# Eclipse Plugin: InnoPhase IoT Software Development Tool for Application Development

InnoPhase IoT Software Development Tool is an Eclipse plugin which simplifies the application development for InnoPhase IoT product using Eclipse IDE.

**Note**:

1. Currently, the plugin supports only one connected device. In case more than one device is connected, the plugin might not function as expected.

2. Steps for manually developing the application are available in section: *Application Development - Manual Procedure*.

## InnoPhase IoT Software Development Tool Installation

Install plugin InnoPhase IoT Software Development Tool onto Eclipse using following steps:

1. Help -> Install New Software

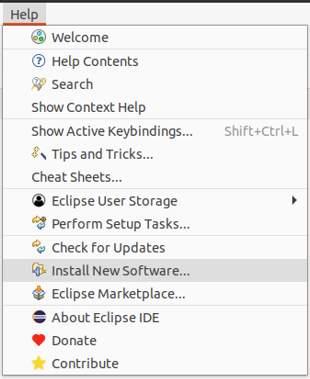


Figure 9: Plugin installation – Install new software

1. Add -> Location to <https://www.innophaseiot.com/eclipse-plugins> ->Add

A screenshot of a computer

Description automatically generated

Figure 10: Plugin installation – Add Repository Location

1. Select the check box: InnoPhase IoT -> Next

A screenshot of a computer

Description automatically generated

Figure 11: Plugin installation – Choose available software to install

1. Once done, Eclipse begins to calculate the requirements and dependencies.

A screenshot of a computer

Description automatically generated

Figure 12: Plugin installation – Calculating requirements and dependencies

1. Select radial button “*Keep my installation the same and modify the items being installed to be compatible*” -> Next

A screenshot of a computer

Description automatically generated

Figure 13: Plugin installation – Install remediation page

1. Press Next to continue.

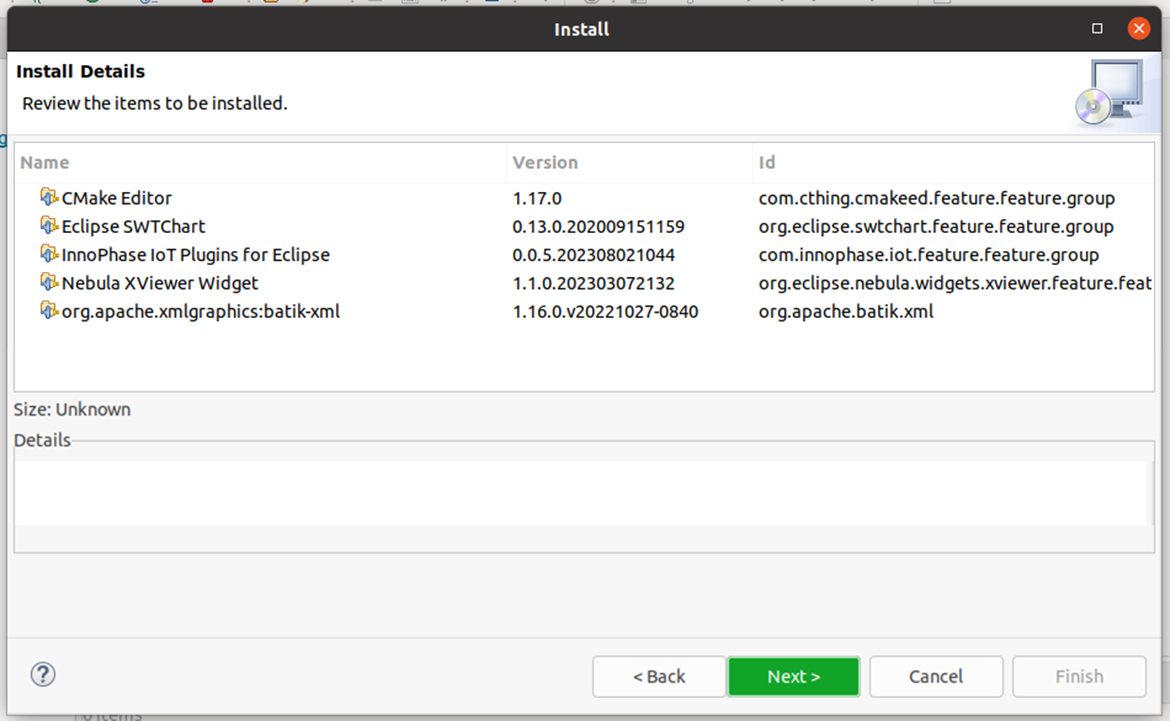


Figure 14: Plugin installation – Install Details

1. Accept terms -> Finish

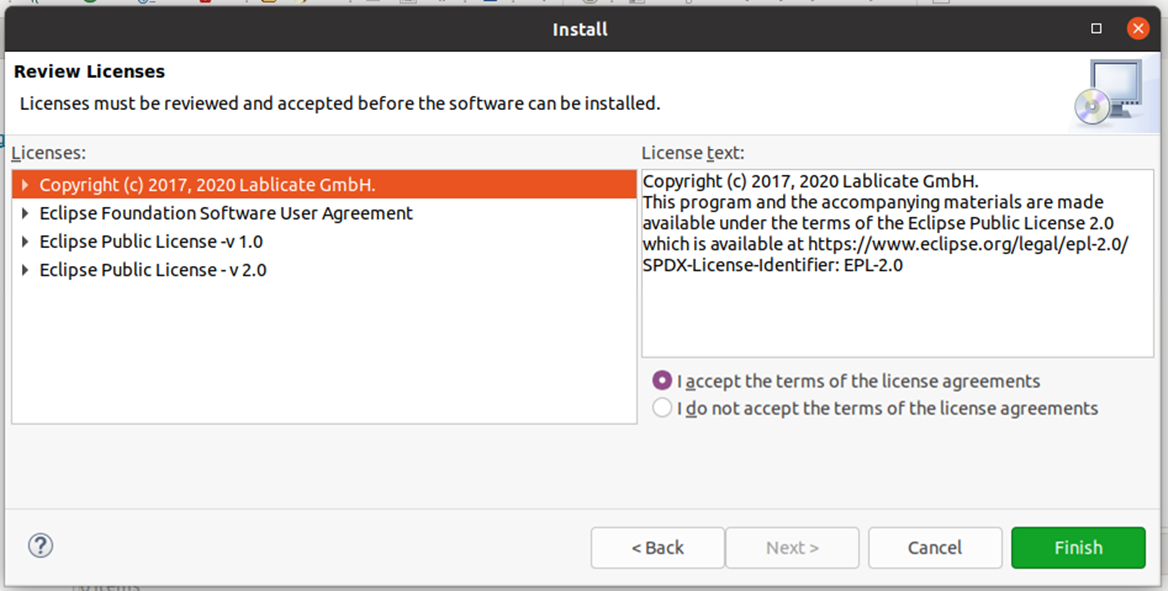
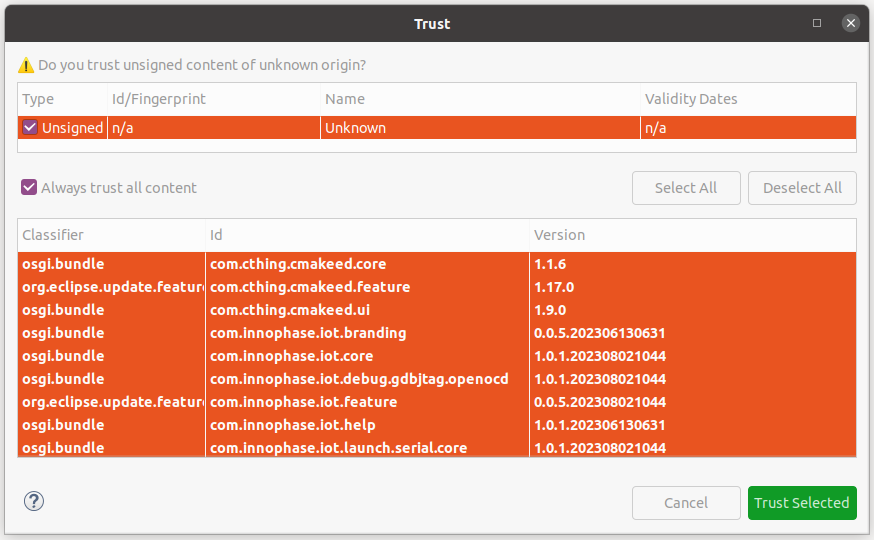


Figure 15: Plugin installation – Review licenses

1. Select the check box Unsigned & Always trust all content and Click on Trust Selected -> Yes, I Accept the Risk



A screenshot of a computer error message

Description automatically generated

Figure 16: Plugin installation – Trust unsigned content

1. Software begins to install, which is indicated in the status bar.

A screen shot of a computer

Description automatically generatedc

Figure 17: Plugin installation – Installing software status

1. Once software installation is complete, a pop-up message for restarting appears. Click on Restart Now.

A screenshot of a software update

Description automatically generated

Figure 18: Plugin installation – Restart Now after installation

1. On successful installation, Eclipse IDE will have a tab for InnoPhase IoT on restarting.

A screenshot of a computer

Description automatically generated

Figure 19: Plugin installation – InnoPhase IoT menu

## Download and Configure Support Files for Plugin

InnoPhase IoT Software Development Tool plugin requires supporting files, which need to be configured before use. Follow the subsequent steps to configure the supporting files:

1. InnoPhase IoT -> Download and Configure Support Files

A screenshot of a computer

Description automatically generated

Figure 20: Download and configure support files menu

1. Browse and choose a directory to download and configure support files for *InnoPhase IoT Software Development Tool* plugin.

A screenshot of a computer

Description automatically generated

Figure 21: Download and configure support files - Input window

**Note**:

1. If the plugin support files are already available, then check the box: Use an existing InnoPhase IoT support directory file system and choose the existing folder.
2. Add 777 permission to file InnoPhase\_IoT\_Eclipse\_T2\_flasher\_Linux in the folder *InnoPhase\_IoT\_plugin\_files* created under the directory chosen for downloading the InnoPhase IoT plugin support files.
3. Click Finish. Support files will be downloaded and saved in the selected folder. A pop-up message will confirm once the download is completed.

A screenshot of a message

Description automatically generated

Figure 22: Download and Configure Support Files – Status message

## Import Talaria TWO Example Project

1. Import an existing example as a make file project in eclipse:
   1. Download and unzip Talaria SDK provided
   2. Open Eclipse
   3. Click on File -> Import

Graphical user interface, application

Description automatically generated

Figure 23: Import an existing example in Eclipse

1. Under C/C++, Click Existing code as Makefile Project and then click Next.

Graphical user interface, text, application, email

Description automatically generated

Figure 24: Import existing code as makefile project

1. Enter the project name, select any of the applications or examples available in SDK package, select the Toolchain as ARM cross GCC and click Finish.
2. Project is imported with the source code and existing make file.

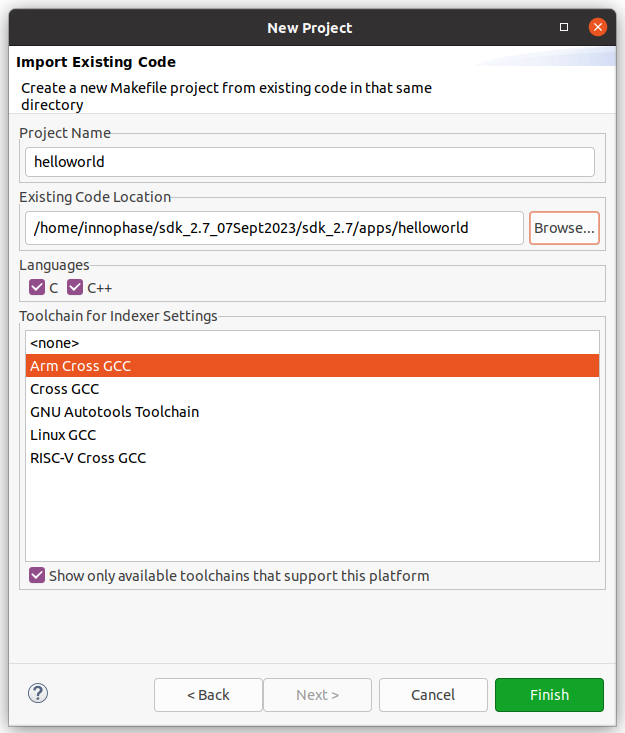


Figure 25: Project imported

## Run Configuration Set-up

InnoPhase IoT Software Development Tool provides *the Run Configuration set-up to Build and Flash the Application to connected Talaria TWO EVB*. Steps for setting up the run configuration is as follows:

1. Right click on Project and select Run As ->Run Configurations.

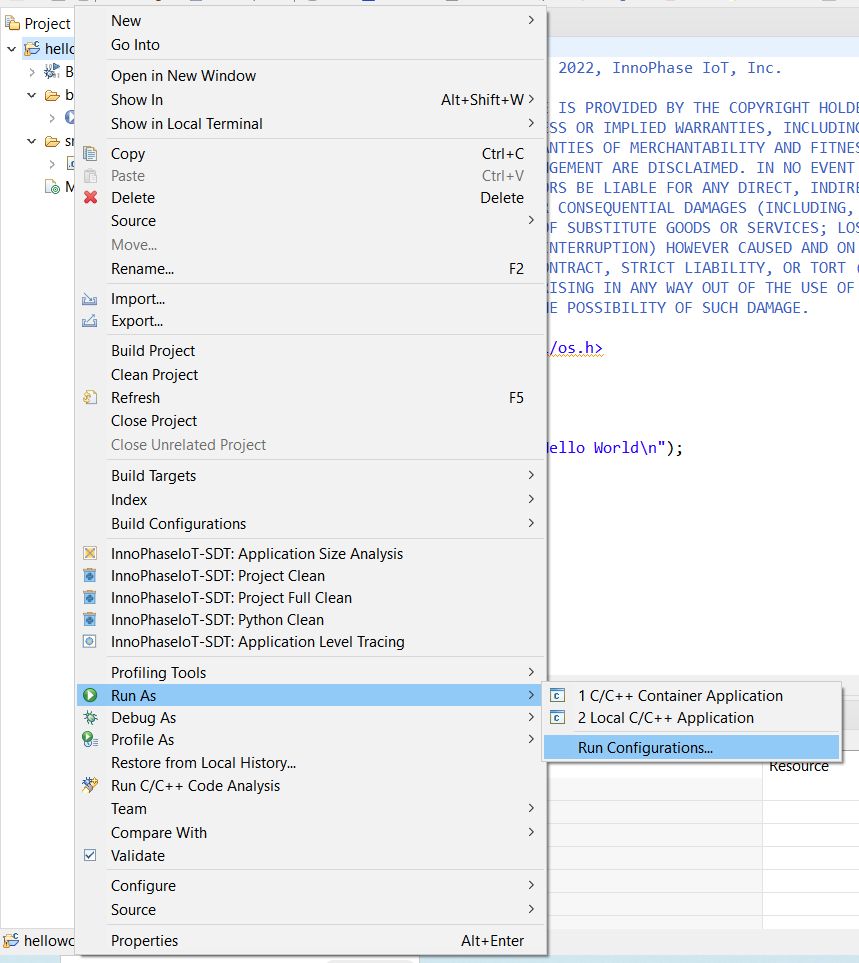


Figure 26: Eclipse Run As – Select Run Configuration

1. Right click on InnoPhase IoT Application Run configuration and select New Configuration.

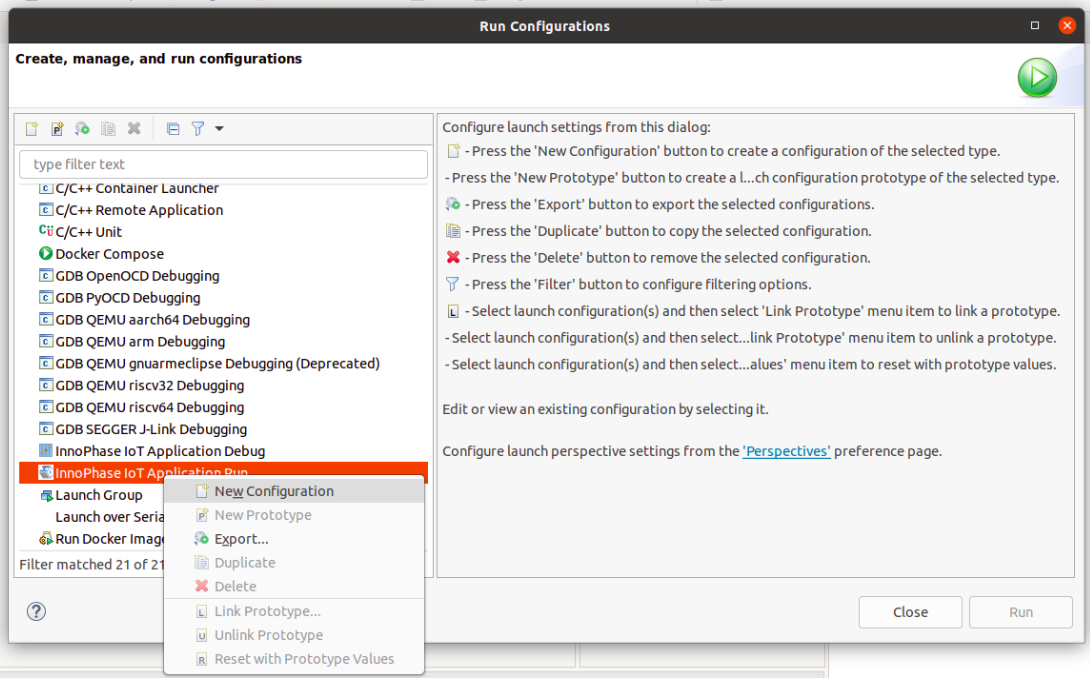


Figure 27: InnoPhase IoT Software Development Tool – New Run Configuration

1. Plugin will create a new configuration, automatically fill the required default values. Other values are required to be filled by the user as per the required the application.

A screenshot of a computer

Description automatically generated

Figure 28: InnoPhase IoT Software Development Tool – Run configuration input window

1. User inputs are required for only the Main tab. Input parameters for this tab are as follows:
   1. **Name:** Name of the run configuration. Automatically filled with default name. Can be changed as per requirement.
   2. **Project:** By default, the active project name will be selected. Use the Browser button to change the project.
   3. **C/C++ Application:** Depending on the project selected, default application file is selected as *project\_path/out/project\_name.elf*.

For example: If the project name is helloworld and the project path is ../helloworld, then the default value will be ../helloworld/out/helloworld.elf.

In case the correct file is not selected, use the Browse button to select the correct file.

* 1. **Working Directory:** Selected by default.
  2. **Talaria TWO Filesystem:** Is empty by default. Select the folder which contains files/folders to be added to the Talaria TWO filesystem.

**Note**: Only files/folders inside the selected folder will be added to the Talaria TWO filesystem. Selected folder will not be added.

1. **Wi-Fi Network Name:** Is empty by default. Add the name of the Wi-Fi API to which Talaria TWO needs to connect.
2. **Wi-Fi Network Password:** Is empty by default. Add the password of the Wi-Fi API provided in the Wi-Fi network name.
3. **Boot Arguments:** Is empty by default. Add the boot argument required by the C/C++ application selected.
4. Click Apply once the inputs are filled.
5. Click Run to build and flash the application to Talaria TWO EVB device.
   1. In case the device is not connected to the system, only the application is built. Application build output can be seen on the console window CDT Build Console.

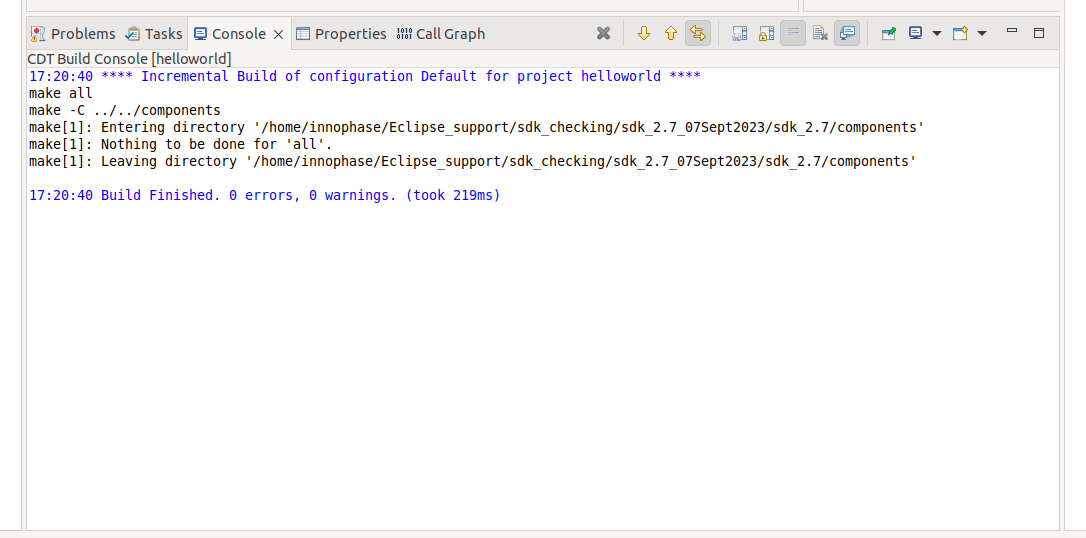


Figure 29: Console Window – CDT build console

**Note**: Use Display Selected Console to select the required console window.

* 1. In case the device is connected to the system, the application flash and device console output can be seen on the console window InnoPhase IoT T2 Device Monitor.

A screenshot of a computer

Description automatically generated

Figure 30: InnoPhase IoT T2 Device Monitor Console Window – Run configuration

**Note**: Currently, the plugin can flash to only one connected device. In case more than one device is connected, the plugin may not function as expected.

## Debug Configuration Set-up

InnoPhase IoT Software Development Tool provides debug configuration set-up to build, flash application to the connected Talaria TWO EVB and debug the application.

Steps to set-up the debug configuration settings are as follows:

1. Right click on Project and select Debug As ->Debug Configurations.

A screenshot of a computer

Description automatically generated

Figure 31: Eclipse Debug As – Select debug configuration

1. Right click on InnoPhase IoT Application Debug configuration and select New Configuration.

A screenshot of a computer

Description automatically generated

Figure 32: InnoPhase IoT Software Development Tool – New debug configuration

1. Plugin will create a new configuration and automatically fill-in the required default values for debugging. As required, the user can fill in the other values.

A screenshot of a computer

Description automatically generated

Figure 33: InnoPhase IoT Software Development Tool – Debug configuration input window

The plugin will automatically fill in the inputs for the Debugger tab. For more details on providing the inputs manually, refer *Debug Configuration Setting up in Eclipse*.

User inputs are required for the Main tab. Input parameters are as follows:

* 1. **Name:** Name of the debug configuration. Filled with default name, the user can change the name as required.
  2. **Project:** By default, the active project name is selected. Use the Browser button to change the project.
  3. **C/C++ Application:** Dependent on the project selected. Default application file is selected as *project\_path/out/project\_name.elf*.

For example: If the project name is helloworld and the project path is *..\helloworld*, then the default value is *../helloworld/out/helloworld.elf*.

In case the correct file is not selected, use the Browse button to select the correct file.

* 1. **Wi-Fi Network Name:** Is empty by default. Add the name of the Wi-Fi AP to which Talaria TWO needs to be connected to.
  2. **Wi-Fi Network Password:** Is empty by default. Add the password of the Wi-Fi AP provided in the Wi-Fi network name.
  3. **Boot Arguments:** Is empty by default.Add the boot argument required by the C/C++ application selected.

1. Click the Apply button once the inputs are filled.
2. Click Debug to build, flash the application onto Talaria TWO EVB and start debugging.
   1. In case the device is not connected to the system, only application will get build. Application build output can be seen in Console window “CDT Build Console”, refer Figure 29.
   2. In case the device is connected to the system:
      1. The application flash and device console output can be seen in the console window InnoPhase IoT T2 Device Monitor, refer Figure 30.
      2. Once the application is flashed successfully onto Talaria TWO EVB, Eclipse may prompt to switch to Debug Perspective. Press Switch to continue debugging.

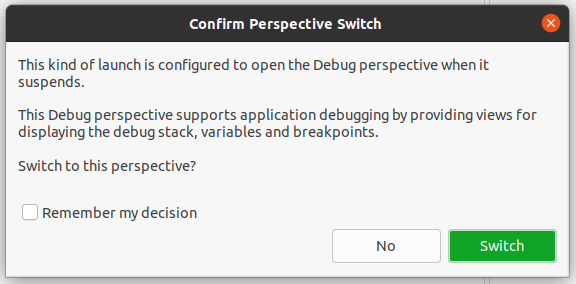


Figure 34: InnoPhase IoT Software Development Tool – Debug perspective switch

* 1. Eclipse will begin to debug and the same is displayed in Debug Perspective as shown in Figure 35.

A screenshot of a computer

Description automatically generated

Figure 35: InnoPhase IoT Software Development Tool – Debug start

* + - 1. Debug the application with Eclipse’s general debugging procedure (refer point 4 to 7 of section *Debugging in Eclipse*).
      2. While debugging the application, the application’s prints will be visible on the console window InnoPhase IoT T2 Device Monitor as shown in Figure 30.