# Prerequisites

1. Talaria TWO FreeRTOS SDK.
2. Ubuntu 20.04 or higher versions.

# JDK Installation

Eclipse IDE requires Java JRE as a prerequisite. JDK includes the JRE plus command-line development tools such as compiler and debuggers that are necessary for developing applets and applications.

Install the java jdk 11 by executing the following commands:

|  |
| --- |
| sudo add-apt-repository ppa:openjdk-r/ppa  sudo apt-get update  sudo apt install openjdk-11-jdk |

After successfully installing the JDK, verify the java JDK version by executing the command java-version in command line as shown in Figure 1.

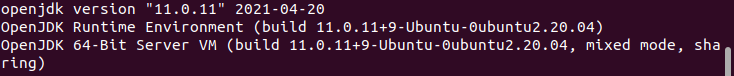


Figure 1: Verifying the java version

# Prerequisites for Eclipse

## Installing Necessary Packages

1. build-essential

The build-essentials packages are necessary for compiling the software. They include the GNU debugger, G++/GNU compiler collection, and some more tools and libraries that are required to compile a program.

1. libc6-armel-cross libc6-dev-armel-cross

These packages are generated for cross compiling. They contain the symlinks, headers, and object files needed to compile and link programs which use the standard C library.

1. binutils-arm-linux-gnueabi

This package provides GNU assembler, linker and binary utilities for the arm-linux-gnueabi target.

1. libncurses5-dev

The ncurses library routines are a terminal-independent method of updating character screens with reasonable optimization.

In terminal window, execute the following commands to install the GCC, G++ cross compilers and support programs:

|  |
| --- |
| sudo apt update  sudo apt install build-essential libc6-armel-cross libc6-dev-armel-cross binutils-arm-linux-gnueabi libncurses5-dev -y |

## Install GNU Toolchain for ARM Embedded

This toolchain contains integrated and validated packages featuring the compiler, libraries, and other tools necessary for software development.

1. Download the recent version GCC ARM toolchain.

|  |
| --- |
| wget <https://developer.arm.com/-/media/Files/downloads/gnu-rm/10.3-2021.10/gcc-arm-none-eabi-10.3-2021.10-x86_64-linux.tar.bz2> |

1. Execute the following command to extract the toolchain from the archive:

|  |
| --- |
| tar xvf gcc-arm-none-eabi-10.3-2021.10-x86\_64-linux.tar.bz2 |

1. Execute the following command to move the toolchain to the common usr folder.

|  |
| --- |
| sudo mv gcc-arm-none-eabi-10.3-2021.10 /usr/ |

**Note**: The following two steps (step 4 and 5) and should be followed **only** while using a virtual machine and **must be skipped** while using the standalone OS.

1. Execute the following commands to set up the toolchain path:

|  |
| --- |
| echo >> ~/.profile  echo 'export PATH=/usr/gcc-arm-none-eabi-10.3-2021.10/bin:$PATH' >> ~/.profile |

1. Reboot the Linux system so that the changes to PATH are applied.

## Installing GDB in Linux

GDB is a source-level debugger which allows the user to debug a program/application by applying break points, step run and watch variables.

In a terminal window, execute the following commands to install GDB:

|  |
| --- |
| sudo apt update  sudo apt-get install gdb  sudo apt-get install gdb-multiarch |

The gdb-multiarch package contains a version of GDB that supports multiple target architectures.

## Installing OpenOCD

OpenOCD is used for on-chip debugging, in-system.

In a terminal window, execute the following command to install OpenOCD:

|  |
| --- |
| sudo apt install openocd -y |