Abstract

The main focus of this tutorial is to describe how to setup the environment, download, compile, and debug the DA14585 IoT MSK Sensors Reference Application using Eclipse/GCC
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 Downloading Eclipse/GCC Example ......................................................................................... 7
5 Importing the Eclipse/GCC Project .............................................................................................. 7
6 Building the Project ..................................................................................................................... 15
7 Debugging ..................................................................................................................................... 17
8 Troubleshooting .......................................................................................................................... 19
9 Creating the Multi-Image ............................................................................................................. 21
10 Code size estimation .................................................................................................................. 21
Revision History ............................................................................................................................... 23

Figures

Figure 1: Download SmartSnippets™ Studio .................................................................................. 5
Figure 2: Install SEGGER J-Link GDB Server ................................................................................ 6
Figure 3: Select SmartSnippets™ Studio Installation Directory ...................................................... 6
Figure 4: Downloading Eclipse/GCC Example ................................................................................. 7
Figure 5: Starting SmartSnippetsTM Studio and Selecting Workspace ........................................... 8
Figure 6: Selecting Workspace Type ............................................................................................... 8
Figure 7: Tools to Be Installed ........................................................................................................ 9
Figure 8: Set SEGGER Ozone & J-Link Installation Folder ............................................................... 9
Figure 9: GCC Tools Installation .................................................................................................... 10
Figure 10: SEGGER SystemView Tool Installation ....................................................................... 11
Figure 11: Tools installation summary ............................................................................................. 11
Figure 12: Starting the IDE ............................................................................................................. 12
Figure 13: Eclipse CDT IDE without Projects ................................................................................ 12
Figure 14: Insert Existing Project into Workspace ......................................................................... 13
Figure 15: Select Project Root Directory ....................................................................................... 13
Figure 16: Selecting the iot585 Project .......................................................................................... 14
Figure 17: iot585 Project Imported ................................................................................................. 15
Figure 18: Build the Project ............................................................................................................ 16
Figure 19: Building Finished ........................................................................................................ 16
Figure 20: Connection between DA14585 IoT MSK and the CIB .................................................... 17
Figure 21: Start Debugging ........................................................................................................... 18
Figure 22: Debug Perspective ........................................................................................................ 19
Figure 23: Launch Debug Configuration ......................................................................................... 20
Figure 24: Import Launch configuration .......................................................................................... 20
Figure 25: Project optimization setting ........................................................................................... 21
Figure 26: GCC console view .......................................................................................................... 22
Terms and Definitions

BLE  Bluetooth Low Energy
CDT  C/C++ Development Tooling
CIB  Communication Interface Board
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
OTP  One Time Programmable (Non Volatile memory)
PC   Personal Computer
SRAM Static Random Access Memory
SOC  System on Chip

1 References

Compiling DA14585 IoT Sensors Reference Application with Eclipse/GCC

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

The provided reference applications can be compiled with Keil μVision and ARMCC toolchain which is the recommended development environment. Many developers would like to use the free GCC compiler/linker and Dialog provides an example Eclipse/GCC project port of the IoT Sensors Reference Application. For more details on this application refer to section 4 of [1].

This document describes how to setup the environment, download, compile, and debug the DA14585 IoT MSK Sensors Reference Application using Eclipse/GCC. It is required that users should already be familiar with Eclipse C/C++ development tooling (CDT) environment.

3 Installation

3.1 Prerequisites
● A personal computer (PC) running Windows 7 or Windows 8/10.
● SmartSnippets Studio v2.0.6.
● A DA14585 IoT MSK.
● A communication interface board (CIB).

3.2 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]. For more information refer to [1] and [2].

SmartSnippets™ Studio contains:
● SmartSnippets™ IDE: Eclipse CDT based integrated development environment (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.
Figure 1: Download SmartSnippets™ Studio

2. Run the SmartSnippets™ Studio installer (.msi). Several required tools are automatically installed, and others need to be manually downloaded and installed.

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.

Figure 3: Select SmartSnippets™ Studio Installation Directory

The SmartSnippets™ Studio will be installed. Please note that some software components during installation may require administrator privileges.
3.3 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_IOTP_v6.160.x.yy\DA14585_IOTP\v6.160.x.yy.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.

4 Importing the Eclipse/GCC Project

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

The SmartSnippets™ Studio will ask for the workspace directory. Select the root directory of the project that has been previously downloaded in section 3.3 and named v6.160.x (for example, v6.160.3.25, Figure 5).
2. Next, when the dialog asking for a workspace type pops up, select "Take no action". This option should be selected only if you know what you are doing. In all other cases users are advised to properly export SDK repository under the selected Workspace path.

3. Figure 7 shows the first page of the wizard, which presents a table of the tools required by this SDK and their versions as well as the installed necessary tools. If the required version of a tool is not matched or not found at all, the wizard will prompt for tool installation. For the selected SDK, installer will guide users to install "Segger JLink", "Segger Ozone", and "Segger SystemView".

4. For tools that are specified as optional, users can always skip those by selecting the "Skip" button and eventually continue with the rest of the installation process. Users are recommended to install all optional tools and then press the "Next" button to continue with the installation dialog, as indicated in Figure 8.
5. Users are required to download and install tools marked as "mandatory" by clicking the "download" button or specifying an installation folder if the tool is already installed externally. The
"Next" button is enabled only after the mandatory tools are successfully installed or a valid installation path is specified.

6. 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 install the required version” and click “Download”. After the download is complete, the green completion mark appears. Then press the “Next” button to continue (Figure 9).

![GCC Tools Installation](image)

**Figure 9: GCC Tools Installation**

7. The SDK Tools Installer will prompt for the installation of SEGGER SystemView (Figure 10).
8. The last page of the wizard shows a summary of all installed tools. Installation of optional tools can be skipped in the wizard. The user’s choice to skip an optional tool is remembered by SmartSnippets™ Studio so that the wizard does not ask again for the installation of this tool the next time the wizard is launched.

9. The SmartSnippets™ Studio environment will now appear. Users should click the IDE icon to switch to the IDE workbench in the C/C++ perspective (Figure 12).
10. Now the Eclipse CDT IDE will open (Figure 13). In the left-side window there are no projects listed, so the next step is to insert the project example.
11. To insert an existing project, click **File > Import**. The Import window now opens, and users should select "**Existing Projects into Workspace**" from the root tree "**General**" (Figure 14) and then click "**Next**".

![Insert Existing Project into Workspace](image)

**Figure 14: Insert Existing Project into Workspace**

12. A window appears, and users should choose "**Browse**" next to "**Select root directory**", then select the root directory of v6.160.4.15 (Figure 15), and press **OK**.
13. A list of projects compatible with Eclipse appears, and users should only select the \texttt{iot585} project by ticking the selection box (Figure 16) and then press \texttt{"Finish"}.

![Figure 15: Select Project Root Directory](image)

![Figure 16: Selecting the iot585 Project](image)
14. After Eclipse imports the project, the **iot585** project appears in the Project Explorer window (Figure 17).

![Figure 17: iot585 Project Imported](image)

### 5 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 18). The progress of building the project is shown in the bottom console window.
At the end of the building procedure (Figure 19), 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.
1.27 mm pitch header (8) on the CIB. Connect the other end of the IDC-10 cable to the debugging port on the bottom of the DA14585 IoT MSK.

![Connection between DA14585 IoT MSK and the CIB](image)

**Figure 20: Connection between DA14585 IoT MSK and the CIB**

**IMPORTANT NOTE**

Before connecting the IoT MSK to the CIB with the IDC cable, turn the side switch of the MSK to OFF to cut off the battery and turn on the CIB switch SW2 to provide power to the MSK.

6 Debugging

Click the **Debug** icon and then select the **RAM_DA14585** debug configuration (**Figure 21**).
Figure 21: Start Debugging

When Eclipse asks to open the Debug perspective, the Debug view opens. This means that the image is written in SRAM and users may now press the Resume button to start running the program (Figure 22). The application is now running, and the IoT MSK is ready to connect to Dialog’s IoT Sensors mobile application. To burn the application image into flash, follow the instructions of section 8 in this document.
7 Troubleshooting

If the RAM_DA14585 debug configuration (Figure 21) cannot be seen, try the following:

1. Click File > Import > Run/Debug > Launch configurations, then Next.
2. Click the **Browse** button to search for a specific folder to import launch configurations from.
8 Creating the Multi-Image

Detailed information on the provided scripts is included in Appendix C of [1]. Users should use the make_image_iot.bat script 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] and [4].

IMPORTANT NOTE

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

9 Code size estimation

This example is compiled in a size optimized manner. The (-Os) option is used which will reduce the size of the created binary. Figure 25 shows the optimization settings for the io585 project.

![Figure 25: Project optimization setting](image)

Information on the size is displayed on the console output. Figure 26 shows the console view.
Compiling DA14585 IoT Sensors Reference Application with Eclipse/GCC

- **text**: code and read-only memory in decimal
- **data**: read-write data in decimal
- **bss**: zero-initialized data in decimal
- **dec**: total of text, data, and bss in decimal
- **hex**: total of text, data, and bss in hex

![GCC console view](image)

Figure 26: GCC console view
Compiling DA14585 IoT Sensors Reference Application with Eclipse/GCC

Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>10-Dec-2018</td>
<td>First release</td>
</tr>
</tbody>
</table>

Status Definitions

<table>
<thead>
<tr>
<th>Status</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRAFT</td>
<td>The content of this document is under review and subject to formal approval, which may result in modifications or additions.</td>
</tr>
<tr>
<td>APPROVED or unmarked</td>
<td>The content of this document has been approved for publication.</td>
</tr>
</tbody>
</table>

Disclaimer

Information in this document is believed to be accurate and reliable. However, Dialog Semiconductor does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information. Dialog Semiconductor furthermore takes no responsibility whatsoever for the content in this document if provided by any information source outside of Dialog Semiconductor.

Dialog Semiconductor reserves the right to change without notice the information published in this document, including without limitation the specification and the design of the related semiconductor products, software and applications.

Applications, software, and semiconductor products described in this document are for illustrative purposes only. Dialog Semiconductor makes no representation or warranty that such applications, software and semiconductor products will be suitable for the specified use without further testing or modification. Unless otherwise agreed in writing, such testing or modification is the sole responsibility of the customer and Dialog Semiconductor excludes all liability in this respect.

Customer notes that nothing in this document may be construed as a license for customer to use the Dialog Semiconductor products, software and applications referred to in this document. Such license must be separately sought by customer with Dialog Semiconductor.

All use of Dialog Semiconductor products, software and applications referred to in this document are subject to Dialog Semiconductor’s Standard Terms and Conditions of Sale, available on the company website (www.dialog-semiconductor.com) unless otherwise stated.

Dialog and the Dialog logo are trademarks of Dialog Semiconductor plc or its subsidiaries. All other product or service names are the property of their respective owners.

© 2019 Dialog Semiconductor. All rights reserved.

Contacting Dialog Semiconductor

United Kingdom (Headquarters)
Dialog Semiconductor (UK) LTD
Phone: +44 1793 757700

Germany
Dialog Semiconductor GmbH
Phone: +49 7021 805-0

The Netherlands
Dialog Semiconductor B.V.
Phone: +31 73 640 8822
Email: enquiry@diasemi.com

North America
Dialog Semiconductor Inc.
Phone: +1 408 845 8500

Japan
Dialog Semiconductor K. K.
Phone: +81 3 5769 5100

Taiwan
Dialog Semiconductor Taiwan
Phone: +886 291 786 222
Web site: www.dialog-semiconductor.com

Hong Kong
Dialog Semiconductor Hong Kong
Phone: +852 2607 4271

Korea
Dialog Semiconductor Korea
Phone: +82 2 3469 8200

China (Shenzhen)
Dialog Semiconductor China
Phone: +86 755 2981 3669

China (Shanghai)
Dialog Semiconductor China
Phone: +86 21 5424 9058