

Application Note

Compiling DA14585 IoT Sensors Reference Application with Eclipse/GCC

AN-B-064

Abstract

The DA14585 IoT Sensors Reference Application can be compiled using the ARM GCC compiler and Dialog Semiconductor provides an example of Eclipse project to how to accomplish this. This document describes the required steps to download and compile this project.

Contents

Abstract	1
Contents	2
Figures.....	2
1 Terms and Definitions.....	3
2 References	3
3 Introduction.....	4
4 Installation.....	4
4.1 Prerequisites	4
4.2 SmartSnippets™ Installation.....	4
4.3 Download the DA14585 IoT MSK Eclipse/GCC Example	7
5 Importing the Eclipse/GCC Project	8
6 Building the Project	18
7 Debugging.....	20
8 Creating the Multi-Image	22
Revision History	23

Figures

Figure 1: Download SmartSnippets™ Studio	5
Figure 2: Use Pre-Existing J-Link.....	5
Figure 3: Select SmartSnippets™ Studio Installation Directory	6
Figure 4: Downloading Eclipse/GCC Example.....	7
Figure 5: Starting SmartSnippets™ Studio and Selecting Workspace.....	8
Figure 6: Selecting Workspace Type	9
Figure 7: Tools to Be Installed.....	9
Figure 8: Set SEGGER Ozone & J-Link Installation Folder	10
Figure 9: GCC Tools Installation	11
Figure 10: Starting the IDE	12
Figure 11: Eclipse CDT IDE without project.....	13
Figure 12: Insert Existing Project into Workspace	14
Figure 13: Select Project Root Directory	15
Figure 14: Selecting the iot585 Project	16
Figure 15: iot585 Project Imported	17
Figure 16: Build the Project	18
Figure 17: Building Finished	19
Figure 18: Start Debugging	20
Figure 19: Debug Perspective.....	21

1 Terms and Definitions

BLE	Bluetooth Low Energy
CDT	C/C++ Development Tooling
GAP	Generic Access Profile
GAPC	Generic Access Profile Controller
GAPM	Generic Access Profile Manager
GATT	Generic Attribute Profile
IDE	Integrated Development Environment
IoT	Internet of Things
MSK	Multi Sensor Development Kit
PC	Personal Computer
RAM	Random Access Memory
SOC	System on Chip

2 References

- [1] UM-B-096, DA14585 IoT Multi Sensor Development Kit Software Reference Applications, User Manual, Dialog Semiconductor.
- [2] UM-B-095, DA14585 IoT Multi Sensor Development Kit Hardware Design, User Manual, Dialog Semiconductor.
- [3] UM-B-057, Smart Snippets™ Studio User Guide, User Manual, Dialog Semiconductor.

3 Introduction

Dialog Semiconductor's IoT multi sensor development kit (MSK) [1], a compact development kit, offers 15 degrees of freedom accompanied by five software reference applications [2] and incorporates versatile sensor management with cloud applications. The IoT MSK is based on DA14585 Bluetooth Low Energy (BLE) system-on-chip (SoC) and a number of motion and environmental sensors.

The provided reference applications can be compiled with Keil μ Vision and ARMCC toolchain which is the recommended development environment. Many developers would like to use the free GCC compiler/linker and Dialog has provided an example Eclipse/GCC project port of the IoT Sensors Reference Application. For more details on this application refer to *section 5* of [1].

This document describes how to setup the environment, download, compile, and debug the IoT Sensors Reference Application using Eclipse/GCC. It is required that users should already be familiar with Eclipse CDT environment.

4 Installation

4.1 Prerequisites

- A Personal Computer running Windows 7 or Windows 10.
- SmartSnippets Studio v2.0.5.
- A DA14585 IoT MSK.
- A Communication Interface Board.

4.2 SmartSnippets™ Installation

Dialog's SmartSnippets™ Studio is a royalty-free software development platform for Smartbond™ devices. For detailed information on SmartSnippets™ Studio refer to [3].

SmartSnippets™ Studio contains:

- SmartSnippets™ IDE: Eclipse CDT based IDE with pre-configured plugins to provide the build/debug environment
- SmartSnippets™ Toolbox which covers all software development requirements, including:
 - programming and loading firmware into SRAM, OTP, and Flash
 - power profiling
- SmartSnippets™ documentation.

The SmartSnippets™ IDE is enabled by an on-board J-Link debugger from SEGGER. This offers standard debug capabilities such as single stepping, setting breakpoints, software download, and many more. For more details on the debugger capabilities, visit <https://www.segger.com>.

The installation procedure for SmartSnippets™ Studio is described in [3]. A summary of the steps is given here:

1. Navigate to Dialog's support portal <https://support.dialog-semiconductor.com> and choose **Connectivity > Products > DA14585 > Software&Tools > Tools** (Figure 1) and download the latest 2.0.x version of SmartSnippets™ Studio.
2. Run the SmartSnippets™ Studio installer (.msi). Several required tools are automatically installed and others need to be manually downloaded and installed.

Compiling DA14585 IoT Sensors Reference Application with Eclipse/GCC

Company Confidential

3. Select the already installed version of SEGGER J-Link GDB server or install the latest version (Figure 2) and click "Next".
4. Select the destination folder (Figure 3) for the SmartSnippets™ Studio and click "Next". The default installation location C:\DiaSemi is recommended.

The SmartSnippets™ Studio will be installed. Please Note that some software components during installation may require administrator privileges.

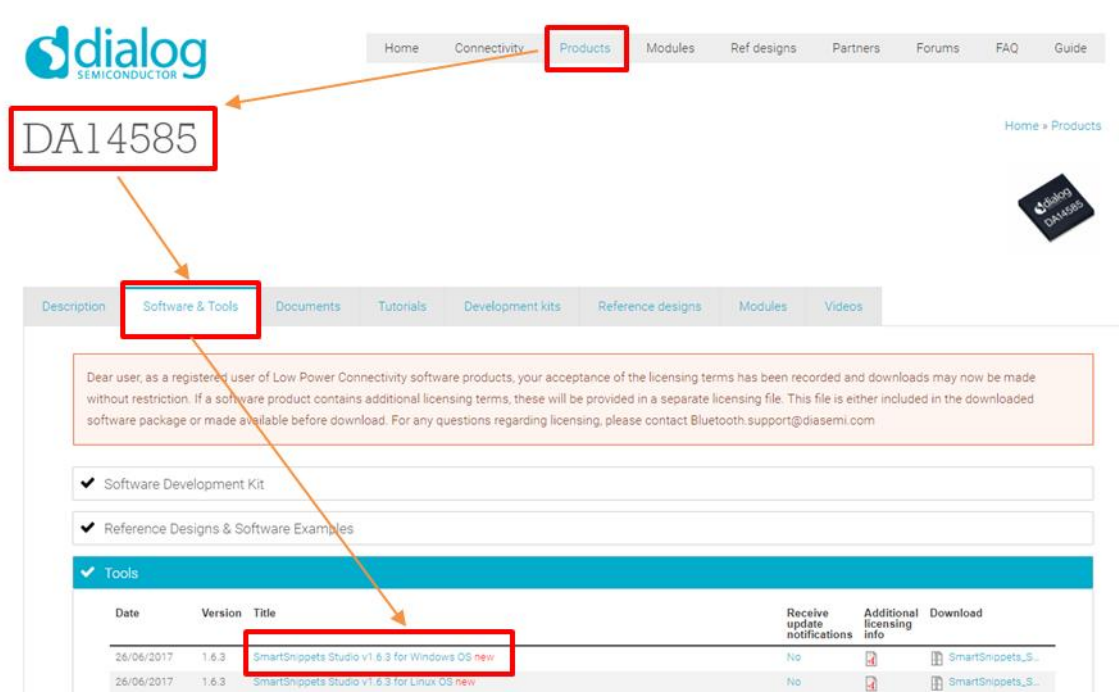


Figure 1: Download SmartSnippets™ Studio

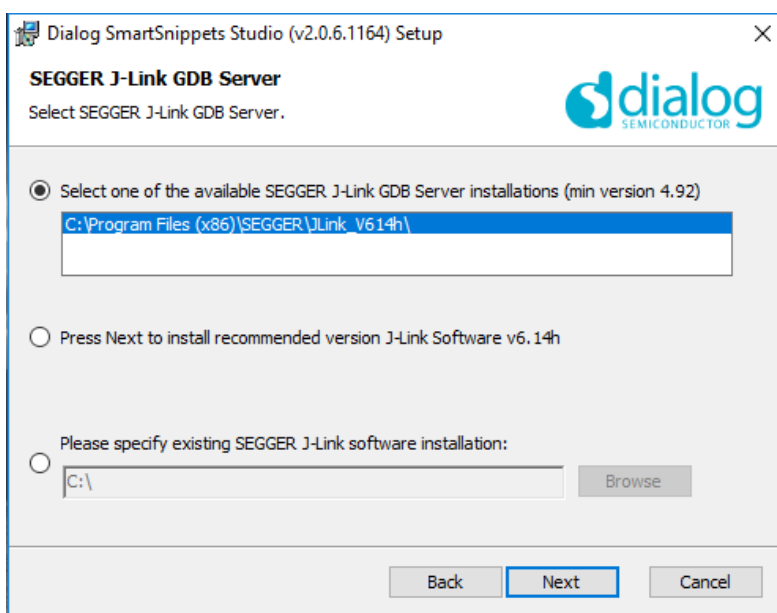


Figure 2: Use Pre-Existing J-Link

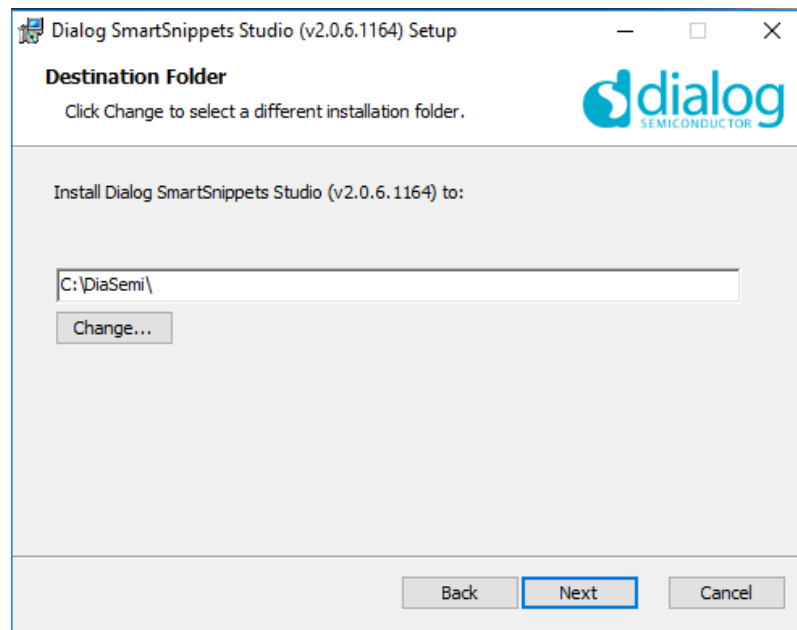


Figure 3: Select SmartSnippets™ Studio Installation Directory

4.3 Download the DA14585 IoT MSK Eclipse/GCC Example

Like [SmartSnippets™ Studio](#), the Eclipse/GCC example can be downloaded from **Connectivity > Products > DA14585 > Reference Designs > DA14585 IoT Multi Sensor Development Kit** (Figure 4) in the support portal.

The filename of the example is `DA14585_IOT_MULTI_SENSOR_DK_GCC_EXAMPLE_v6.160.x.zip` and includes only the GCC port of the software. The release folder `v6.160.x` included in the zipped file should be extracted by users and placed in a convenient path.

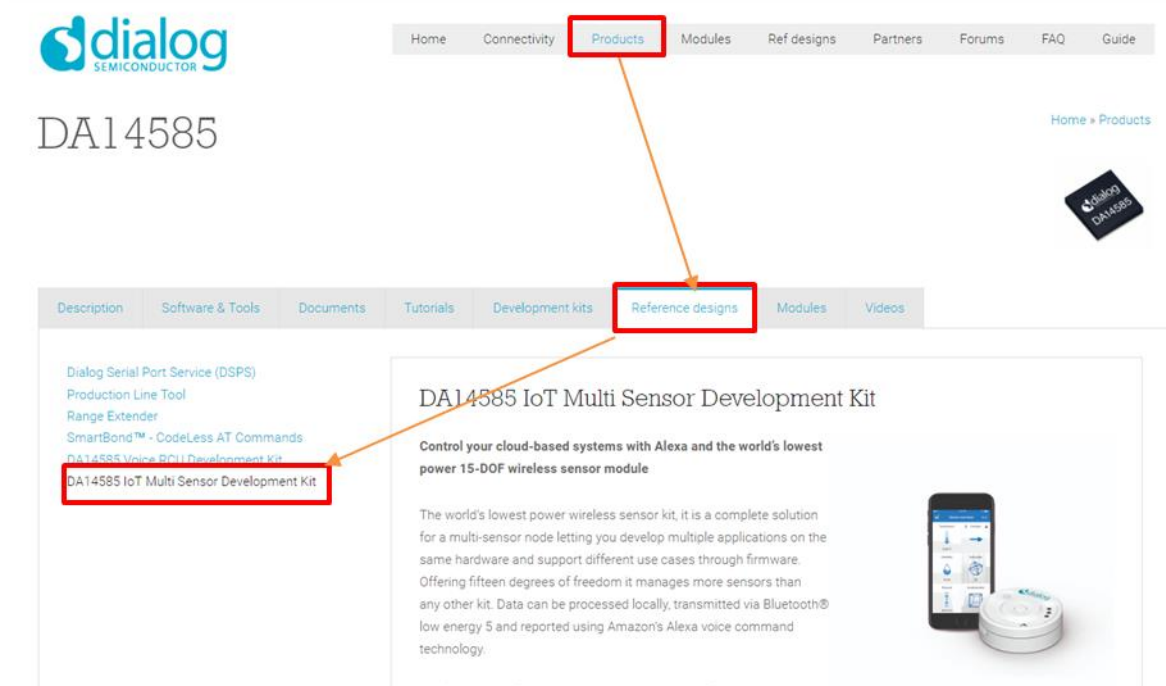


Figure 4: Downloading Eclipse/GCC Example

5 Importing the Eclipse/GCC Project

1. Start SmartSnippets™ Studio by clicking on the icon.

The SmartSnippets™ Studio will ask the workspace directory. Select the root directory of the project that is named v6.160.x (for example, v6.160.4.15, [Figure 5](#)).

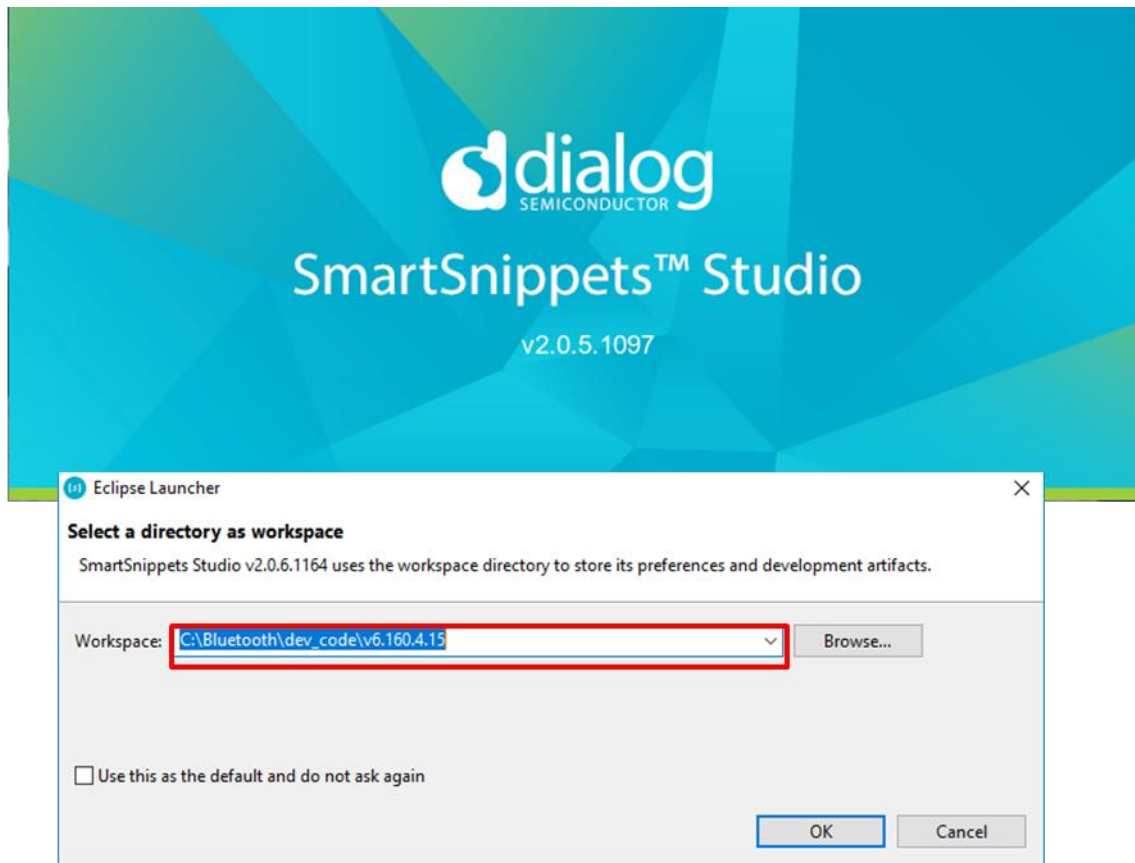


Figure 5: Starting SmartSnippets™ Studio and Selecting Workspace

2. Next, when the dialog asking for a workspace type pops up, select **"Take no action"**.

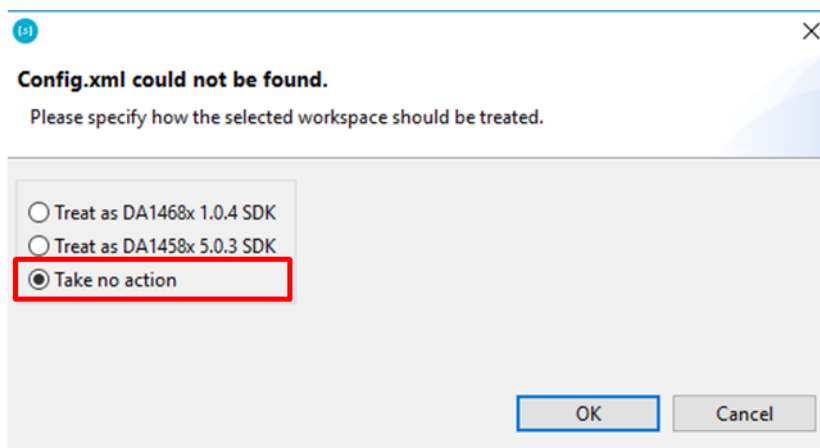


Figure 6: Selecting Workspace Type

3. To proceed, a number of development tools should be installed (Figure 7), or their installation directory should be given (Figure 8) if they are already installed.

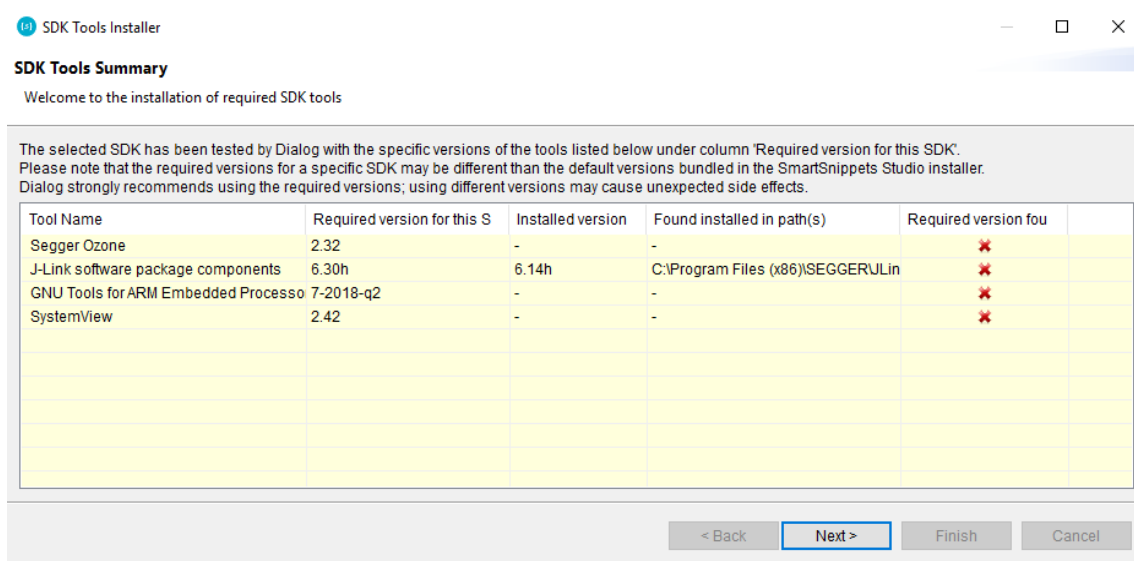


Figure 7: Tools to Be Installed

Compiling DA14585 IoT Sensors Reference Application with Eclipse/GCC

Company Confidential

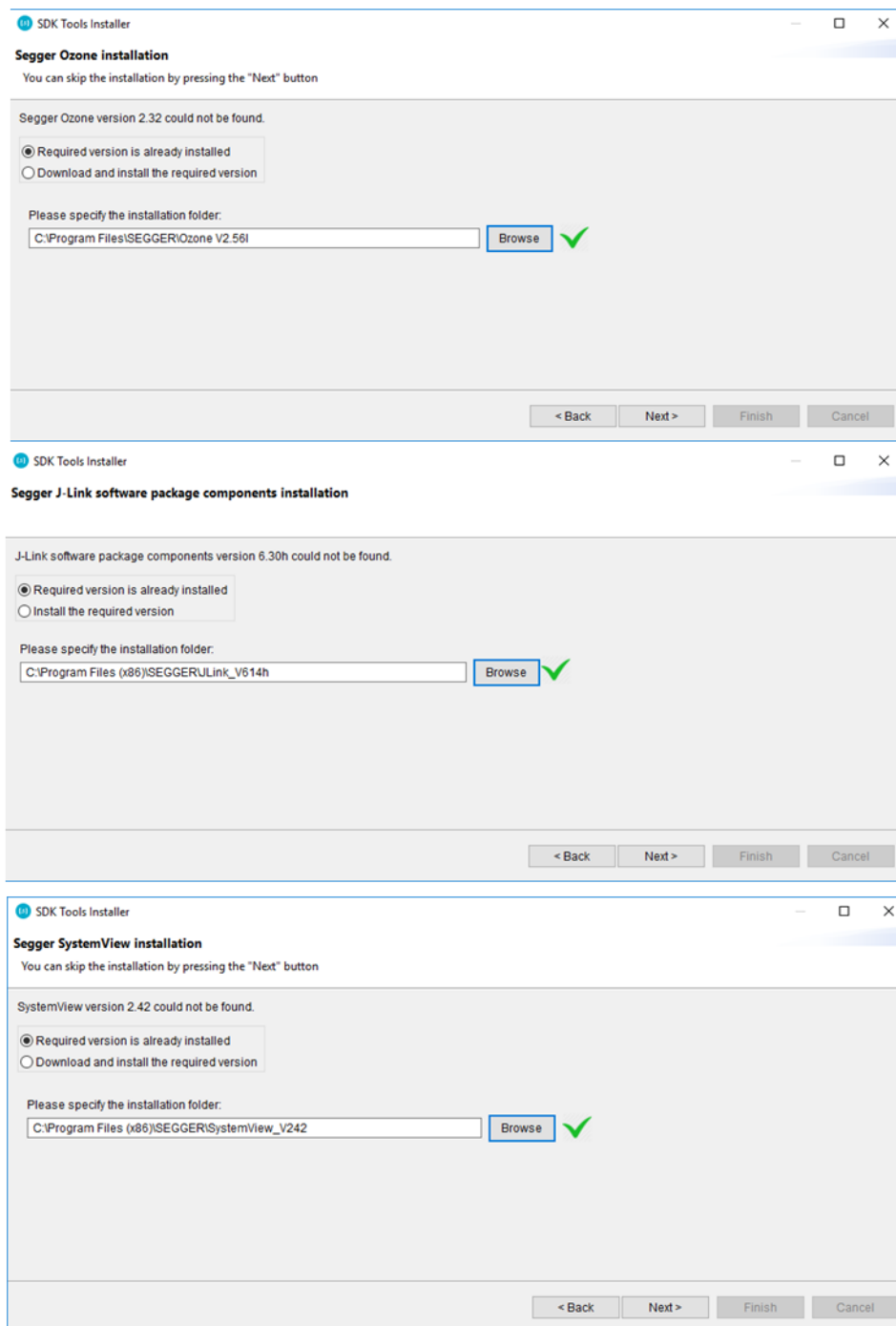


Figure 8: Set SEGGER Ozone & J-Link Installation Folder

Compiling DA14585 IoT Sensors Reference Application with Eclipse/GCC

Company Confidential

4. The **SDK Tools Installer** will prompt for the installation of GCC and the default version 7-2018-q2 is recommended to download. Select **Download and installed the required version** and click **Download** (Figure 9). After the download is complete the green completion mark is shown and the **Next** may be pressed to continue (Figure 9).

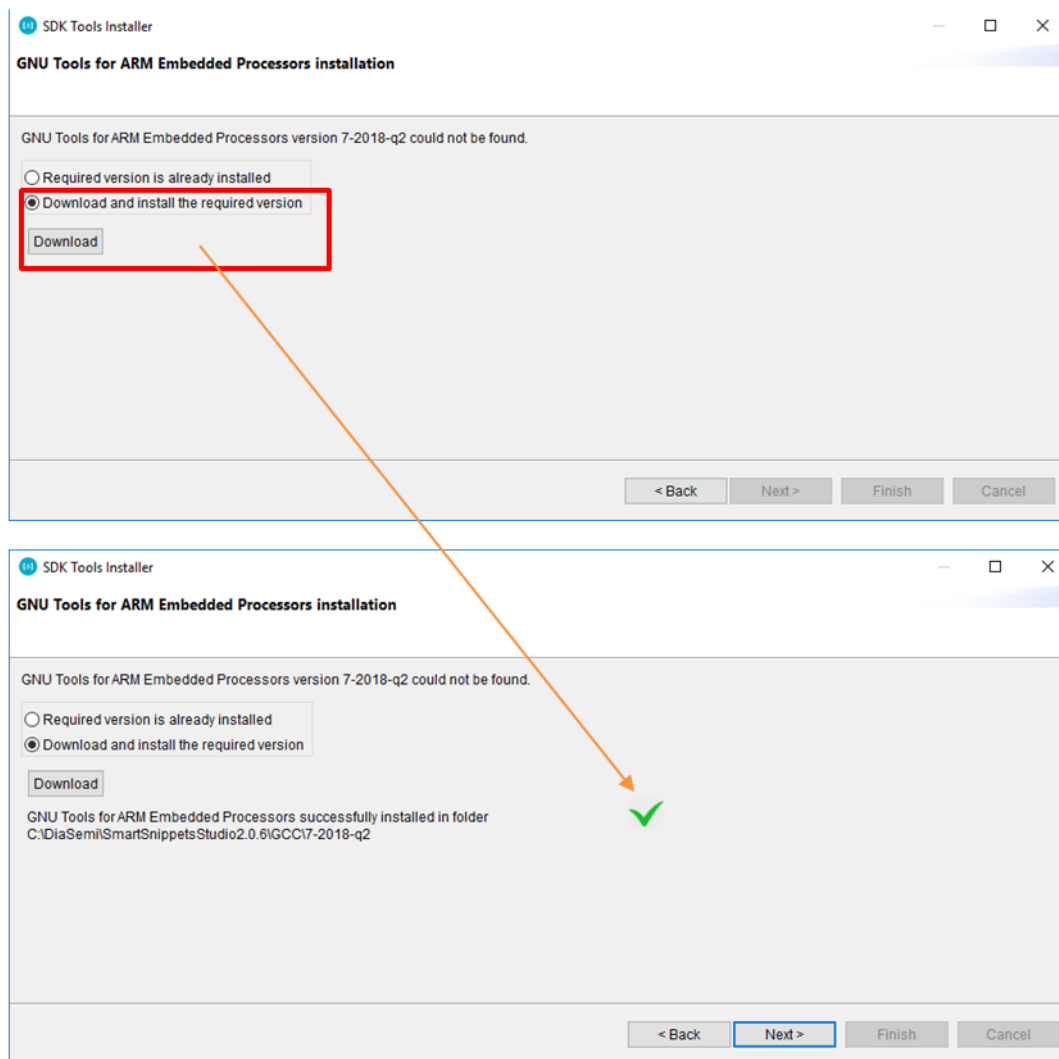


Figure 9: GCC Tools Installation

Compiling DA14585 IoT Sensors Reference Application with Eclipse/GCC

Company Confidential

5. The SmartSnippets™ Studio environment will now appear. Users should click the IDE icon to start the Eclipse CDT IDE (Figure 10).

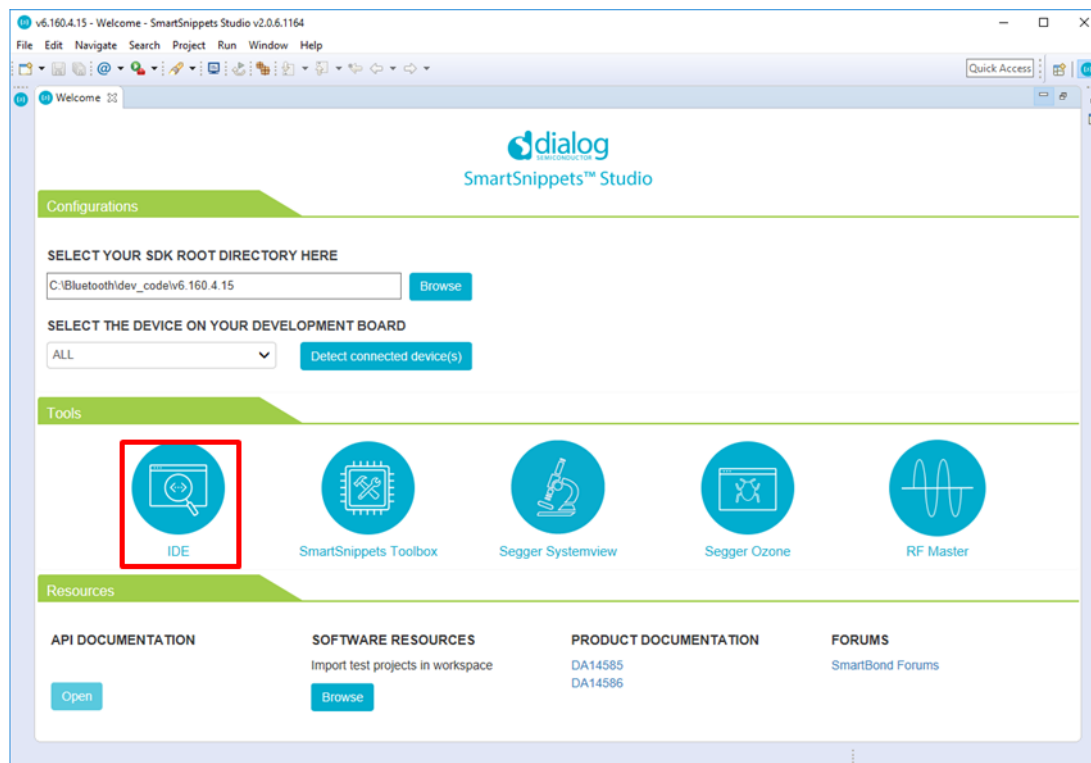


Figure 10: Starting the IDE

Compiling DA14585 IoT Sensors Reference Application with Eclipse/GCC

Company Confidential

6. Now the Eclipse CDT IDE will open ([Figure 11](#)). In the left-side window there are no projects listed, so the next step is to insert the project example.

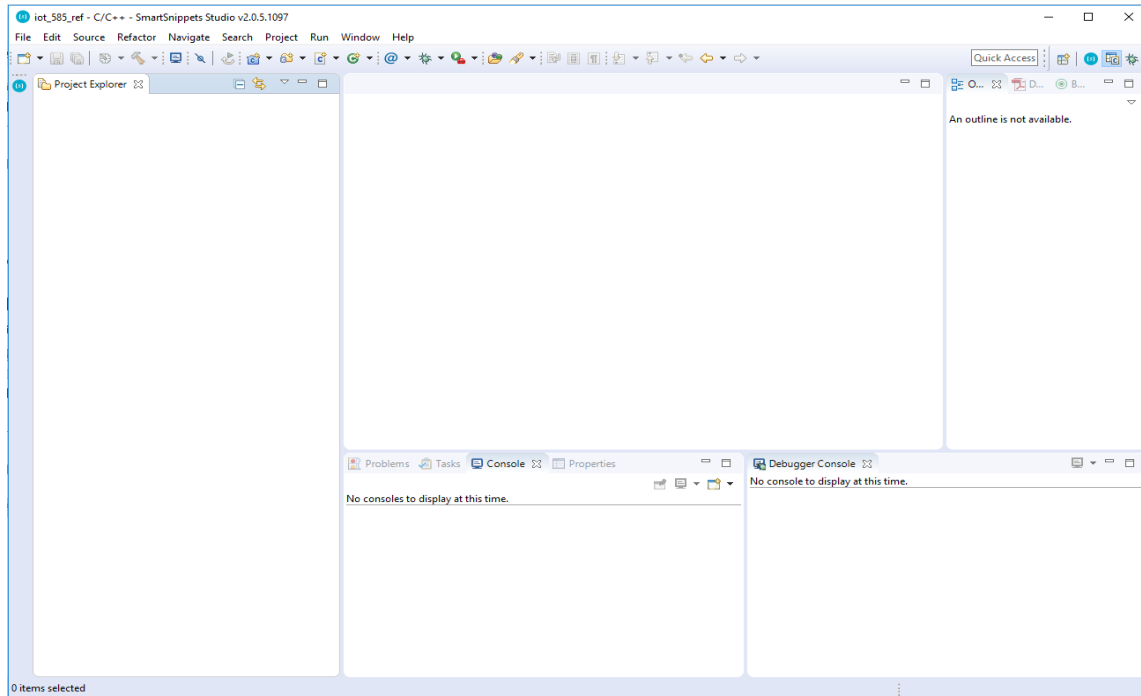


Figure 11: Eclipse CDT IDE without project

Compiling DA14585 IoT Sensors Reference Application with Eclipse/GCC

Company Confidential

7. To insert an existing project, click **File > Import**. The Import window now will open and users should select "**Existing Projects into Workspace**" from the root tree "**General**" (Figure 12) and then click "**Next**".

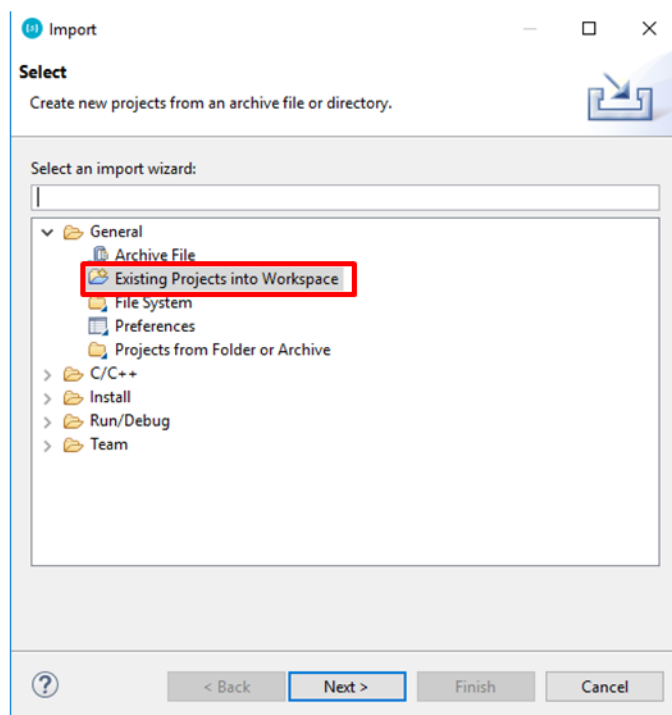


Figure 12: Insert Existing Project into Workspace

8. A window will appear and users should choose "**Browse**" next to "**Select root directory**", then select the root directory of v6.160.4.15 (Figure 13), and press **OK**.

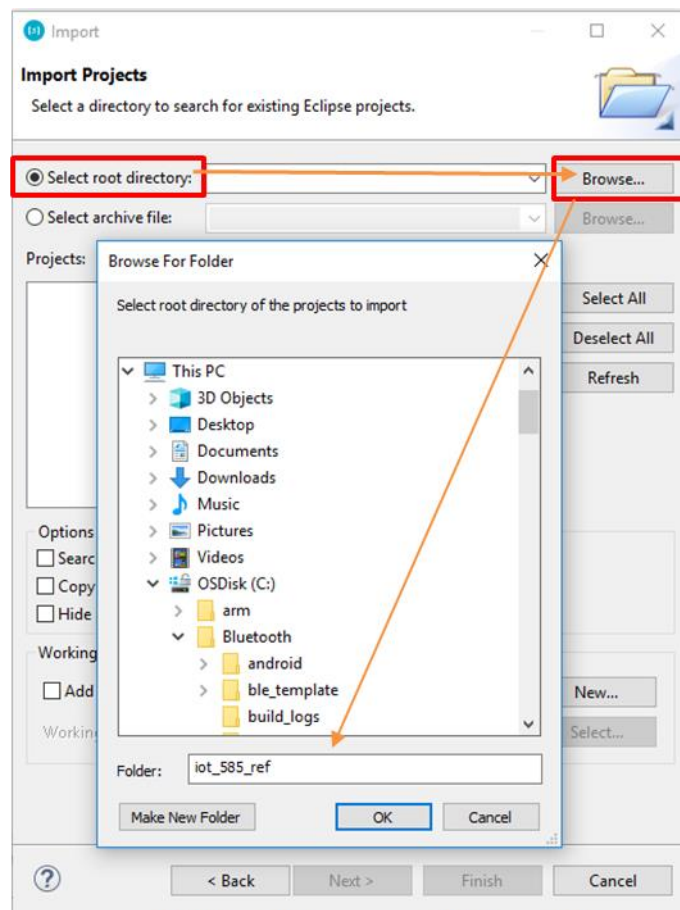


Figure 13: Select Project Root Directory

9. A list of projects compatible with Eclipse will appear, and users should only select the **iot585** project by ticking the selection box (Figure 14) and then press "**Finish**".

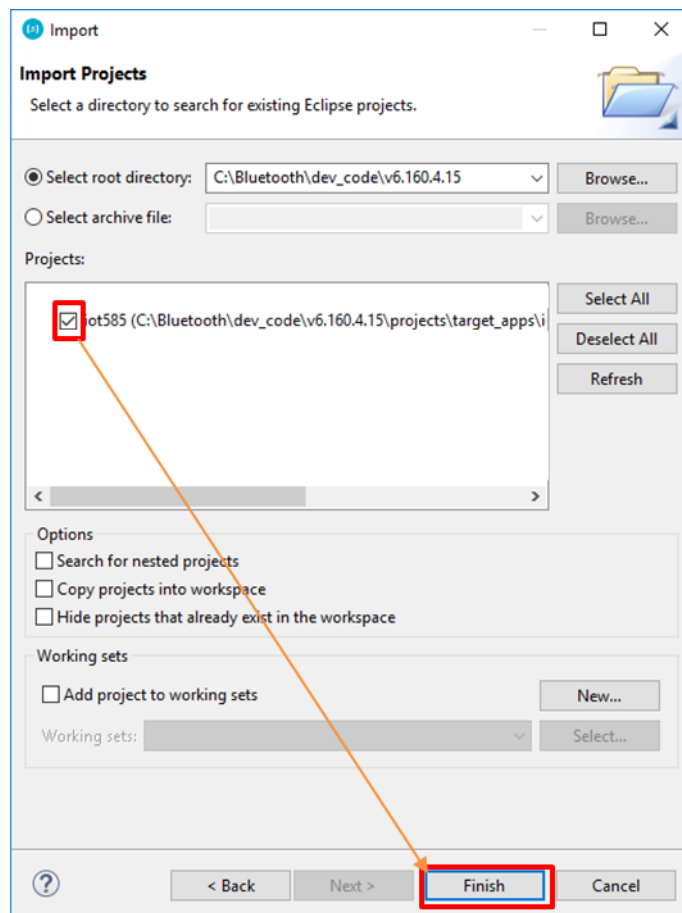


Figure 14: Selecting the iot585 Project

Compiling DA14585 IoT Sensors Reference Application with Eclipse/GCC

Company Confidential

10. After Eclipse imports the project, the **iot585** project will appear in the Project Explorer window (Figure 15).

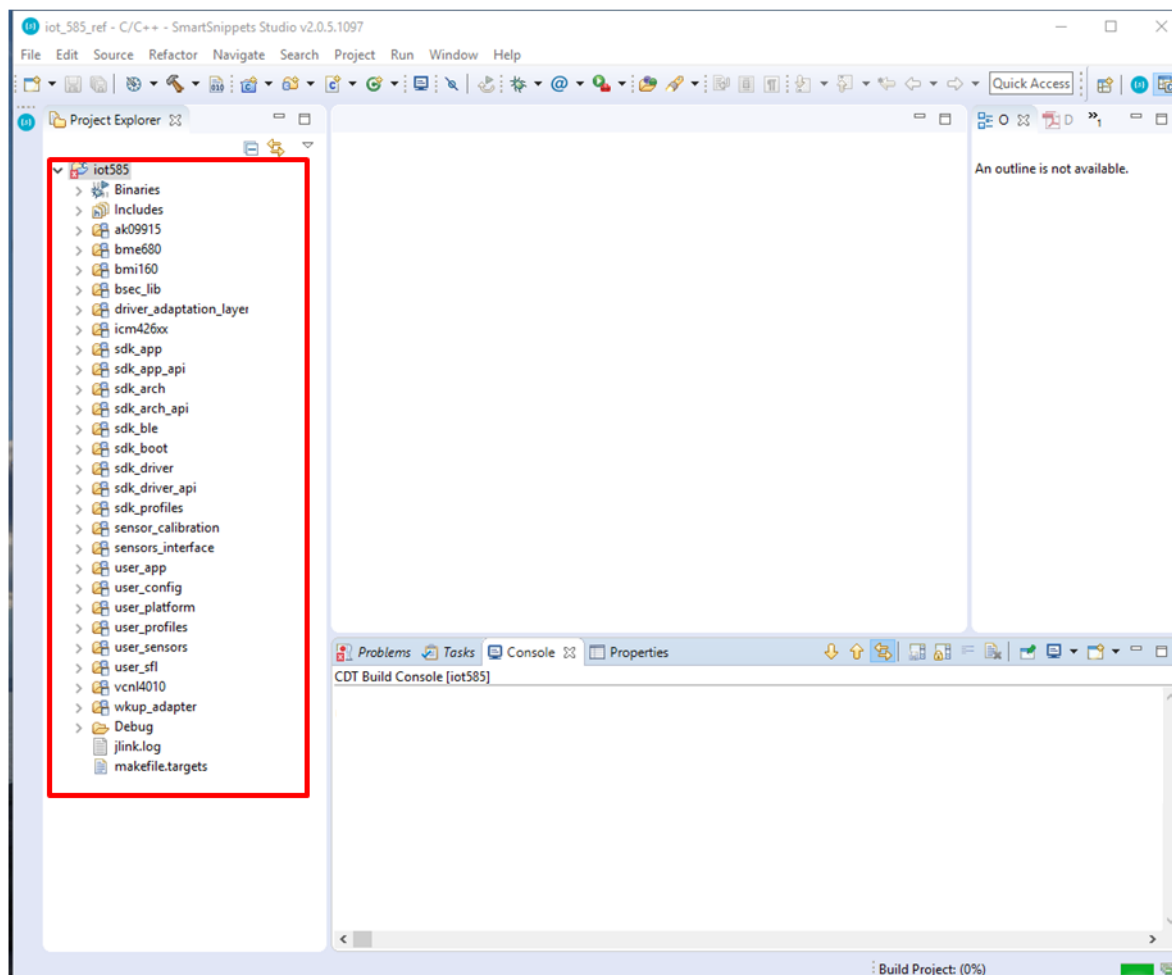


Figure 15: iot585 Project Imported

6 Building the Project

After the abovementioned steps, an existing project is imported into Eclipse CDT environment but it is not built. To build the project, click the "build" icon on the menu bar (Figure 16). The progress of building the project is shown in the bottom console window.

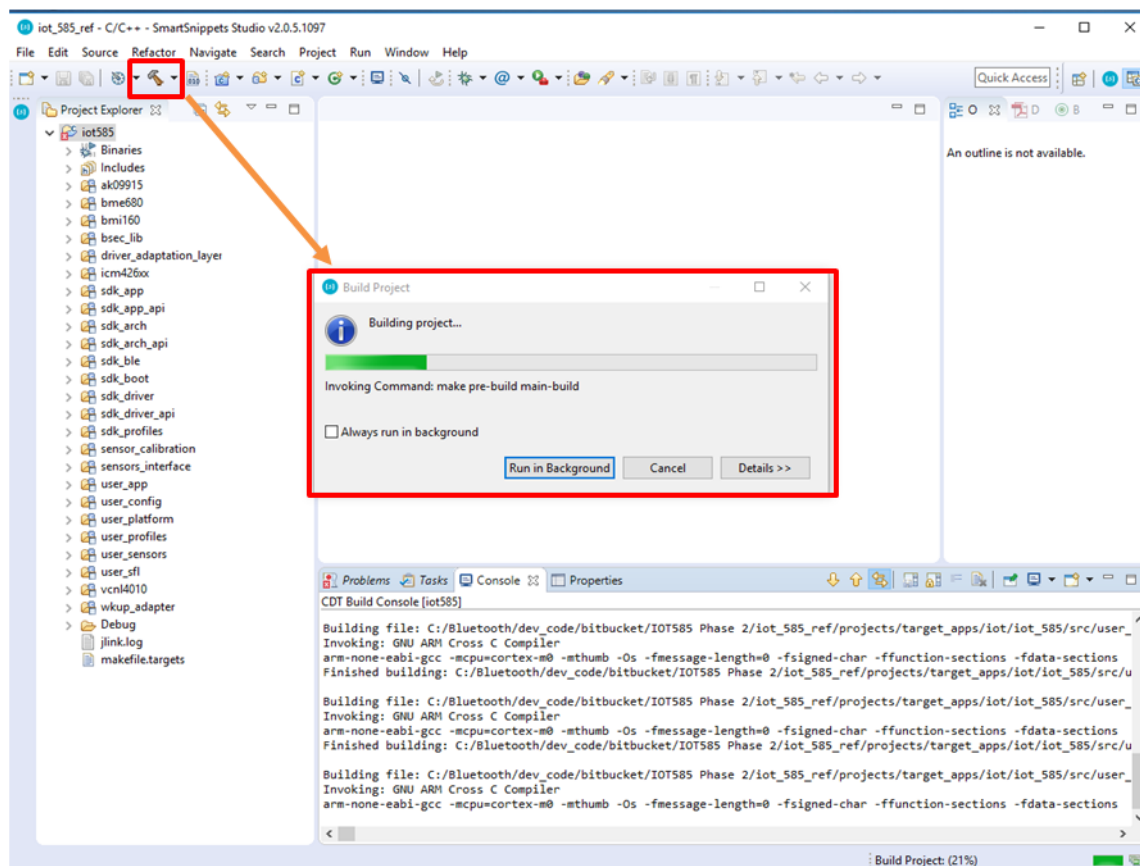


Figure 16: Build the Project

Compiling DA14585 IoT Sensors Reference Application with Eclipse/GCC

Company Confidential

At the end of the building procedure (Figure 17), output files `iot585.elf` and `iot585.hex` are generated. These files are located inside the folder `projects\target_apps\iot\iot_585\Eclipse\Debug`. The `iot585.elf` is the linker output and can be used for debugging. The `iot585.hex` is the stripped binary and will be used to create the multi-image file that can be burned directly into flash.

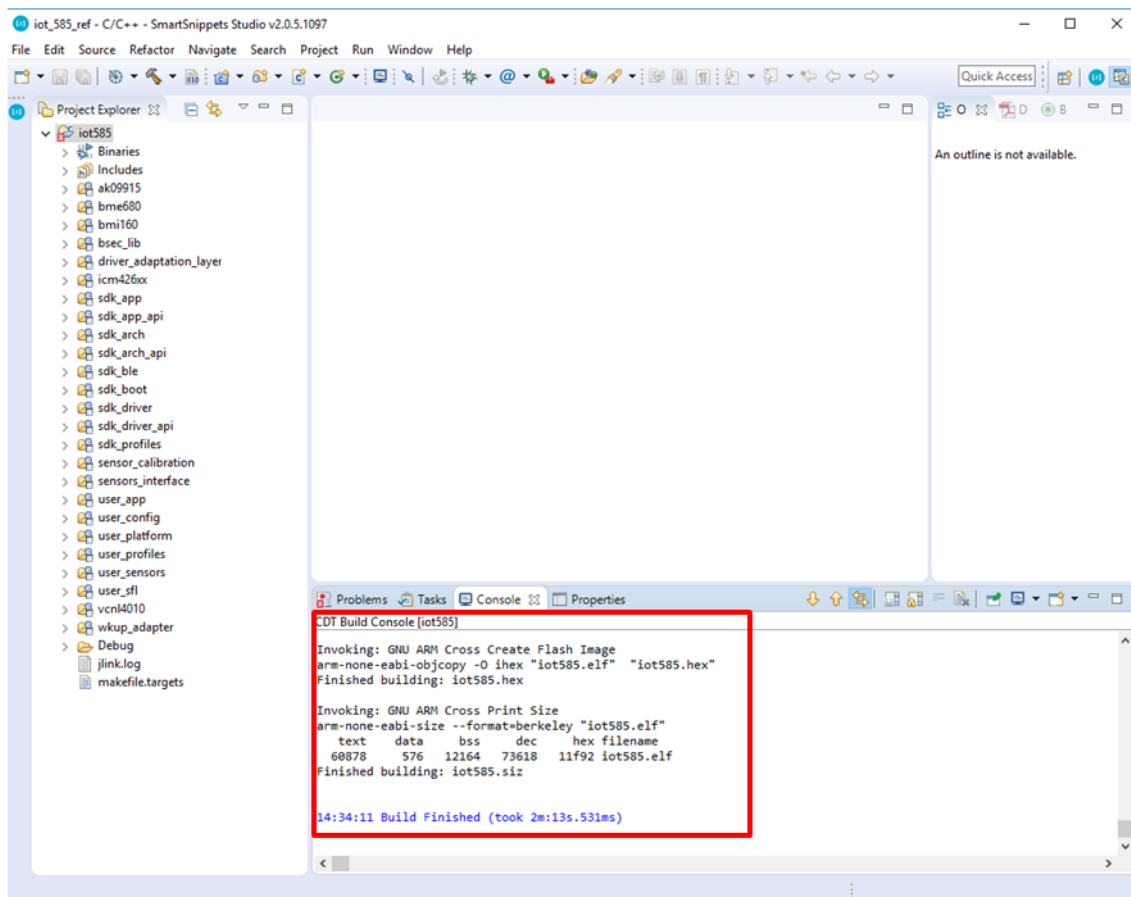


Figure 17: Building Finished

7 Debugging

For the purpose of debugging, click the **Debug** icon and then select the **RAM_DA14585** debug configuration (Figure 18).

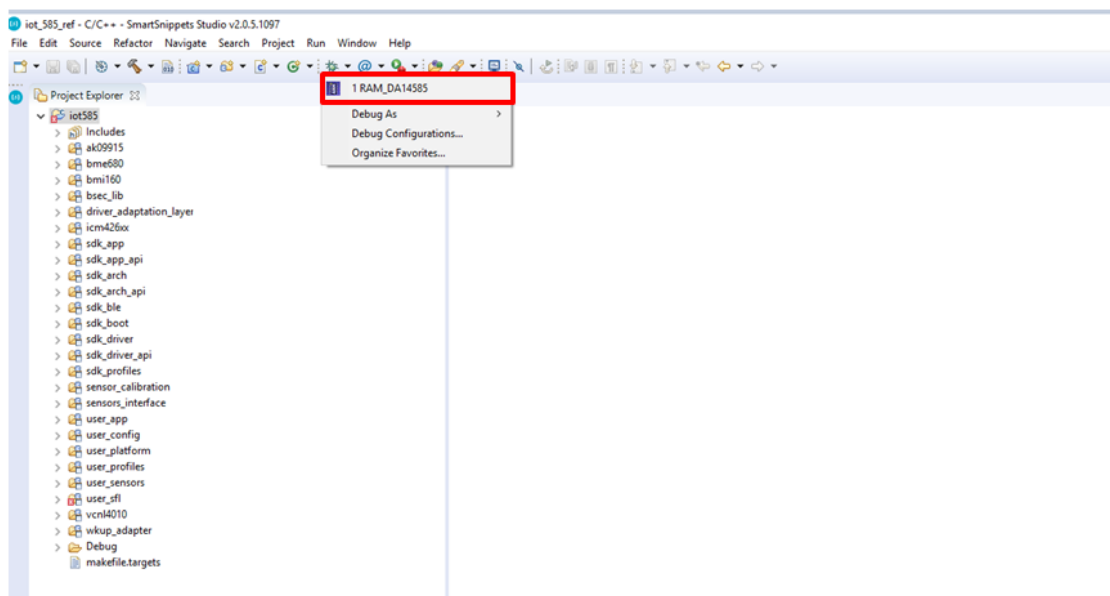


Figure 18: Start Debugging

Compiling DA14585 IoT Sensors Reference Application with Eclipse/GCC

Company Confidential

Accept it when Eclipse asks to open the Debug perspective and then the Debug view will open. Press the **Resume** button to start running the program (Figure 19).

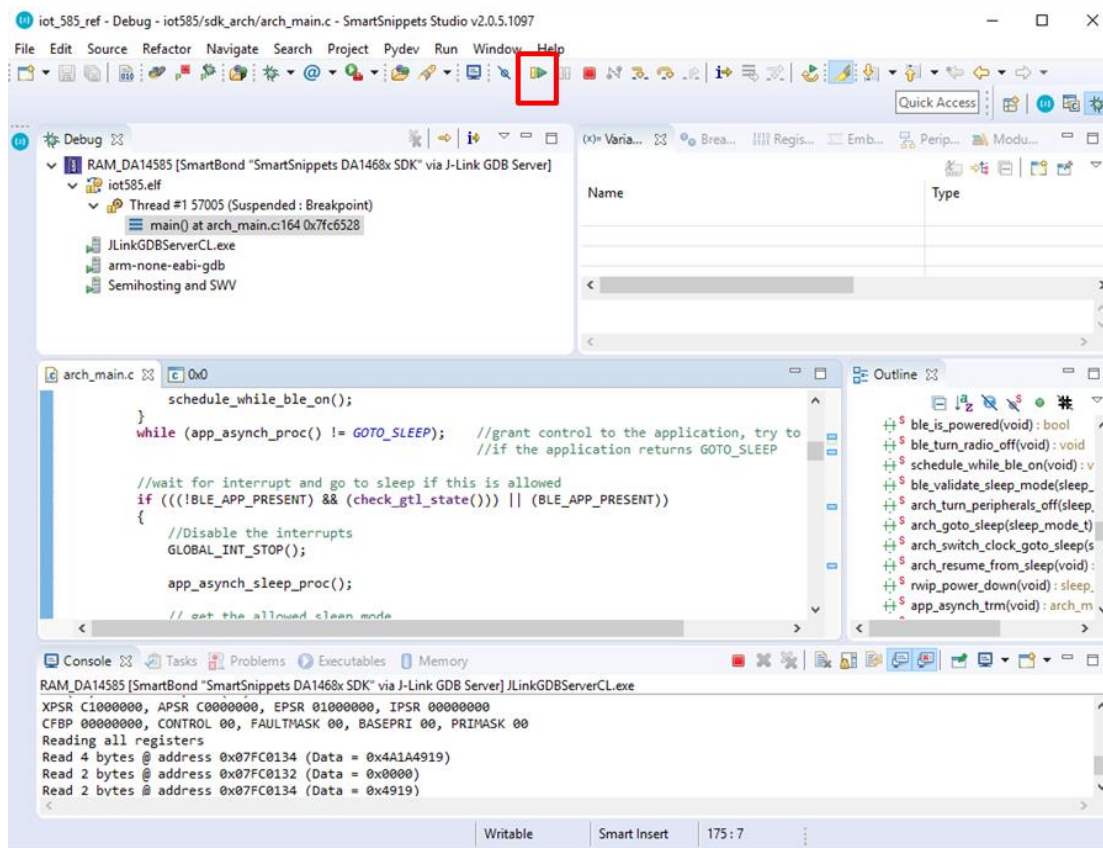


Figure 19: Debug Perspective

8 Creating the Multi-Image

Detailed information on the provided scripts is included in *Appendix C* of [1]. Users should use the script `make_image_iot.bat` that is placed in “`..\utilities\mkimage_utils_scripts`”. The `.hex` extension is added automatically and the command should be “`make_image_iot.bat io585`”. The output file `multi_iot585.bin` can be used by the Flash Programmer of [SmartSnippets™ Toolbox](#). Detailed instructions of how to burn the multi-image `multi_iot585.bin` using [SmartSnippets™ Toolbox](#) are included in *Appendix D.2* of [1].

IMPORTANT NOTE

Because a secondary bootloader is used (*Appendix C* of [1]) that selects the most recent/valid image between two images, `io585.hex` cannot be burnt directly into flash, but the multi-image should be created instead. This functionality is necessary to support firmware updates over the air (SUOTA).

Revision History

Revision	Date	Description
1.1	17-Jan-2022	Updated logo, disclaimer, copyright.
1.0	10-Oct-2018	Initial version.

Compiling DA14585 IoT Sensors Reference Application with Eclipse/GCC

Company Confidential

Status Definitions

Status	Definition
DRAFT	The content of this document is under review and subject to formal approval, which may result in modifications or additions.
APPROVED or unmarked	The content of this document has been approved for publication.