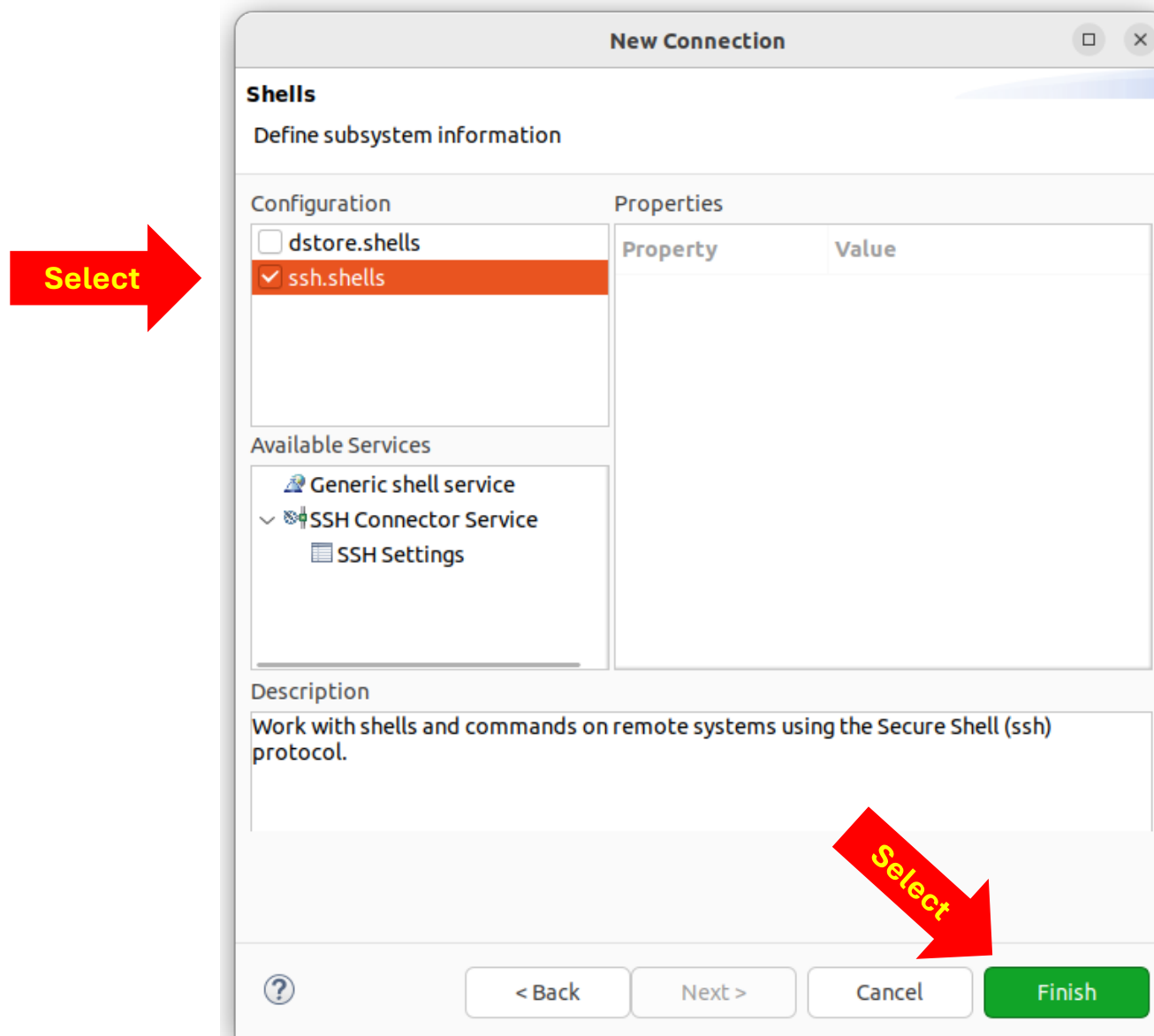
Property	Value
----------	-------

**Description**

This configuration allows you to work with processes on remote linux systems using any contributed Shell subsystem.

**Navigation:** ? < Back Next > Cancel Finish (disabled)

## Define and Make a Remote System Connection



**New Connection**

**Shells**  
Define subsystem information

Configuration	Properties		
<input type="checkbox"/> dstore.shells	<table border="1"><thead><tr><th>Property</th><th>Value</th></tr></thead><tbody></tbody></table>	Property	Value
Property	Value		
<input checked="" type="checkbox"/> ssh.shells			

**Available Services**

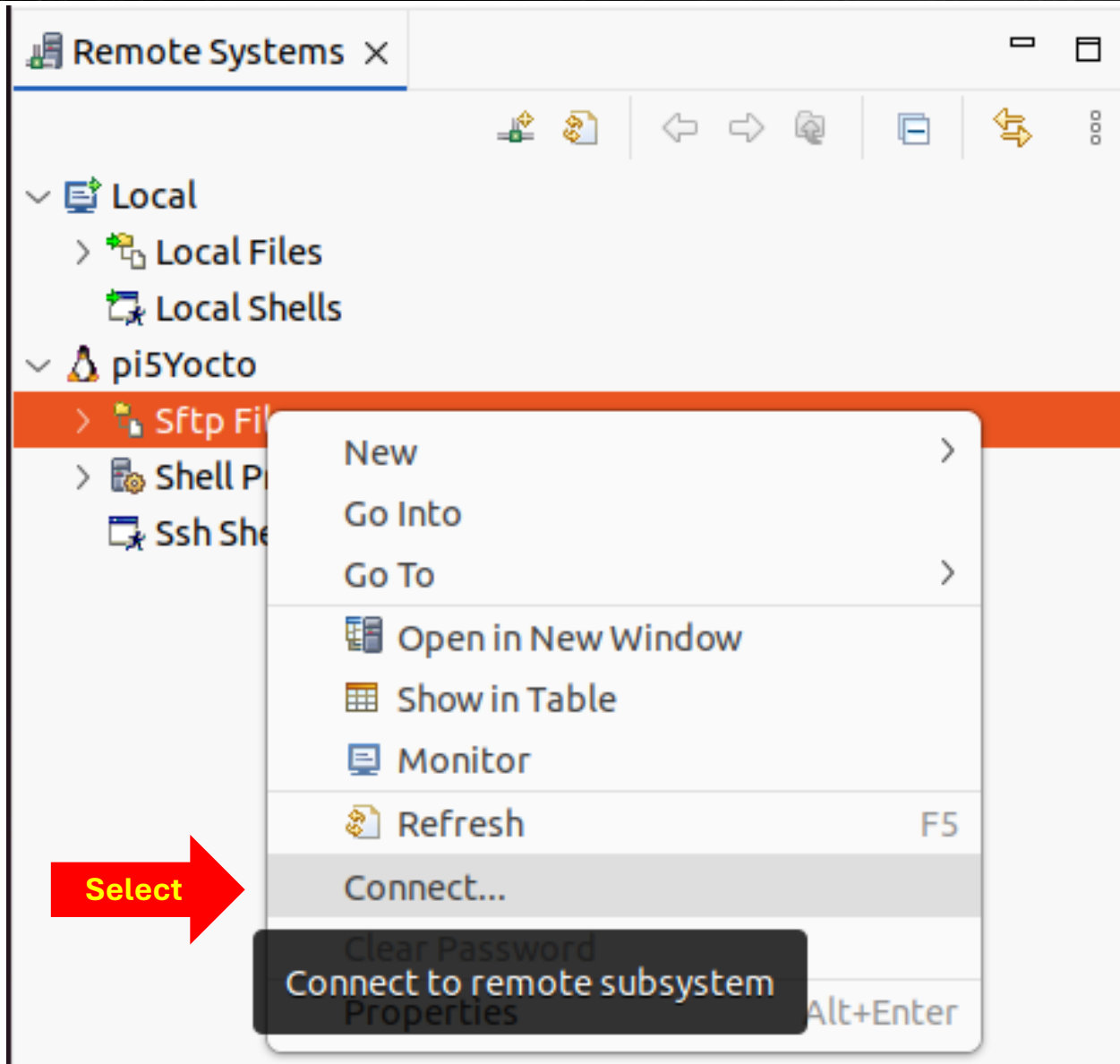
- Generic shell service
- ✓ SSH Connector Service
  - SSH Settings

**Description**  
Work with shells and commands on remote systems using the Secure Shell (ssh) protocol.

**Buttons:** ? < Back Next > Cancel **Finish**

## Define and Make a Remote System Connection

Mouse Right Click



Select



## Define and Make a Remote System Connection

Enter Password

System type: Linux

Host name: 192.168.1.186

Connection name: pi5Yocto

User ID:

Password (optional):

☒ Save user ID

☐ Save password

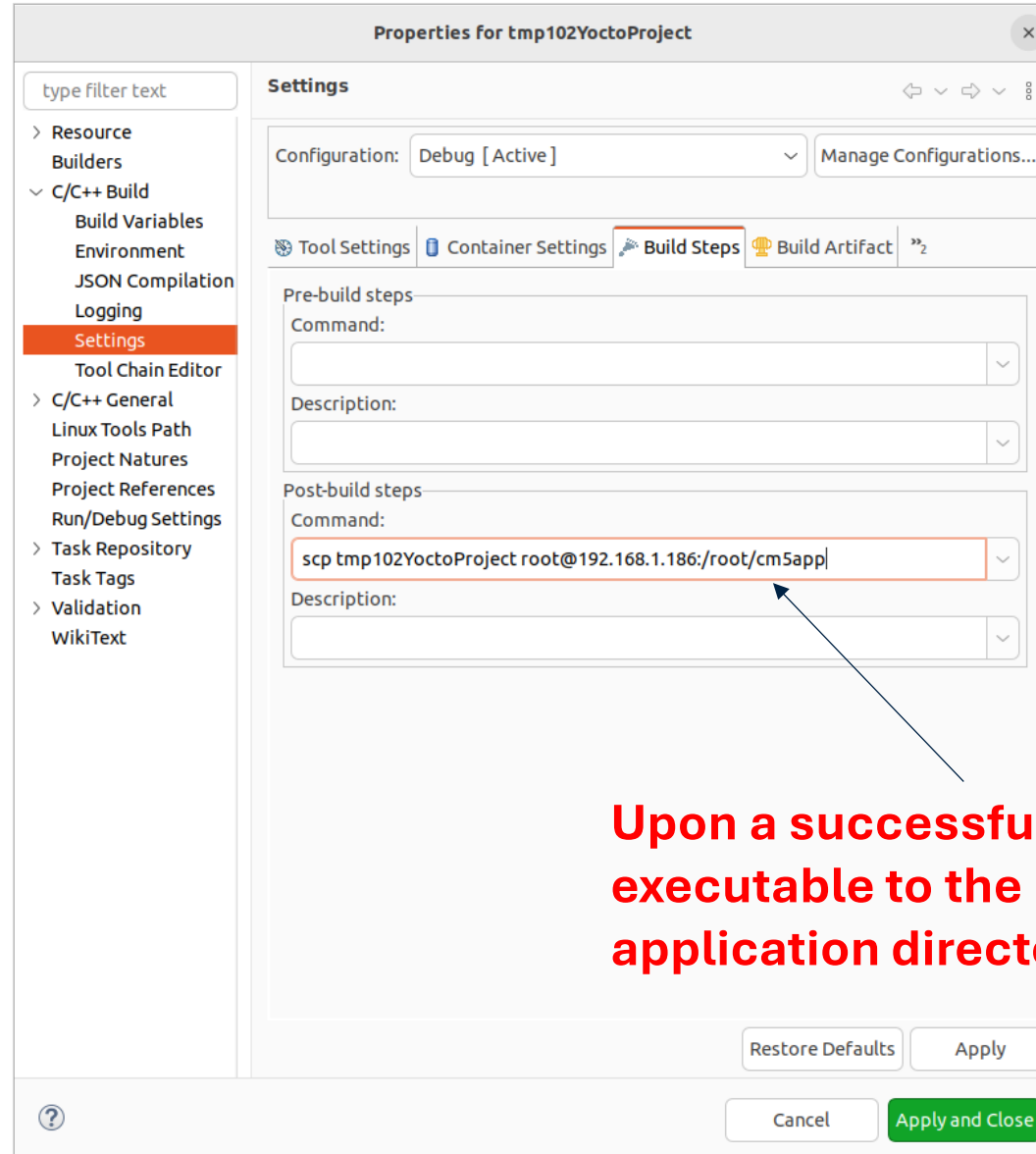
Cancel

OK

Enter User ID of root

Do not enter a password.

## Define and Make a Remote System Connection



Properties for tmp102YoctoProject

Settings

Configuration: Debug [ Active ] Manage Configurations...

Tool Settings Container Settings Build Steps Build Artifact

Pre-build steps

Command:

Description:

Post-build steps

Command:

scp tmp102YoctoProject root@192.168.1.186:/root/cm5app|

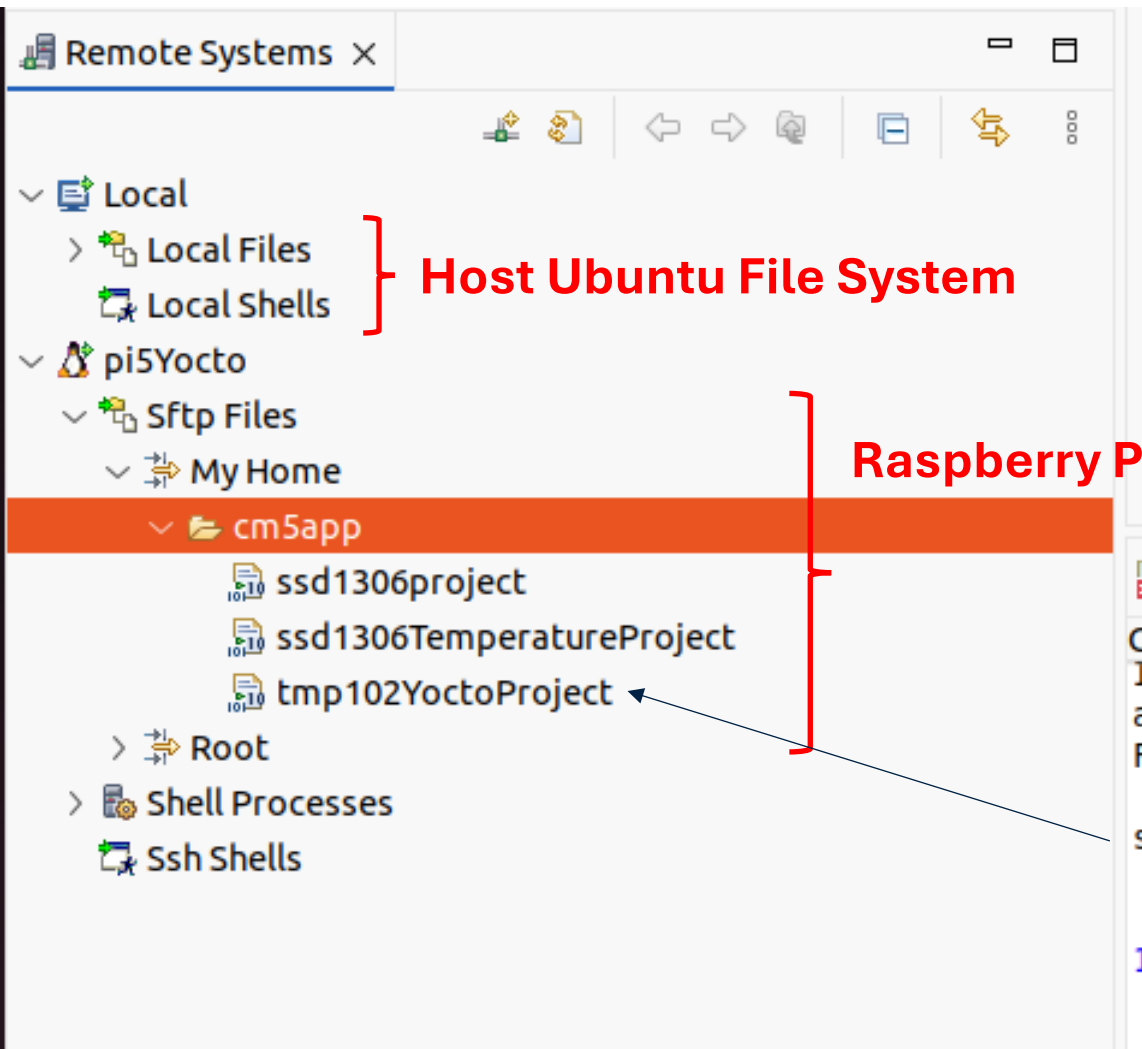
Description:

Restore Defaults Apply

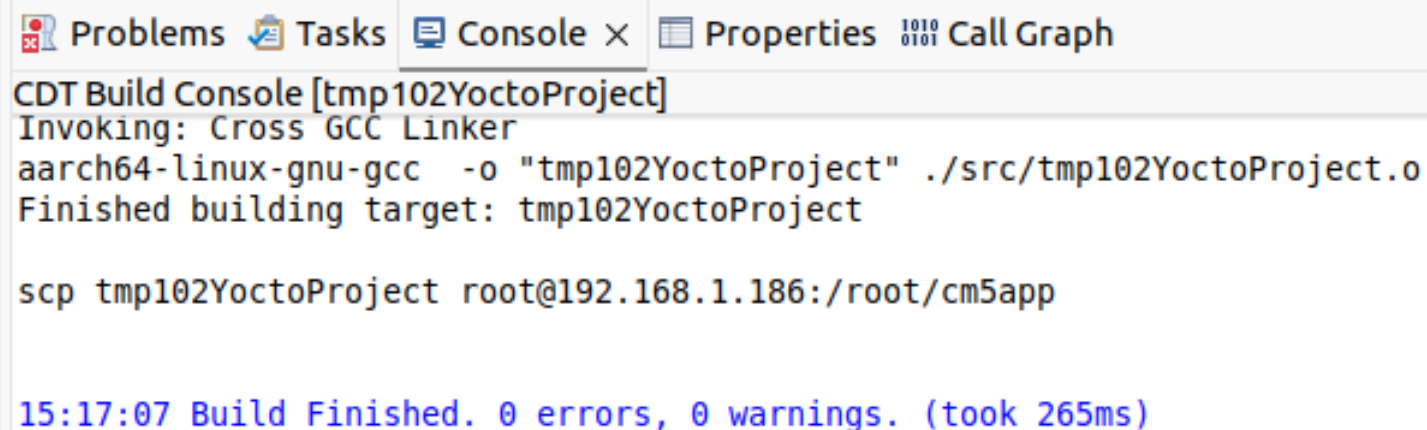
Cancel Apply and Close

**Upon a successful compilation, copy the executable to the user-defined Pi 5 application directory.**

## Define and Make a Remote System Connection



The **cm5app** directory can be created invoking the “***mkdir cm5app***” command from the **picocom** terminal window or by right clicking on **My Home** -> **New** -> **Folder**.



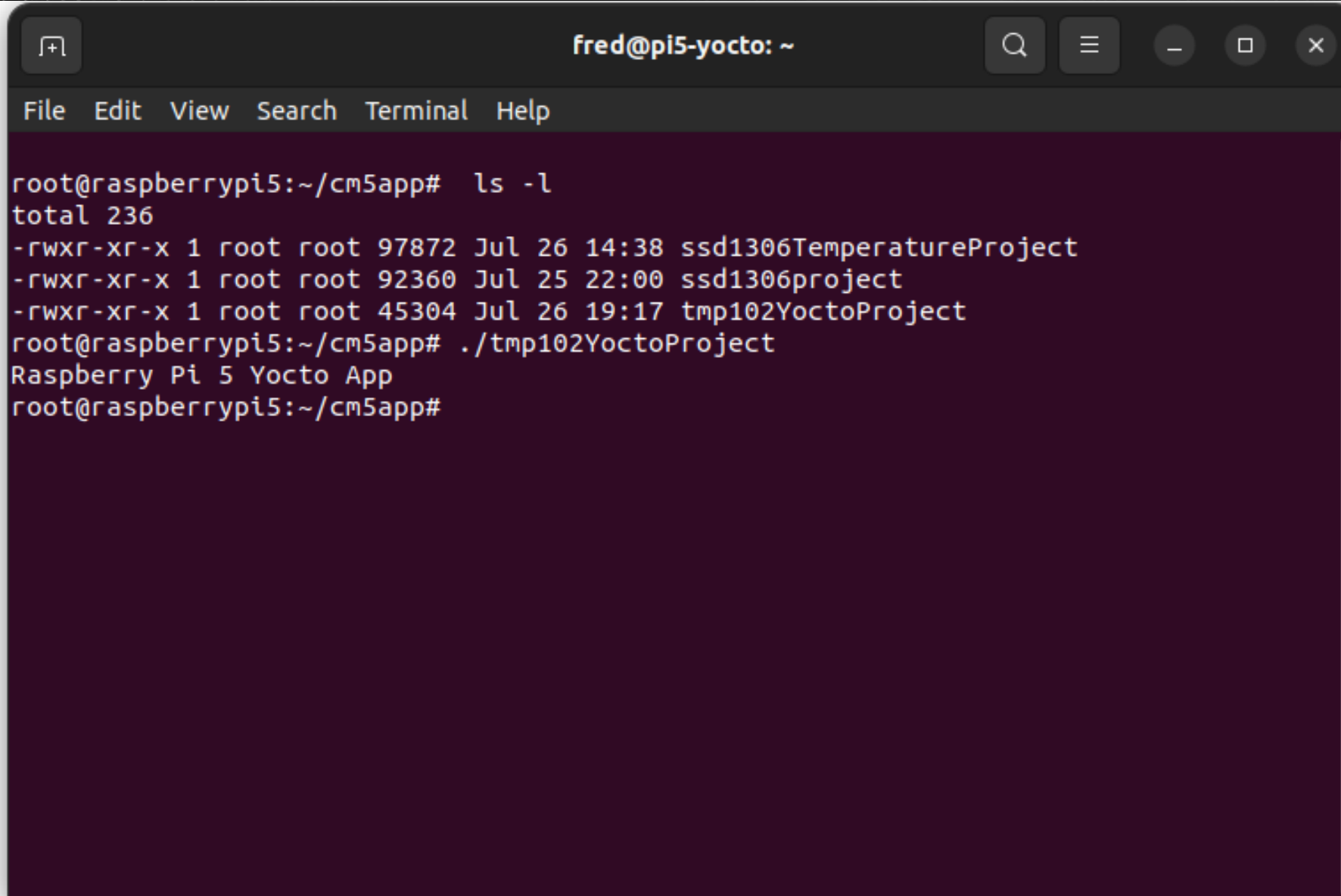
The screenshot shows the 'Console' view in Eclipse. The output is as follows:

```
CDT Build Console [tmp102YoctoProject]
Invoking: Cross GCC Linker
aarch64-linux-gnu-gcc -o "tmp102YoctoProject" ./src/tmp102YoctoProject.o
Finished building target: tmp102YoctoProject

scp tmp102YoctoProject root@192.168.1.186:/root/cm5app

15:17:07 Build Finished. 0 errors, 0 warnings. (took 265ms)
```

## Define and Make a Remote System Connection



```
fred@pi5-yocto: ~  
File Edit View Search Terminal Help  
root@raspberrypi5:~/cm5app# ls -l  
total 236  
-rwxr-xr-x 1 root root 97872 Jul 26 14:38 ssd1306TemperatureProject  
-rwxr-xr-x 1 root root 92360 Jul 25 22:00 ssd1306project  
-rwxr-xr-x 1 root root 45304 Jul 26 19:17 tmp102YoctoProject  
root@raspberrypi5:~/cm5app# ./tmp102YoctoProject  
Raspberry Pi 5 Yocto App  
root@raspberrypi5:~/cm5app#
```