# 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.

A screenshot of a computer

Description automatically generated

Figure 37: 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.

A screenshot of a computer

Description automatically generated

Figure 38: Include toolchain properties

1. Click on OpenOCD Path, provide the OpenOCD as the executable, click on Browse and point to the OpenOCD installed location.

A screenshot of a computer

Description automatically generated

Figure 39: Add OpenOCD path

## Building Application in Eclipse

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

A screenshot of a computer

Description automatically generated

Figure 40: Building the Application

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

A screenshot of a computer program

Description automatically generated

Figure 41: Build Console

## Debug Configuration Setting 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 42: Selecting debug configuration

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 as shown in Figure 43.

A screenshot of a computer

Description automatically generated

Figure 43: Changing debug configuration

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

A screenshot of a computer

Description automatically generated

Figure 44: Debugger config window

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

|  |
| --- |
| -s /home/<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 |

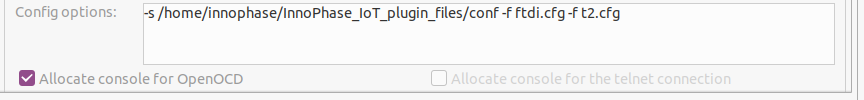


Figure 45: Debugger config window for SWD Configuration

1. In GDB client setup, select the variable and select the project\_loc.

**Note**: Copy the gdbinit files from apps folder to the application directory for debugging the particular application.

A screenshot of a computer

Description automatically generated

Figure 46: Selecting the variable

Using project\_path variable, locate the gdb-multiarch for the GDB client setup. For example:

|  |
| --- |
| /usr/bin/${ project\_path: <location of apps directory in sdk>}gdb-multiarch |

1. Click Apply.

## Programming Talaria TWO EVB

Program the ELF onto Talaria TWO using the Download tool.

For more information on flashing the ELF using the Download Tool, refer: UG\_Download\_Tool.pdf (path: *sdk\_x.y\pc\_tools\Download\_Tool\doc*).

**Note**: x and y refer to the SDK release version. For example: sdk\_2.4\doc.

Click on Clear Flash on the Download Tool and ensure the output is as shown in Figure 47.

Text

Description automatically generated

Figure 47: Download Tool Console

## Debugging in Eclipse

The debugger provides for the 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

Click on Debug in the Start-up tab.

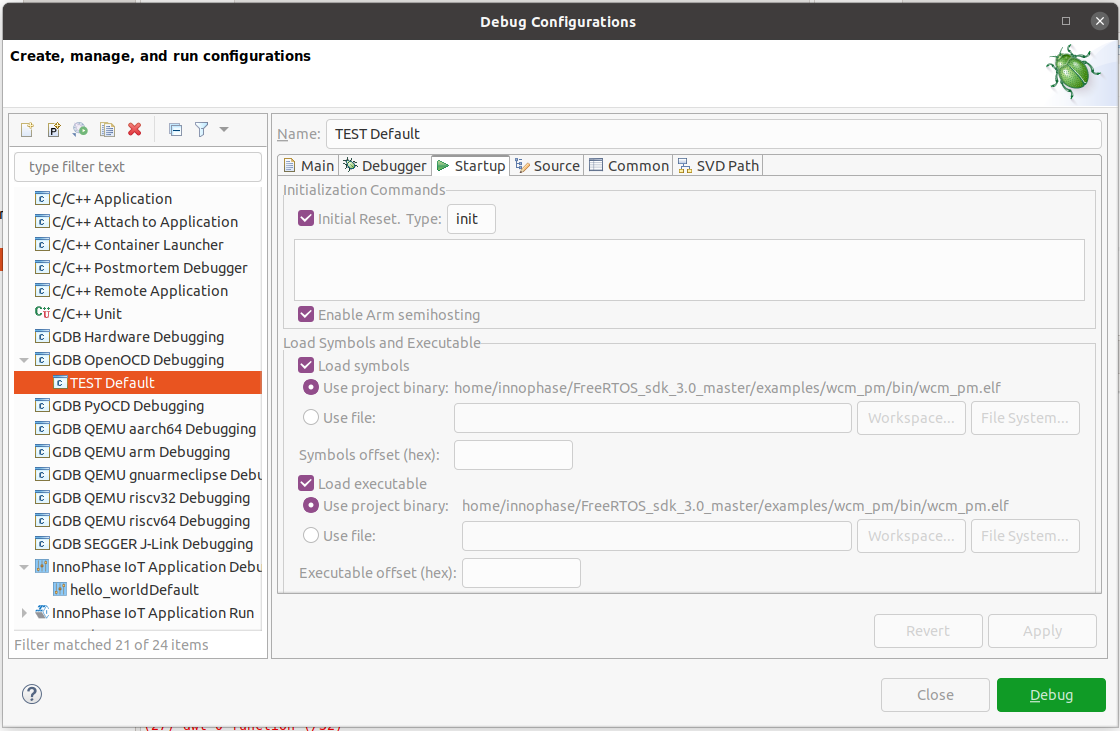


Figure 48: Debug startup window

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

A screenshot of a computer

Description automatically generated

Figure 49: Confirm Perspective Switch

1. The Debug perspective appears with the iperf3.elf application window open. The eclipse IDE repositions into debug perspective.

A screenshot of a computer

Description automatically generated

Figure 50: GDB OpenOCD Debugging – Start-up

1. Break Points:

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 Resume ( ).

A screenshot of a computer

Description automatically generated

Figure 51: Setting breakpoints

1. Execution will stop at the first breakpoint added to the c file. As shown in Figure 51, breakpoint is set at line 5.
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 the execution of the next line.
4. While debugging the application, the application’s prints will be visible on the Download Tool’s console window.

A computer screen shot of a computer screen

Description automatically generated

Figure 52: Application prints in the Download Tool console