



DesignNews

Raspberry Pi 4B Application Development Using the C Programming Language

DAY 1: Install the Raspberry Pi Cross Compilation Toolset

Sponsored by



Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Group Chat’ by maximizing the chat widget in your dock.
- Submit questions for the lecturer using the Q&A widget. They will follow-up after the lecture portion concludes.

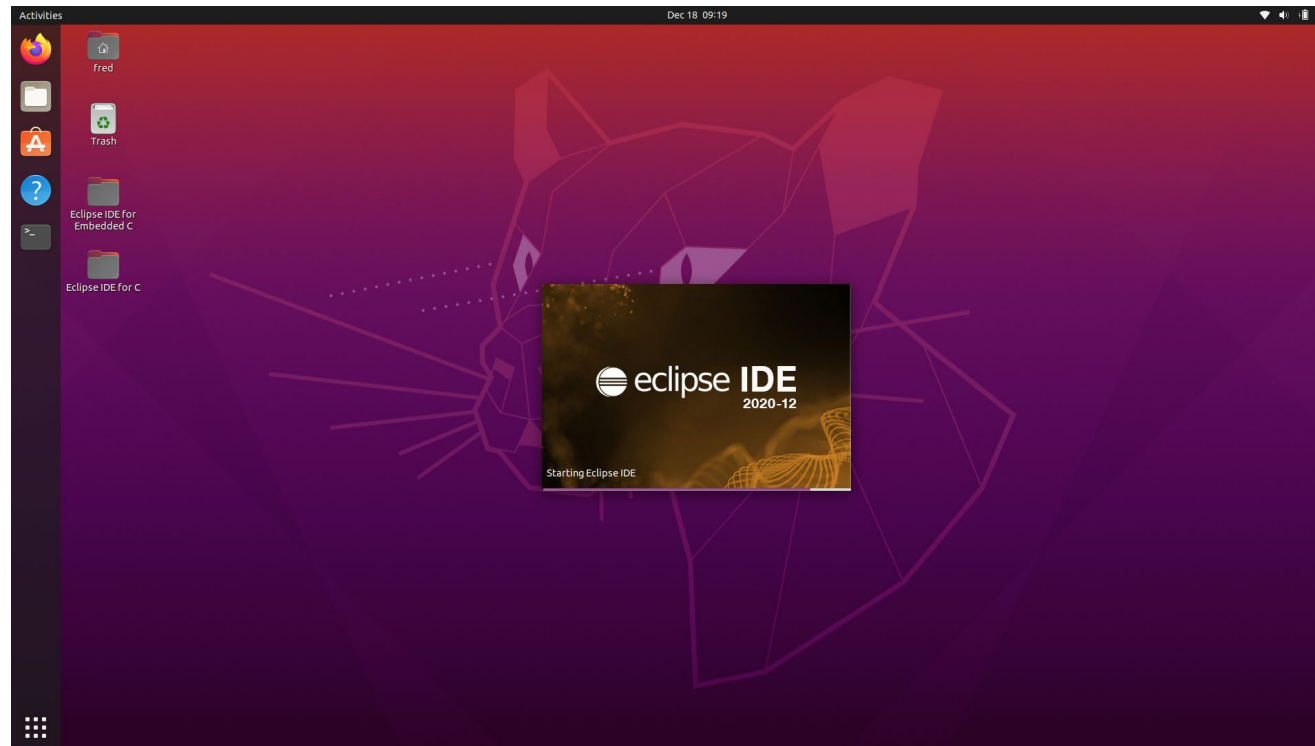


Fred Eady

Visit 'Lecturer Profile' in your console for more details.

AGENDA

- **Raspberry Pi 4B Setup**
- **Load the Cross Development Toolset**
- **Install Eclipse IDE for C/C++ Developers**
- **Using Eclipse**



Raspberry Pi 4B Setup

```
pi@cec: ~  
Raspberry Pi 4 Model B Rev 1.2  
Raspberry Pi Software Configuration Tool (raspi-config)  
1 System Options      Configure system settings  
2 Display Options     Configure display settings  
3 Interface Options   Configure connections to peripherals  
4 Performance Options Configure performance settings  
5 Localisation Options Configure language and regional settings  
6 Advanced Options    Configure advanced settings  
8 Update              Update this tool to the latest version  
9 About raspi-config  Information about this configuration tool  
  
<Select>           <Finish>
```

Load the Cross Development Toolset

```
fred@ubuntuLaptop: ~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
fred@ubuntuLaptop:~$ sudo apt-get update  
[sudo] password for fred:  
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease  
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]  
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [109 kB]  
Get:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]  
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [263 kB]  
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/main DEP-11 48x48 Icons [54.3 kB]  
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [205 kB]  
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [2,468 B]  
Get:9 http://us.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [1,768 B]  
Get:10 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [24.3 kB]  
Get:11 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [56.5 kB]  
Fetched 931 kB in 1s (699 kB/s)  
Reading package lists... Done  
fred@ubuntuLaptop:~$
```

Load the Cross Development Toolset

```
fred@ubuntuLaptop: ~  
fred@ubuntuLaptop:~$ sudo apt-get install build-essential  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following package was automatically installed and is no longer required:  
  libfprint-2-tod1  
Use 'sudo apt autoremove' to remove it.  
The following additional packages will be installed:  
  binutils binutils-common binutils-x86-64-linux-gnu dpkg-dev fakeroot g++ g++-9 gcc gcc-9  
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1  
  libbinutils libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd0 libctf0 libfakeroot libgcc-9-dev  
  libitm1 liblsan0 libquadmath0 libstdc++-9-dev libtsan0 libubsan1 linux-libc-dev make  
  manpages-dev  
Suggested packages:  
  binutils-doc debian-keyring g++-multilib g++-9-multilib gcc-9-doc gcc-multilib autoconf automake  
  libtool flex bison gcc-doc gcc-9-multilib gcc-9-locales glibc-doc libstdc++-9-doc make-doc  
The following NEW packages will be installed:  
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev fakeroot g++ g++-9  
  gcc gcc-9 libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5  
  libatomic1 libbinutils libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd0 libctf0 libfakeroot  
  libgcc-9-dev libitm1 liblsan0 libquadmath0 libstdc++-9-dev libtsan0 libubsan1 linux-libc-dev  
  make manpages-dev  
0 upgraded, 32 newly installed, 0 to remove and 1 not upgraded.  
Need to get 31.4 MB of archives.  
After this operation, 143 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

Load the Cross Development Toolset

```
fred@ubuntuLaptop: ~  
fred@ubuntuLaptop:~$ sudo apt-cache search gnuabihi  
binutils-arm-linux-gnueabihi - GNU binary utilities, for arm-linux-gnueabihi target  
binutils-arm-linux-gnueabihi-dbg - GNU binary utilities, for arm-linux-gnueabihi target (debug symbols)  
cpp-9-arm-linux-gnueabihi - GNU C preprocessor  
cpp-arm-linux-gnueabihi - GNU C preprocessor (cpp) for the armhf architecture  
g++-9-arm-linux-gnueabihi - GNU C++ compiler (cross compiler for armhf architecture)  
g++-arm-linux-gnueabihi - GNU C++ compiler for the armhf architecture  
gcc-9-arm-linux-gnueabihi - GNU C compiler (cross compiler for armhf architecture)  
gcc-9-arm-linux-gnueabihi-base - GCC, the GNU Compiler Collection (base package)  
gcc-arm-linux-gnueabihi - GNU C compiler for the armhf architecture  
cpp-10-arm-linux-gnueabihi - GNU C preprocessor  
cpp-8-arm-linux-gnueabihi - GNU C preprocessor  
g++-10-arm-linux-gnueabihi - GNU C++ compiler (cross compiler for armhf architecture)  
g++-10-multilib-arm-linux-gnueabihi - GNU C++ compiler (multilib support) (cross compiler for armhf architecture)  
g++-8-arm-linux-gnueabihi - GNU C++ compiler (cross compiler for armhf architecture)  
g++-8-multilib-arm-linux-gnueabihi - GNU C++ compiler (multilib support) (cross compiler for armhf architecture)  
g++-9-multilib-arm-linux-gnueabihi - GNU C++ compiler (multilib support) (cross compiler for armhf architecture)  
g++-multilib-arm-linux-gnueabihi - GNU C++ compiler for the armhf architecture  
gcc-10-arm-linux-gnueabihi - GNU C compiler (cross compiler for armhf architecture)  
gcc-10-arm-linux-gnueabihi-base - GCC, the GNU Compiler Collection (base package)  
gcc-10-multilib-arm-linux-gnueabihi - GNU C compiler (multilib support) (cross compiler for armhf architecture)  
gcc-10-plugin-dev-arm-linux-gnueabihi - Files for GNU GCC plugin development.  
gcc-8-arm-linux-gnueabihi - GNU C compiler (cross compiler for armhf architecture)  
gcc-8-arm-linux-gnueabihi-base - GCC, the GNU Compiler Collection (base package)  
gcc-8-multilib-arm-linux-gnueabihi - GNU C compiler (multilib support) (cross compiler for armhf architecture)  
gcc-8-plugin-dev-arm-linux-gnueabihi - Files for GNU GCC plugin development.  
gcc-9-multilib-arm-linux-gnueabihi - GNU C compiler (multilib support) (cross compiler for armhf architecture)  
gcc-9-plugin-dev-arm-linux-gnueabihi - Files for GNU GCC plugin development.  
gcc-multilib-arm-linux-gnueabihi - GNU C compiler for the armhf architecture
```


Load the Cross Development Toolset

```
fred@ubuntuLaptop: ~  
fred@ubuntuLaptop:~$ sudo apt-get install g++-arm-linux-gnueabi  
[sudo] password for fred:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following package was automatically installed and is no longer required:  
  libfprint-2-tod1  
Use 'sudo apt autoremove' to remove it.  
The following additional packages will be installed:  
  binutils-arm-linux-gnueabi cpp-9-arm-linux-gnueabi cpp-arm-linux-gnueabi g++-9-arm-linux-gnueabi  
  gcc-10-cross-base gcc-9-arm-linux-gnueabi gcc-9-arm-linux-gnueabi-base gcc-9-cross-base gcc-arm-linux-gnueabi  
  libasan5-armhf-cross libatomic1-armhf-cross libc6-armhf-cross libc6-dev-armhf-cross libgcc-9-dev-armhf-cross  
  libgcc-s1-armhf-cross libgomp1-armhf-cross libstdc++-9-dev-armhf-cross libstdc++6-armhf-cross libubsan1-armhf-cross  
  linux-libc-dev-armhf-cross  
Suggested packages:  
  binutils-doc gcc-9-locales cpp-doc g++-9-multilib-arm-linux-gnueabi gcc-9-doc gcc-9-multilib-arm-linux-gnueabi  
  autoconf automake libtool flex bison gdb-arm-linux-gnueabi gcc-doc  
The following NEW packages will be installed:  
  binutils-arm-linux-gnueabi cpp-9-arm-linux-gnueabi cpp-arm-linux-gnueabi g++-9-arm-linux-gnueabi  
  g++-arm-linux-gnueabi gcc-10-cross-base gcc-9-arm-linux-gnueabi gcc-9-arm-linux-gnueabi-base gcc-9-cross-base  
  gcc-arm-linux-gnueabi libasan5-armhf-cross libatomic1-armhf-cross libc6-armhf-cross libc6-dev-armhf-cross  
  libgcc-9-dev-armhf-cross libgcc-s1-armhf-cross libgomp1-armhf-cross libstdc++-9-dev-armhf-cross libstdc++6-armhf-cross  
  libubsan1-armhf-cross linux-libc-dev-armhf-cross  
0 upgraded, 21 newly installed, 0 to remove and 1 not upgraded.  
Need to get 32.5 MB of archives.  
After this operation, 117 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```



Load the Cross Development Toolset

```
fred@ubuntuLaptop: /usr/bin
fred@ubuntuLaptop:~$ cd /usr/bin
fred@ubuntuLaptop:/usr/bin$ ls arm*
arm2hpd1
arm-linux-gnueabi-hf-addr2line
arm-linux-gnueabi-hf-ar
arm-linux-gnueabi-hf-as
arm-linux-gnueabi-hf-c++filt
arm-linux-gnueabi-hf-cpp
arm-linux-gnueabi-hf-cpp-9
arm-linux-gnueabi-hf-dwp
arm-linux-gnueabi-hf-elfedit
arm-linux-gnueabi-hf-g++
arm-linux-gnueabi-hf-g++-9
arm-linux-gnueabi-hf-gcc
arm-linux-gnueabi-hf-gcc-9
fred@ubuntuLaptop:/usr/bin$
arm-linux-gnueabi-hf-gcc-ar
arm-linux-gnueabi-hf-gcc-ar-9
arm-linux-gnueabi-hf-gcc-nm
arm-linux-gnueabi-hf-gcc-nm-9
arm-linux-gnueabi-hf-gcc-ranlib
arm-linux-gnueabi-hf-gcc-ranlib-9
arm-linux-gnueabi-hf-gcov
arm-linux-gnueabi-hf-gcov-9
arm-linux-gnueabi-hf-gcov-dump
arm-linux-gnueabi-hf-gcov-dump-9
arm-linux-gnueabi-hf-gcov-tool
arm-linux-gnueabi-hf-gcov-tool-9
arm-linux-gnueabi-hf-gprof
arm-linux-gnueabi-hf-ld
arm-linux-gnueabi-hf-ld.bfd
arm-linux-gnueabi-hf-ld.gold
arm-linux-gnueabi-hf-nm
arm-linux-gnueabi-hf-objcopy
arm-linux-gnueabi-hf-objdump
arm-linux-gnueabi-hf-ranlib
arm-linux-gnueabi-hf-readelf
arm-linux-gnueabi-hf-size
arm-linux-gnueabi-hf-strings
arm-linux-gnueabi-hf-strip
```


Install Eclipse IDE for C/C++ Developers

eclipseinstaller by Oomph ★ DONATE

type filter text

-  **Eclipse IDE for Java Developers**
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration
-  **Eclipse IDE for Enterprise Java Developers**
Tools for developers working with Java and Web applications, including a Java IDE, tools for Web Services, JPA and Data Tools, JavaServer Pages and Faces, Mylyn,...
-  **Eclipse IDE for C/C++ Developers**
An IDE for C/C++ developers.
-  **Eclipse IDE for Embedded C/C++ Developers**
An IDE for Embedded C/C++ developers. It includes managed cross build plug-ins (Arm and RISC-V) and debug plug-ins (SEGGER J-Link, OpenOCD, and QEMU), plus a...
-  **Eclipse IDE for Web and JavaScript Developers**
The essential tools for any JavaScript developer, including JavaScript, TypeScript, HTML, CSS, XML, Yaml, Markdown... languages support; Kubernetes, Angular and...
-  **Eclipse IDE for PHP Developers**

eclipseinstaller by Oomph ★ DONATE

 **Eclipse IDE for C/C++ Developers** [details](#)
An IDE for C/C++ developers.

Java 11+ VM

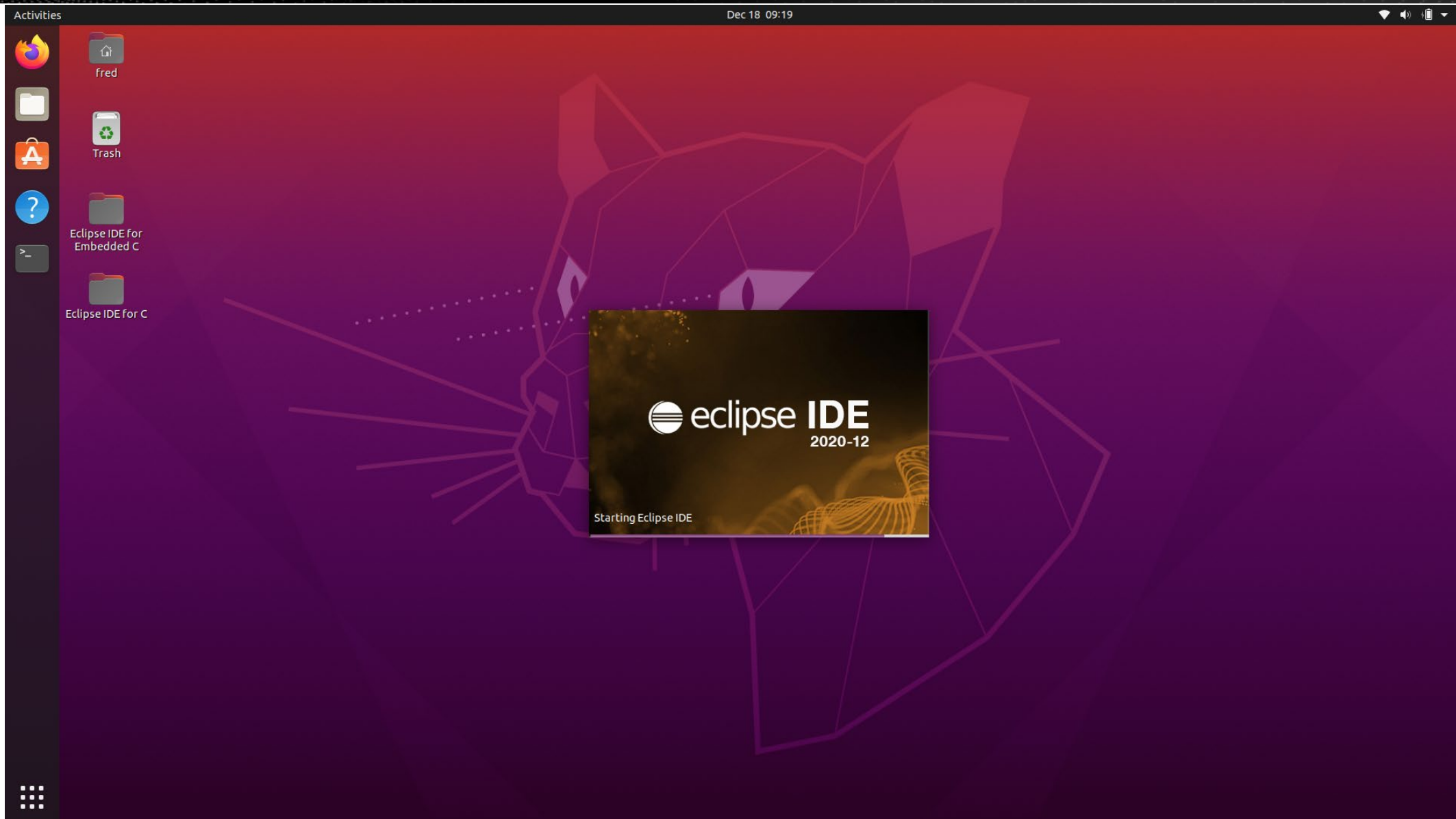
Installation Folder

- create start menu entry
- create desktop shortcut

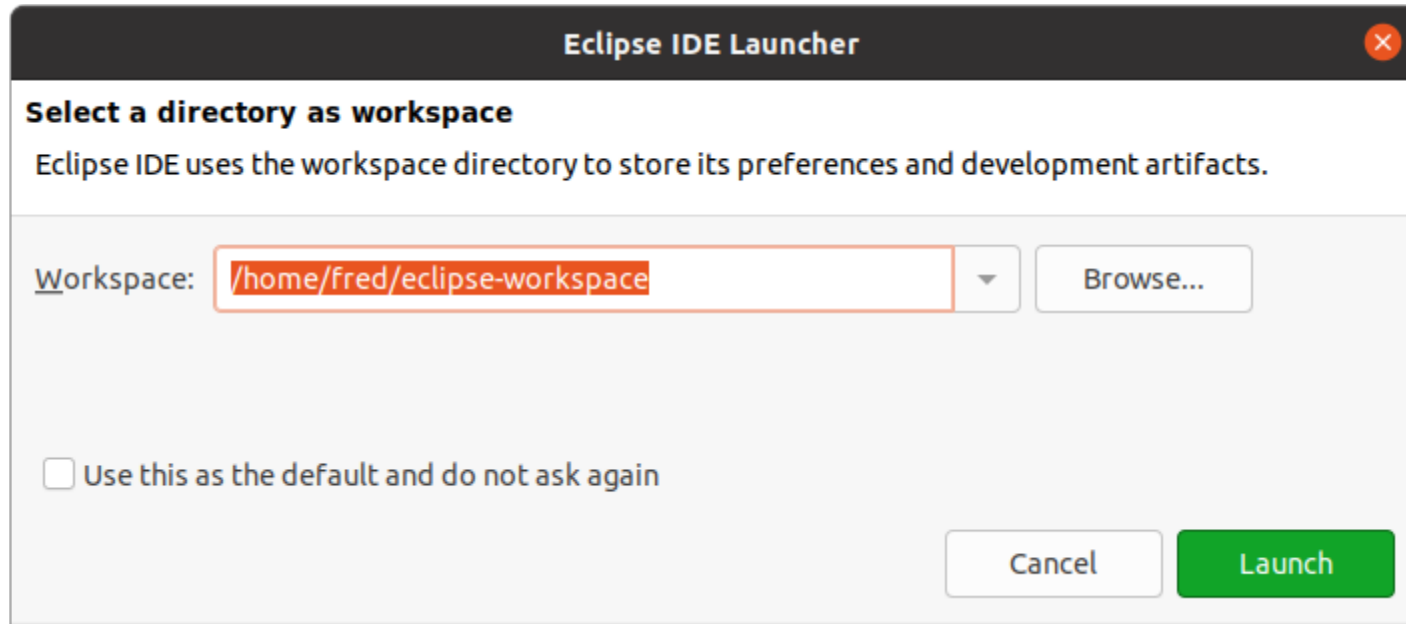
[↓ INSTALL](#)

[← BACK](#)

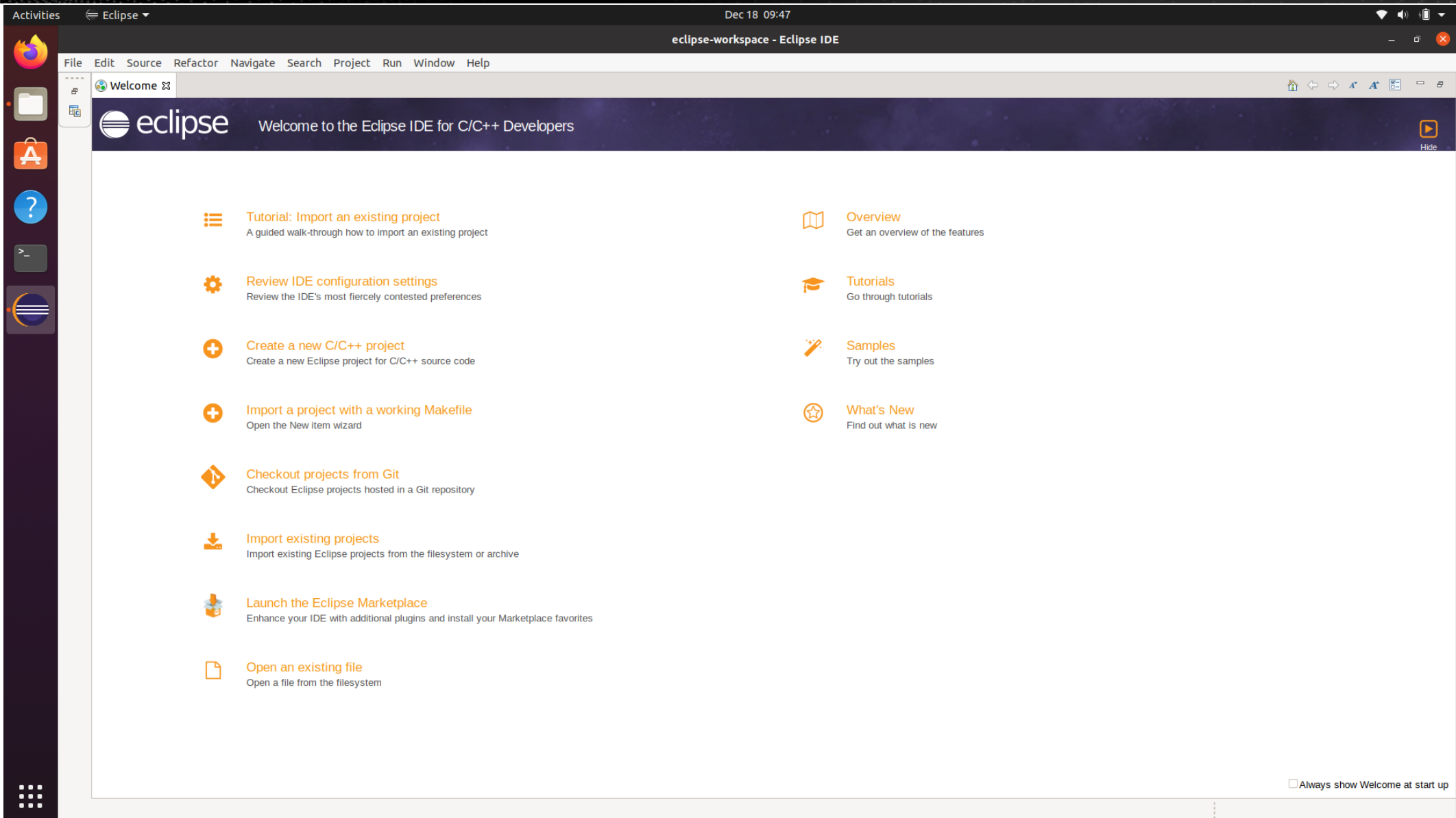
Install Eclipse IDE for C/C++ Developers



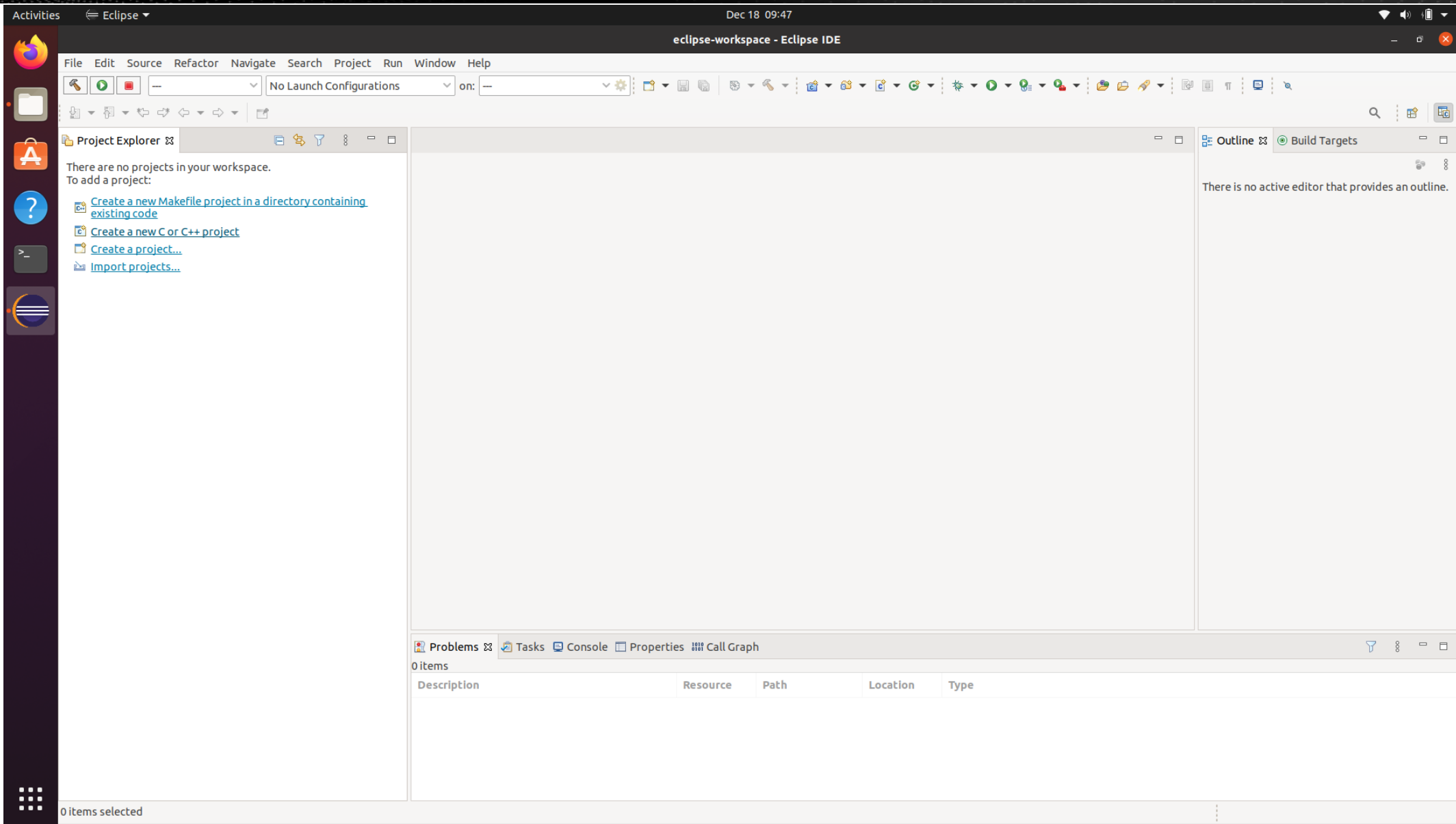
Using Eclipse – Launch the Eclipse IDE



Using Eclipse – Launch the Eclipse IDE



Using Eclipse – Create a C Project

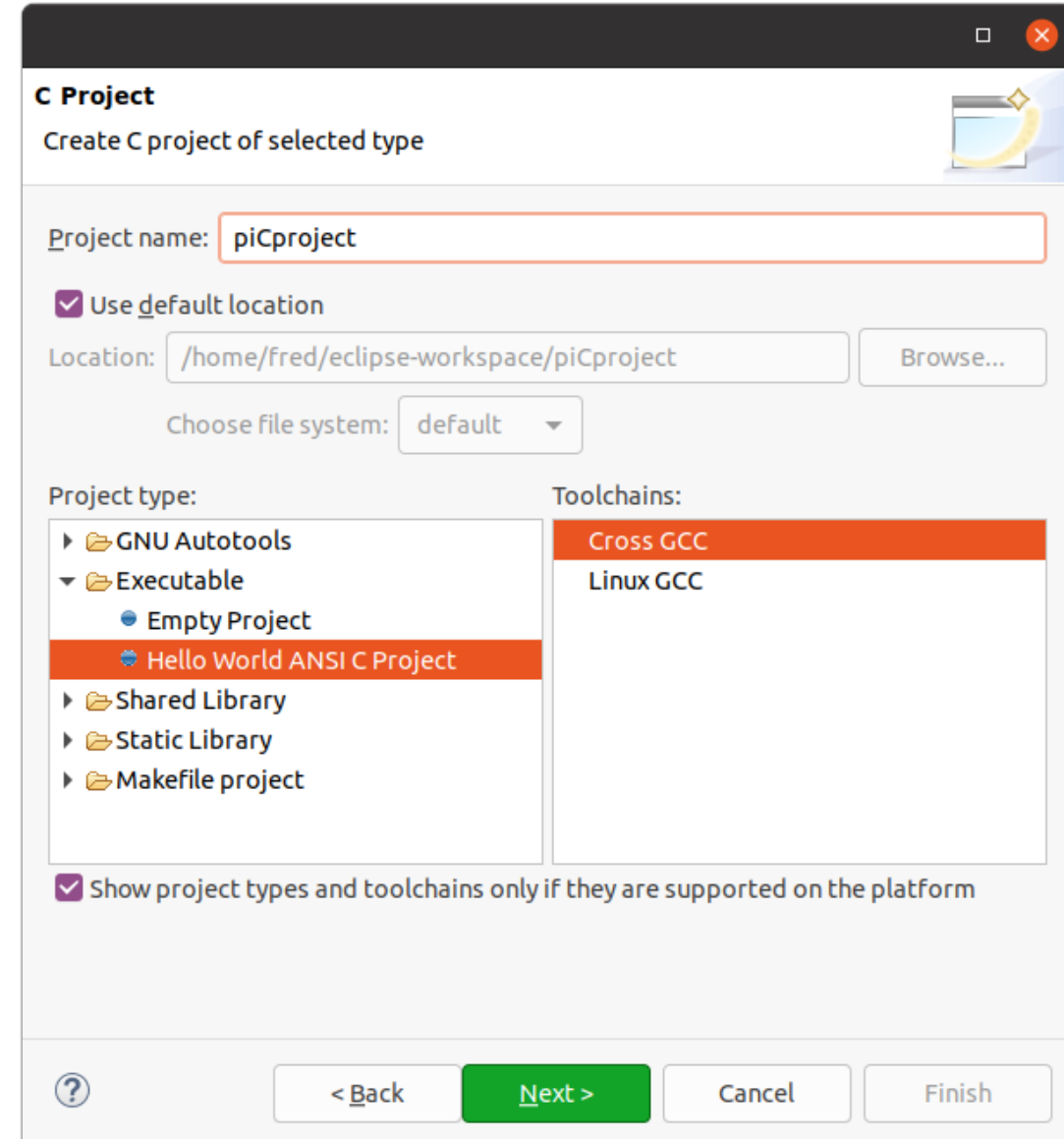
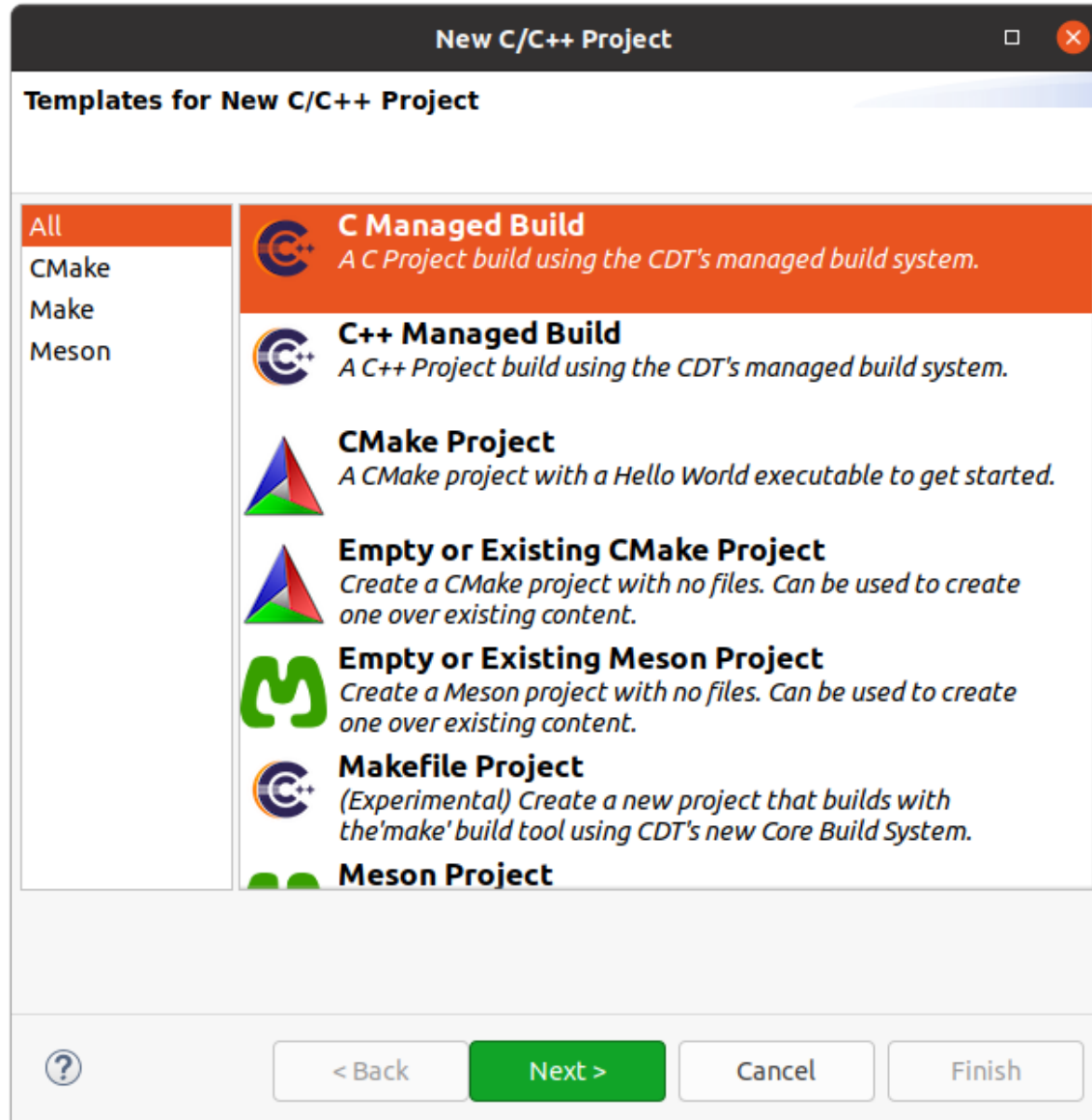


The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar contains various icons for file operations and development. The Project Explorer on the left displays the message: "There are no projects in your workspace. To add a project:" followed by four options: "Create a new Makefile project in a directory containing existing code", "Create a new C or C++ project", "Create a project...", and "Import projects...". The Outline view on the right shows "Build Targets" and the message: "There is no active editor that provides an outline." The bottom status bar shows "0 items" and a table with columns: Description, Resource, Path, Location, and Type.

Description	Resource	Path	Location	Type
0 items				

0 items selected

Using Eclipse – Create a C Project



Using Eclipse – Create a C Project


Basic Settings
Basic properties of a project

Author: FE

Copyright notice: Your copyright notice



Hello world greeting: !!!Hello CEC!!!

Source: src




Select Configurations
Select platforms and configurations you wish to deploy on

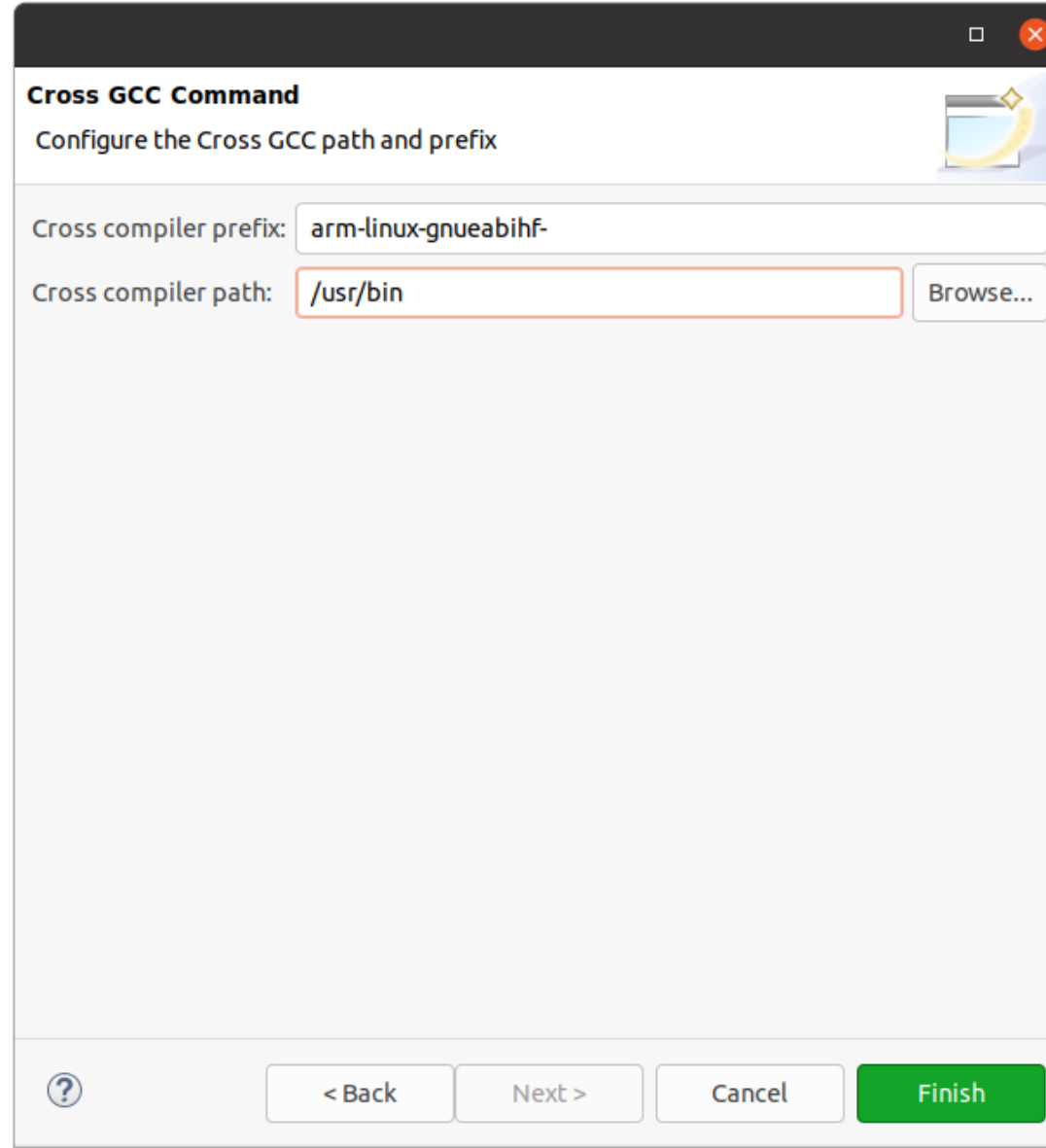
Project type: Executable
Toolchains: Cross GCC
Configurations:

-  Debug
-  Release

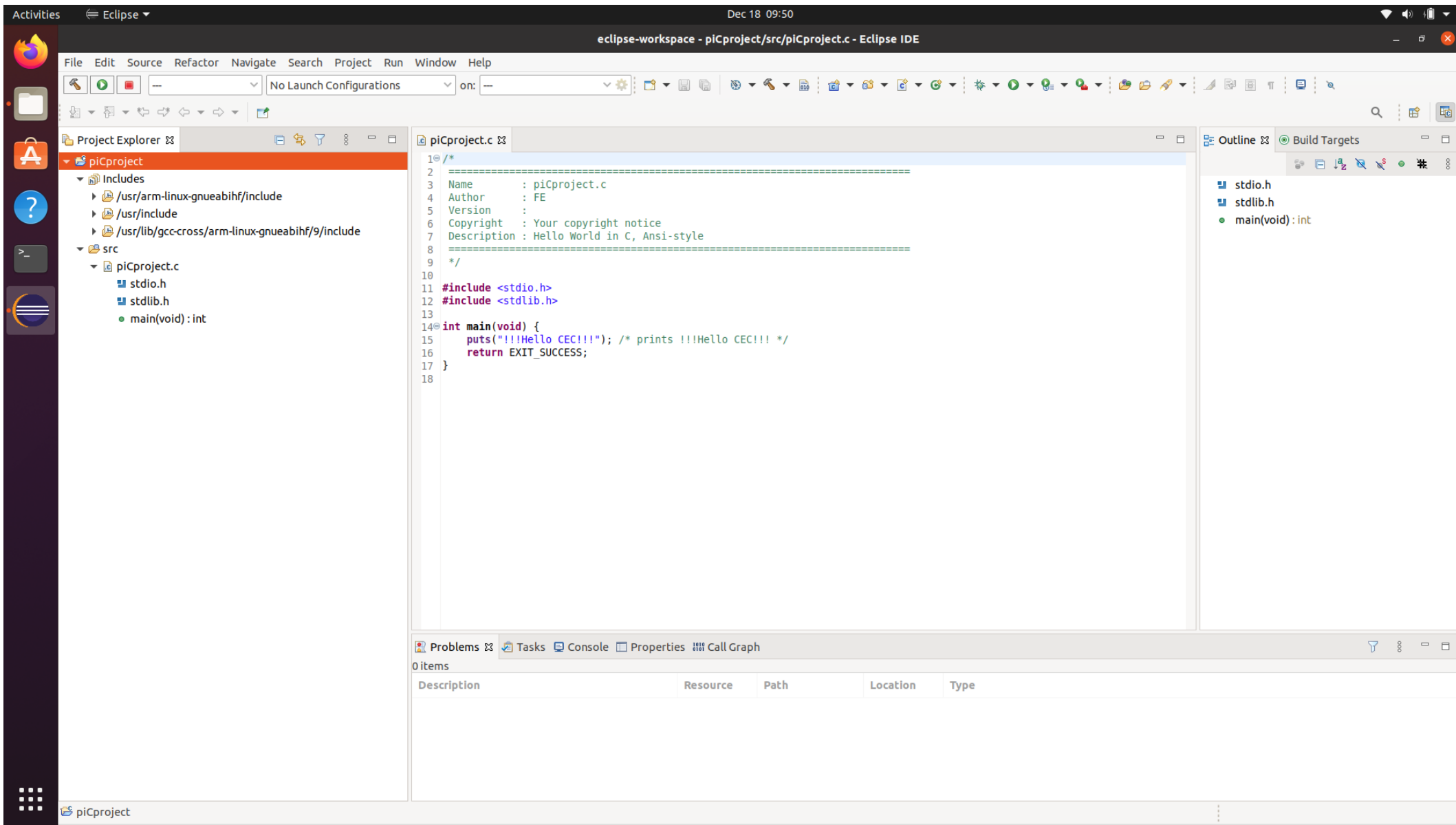
Use "Advanced settings" button to edit project's properties.
Additional configurations can be added after project creation.
Use "Manage configurations" buttons either on toolbar or on property pages.



Using Eclipse – Create a C Project



Using Eclipse – Create a C Project

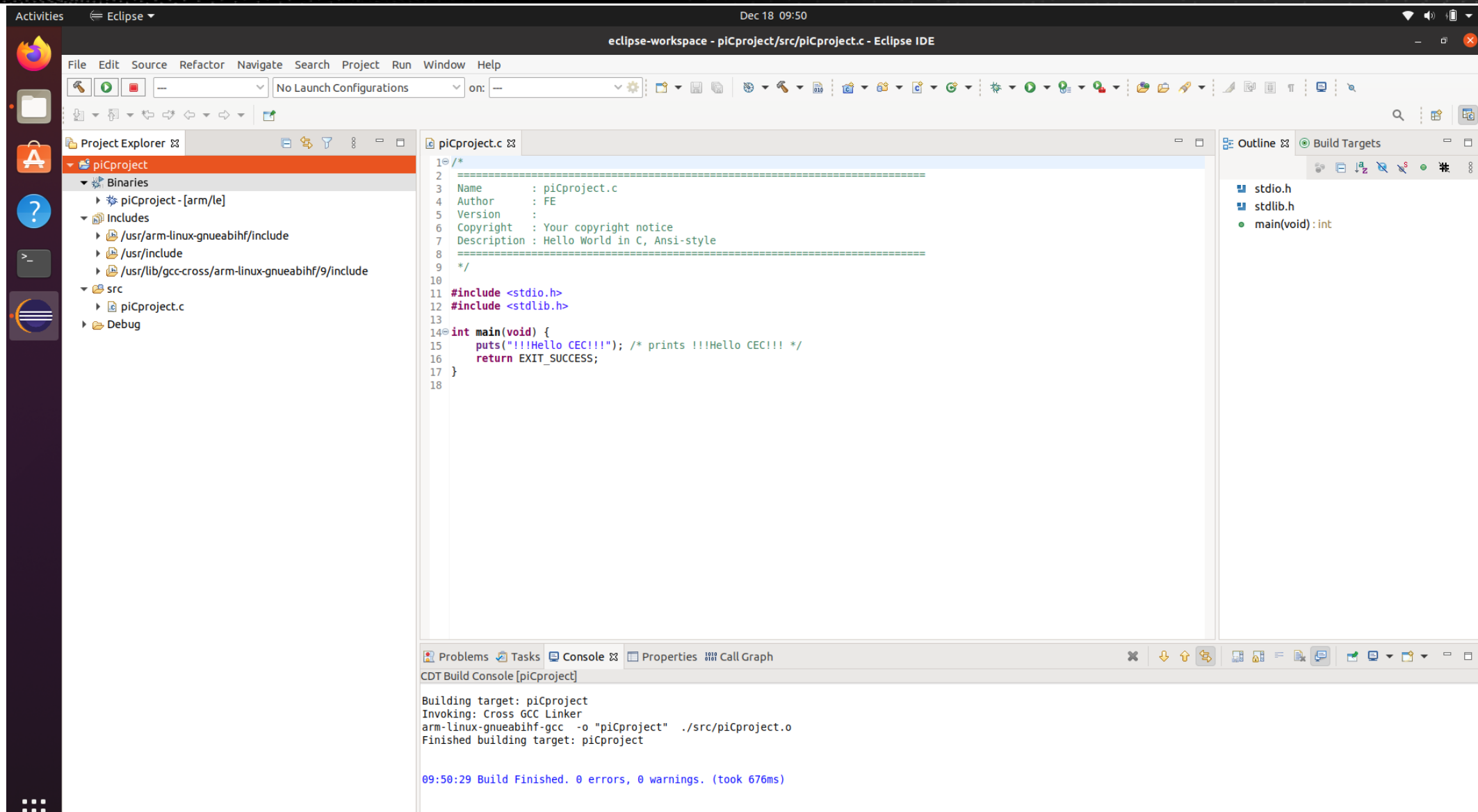


The screenshot shows the Eclipse IDE interface with the following components:

- Project Explorer:** Shows the project structure for 'piCProject', including 'Includes' (with paths like /usr/arm-linux-gnueabi/include) and 'src' (containing piCProject.c, stdio.h, and stdlib.h). A build target 'main(void): int' is listed.
- Editor:** Displays the source code for 'piCProject.c':

```
1 /*
2
3 Name      : piCProject.c
4 Author    : FE
5 Version   :
6 Copyright : Your copyright notice
7 Description: Hello World in C, Ansi-style
8
9 */
10
11 #include <stdio.h>
12 #include <stdlib.h>
13
14 int main(void) {
15     puts("!!!Hello CEC!!!"); /* prints !!!Hello CEC!!! */
16     return EXIT_SUCCESS;
17 }
18
```
- Outline:** Shows the project structure: stdio.h, stdlib.h, and main(void): int.
- Problems:** Shows 0 items.

Using Eclipse – Test Build



Activities ← Eclipse ▾ Dec 18 09:50
eclipse-workspace - piCProject/src/piCProject.c - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

No Launch Configurations ▾ on: ▾

Project Explorer

- piCProject
 - Binaries
 - piCProject - [arm/le]
 - Includes
 - /usr/arm-linux-gnueabi/hf/include
 - /usr/include
 - /usr/lib/gcc-cross/arm-linux-gnueabi/hf/9/include
 - src
 - piCProject.c
 - Debug

piCProject.c

```
1 /*
2 =====
3 Name      : piCProject.c
4 Author    : FE
5 Version   :
6 Copyright : Your copyright notice
7 Description: Hello World in C, Ansi-style
8 =====
9 */
10
11 #include <stdio.h>
12 #include <stdlib.h>
13
14 int main(void) {
15     puts("!!!Hello CEC!!!"); /* prints !!!Hello CEC!!! */
16     return EXIT_SUCCESS;
17 }
18
```

Outline

- Build Targets
 - stdio.h
 - stdlib.h
 - main(void): int

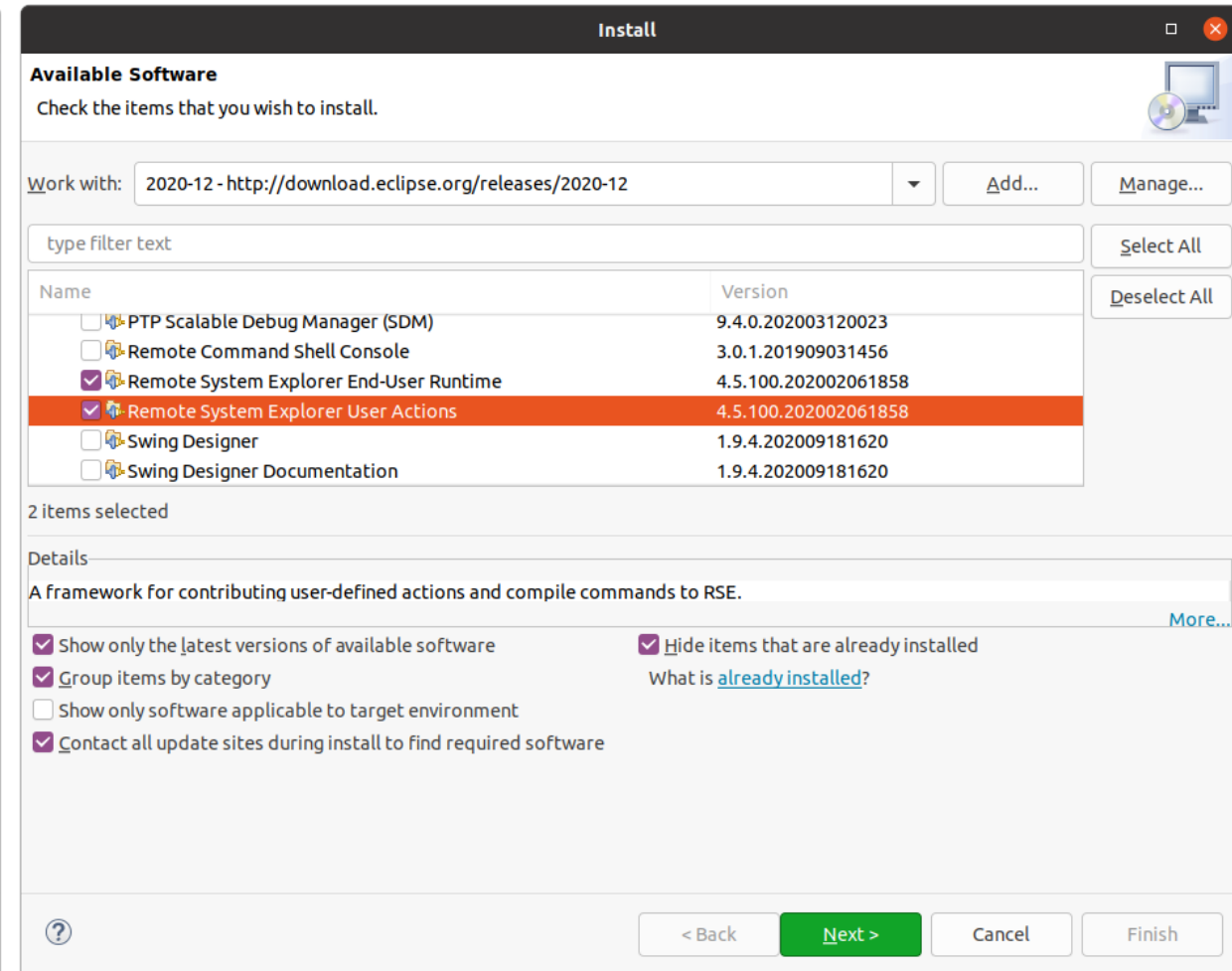
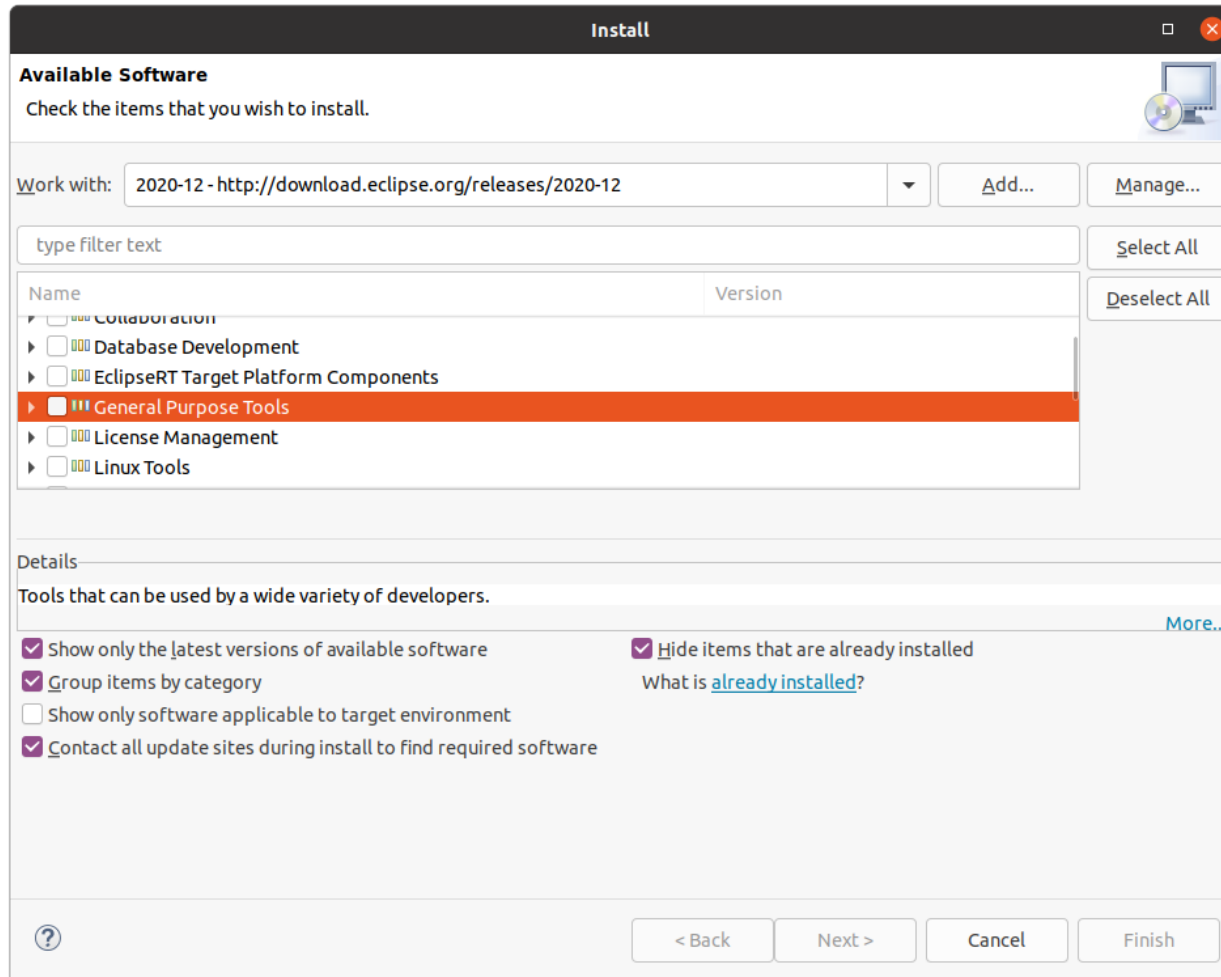
Problems Tasks Console Properties Call Graph

CDT Build Console [piCProject]

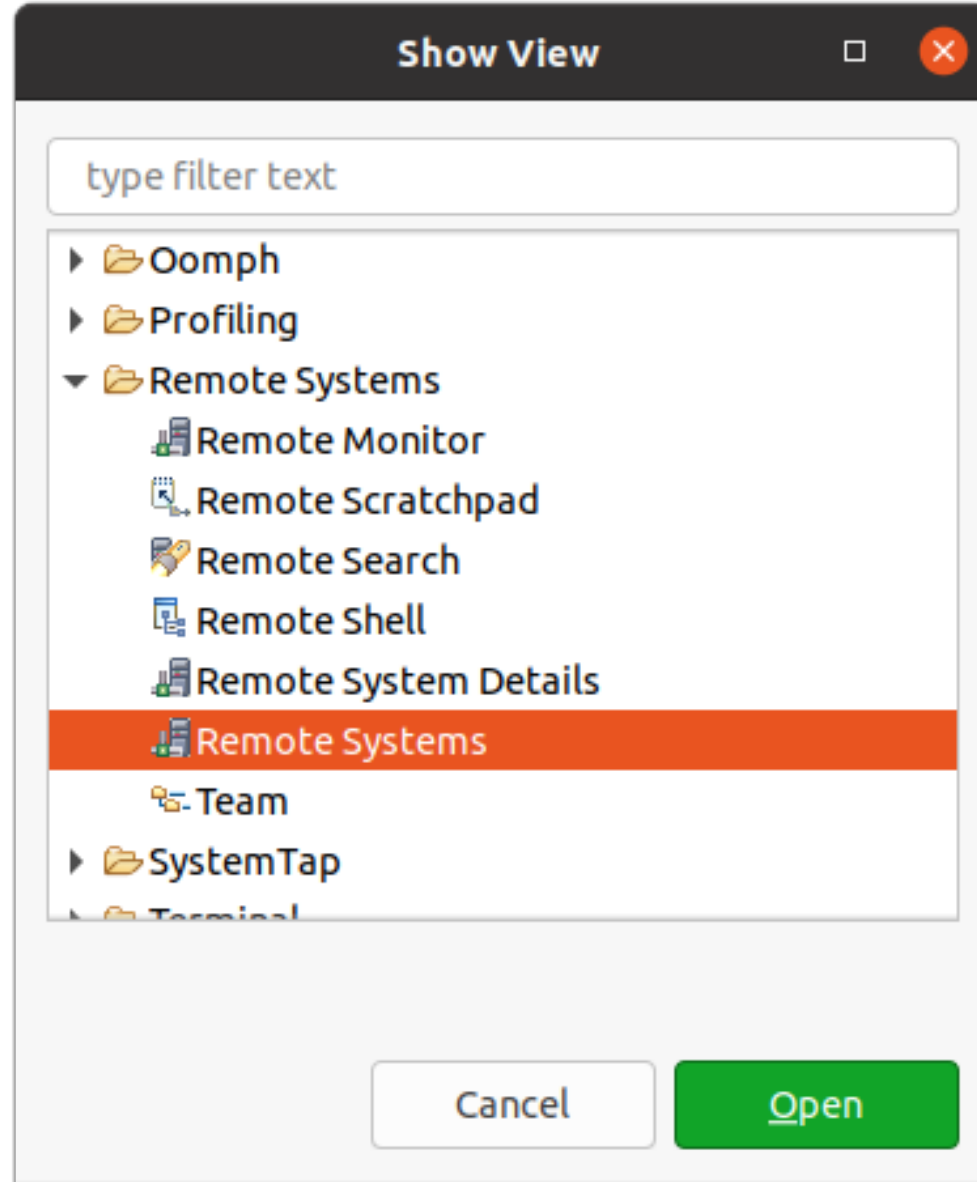
```
Building target: piCProject
Invoking: Cross GCC Linker
arm-linux-gnueabi-hf-gcc -o "piCProject" ./src/piCProject.o
Finished building target: piCProject

09:50:29 Build Finished. 0 errors, 0 warnings. (took 676ms)
```

Using Eclipse – Connect to the Pi



Using Eclipse – Connect to the Pi



Using Eclipse – Connect to the Pi

New Connection

Remote Linux System Connection
Define connection information


Parent profile: ubuntuLaptop

Host name: 192.168.1.20

Connection name: pi

Description:

Verify host name
[Configure proxy settings](#)




New Connection

Select Remote System Type
Any distribution of Linux

System type:
type filter text

- General
 - FTP Only
 - Linux**
 - Local
 - SSH Only
 - Telnet Only (Experimental)
- Unix Unix
- Windows



Using Eclipse – Connect to the Pi

New Connection

Files
Define subsystem information

Configuration

dstore.files
 ftp.files
 ssh.files

Properties

Property	Value

Available Services

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

Description

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

[?](#) < Back Next > Cancel Finish

New Connection

Processes
Define subsystem information

Configuration

dstore.processes
 processes.shell.linux

Properties

Property	Value

Available Services

- Shell Process Service

Description

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

[?](#) < Back Next > Cancel Finish

New Connection

Shells
Define subsystem information

Configuration

dstore.shells
 ssh.shells

Properties

Property	Value

Available Services

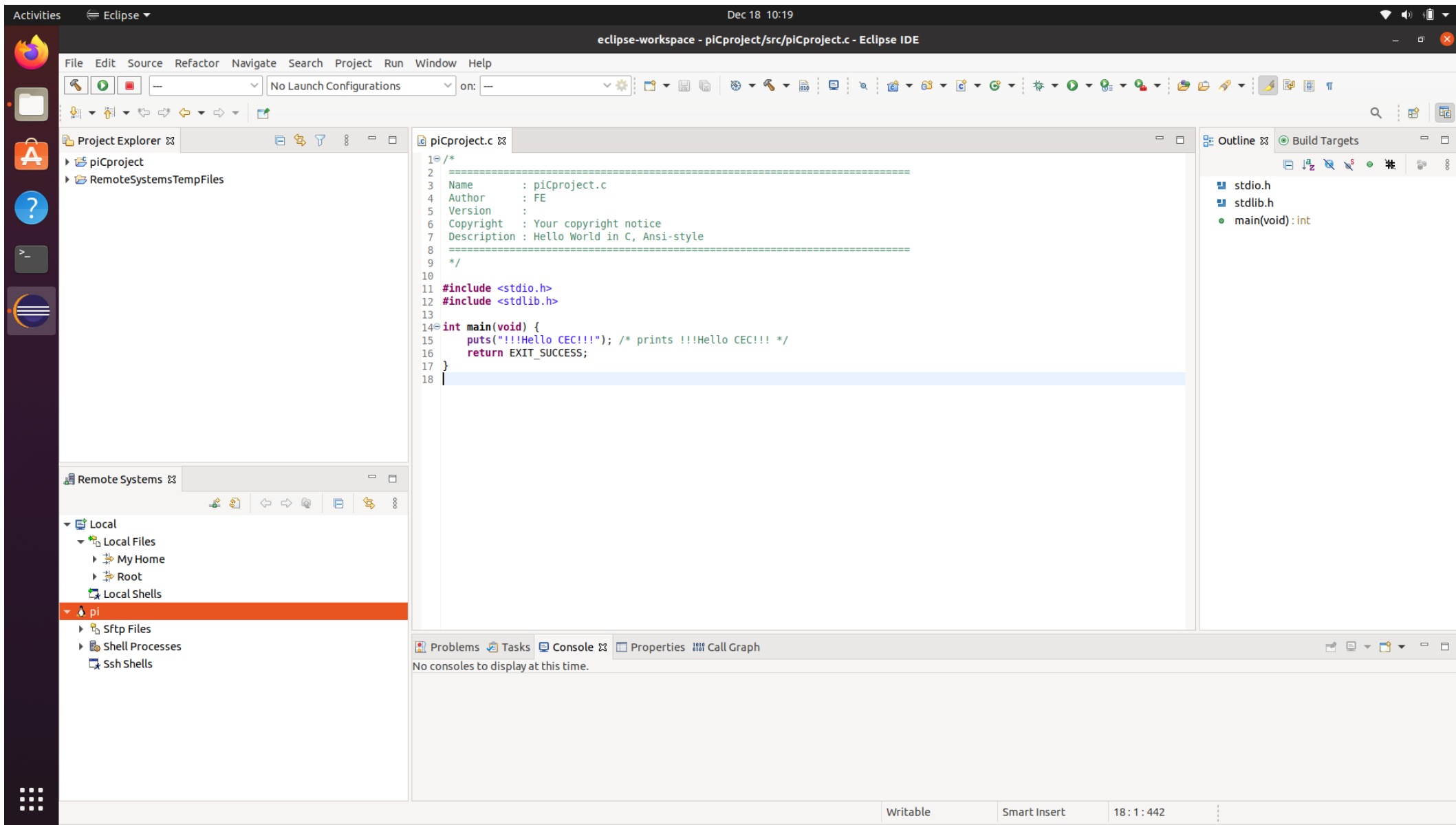
- Generic shell service
- SSH Connector Service
 - SSH Settings

Description

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

[?](#) < Back Next > Cancel Finish

Using Eclipse – Connect to the Pi



The screenshot displays the Eclipse IDE interface. The main editor window shows the source code for `piCproject.c`. The code includes standard headers and a `main` function that prints "!!!Hello CEC!!!". The Project Explorer on the left shows the project structure, and the Remote Systems view at the bottom left shows a connection to a Raspberry Pi. The Outline view on the right shows the project's structure, including `stdio.h`, `stdlib.h`, and the `main` function.

```
1 /*
2 =====
3 Name      : piCproject.c
4 Author    : FE
5 Version   :
6 Copyright : Your copyright notice
7 Description: Hello World in C, Ansi-style
8 =====
9 */
10
11 #include <stdio.h>
12 #include <stdlib.h>
13
14 int main(void) {
15     puts("!!!Hello CEC!!!"); /* prints !!!Hello CEC!!! */
16     return EXIT_SUCCESS;
17 }
18
```

Using Eclipse – Connect to the Pi

Enter Password ✕

System type: Linux

Host name: 192.168.1.20

Connection name: pi

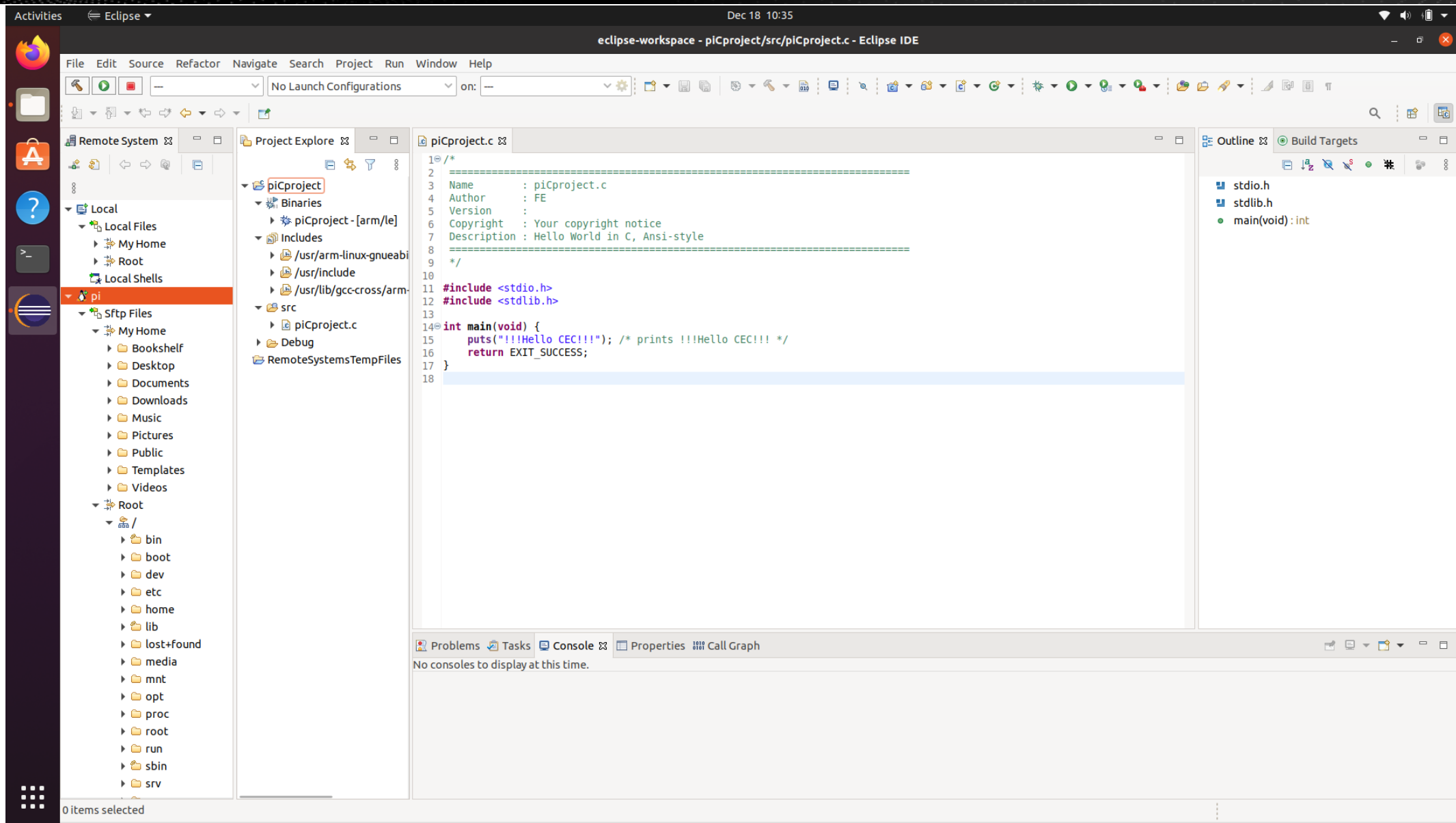
User ID:

Password (optional):

Save user ID

Save password

Using Eclipse – Connect to the Pi



The screenshot shows the Eclipse IDE interface with the following components:

- Remote System:** A sidebar on the left showing a tree view of the Raspberry Pi's file system. The 'pi' system is selected, and the 'src' directory is expanded to show 'piCproject.c'.
- Project Explorer:** A sidebar showing the project structure for 'piCproject', including 'Binaries', 'Includes', and 'src'.
- Editor:** The main workspace displays the source code for 'piCproject.c'. The code includes headers for `stdio.h` and `stdlib.h`, and contains a `main` function that prints "!!!Hello CEC!!!".
- Outline:** A sidebar on the right showing the project's build targets, including `stdio.h`, `stdlib.h`, and `main(void):int`.
- Console:** The bottom status bar indicates "No consoles to display at this time."

```
1 /*
2 =====
3 Name      : piCproject.c
4 Author    : FE
5 Version   :
6 Copyright : Your copyright notice
7 Description: Hello World in C, Ansi-style
8 =====
9 */
10
11 #include <stdio.h>
12 #include <stdlib.h>
13
14 int main(void) {
15     puts("!!!Hello CEC!!!"); /* prints !!!Hello CEC!!! */
16     return EXIT_SUCCESS;
17 }
18
```

Thank you for attending

Please consider the resources below:

- <https://www.raspberrypi.org>
- <https://ubuntu.com>
- <https://www.eclipse.org>



Thank You

Sponsored by

