# Application Development - Manual Procedure

This section describes the manual procedure for developing the application for InnoPhase IoT product using Eclipse IDE.

## Import Talaria TWO Example Project

Procedure to import an example project remains the same as *Import Talaria TWO Example Project* for Eclipse plugin.

## Configuring the Eclipse Project

1. Click on your project, right click and select Properties.

Graphical user interface, application

Description automatically generated

Figure 42: Select Properties

1. Select Settings under C/C++ build and click on Toolchains. Add the tool chain path and build tools path through global, workspace and project links. Click Apply and Close.

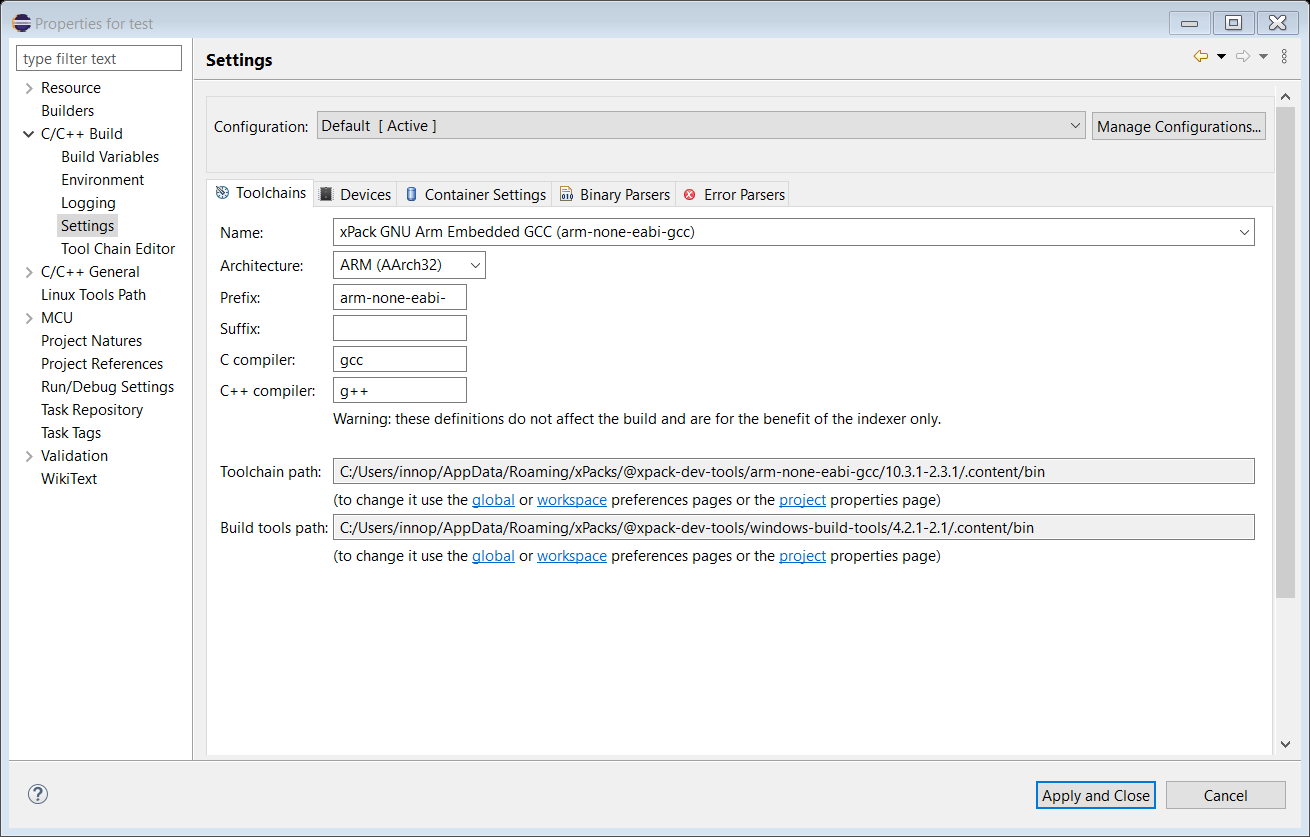


Figure 43: Settings Window

1. Click on ARM Toolchains Path option available under MCU.
2. Click on xPack. and a window opens as shown in the picture below.
3. Select the version of the toolchain and click ‘OK’.
4. Click ‘Apply’.

A screenshot of a computer

Description automatically generated

Figure 44: Adding ARM Tool Chain Path

1. Click on Build Tools Path option available under MCU.
2. Click on xPack and a window opens as shown in the picture below.
3. Select the version of the buildtools and click OK.
4. Click Apply.

A screenshot of a computer

Description automatically generated

Figure 45: Adding Buildtools Path

1. Click on OpenOCD Path, provide the OpenOCD as the executable.
2. Click on xPack, select the version of OpenOCD and click OK.
3. Click on ‘Apply’ and then ‘Apply and Close’.

A screenshot of a computer

Description automatically generated

Figure 46: Adding OpenOCD Path

1. Includes directory of the project is added.

Graphical user interface, text, application

Description automatically generated

Figure 47: Includes directory of the project

## Building Application in Eclipse

1. To build a project, click Project -> Build Project.

Graphical user interface, text, application

Description automatically generated

Figure 48: Building the Application

1. On successfully building the application, an out directory containing the .elf file is created inside the application.

Graphical user interface, text, application

Description automatically generated

Figure 49: Build console

## Debug Configuration Set-up in Eclipse

1. To start debugging the application, select the project and right click on it. Choose Debug As -> Debug Configuration.

A screenshot of a computer

Description automatically generated

Figure 50: Debug configuration settings

1. Double click on GDB OpenOCD Debugging and the debug configuration of the project is seen. Select the configuration and then point to the generated ELF of the application present in the out directory.

Graphical user interface, text, application, email

Description automatically generated

Figure 51: GDB OpenOCD Debugging settings

1. Under Debugger Settings, select the executable path as an OpenOCD installed path by clicking on Browse.

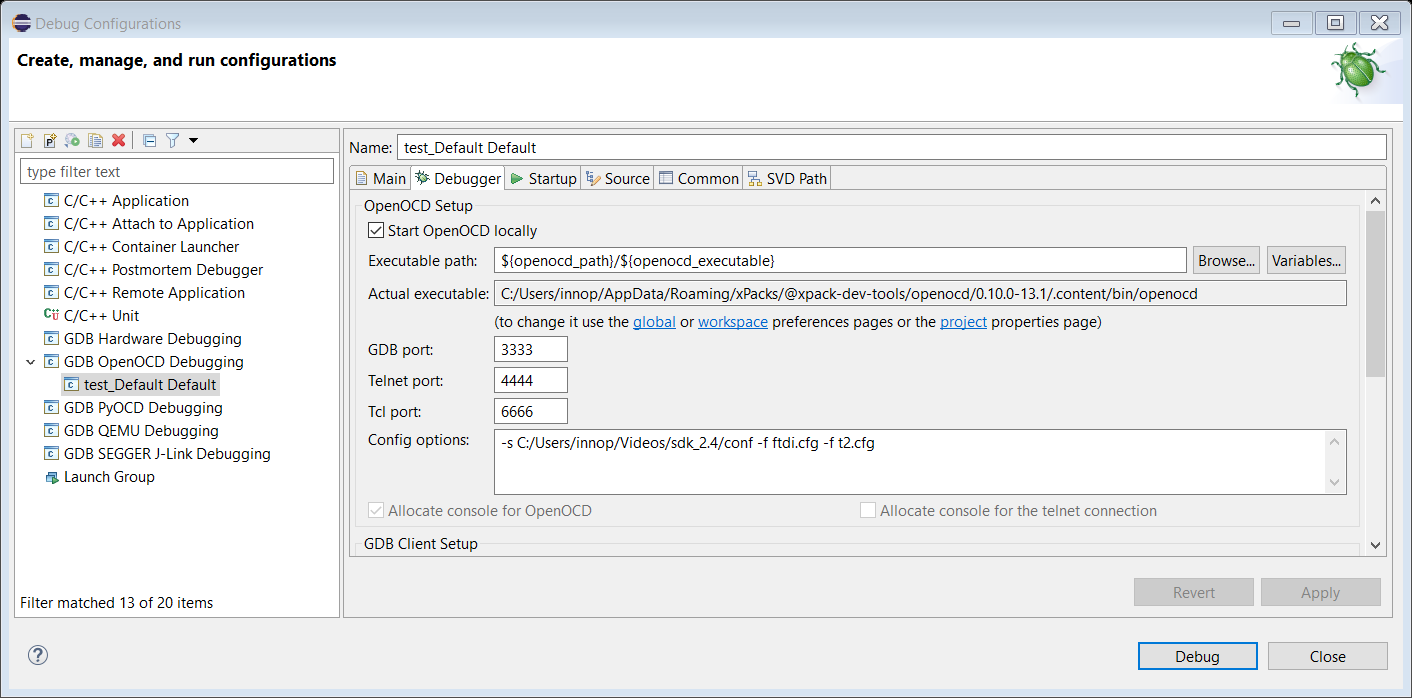


Figure 52: Debugger settings – OpenOCD installed path

In Config options, enter the path of configuration files available in conf directory of the SDK. These two files contain the configuration settings of Talaria TWO EVB. Configuration path should be as follows:

|  |
| --- |
| -s C:/<PATH TO SDK>/sdk\_x.y/conf -f ftdi.cfg -f t2.cfg |

**Note:** For SWD debugging, use the following configuration in Config options:

|  |
| --- |
| -s C:/<PATH TO SDK>/sdk\_x.y/conf -f ftdi\_swd.cfg -f t2\_swd.cfg |

A screenshot of a computer

Description automatically generated

Figure 53: Debugger config window for SWD Configuration

1. In GDB client setup, Select the actual executable arm-none-eabi-gdb as shown in Figure 54. Add the following commands in commands tab:

|  |
| --- |
| set mem inaccessible-by-default off  set substitute-path /tmp/build-2\_gctshx/ "C:/data/"  mem 0 0x40000 ro  mem 0x40000 0xc0000 rw  mem 0x100000 0x200000 ro  mem 0xfc0000 0x1000000 rw  mem 0x2000000 0x2100000 rw |

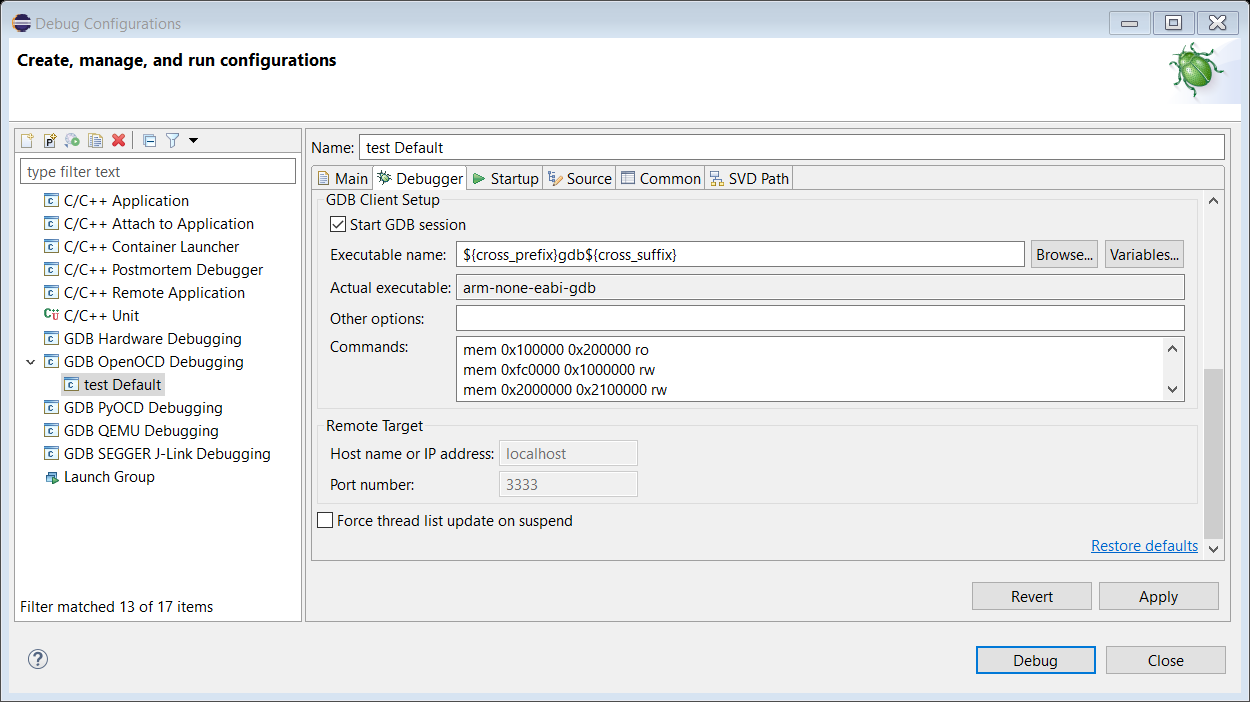


Figure 54: Debugger settings

1. Click on Apply.

## Programming Talaria TWO EVB

Flash the ELF onto Talaria TWO using the Download tool. Point to the elf file and click on PROG Flash to flash the application.

Ensure that the output is as shown in Figure 56. If not, Click on Clear Flash on the Download Tool and program the elf onto Talaria TWO again.

Graphical user interface, text, application, email

Description automatically generated

Figure 56: Download Tool Console

## Debugging in Eclipse

To debug the application the debugger provides control of program execution by setting breakpoints, suspending executed programs, stepping through the code and examining the contents of variables.

1. Click on Start-up tab and select the following options:
   1. Initial Reset
   2. Load symbols
   3. Load executable
   4. Debug in RAM
   5. Click on Debug in the start-up tab

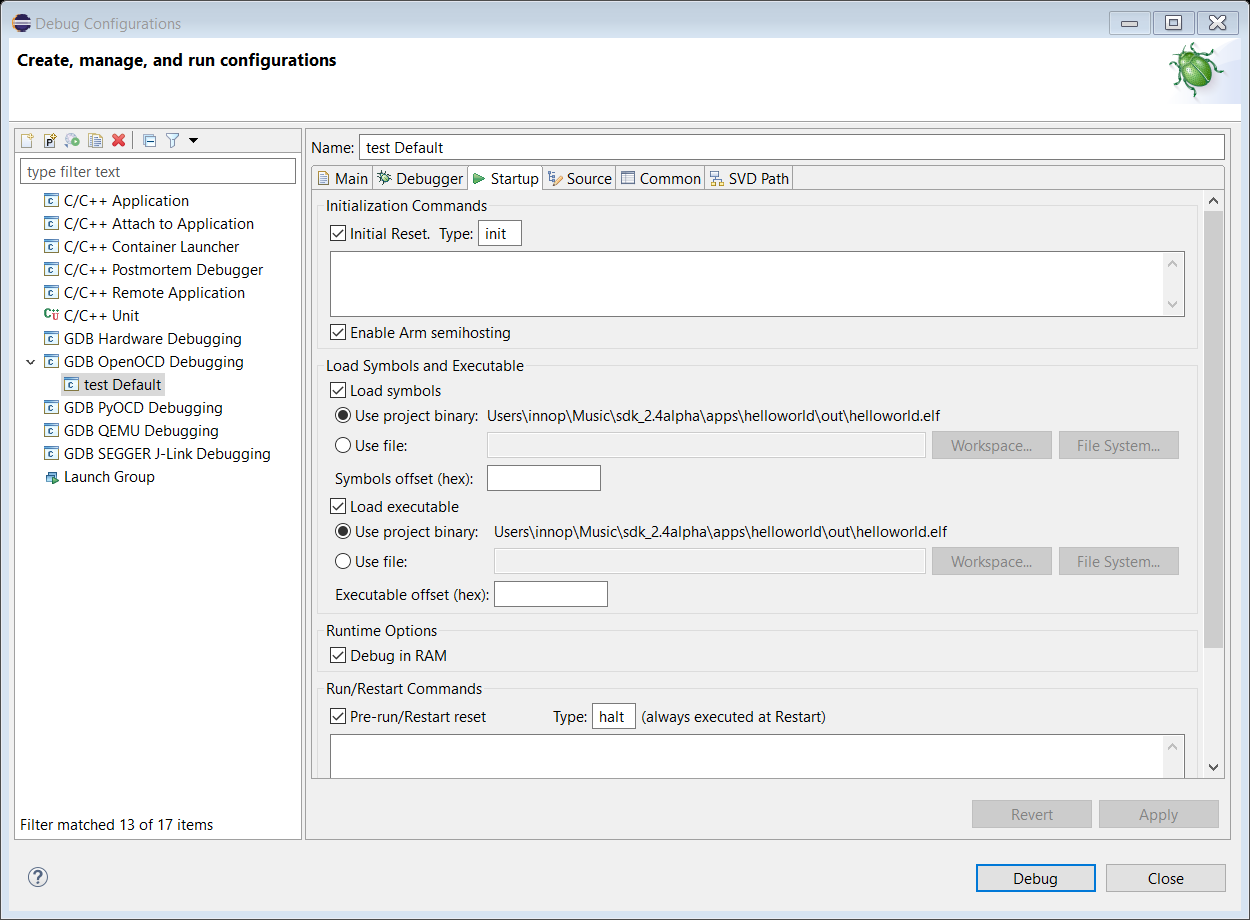


Figure 57: Debug start-up settings

1. On being prompted to switch to the Debug perspective, click Switch.

Graphical user interface, text, application, email

Description automatically generated

Figure 58: Confirm Perspective Switch

1. The Debug perspective appears with the application window open. Eclipse IDE re-positions into debug perspective.

Graphical user interface, text, application

Description automatically generated

Figure 59: Debug perspective

1. Breakpoints can be set by double-clicking to the left of the line number. The blue circle indicates ( ) that the breakpoint is set. Similarly, multiple breakpoints can be added. To start debugging, click on ( ).

Graphical user interface, text, application

Description automatically generated

Figure 60: Setting breakpoint

1. The execution will stop at the first breakpoint added to the c file. As shown in Figure 60, breakpoint is set at line 6.
2. To continue execution, click the Resume button ( ) on the toolbar of the Debug view. This will resume execution of the program and stop at the next breakpoint.
3. Step into and step over can also be used to continue execution of the next line.
4. While debugging the application, the application’s prints will be visible on Download Tool’s console window.

Graphical user interface, text, application, email

Description automatically generated

Figure 61: Step run output in console