NM180100 Software Development Toolchain Setup

1. Contents

2. Toolchain Installation Steps

To develop software for the NM180100, you will need to install and setup a toolchain that can build, program, and debug code for the Ambiq Micro Apollo 3 processor in the NM180100. This guide will help you setup the development environment on your system.

2.1. Linux

This section assumes that you are running Ubuntu or similar Debian environment.

2.1.1. Pre-requisite

Check if the default development tools are installed. If not, issue the following commands in a terminal:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt install build-essential make openocd gdb-multiarch
sudo apt install python3
```
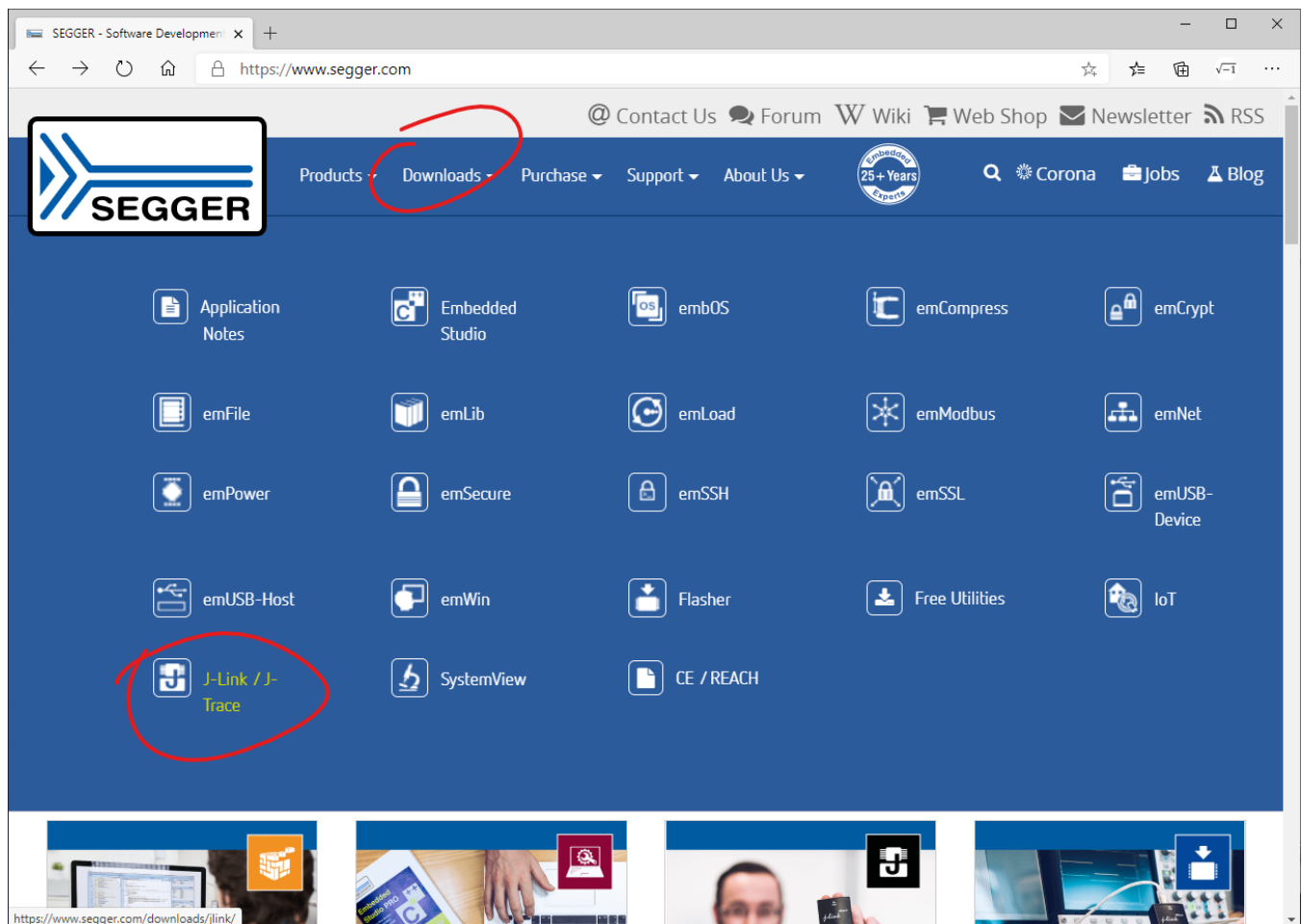
2.1.2. Java Development Kit

This is optional and is only required if you plan on developing with Eclipse.

```
sudo apt install default-jdk
```

2.1.3. SEGGER J-Link Installation

- Visit https://www.segger.com
- Select J-Link / J-Trace under Downloads

- Scroll to J-Link Software and Documentation and expand the section by clicking on the + sign.

- Download the appropriate file for your OS. If you are running Ubuntu 64-bit for example, download the Linux, DEB installer, 64-bit.

- In a terminal, navigate to the directory where you have downloaded the installer and type the following

```
sudo apt install installer_file_name.deb
```

In Ubuntu, this will install JLink under /opt/SEGGER/JLink. You can optionally add this directory to your search path so that J-Link is accessible anywhere.

2.1.4. GNU ARM Embedded Toolchain Installation

- Visit https://developer.arm.com/open-source/gnu-toolchain/gnu-rm/downloads
- Download the corresponding file for your OS.

- In a terminal, extract the file to a location of your choice. For example, if the file is downloaded to your Downloads directory under your home folder and you want to install the toolchain in /opt, then type the following:

```
cd /opt
```

```
sudo bunzip2 ~/Downloads/gcc-arm-none-eabi-9-2020-q2-update-x86_64-
linux.tar.bz2
```

2.1.5. Eclipse Installation

- To install Eclipse, open a terminal and type the following

```
sudo snap install eclipse --classic
```

- If you already have Eclipse installed and need to update to the latest version, type the following. Otherwise skip to the next step.

```
sudo snap refresh eclipse
```

- Once Eclipse is installed, start the program. If this is the first time you are running Eclipse, it will ask you for a location for the workspace. Mark the checkbox and select "Launch"
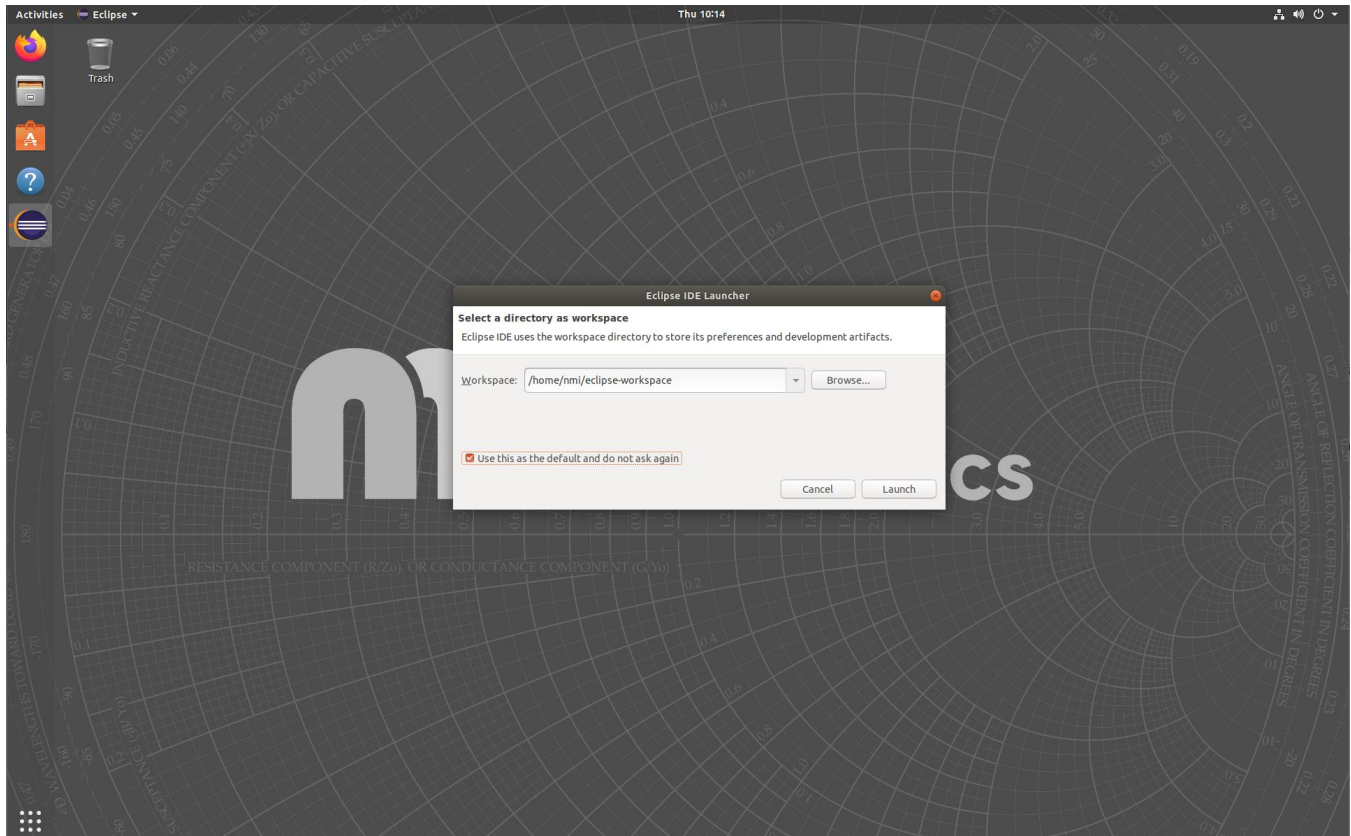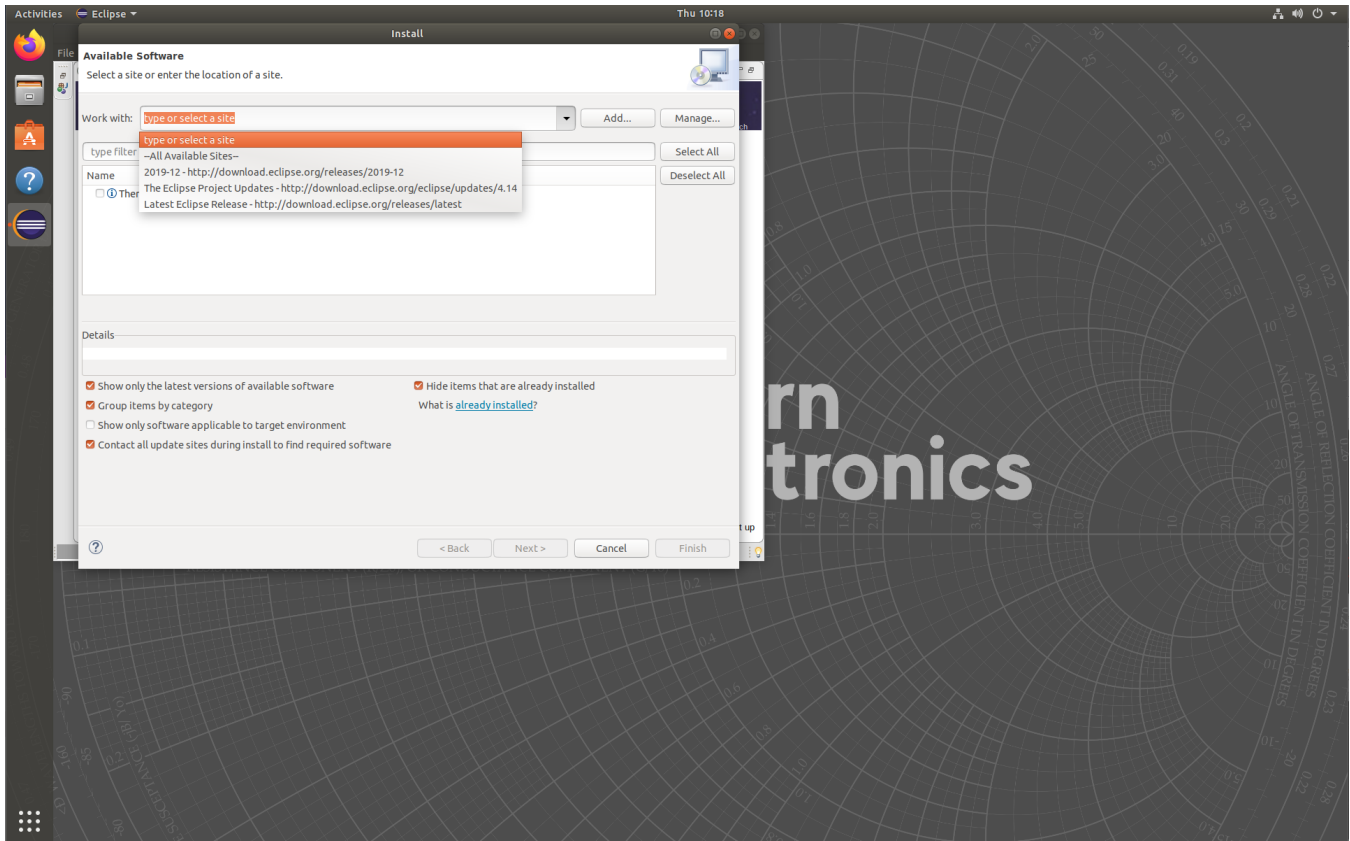
- We will now install the C/C++ Development Tools necessary for firmware development. With Eclipse opened, click "Help" in the menu bar and then select "Install New Software…"
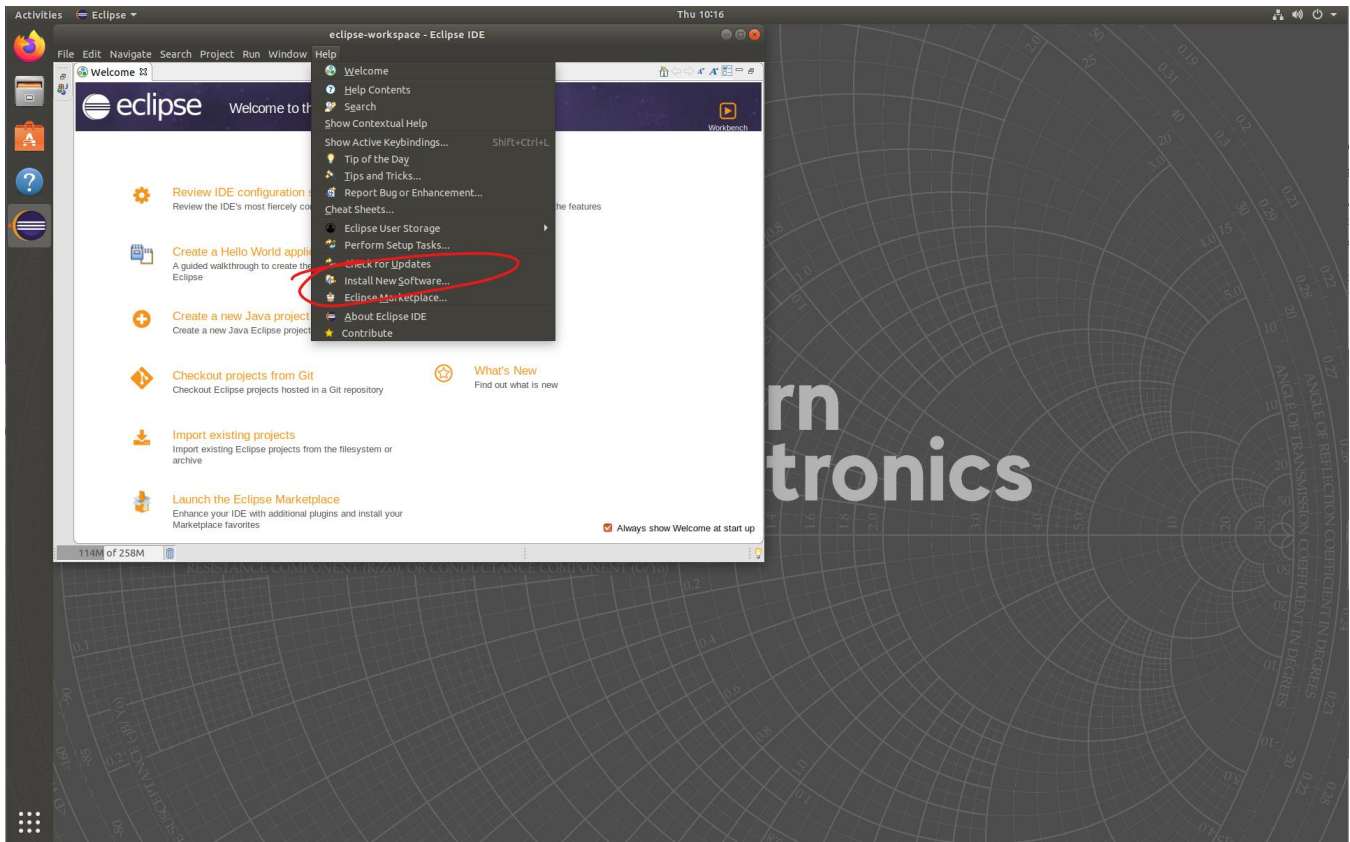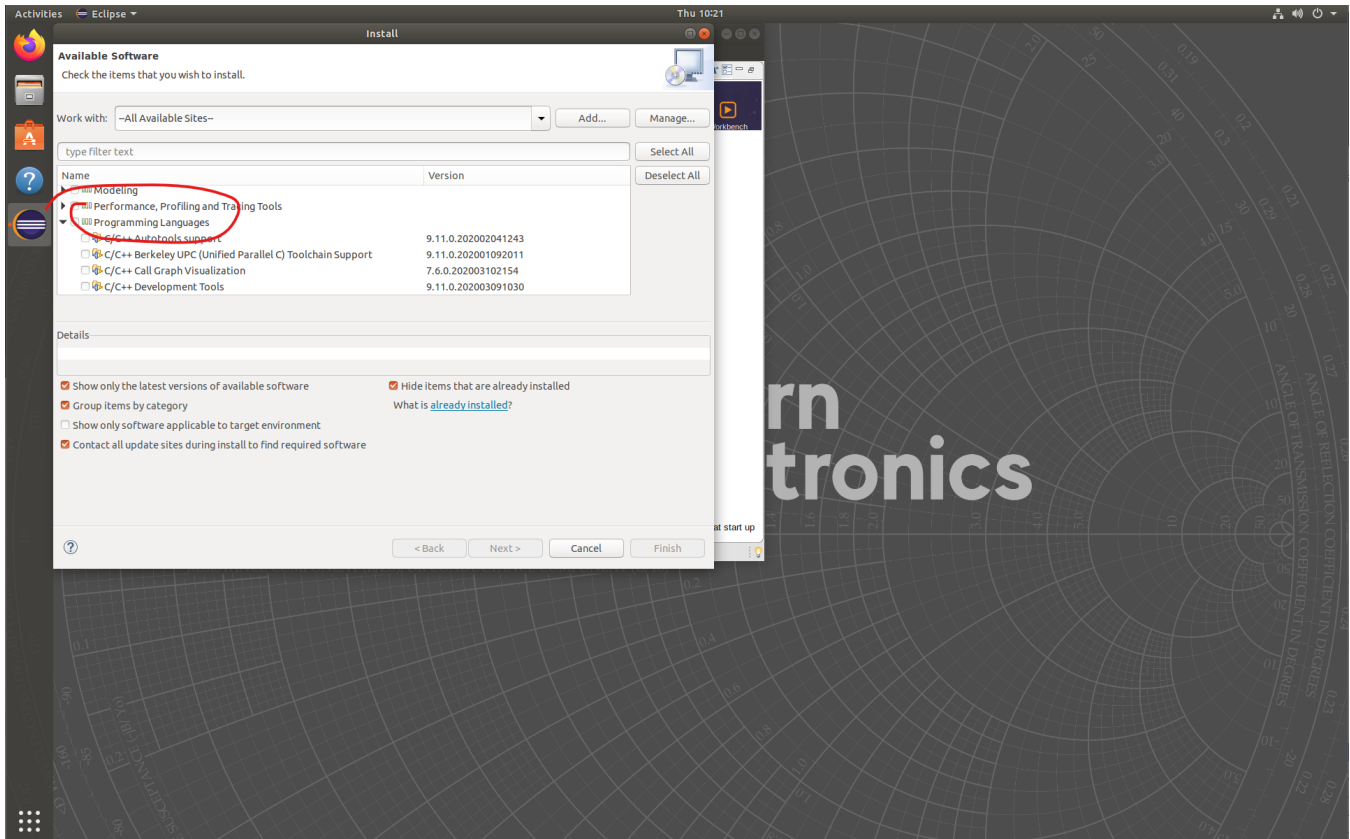
- This will open a new window. Click the "Work with" drop down box and select "—All Available Sites—"

- Once the loading is complete, scroll to the line called "Programming Languages" and expand the item.


- Mark the checkbox for "C/C++ Development Tools"
- Click Next to proceed and follow the on-screen instructions to complete.
- Once the installation is completed, Eclipse will prompt for a restart. Click "Restart Now" to continue.
- We will now install the GNU MCU Eclipse plugin for cross platform embedded ARM development. With Eclipse open, click "Help" in the menu bar and then select "Eclipse Marketplace…" A new window will open.
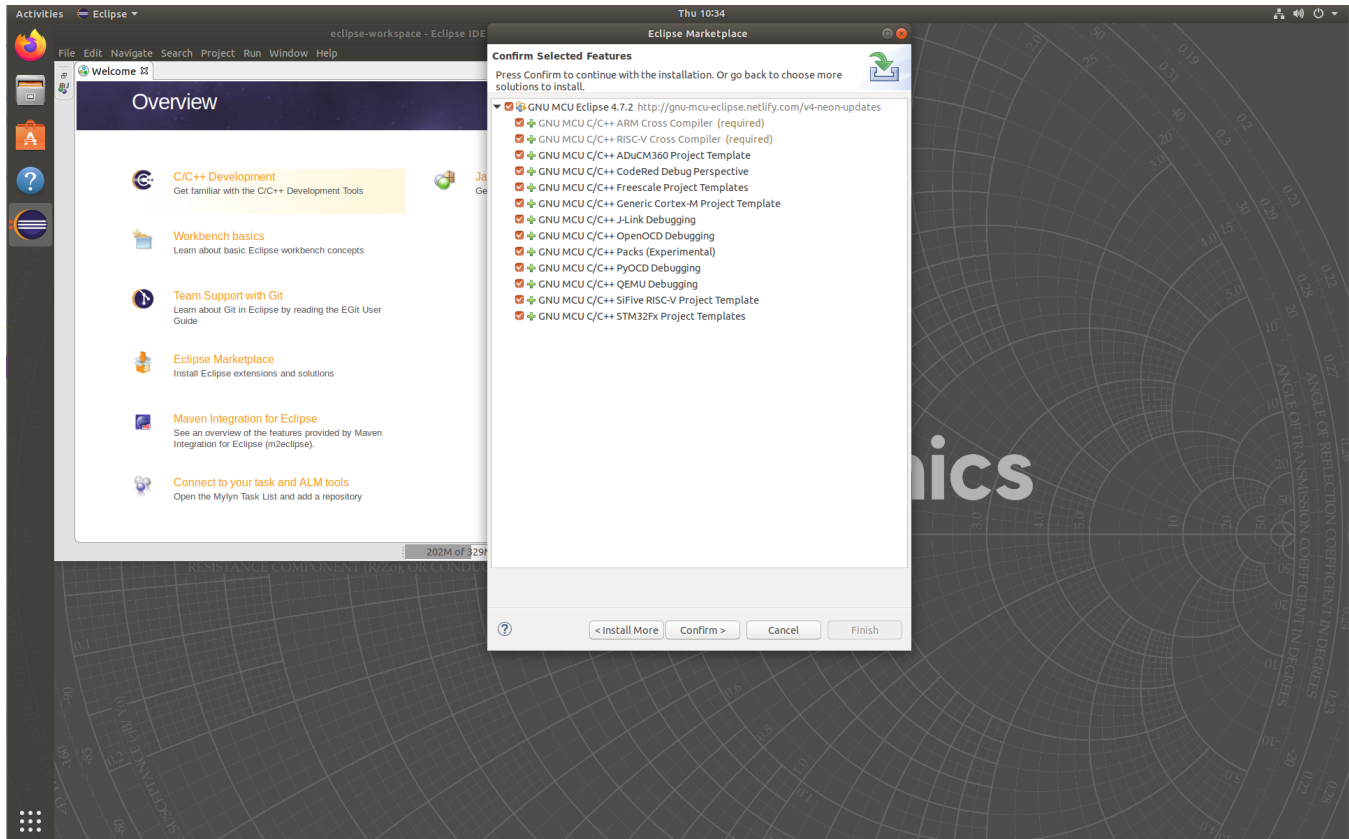
- In the Find text box, type "GNU MCU Eclipse" without the quotes and press ENTER.
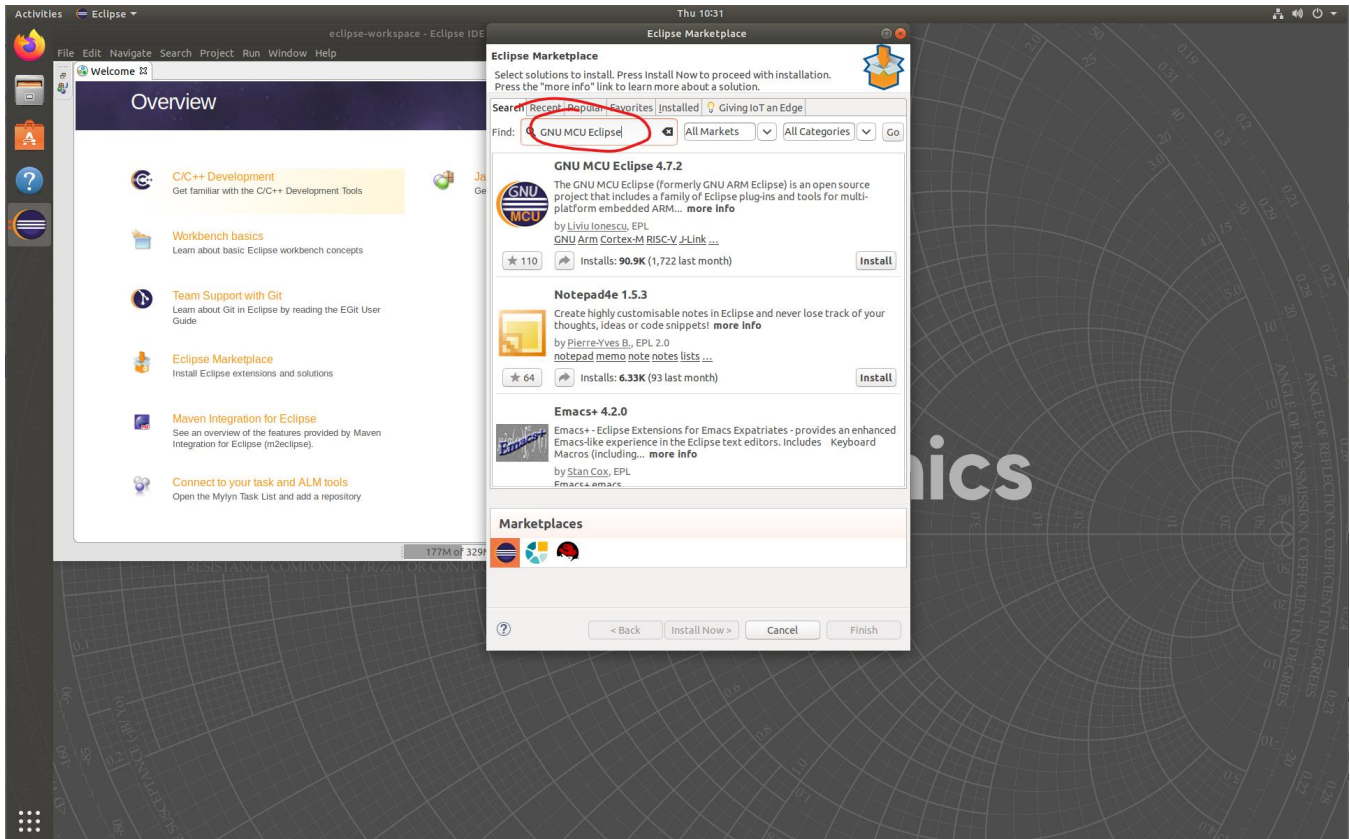
- When the search results are loaded, find the entry "GNU MCU Eclipse" and click on the "Install" button
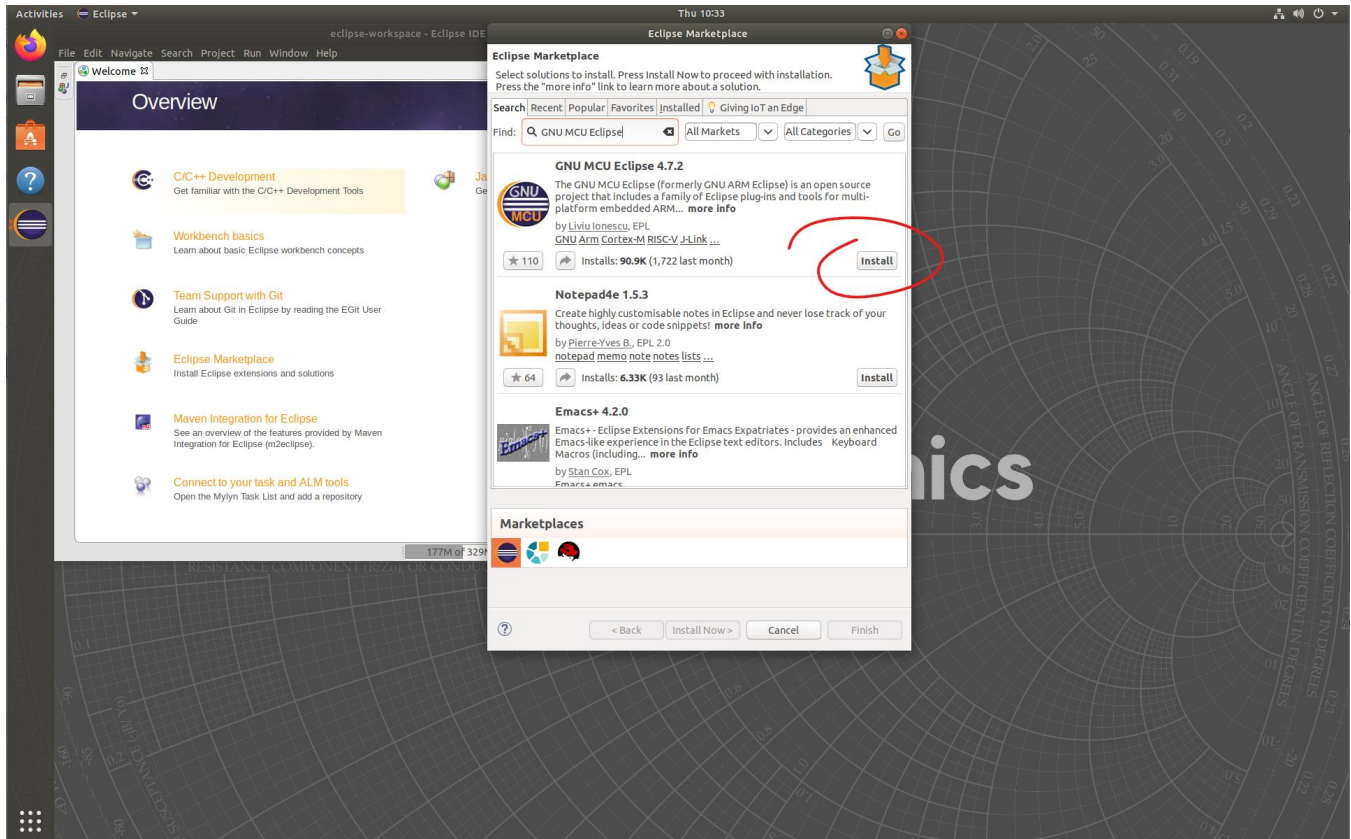
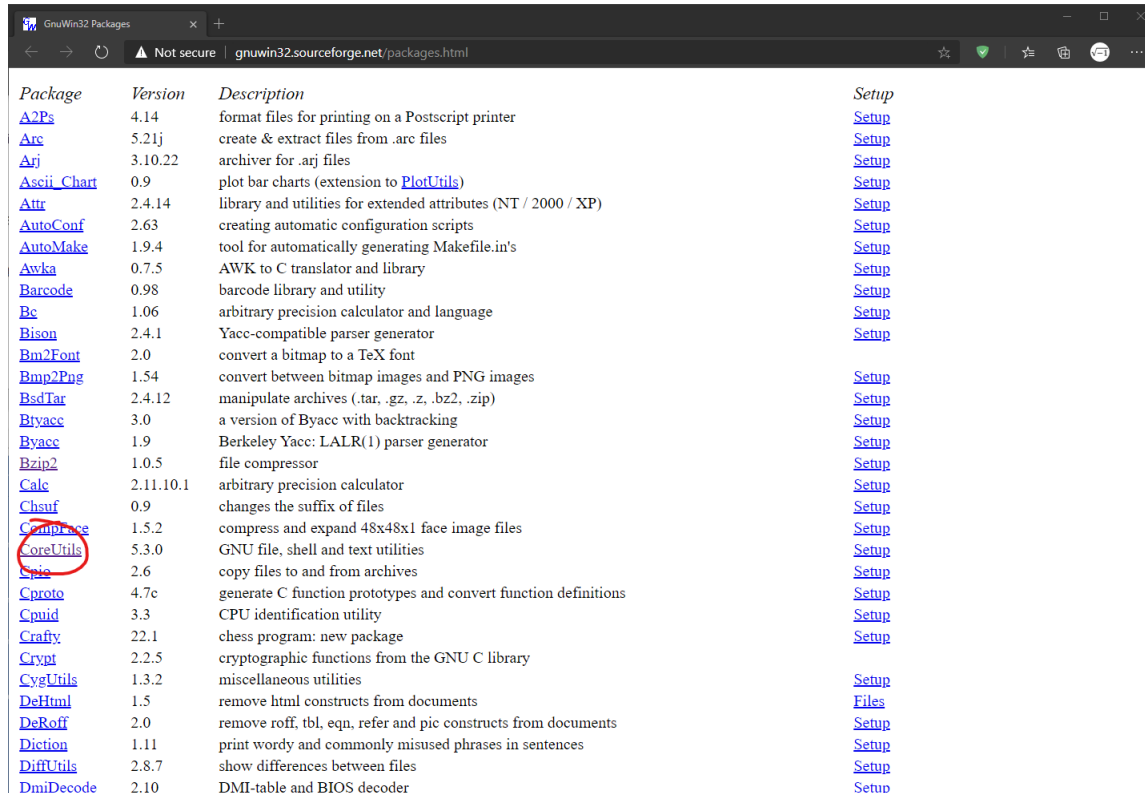- Accept the default selection and click "Confirm"

- Follow the on-screen instructions to complete. A security warning may pop-up indicating that the plugin contains unsigned content. Select "Install anyway" to complete the installation.
- Once the plugin is completed, Eclipse will prompt you to restart. Click "Restart now" to complete the eclipse development environment and toolchain installation.

2.2. Windows

2.2.1. Pre-requisite

There are four commands required from GnuWin32: cp, mkdir, rmdir, and make. They can be obtained from the packages CoreUtils and Make. There are other alternatives including but not limited to Cygwin, MinGW, or Windows Subsystem for Linux.

- Visit http://gnuwin32.sourceforge.net/packages.html
- Click on the package CoreUtils

- Download the setup package. If you choose manual installation, make sure that you download both the binaries and the dependencies.

- Repeat the above two steps for Make
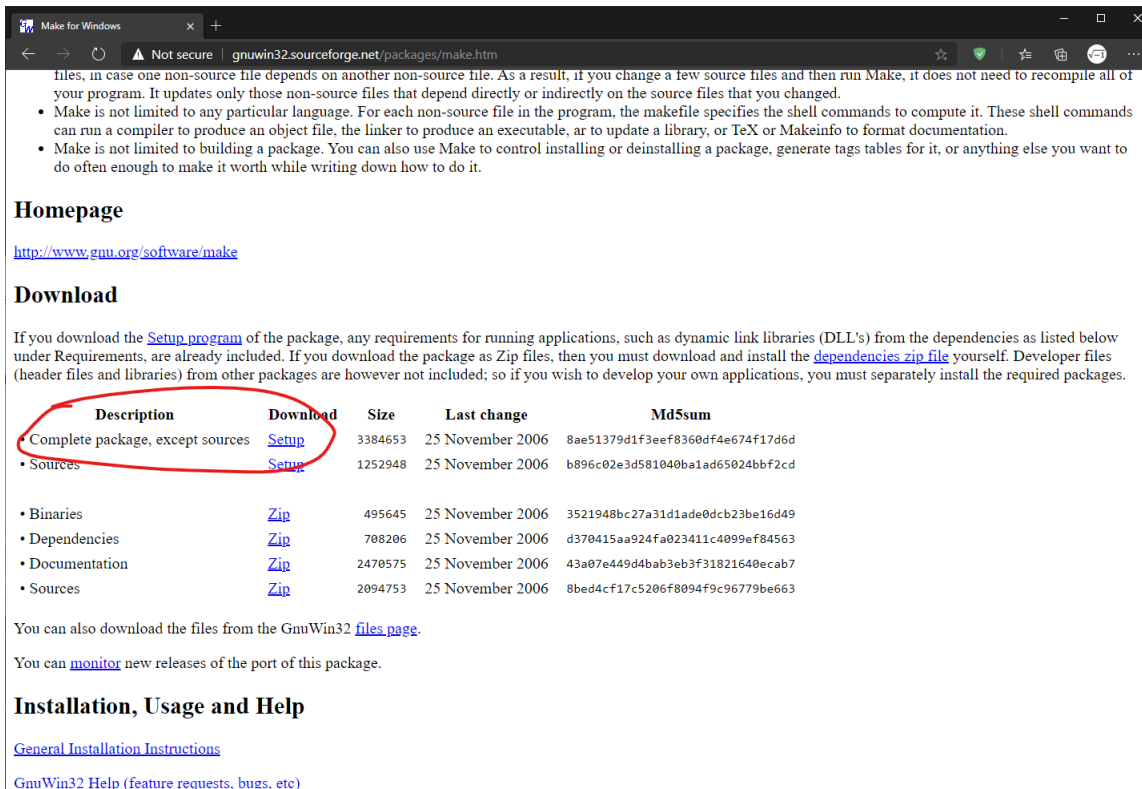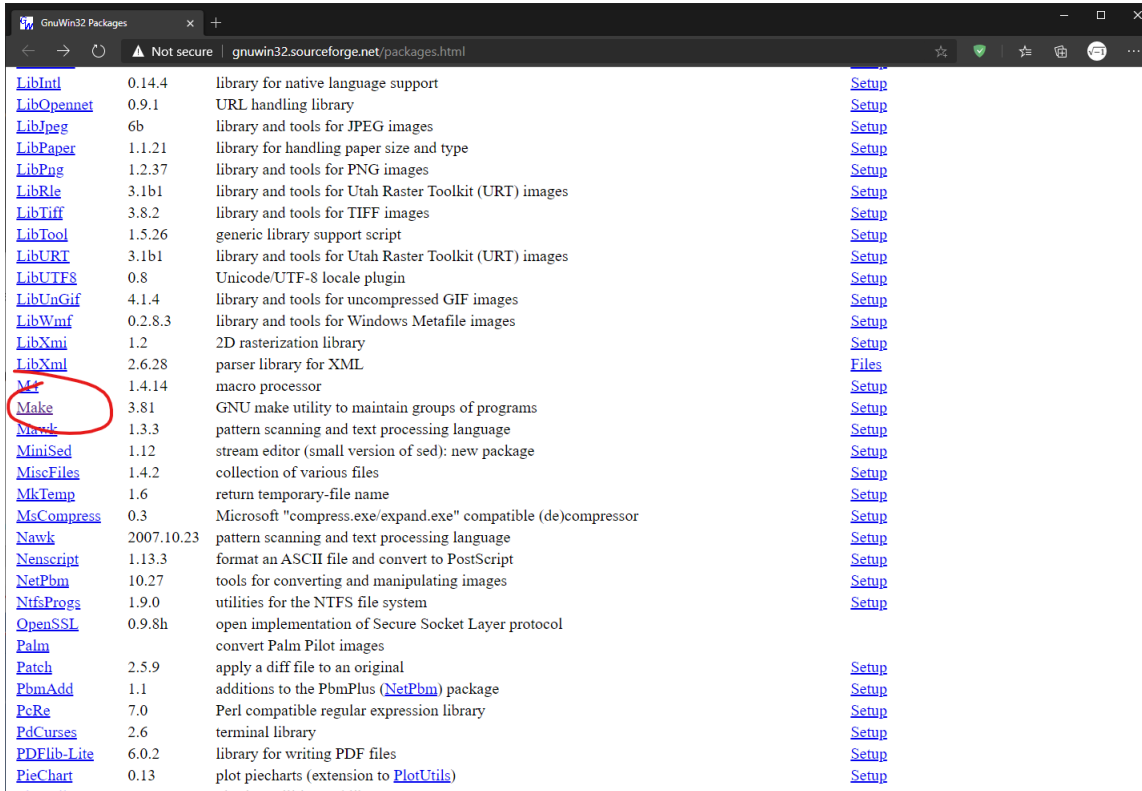
2.2.2. Python Installation

Python is needed to dynamically generate the board support package including the pin definition header and source files.

- Visit https://www.python.org/downloads/
- Download the latest version for Windows and follow the on-screen instructions.
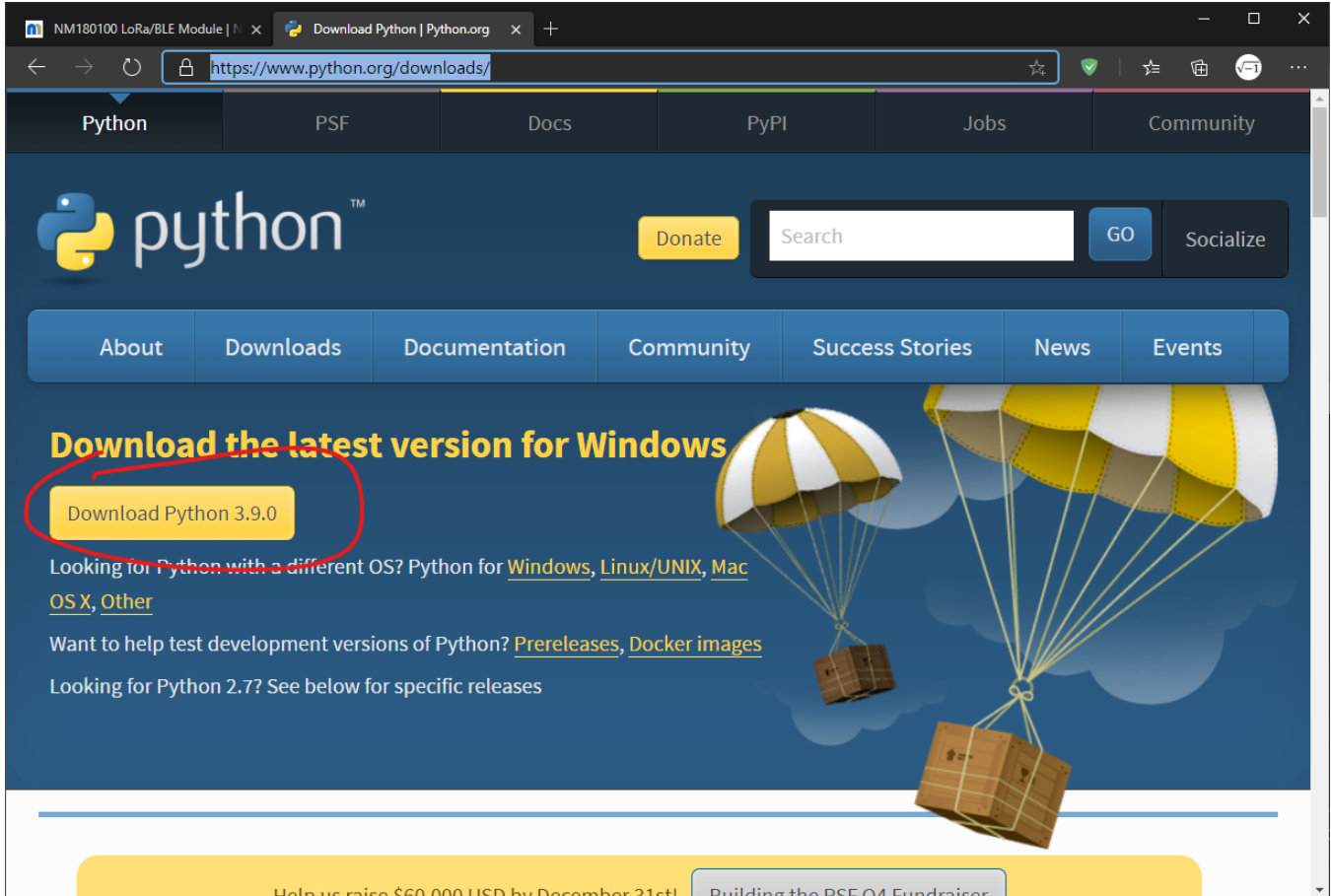
2.2.3. SEGGER J-Link Installation

Follow the steps in 2.1.3 and download the Windows version of J-Link at

https://www.segger.com/downloads/jlink/#J-LinkSoftwareAndDocumentationPack
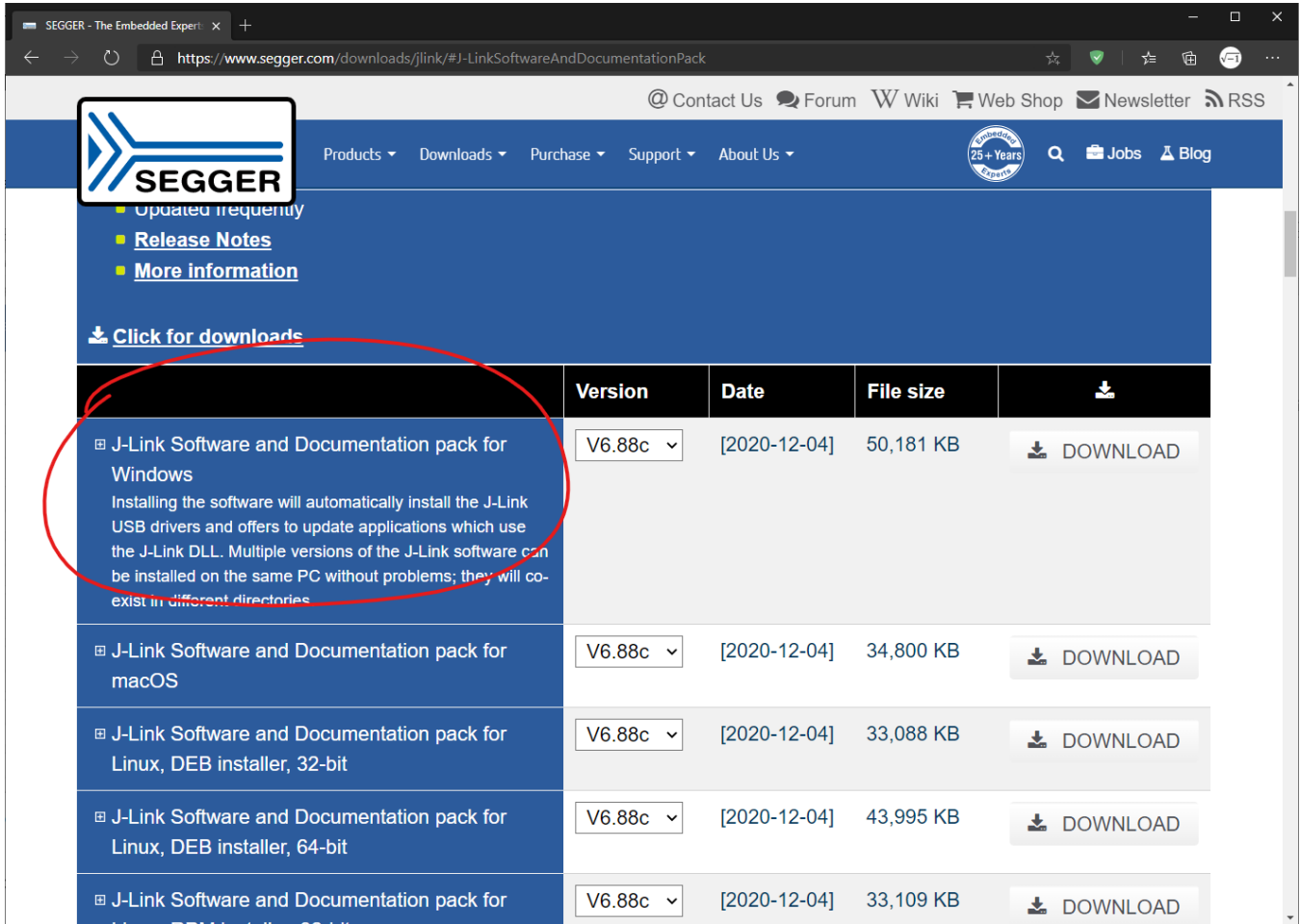
2.2.4. GNU ARM Embedded Toolchain Installation

Follow the steps in 2.1.4 and download the Windows version of GNU toolchain at

https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads

2.2.5. Executable Search Path

- Finally ensure that the paths to all the executables are added to your search path.
  1. Type "View Advanced System Settings" in the Windows search bar
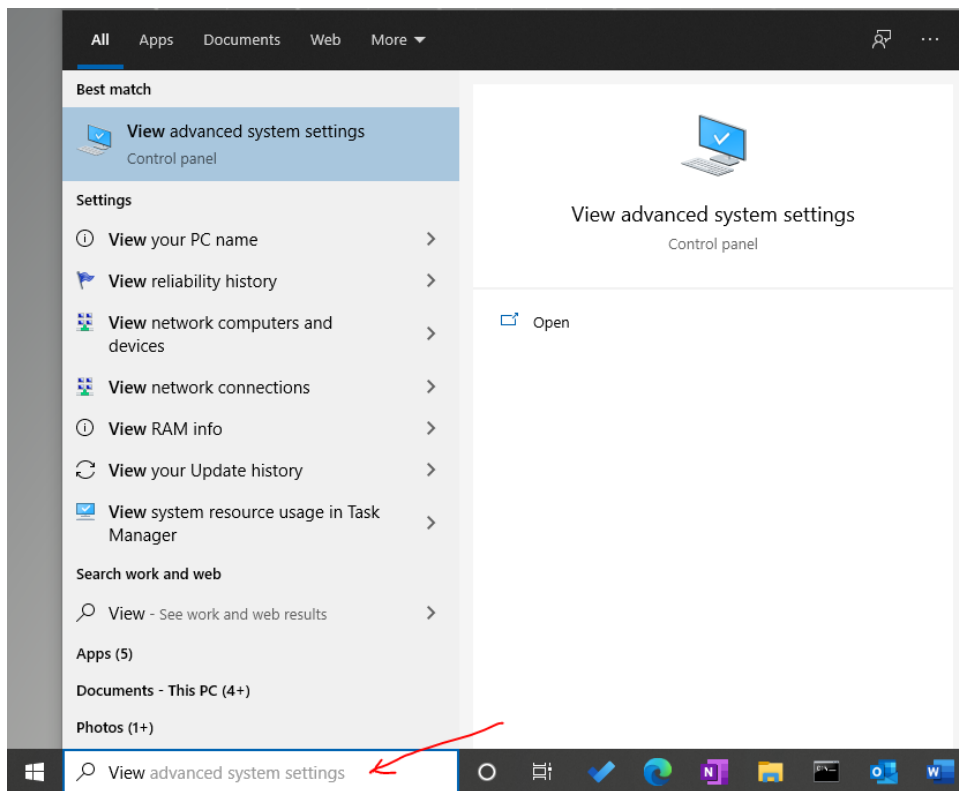
2. Click on Environment Variables

northern
mechatronics
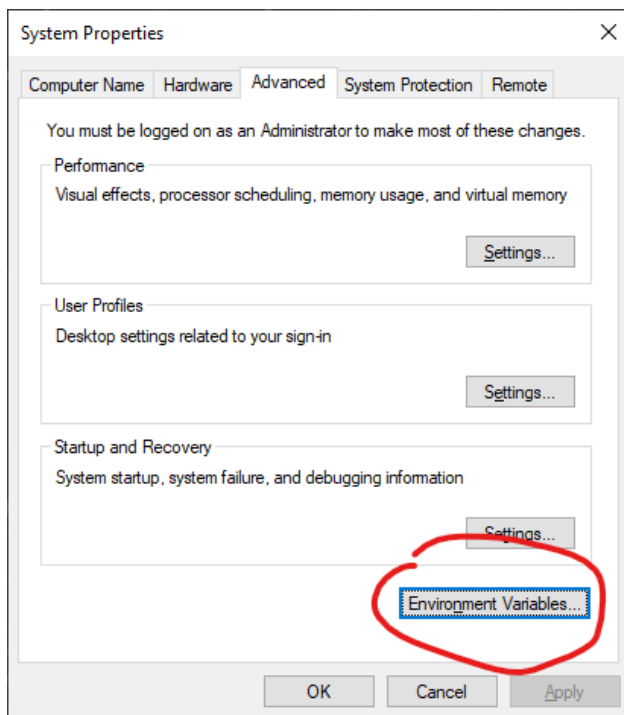
3. Select Path under System variables and click "Edit…"
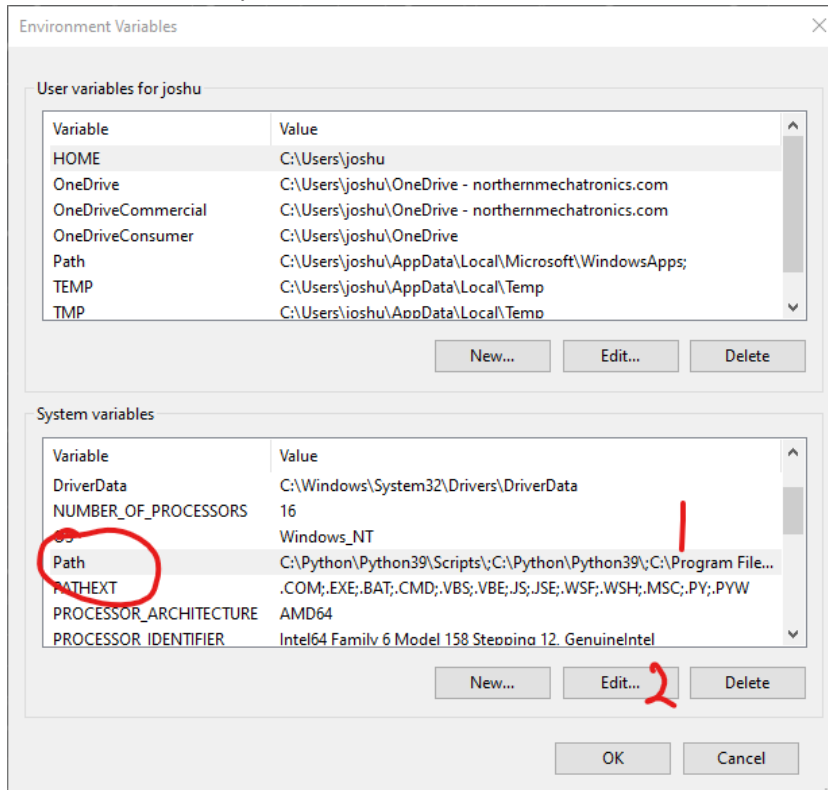
4. If not done already, add the paths where you have installed Python, GnuWin32, J-Link and Arm GNU Toolchain
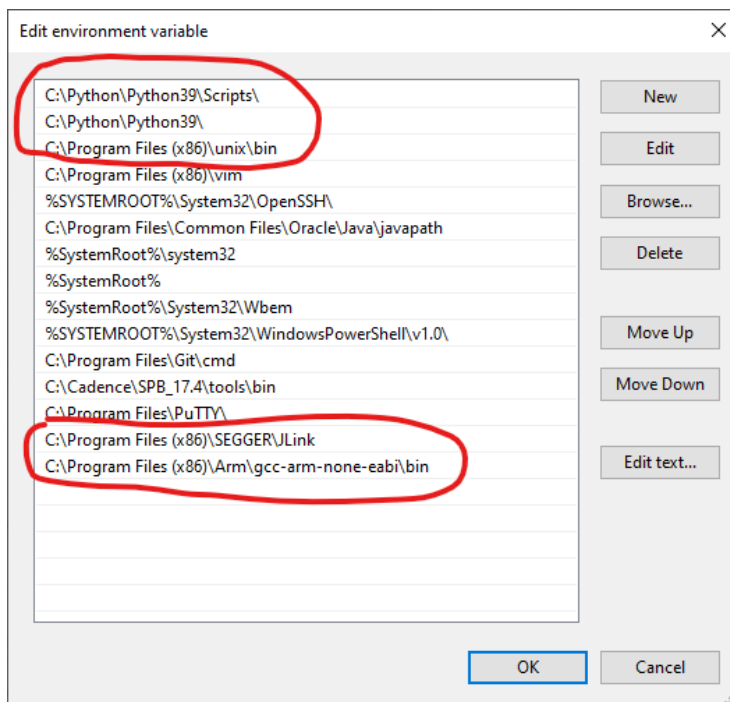
3. Eclipse Configuration

3.1. C/C++ Environment Setup

Eclipse defaults to the Java language development environment. You can change this by opening the C/C++ perspective.

- With Eclipse opened, click on "Window", "Perspective", "Open Perspective", and finally "Other…"
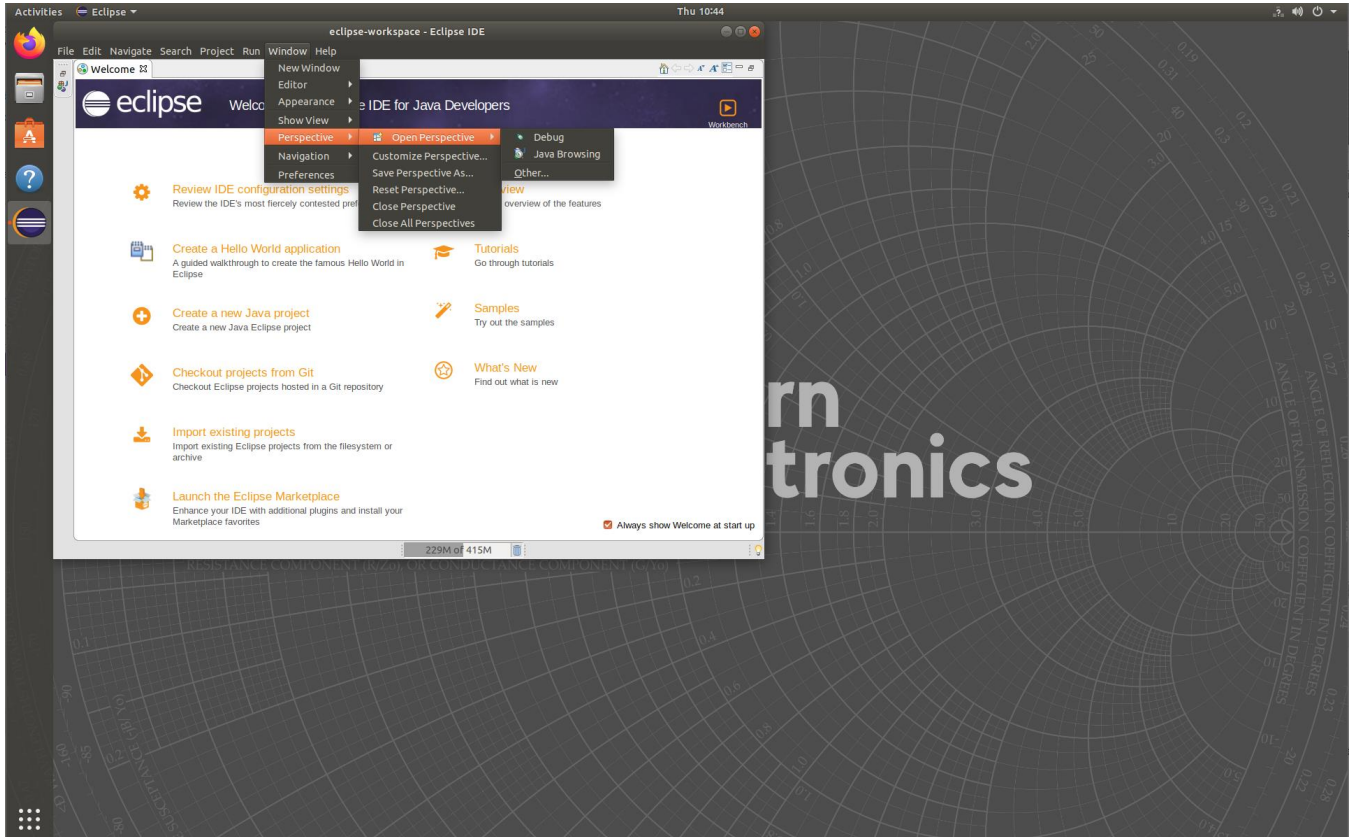
- This opens a new Window. Select "C/C++" and click "Open"
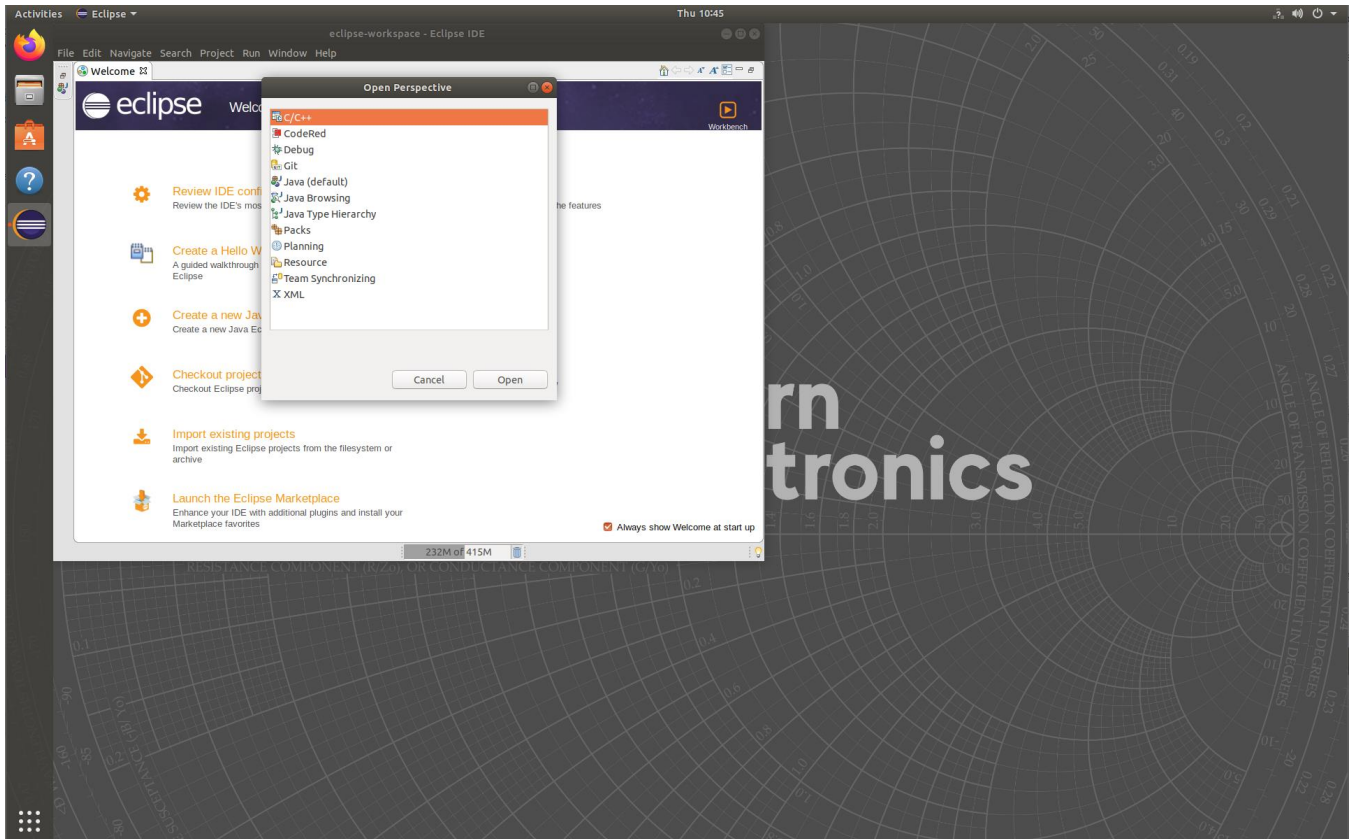
- Close or minimize the "Welcome" tab and you will be presented with the C/C++ environment.
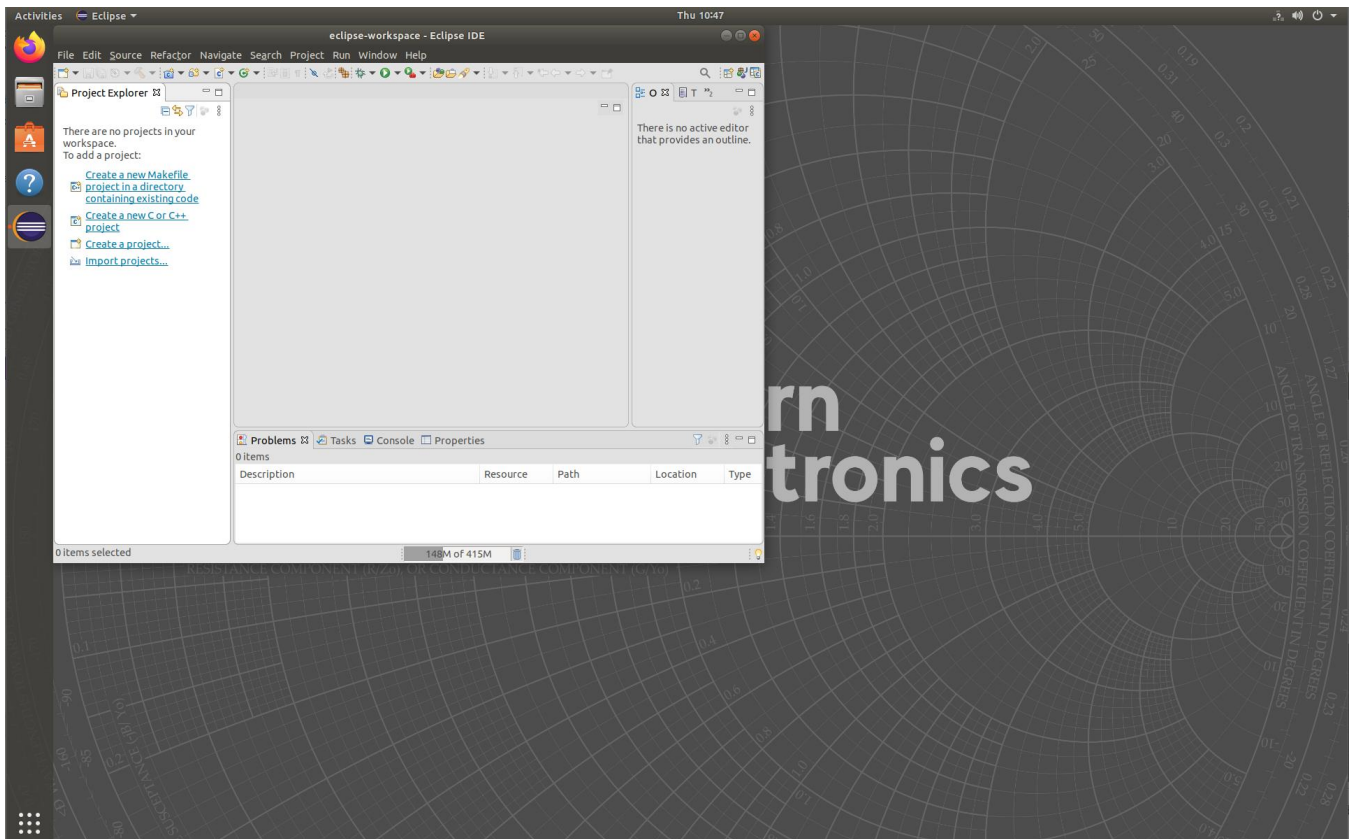
3.2. Eclipse GNU MCU Configuration

Finally, we will configure the GNU MCU plugin by specifying the location of the GNU ARM toolchain installation location.

- With Eclipse open, click "Window" in the menu bar and select "Preferences"
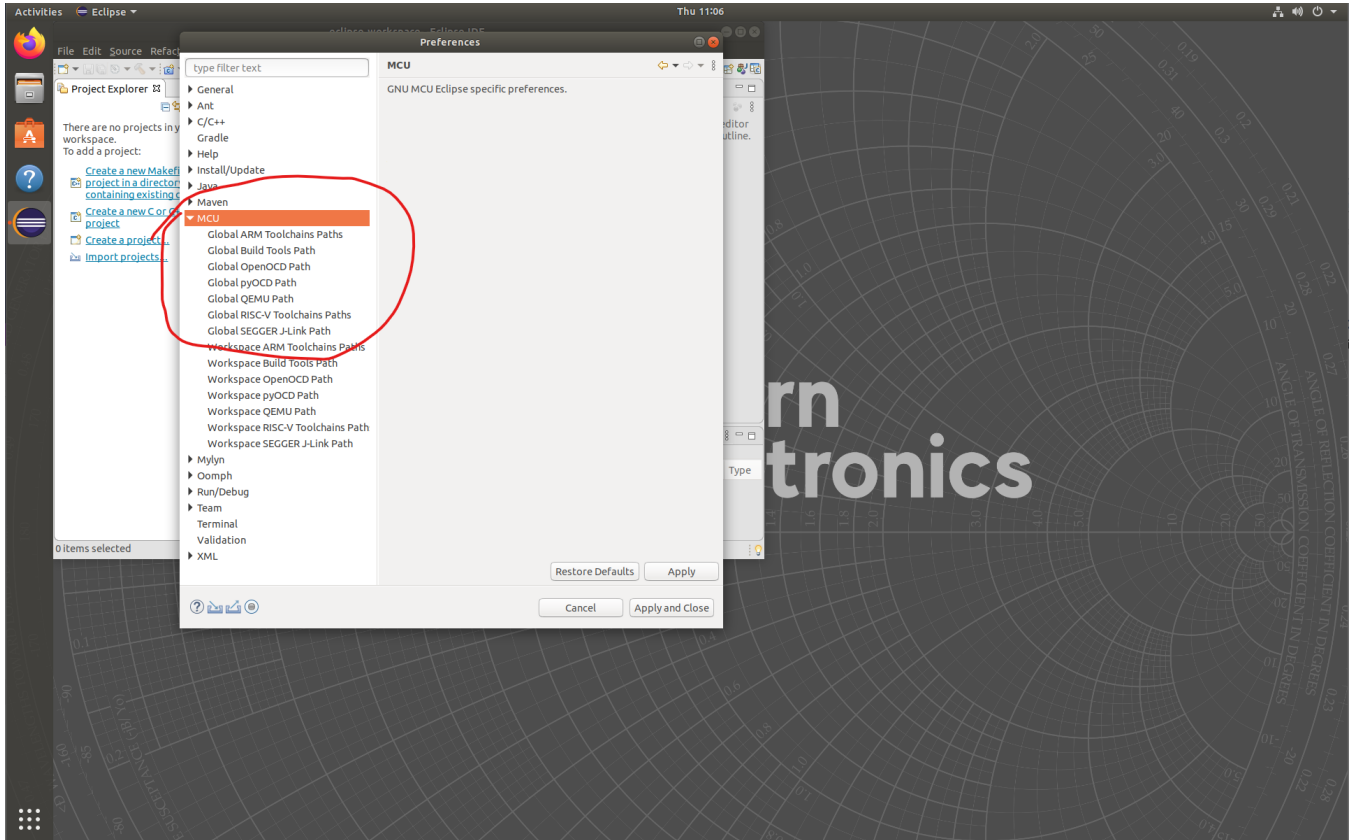
- Find and expand the row containing "MCU"
- The items under "MCU" are prefixed by either "Global" and "Workspace". We will focus on the Global settings right now. These settings are the global default.
- There are two paths we need to configure:
  - Global ARM Toolchains Paths
  - Global SEGGER J-Link Path

3.2.1. Global ARM Toolchains Paths Configuration

- With the Preference dialog window open, select "Global ARM Toolchains Paths" under MCU and click on "Browse"
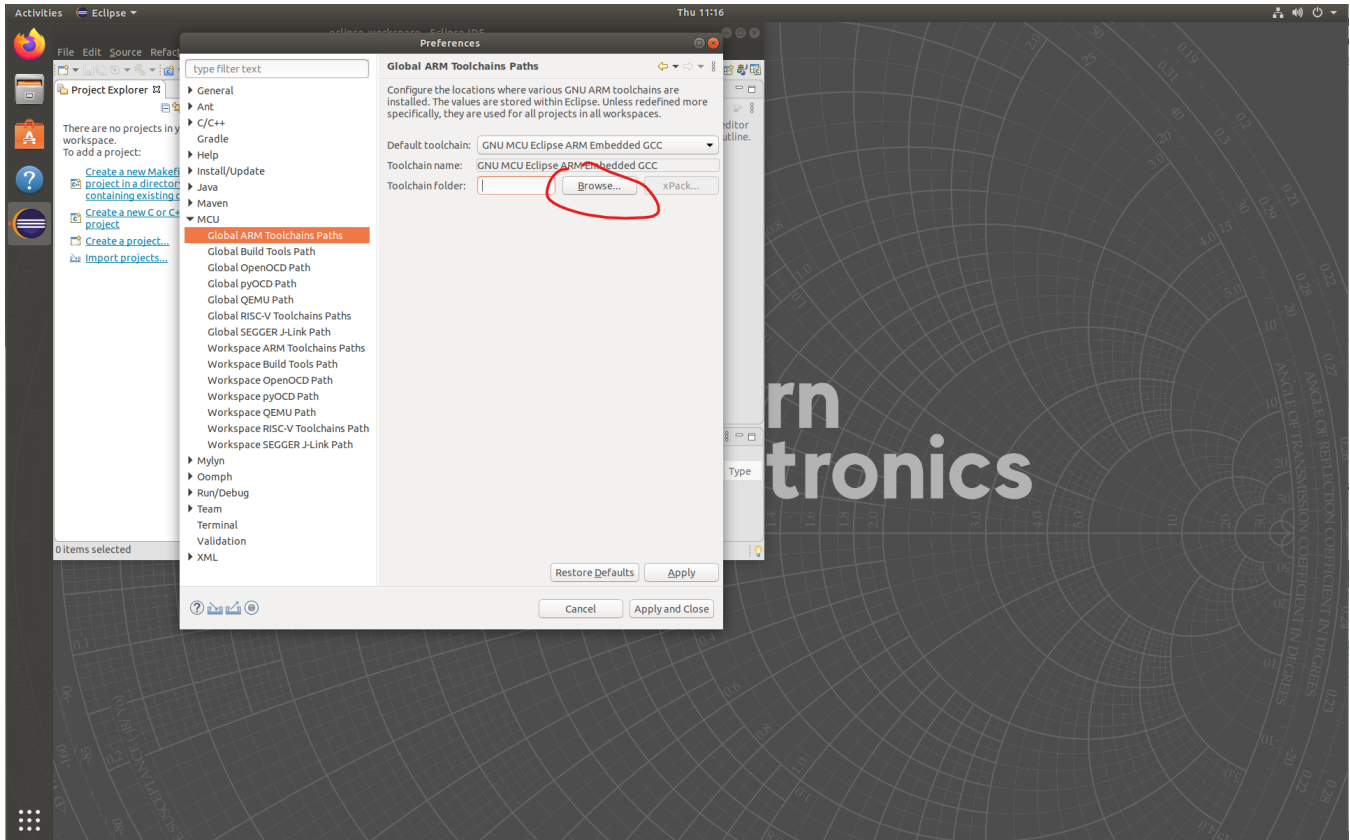
- Browse to the location where the ARM toolchain was installed in 2.1.4. In our example, this is located under

  `/opt/gcc-arm-none-eabi-9-2020-q2-update/bin`

- Click "Apply" to apply the changes.

3.2.2. Global SEGGER J-Link Path Configuration

- With the Preference dialog window open, select "Global SEGGER J-Link Path" under MCU.
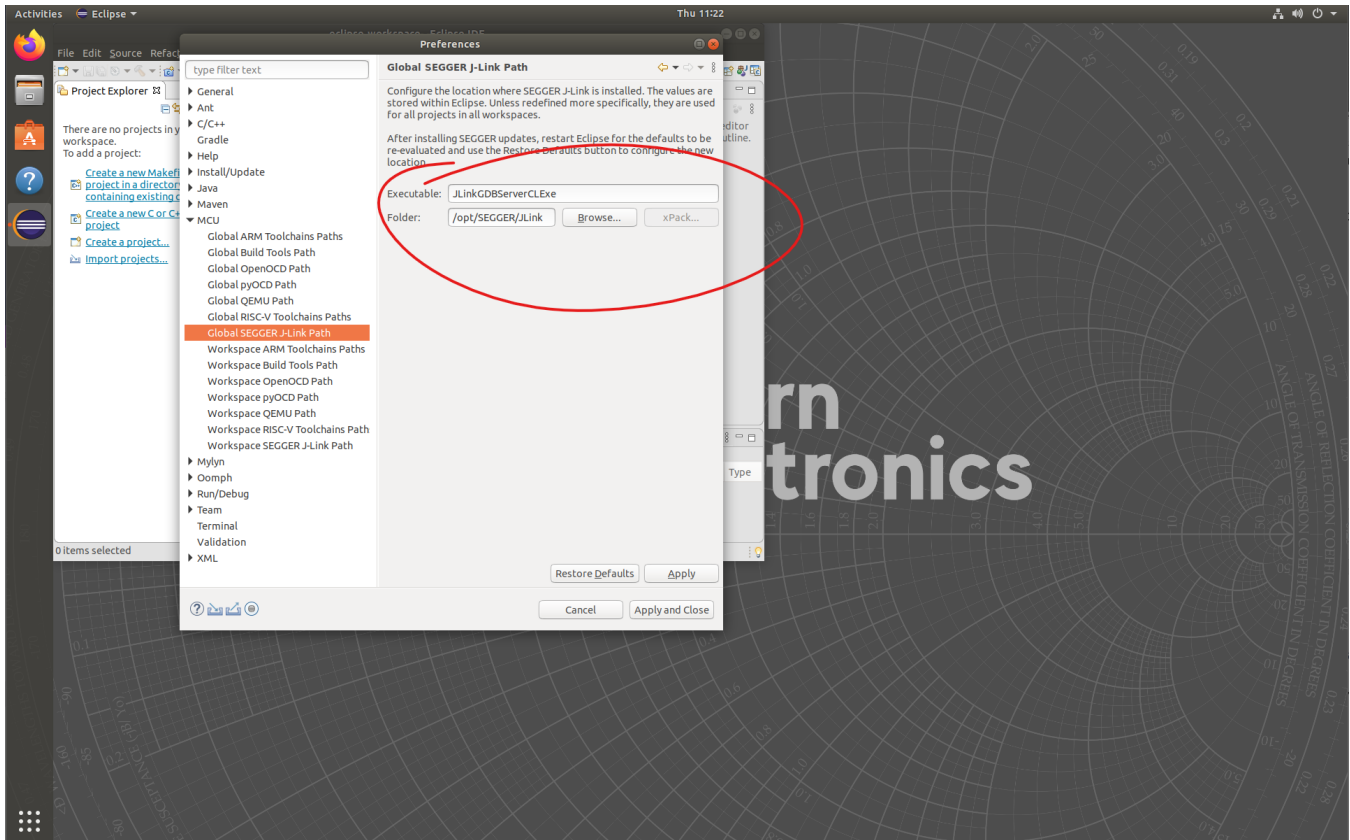

- The plugin should automatically detect the SEGGER J-Link installation. If not, enter the values into the Executable and Folder text boxes as shown in the screenshot.

  Executable: JLinkGDBServerCLExe

  Folder: /opt/SEGGER/JLink (or the path where you have installed SEGGER Jlink)

- Click "Apply and Close" to apply the changes.
- The Eclipse development environment is now completely configured.

4. Document Details

Parameter	Value
Name	NM180100 Software Development Toolchain Setup
Number	2000012
Revision	A.7
Life Cycle State	Released