ML63Q2500
Software Development Environment Tutorial

Rev.1
Issue Date: Dec. 26, 2025

Notice

Precaution on using ROHM Products

1) When using ROHM Products, refer to the latest product information and ensure that usage conditions (absolute maximum ratings[*1], recommended operating conditions, etc.) are within the ranges specified. ROHM disclaims any and all liability for any malfunctions, failure or accident arising out of or in connection with the use of ROHM Products outside of such usage conditions specified ranges, or without observing precautions. Even if it is used within such usage conditions specified ranges, semiconductors can break down and malfunction due to various factors. Therefore, in order to prevent personal injury, fire or the other damage from break down or malfunction of ROHM Products, please take safety at your own risk measures such as complying with the derating characteristics, implementing redundant and fire prevention designs, and utilizing backups and fail-safe procedures.
   *1: Absolute maximum ratings: a limit value that must not be exceeded even momentarily.

2) The Products specified in this document are not designed to be radiation tolerant.

3) Descriptions of circuits, software and other related information in this document are provided only to illustrate the standard operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. And the peripheral conditions must be taken into account when designing circuits for mass production. ROHM disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, and other related information.

4) No license, expressly or implied, is granted hereby under any intellectual property rights or other rights of ROHM or any third party with respect to ROHM Products or the information contained in this document (including but not limited to, the Product data, drawings, charts, programs, algorithms, and application examples, etc.). Therefore, ROHM shall have no responsibility whatsoever for any dispute, concerning such rights owned by third parties, arising out of the use of such technical information.

5) ROHM intends our Products to be used in a way indicated in this document. Please be sure to contact a ROHM sales office if you consider the use of our Products in different way from original use indicated in this document. For use of our Products in medical systems, please be sure to contact a ROHM representative and must obtain written agreement. Do not use our Products in applications which may directly cause injuries to human life, and which require extremely high reliability, such as aerospace equipment, nuclear power control systems, and submarine repeaters, etc. ROHM disclaims any and all liability for any losses and damages incurred by you or third parties arising by using the Product for purposes not intended by us without our prior written consent.

6) Please use the Products in accordance with any applicable environmental laws and regulations, such as the RoHS Directive. ROHM shall have no responsibility for any damages or losses resulting non-compliance with any applicable laws or regulations.

7) When providing our Products and technologies contained in this document to other countries, you must abide by the procedures and provisions stipulated in all applicable export laws and regulations, including without limitation the US Export Administration Regulations and the Foreign Exchange and Foreign Trade Act.

8) Please contact a ROHM sales office if you have any questions regarding the information contained in this document or ROHM's Products.

9) This document, in part or in whole, may not be reprinted or reproduced without prior consent of ROHM.

(Note) “ROHM” as used in this document means ROHM Co., Ltd.

Other Precaution

1) All information contained in this document is subject to change for the purpose of improvement, etc. without any prior notice. Before purchasing or using ROHM Products, please confirm the latest information with a ROHM sales office.

2) ROHM has used reasonable care to ensure the accuracy of the information contained in this document, however, ROHM shall have no responsibility for any damages, expenses or losses arising from inaccuracy or errors of such information.

TSZ72037・01・001

Table of Contents

1. Introduction

This document is a tutorial for the integrated development environment LEXIDE-Ω, used to develop software for the Solist-AI™ microcontroller.
By following this document, you will learn how to use each tool in the development environment. As an example, this tutorial uses the ML63Q2557 to create a simple program and run it on actual hardware to deepen understanding. If you are using a microcontroller other than ML63Q2557, please make the necessary adjustments as you follow along.

[Note]
The screens shown in this document are examples at the time of creation and may differ from the actual screens due to version upgrades or other changes.
The contents of this document serve as supplementary material to the official manuals of the development tools and software. For detailed instructions, please refer to the user manuals of the development tools you are using.

1.1 Preparation

Before proceeding with this document, please prepare the following items.

Item	Description
LAPIS Development Tools LEXIDE-Ω (2.1.0 or later)	This is an integrated development environment (IDE) for our microcontrollers equipped with Arm™ Cortex™-M cores (*1).
ML63Q2500 Software Pack	This is the device information file for the ML63Q2500 group(*1).
CMSIS-Core(M)	This is a device information file for Arm™ Cortex™-M core-based microcontrollers (*2).
Solist-AI™ SimAnomaly Detection Version (V1.00.04)	This is a simulation tool for testing Solist-AI™ functionality on a PC (*1). It is designed for anomaly detection using unsupervised learning. By training the system with a set of numerical data representing normal conditions, you can detect deviations from the learned state as anomalies.
Solist-AI™ SimSupervised Learning Version (SLV1.00.04)	This is a simulation tool that allows you to verify the functionality of Solist-AI™ on a PC (*1). In addition to anomaly detection using unsupervised learning, please use this tool when considering applications such as object recognition using sensor data or battery degradation prediction through supervised learning for forecasting and parameter estimation.
ML63Q2500 Group Reference Software	This includes peripheral drivers for ML63Q2500 group and sample programs that run on the reference board (*1).
Solist-AI™ Scope	This is a real-time waveform display tool for visualizing the output of Solist-AI™(*1). To use this tool, a USB-SPI conversion module is required. Note: Instead of Solist-AI™ Scope, you can also use the LEXIDE Scope included with LEXIDE-Ω. In that case, the USB-SPI conversion module is not required. For details, please refer to “4.3 Waveform Display Using LEXIDE Scope.”
Reference Board	This is a board equipped with a sample microcontroller. By adding the necessary components as needed, you can evaluate the operation of the ML63Q25xx microcontroller.
Debug Adapter	This is used for software debugging and programming to Flash memory. Please use a J-Link or a CMSIS-DAP compliant debug adapter. In this tutorial, J-Link will be used.
USB-SPI Conversion Module (Optional)	By connecting to a PC via a USB-to-SPI converter (FT232H), you can monitor AI operations. We recommend using the MM-FT232H or MM-FT232HC modules (hereafter referred to as MM-FT232H and MM-FT232HC), which are equipped with the FT232H chip from FTDI and manufactured by Sunhayato Corp. Note: It is not required if Solist-AI™ Scope is not used.

*1：You can download it from the following page on the ROHM website.
https://www.rohm.com/products/micon/solist-ai/ml63q2500-group
*2：Available for download from the following link.
https://www.keil.arm.com/packs/cmsis-arm/versions/

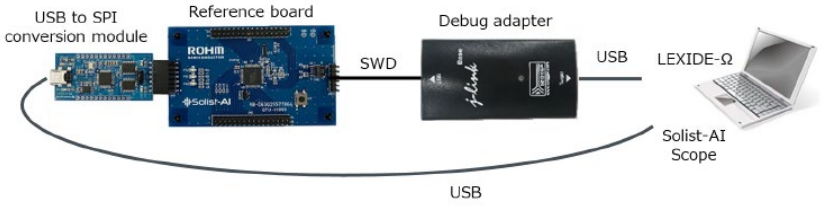

Figure 1-1. Software Development Environment Configuration

1.2 Related Manuals

The related manuals for this document are listed below. Please refer to them as needed.

Manual Title	Description
ML63Q2500 Group User's Manual	This is the hardware manual for the ML63Q2500 group of microcontrollers (*1).
LEXIDE-Ω for ARM Core Microcontrollers User's Manual	This manual (*2) describes how to use the integrated development environment LEXIDE-Ω for developing programs for LSIs equipped with Arm cores. Please refer to it when you need to check how to create projects or configure settings for debugging in LEXIDE-Ω.
LEXIDE C Compiler for ARM User's Manual	This manual (*3) describes how to use the LCC-ARM toolchain, including the C compiler, assembler, linker, and other components.
ML63Q2500 Reference Software Startup Manual	This manual (*4) describes the peripheral drivers for the ML63Q2500 group and sample programs that run on the reference board.
ML63Q2500 Reference Software Peripheral Driver API Specification	This is an HTML help-format document (*4) that describes the peripheral driver APIs for the ML63Q2500 group.
Solist-AI™ Scope Users Guide	This manual (*5) describes how to use the Windows debugging application Solist-AI™ Scope, which enables real-time visualization of memory information in graphical format.

*1 : You can download it from the following page on ROHM's website.
https://www.rohm.com/products/micon/solist-ai/ml63q2500-group
*2 : You can refer to it from [LAPIS LEXIDE Software Tools Documentation] in the Windows Start Menu.
*3 : You can refer to it from [LAPIS LEXIDE Software Toolchain Ver.*xxxxxx*] in the Windows Start Menu.
*4 : You can refer to it from the Document folder of the ML63Q2500 reference software.
*5 : You can refer to it from the Document folder of the Solist-AI™ Scope.

2. Installation

To use the development environment, install the following tools on your PC. (Please refer to "1.1 Preparation" for the download link.)

- LAPIS Development Tools LEXIDE-Ω
- ML63Q2500 Group Reference Software
- Device Information Files (ML63Q2500 Software Pack, CMSIS-Core(M))

2.1 Installing LAPIS Development Tools LEXIDE-Ω

Extract the downloaded file LAPIS_LEXIDE_V*xxxx*.zip (where V*xxxx* represents the version string) to a folder of your choice, then double-click LexideInstaller_*yyyymmdd*.exe (where *yyyymmdd* indicates the release version) inside the folder.
Follow the on-screen instructions provided by the installer to complete the installation.

2.2 Installing the ML63Q2500 Group Reference Software

Extract the downloaded file ML63Q2500_ReferenceSoftware_v*xxx*.zip (where v*xxx* represents the release version) to a folder of your choice.
When creating your own program, if you wish to reuse a sample program from the reference software, please copy it before use.

2.3 Installing Device Information Files

Installation of the device information files (ML63Q2500 Software Pack, CMSIS-Core(M)) is performed via LEXIDE-Ω. First, launch LEXIDE-Ω.

From the Windows Start Menu, select:
Start > All Programs > LAPIS LEXIDE Tools > LEXIDE-Ω

After a brief moment, the [Select a directory as workspace] dialog will appear.
Set the workspace to C:\lexide\workspace_omega_v2_arm, then click the [Launch] button.

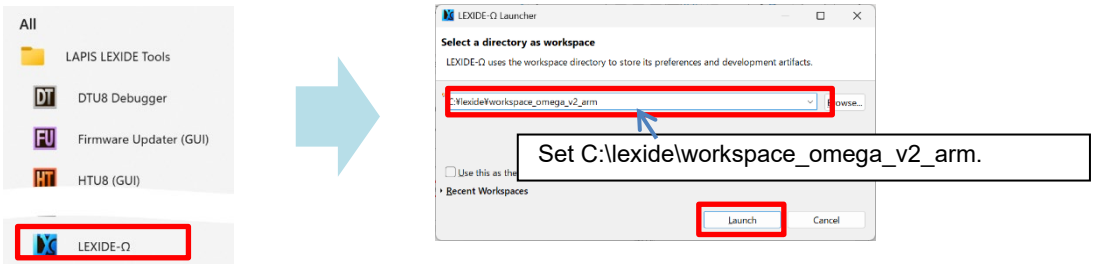

Figure 2-1. Launching LEXIDE-Ω

Once LEXIDE-Ω has launched, open the CMSIS-Pack Manager.
To do this, click the icon in the upper-right corner of the LEXIDE-Ω window to open the [Open Perspective] dialog.
From the dialog, select [CMSIS-Pack Manager] and click the [Open] button.

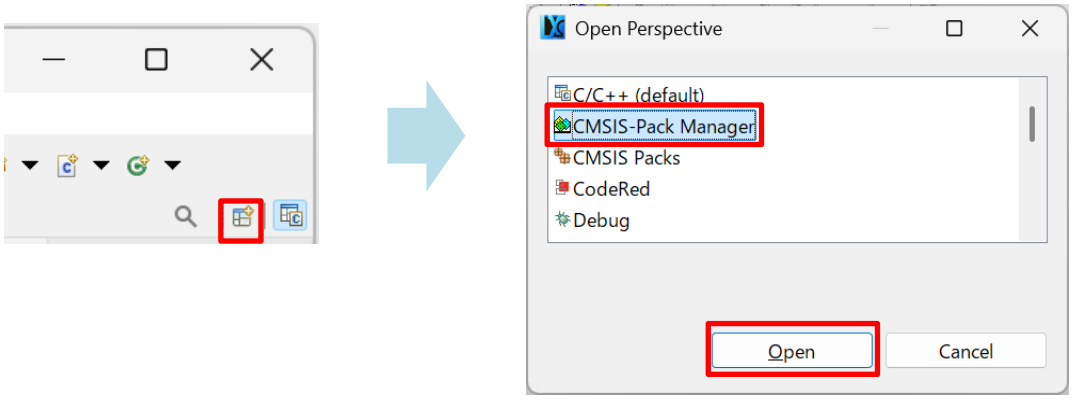

Figure 2-2. How to display the CMSIS-Pack Manager

In the displayed [CMSIS-Pack Manager] perspective (*1), click [Import existing packs…].
In the [Import Packs] dialog that appears, specify the downloaded device information file (in this case, ROHM.ML63Q25x7_DFP1.01.pack) and click the [Open] button.

*1: A perspective defines the layout of views, including commonly used menu bars and toolbars, tailored to specific tasks.

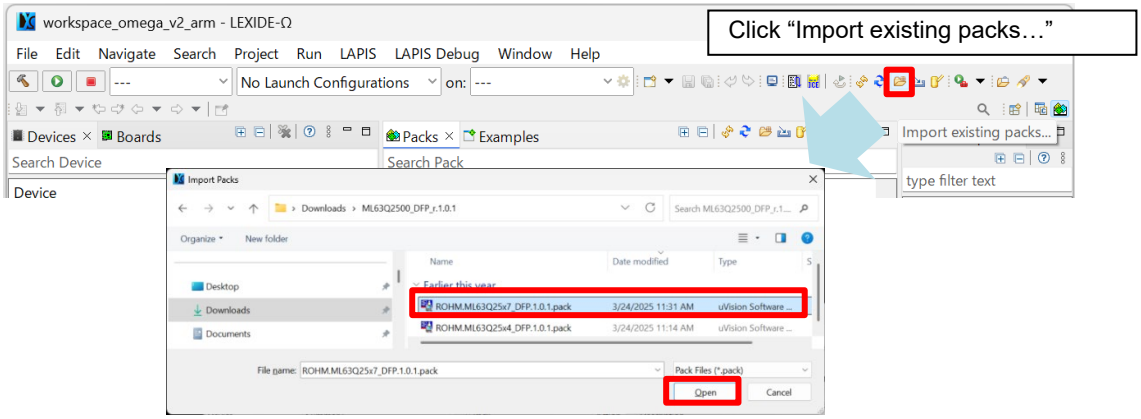

Figure 2-3. Installing the ML63Q2500 Software Pack (ML63Q25x7_DFP)

Once the installation is complete, the installed device information file will be displayed as shown below, and it will be available for use within LEXIDE-Ω.

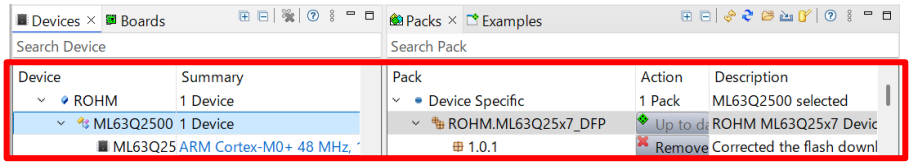

Figure 2-4. Display of Installed Device Information File

Similarly, specify CMSIS-Core(M) in the [Import Packs] dialog and install it.

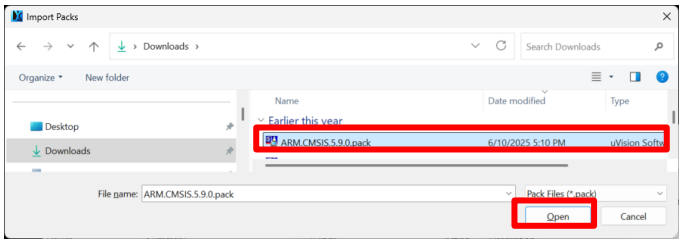

Figure 2-5. Installation of CMSIS-Core(M)

3. Program Creation and Debugging

In this section, we will use a sample program included in the ML63Q2500 reference software.
The program used is BlinkLED, which blinks the LED on the reference board.
The blinking timing is controlled by interruptions from the low-speed time base counter.

3.1 Steps to Create the Program

In this tutorial, we will create the program following the steps below.

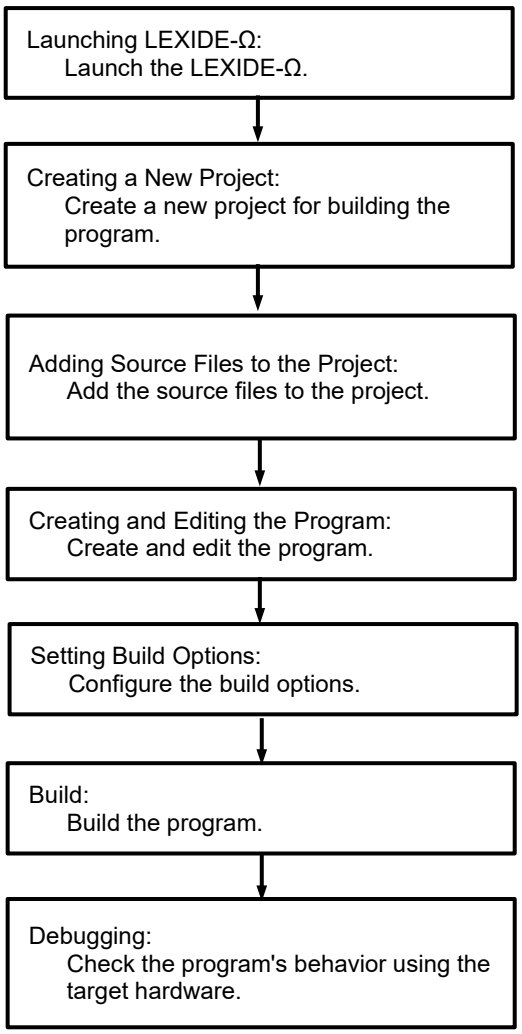

Figure 3-1. Steps for Creating the Program

3.2 Launching LEXIDE-Ω

Select “Windows Start > All > LAPIS LEXIDE Tools > LEXIDE-Ω”.
After a short while, the [Select a directory as workspace] dialog will appear.
Here, set the directory to “C:\lexide\workspace_omega_v2_arm” and click the [Launch] button.

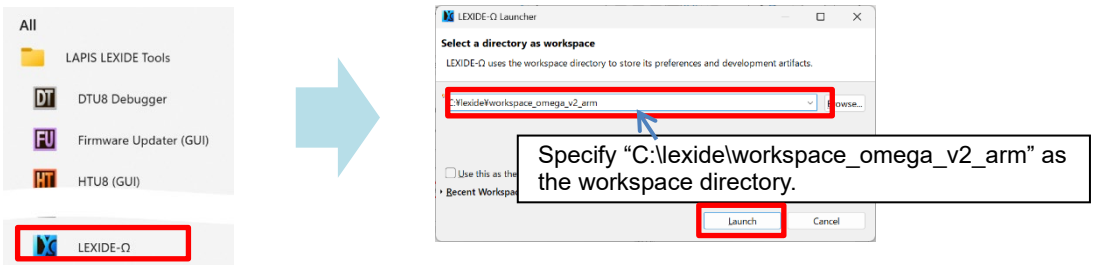

Figure 3-2. Launching LEXIDE-Ω

After a short while, LEXIDE-Ω will launch.
The screen displayed upon starting LEXIDE-Ω is shown below.

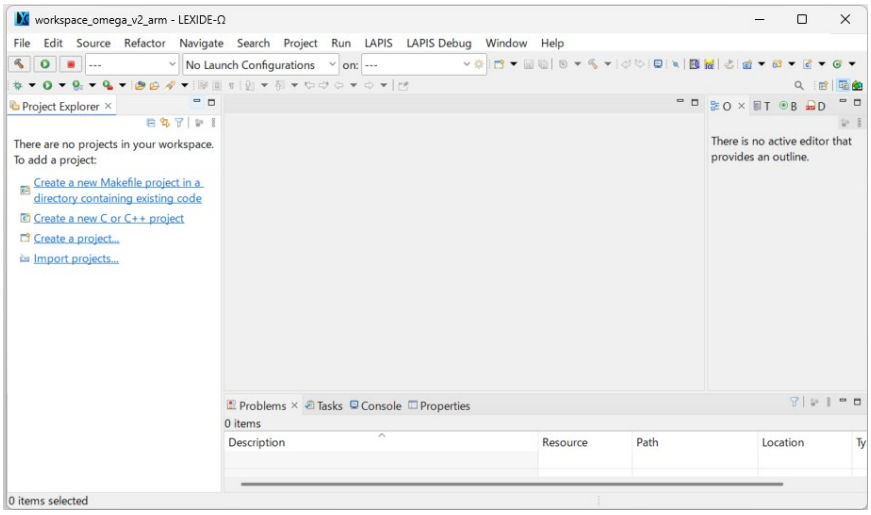

Figure 3-3. LEXIDE-Ω Initial Screen

If your screen does not match Figure 3-3, click the  icon to change to the [C/C++] perspective.

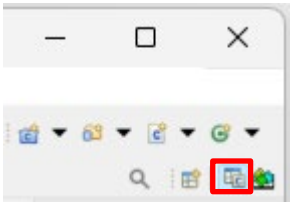

Figure 3-4. Switching to the [C/C++] Perspective

When LEXIDE-Ω starts, a folder named .metadata is created under the workspace directory (C:\lexide\workspace_omega_v2_arm), as shown in the figure below.

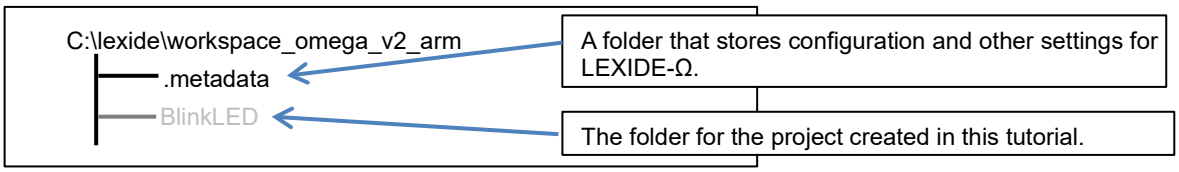

Figure 3-5. Folder Structure in the LEXIDE-Ω Workspace

A workspace is a working folder used to store LEXIDE-Ω configuration information and projects.
Multiple projects can be created within a single workspace.
In this tutorial, we will create folders for the project and source files under the workspace directory (C:\lexide\workspace_omega_v2_arm).

3.3 Creating a New LEXIDE-Ω Project

Create a new project.
Select [File] > [New] > [Project] to open the [New Project] dialog.

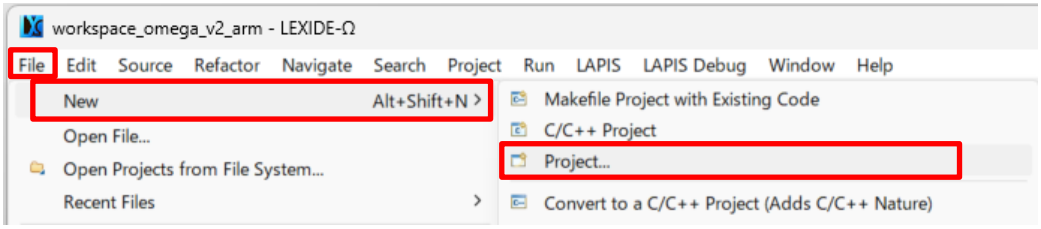

Figure 3-6. How to Display the [New Project] Dialog

In the [New Project] dialog, select [C/C++] > [C Project], then click the [Next] button.

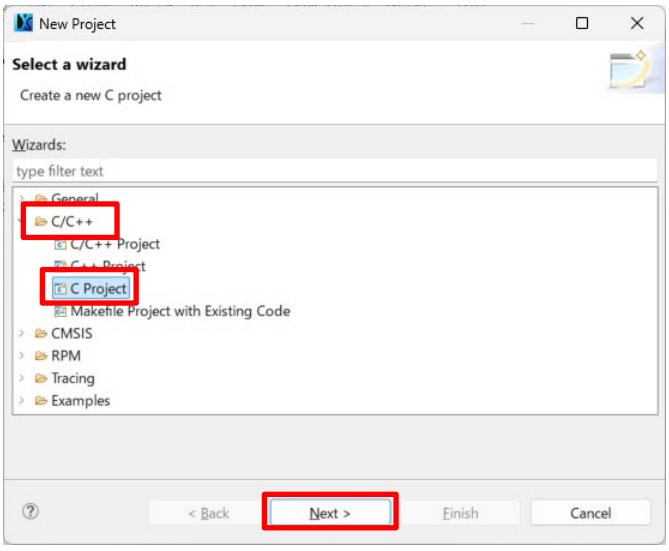

Figure 3-7. Project Selection

On the [C Project] page, configure the project name, project type, and toolchain, then click the [Next] button.
In this tutorial, we will create a new project named BlinkLED.

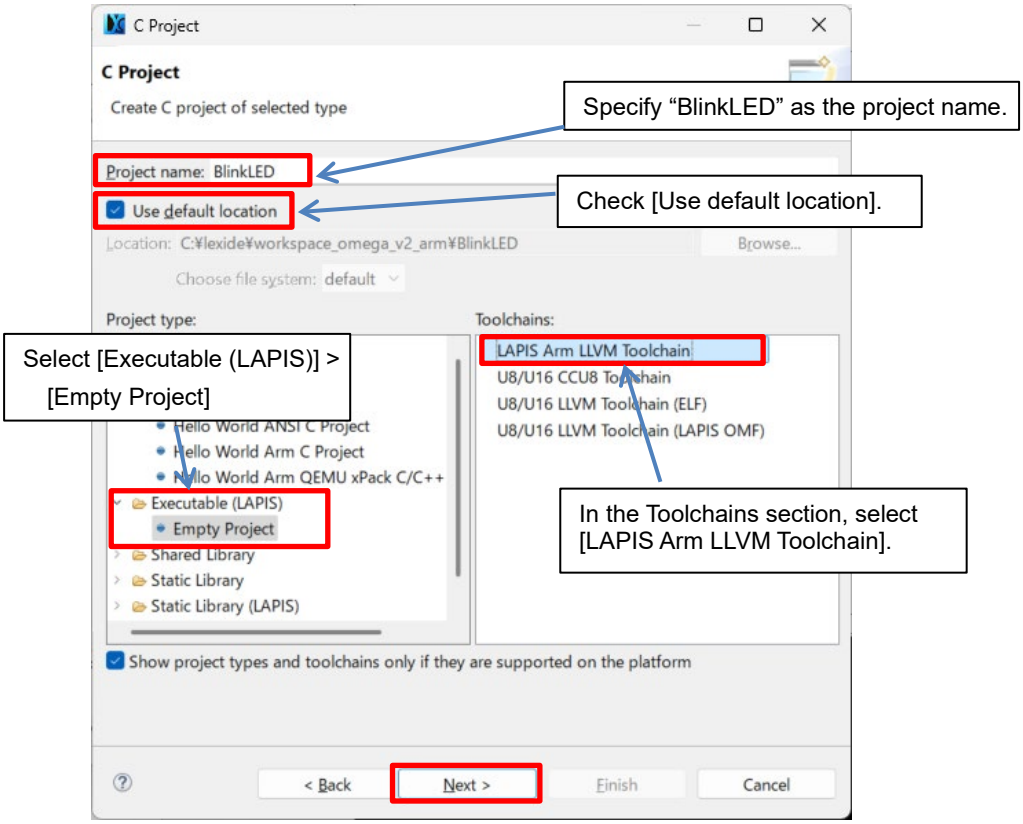

Figure 3-8. Project Selection

On the [Select Configurations] page, select a build configuration and click the [Next] button. In this tutorial, we will select [Debug].

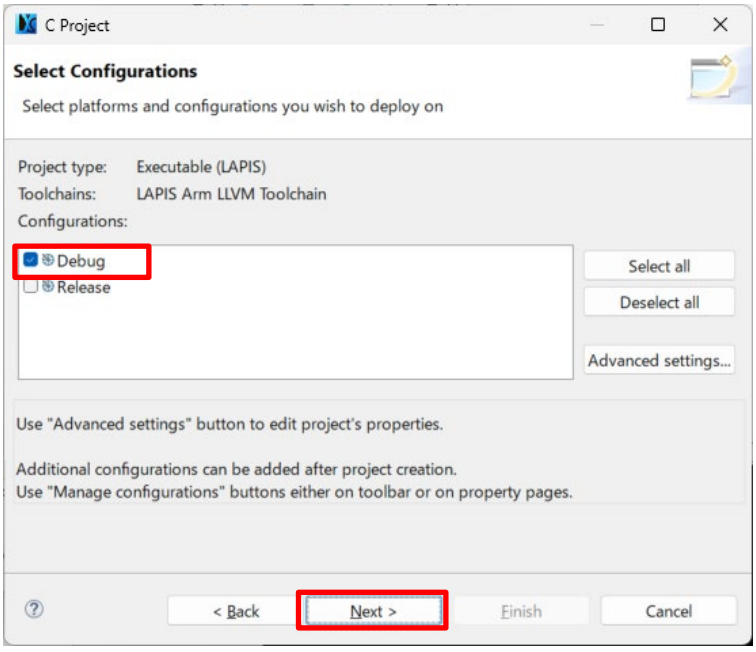

Figure 3-9. Build Configuration Settings

On the [Select LAPIS Arm Device] page, select the target MCU.
In this tutorial, select [ML63Q2500] for [Family], and ML63Q25x7 for [Device].

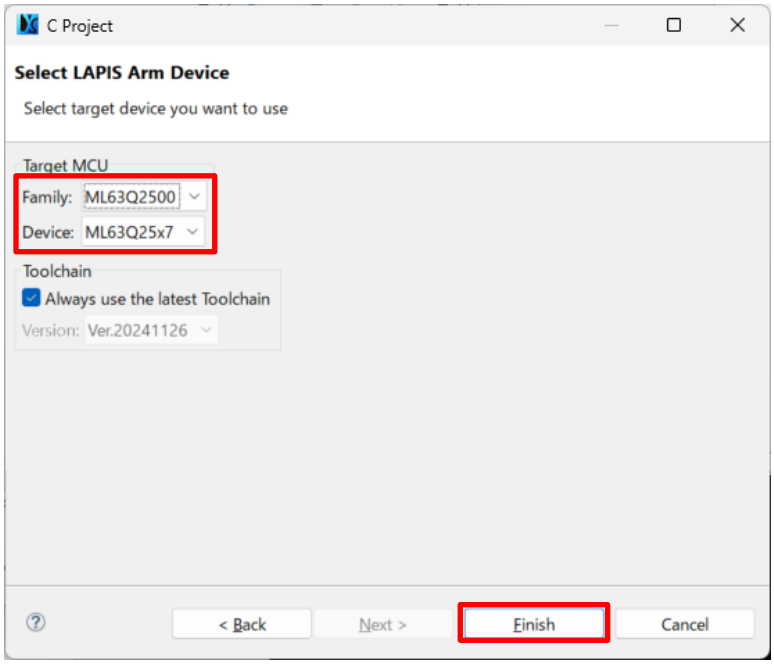

Figure 3-10. Selecting the Target MCU

After clicking the [Finish] button, the Project Wizard will be completed, and the project will be generated.

In LEXIDE-Ω's [Project Explorer], the newly created project "BlinkLED" will be displayed.
The project will include files such as ML63Q25x7.h (which defines the peripheral registers of the target MCU), startup_ML63Q25x7.c (startup file), and linker script files.

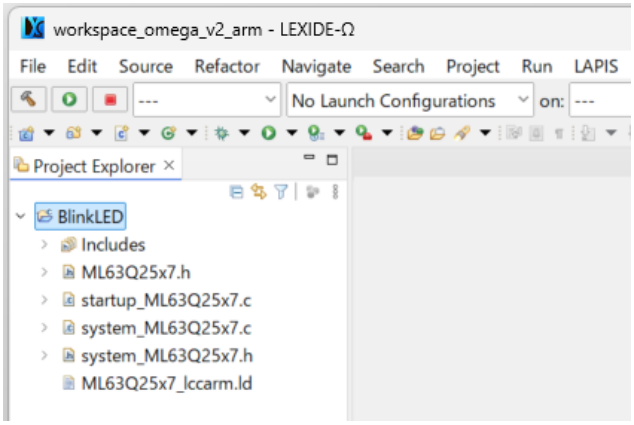

Figure 3-11. State Immediately After Creating a New Project

3.4 Adding Source Files to the Project

Add the source files of the LED blinking program (BlinkLED), which is included in the reference software, into LEXIDE-Ω.
Each sample program in the reference software includes a project for LEXIDE-Ω. However, since virtual folders are used, it has the disadvantage of making it difficult to change the file or folder structure.
Therefore, in this tutorial, the sample program projects will be used only as a reference to check the file structure, and the source files of the sample programs will be manually added to the project.
The file structure of the BlinkLED sample program in the reference software is as follows:

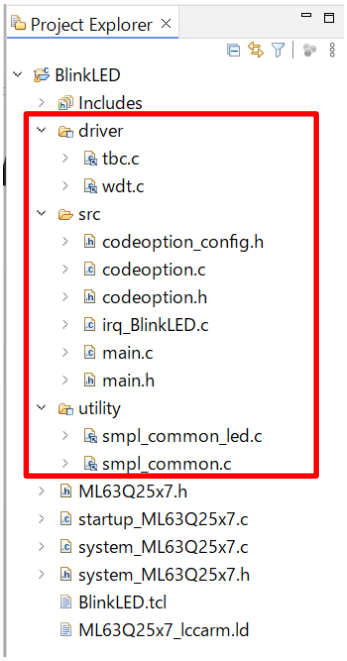

Figure 3-12. File Structure of the BlinkLED Sample Program in the Reference Software

In this tutorial, we will refer to the file structure shown above and add the source files of the BlinkLED sample program, enclosed in red, to the project.

Select the src folder located under SourceCode > samples > Common > BlinkLED in the reference software, and drag and drop it into the BlinkLED project in the [Project Explorer].

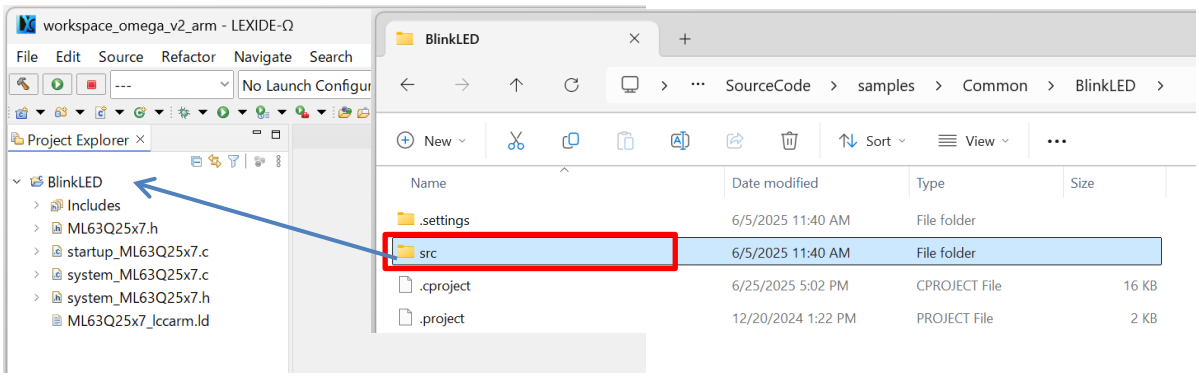

Figure 3-13. Drag and drop the src Folder of the BlinkLED Sample Program into the Project

When the following dialog appears, select [Copy files and folders] and click the [OK] button.

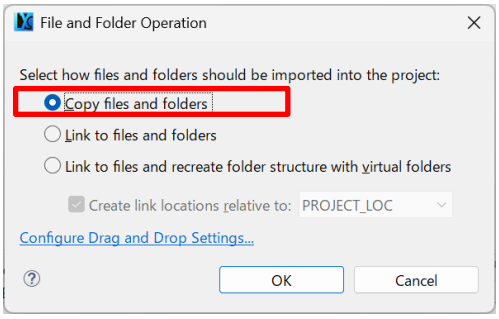

Figure 3-14. Confirmation Dialog for File and Folder Operation

The dragged and dropped folder and source files will be copied to the workspace folder and displayed in the project within the [Project Explorer].

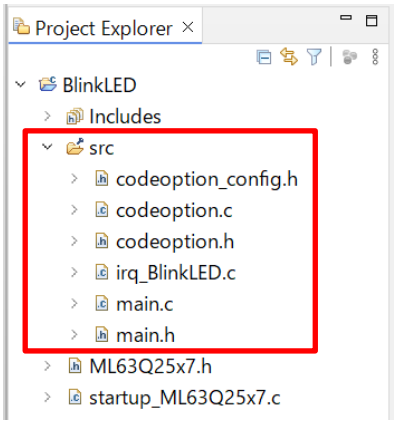

Figure 3-15. Folder and Source Files Added in the Project

Similarly, select the driver and utility folders under SourceCode in the reference software, and drag and drop them into the project in the [Project Explorer].

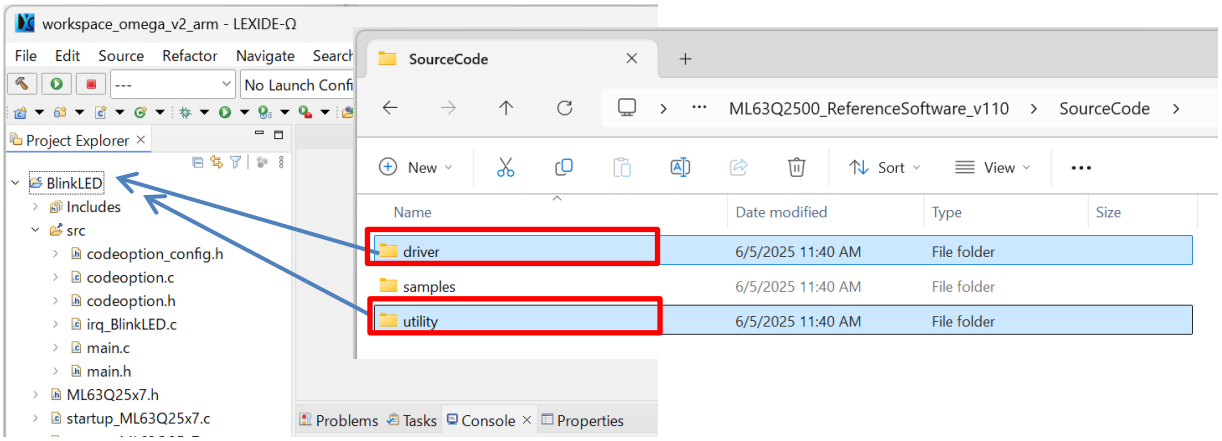

Figure 3-16. Drag and drop the driver and utility Folders into the Project

When the following dialog appears, select [Copy files and folders] and click the [OK] button.

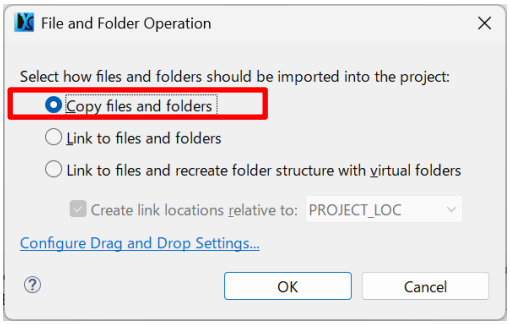

Figure 3-17. Confirmation Dialog for File and Folder Operations

Next, we will exclude unnecessary files from the build target of the BlinkLED sample program.
By referring to “Figure 3-12. File Structure of the BlinkLED Sample Program in the Reference Software,” we can see that the source files used from the driver folder are tbc.c and wdt.c, and from the utility folder are smpl_common.c and smpl_common_led.c.
Therefore, files other than tbc.c and wdt.c in the driver/src folder will be excluded from the build target.
The driver/lib folder contains library files and will not be linked unless explicitly called from the program, so it will remain unchanged.
Select all files in the driver/src folder except tbc.c and wdt.c while holding down the Ctrl key, then right-click and choose [Resource Configurations] > [Exclude from Build…] from the popup menu.

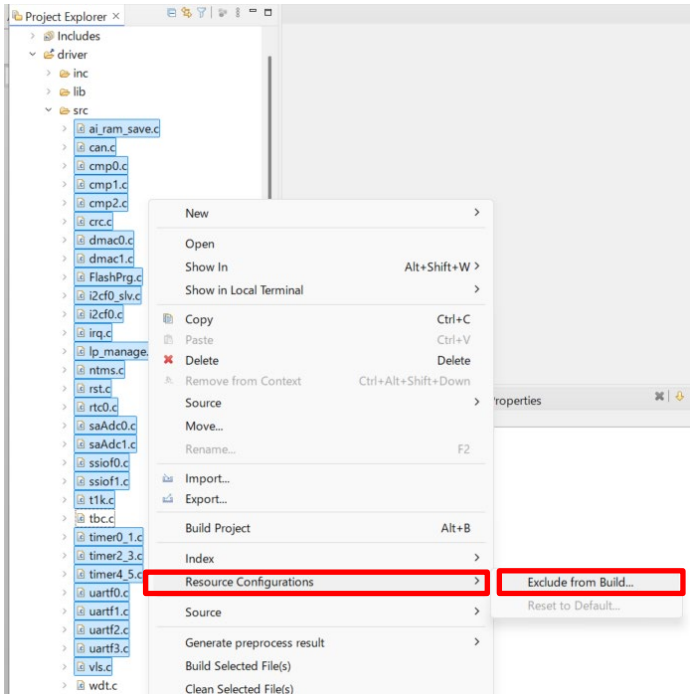

Figure 3-18. Excluding Unnecessary Files from the Build Target

When the following dialog appears, check [Debug] and click the [OK] button.

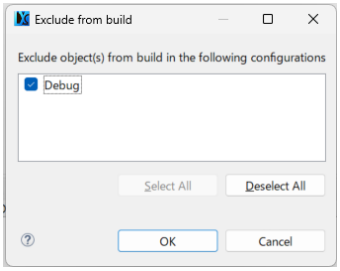

Figure 3-19. Confirmation Dialog When Excluding from Build Target

As a result, the selected files will be excluded from the build target.
Files excluded from the build target will appear grayed out.

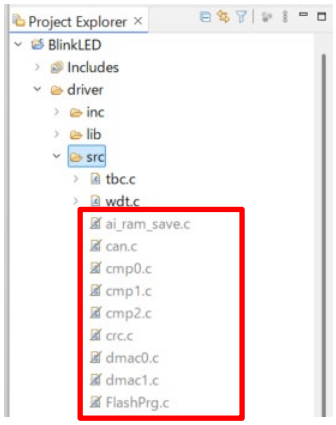

Figure 3-20. Files Excluded from the Build Target

This completes the file registration to the project.

3.5 Cleating and Editing the Program

The sample program used in this tutorial, BlinkLED, repeatedly switches the lighting pattern of the LEDs (P50, P51, P52) on the reference board at fixed time intervals.
The timing for switching the lighting pattern is controlled by an interruption from the Low-Speed Time Base Counter (TBC).

In this tutorial, the program will be used as-is without modifying its behavior.

To edit a source file, double-click the file name displayed under the project in the [Project Explorer]. The source file will open and become editable.

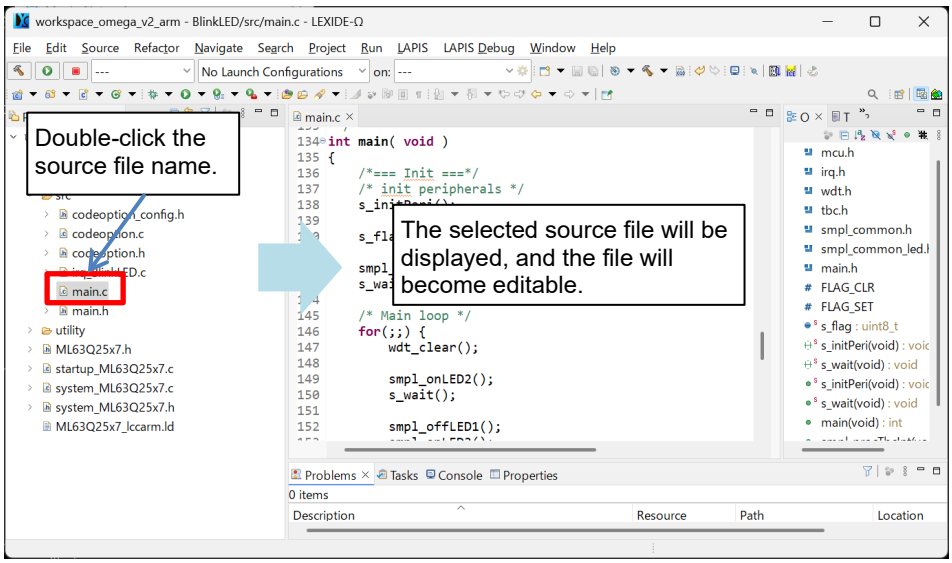

Figure 3-21. How to Display a Source File

3.6 Setting Build Options

Set build options such as include paths, macros, and optimization settings.
Build options are configured in the [Properties] dialog.
To open the [Properties] dialog, select the BlinkLED project in the [Project Explorer], then choose [Project] > [Properties] from the menu.

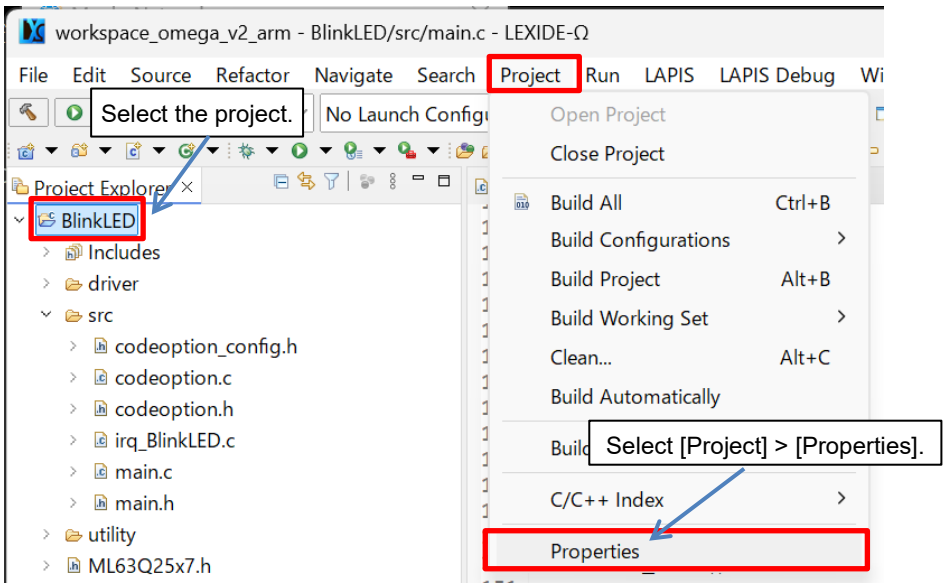

Figure 3-22. How to Display the [Properties] Dialog

Build options are configured by selecting [C/C++ Build] > [Settings] in the [Properties] dialog shown below.

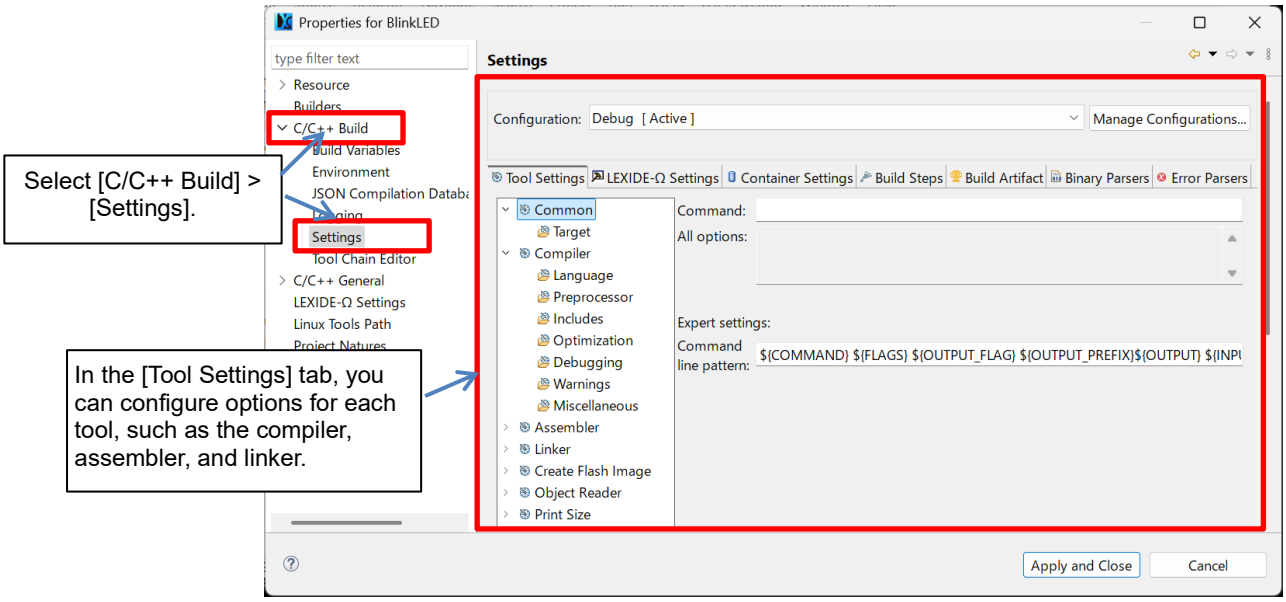

Figure 3-23. [Properties] Dialog – Build Options Settings Screen

In this tutorial, the following settings will be configured:

- Macro settings
- Include path settings
- Optimization options settings

3.6.1 Macro Settings

The reference software includes sections where settings are switched depending on the device using macros. Therefore, you need to configure the macro corresponding to microcontroller you are using.
Select [Compiler] > [Preprocessor], click the [Add] button , and in the dialog that appears, specify the macro corresponding to your microcontroller.
In this tutorial, ML63Q2557 is specified.

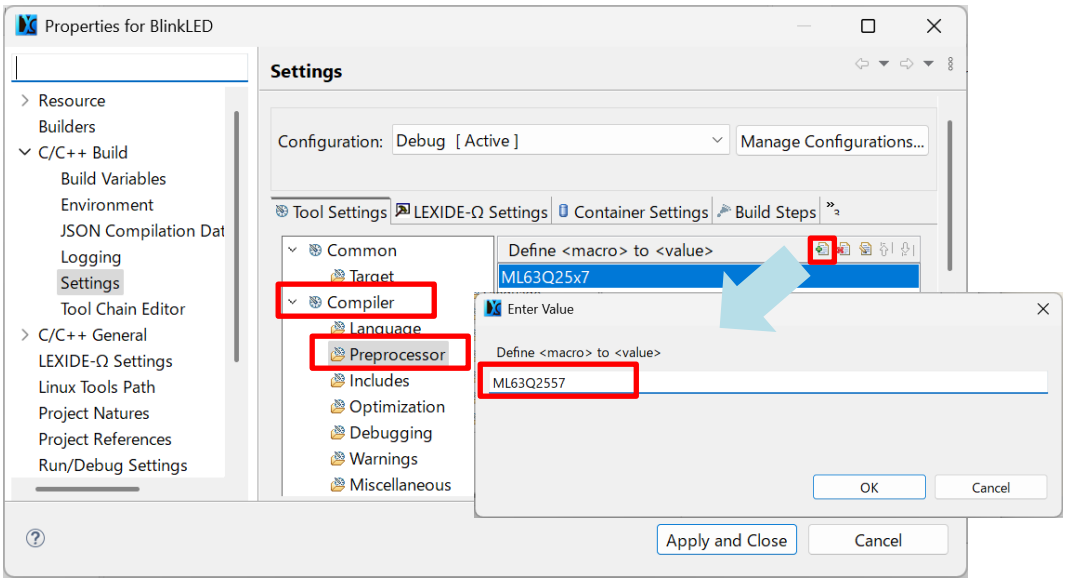

Figure 3-24. Macro Settings

3.6.2 Include Path Settings

Header files (*.h) are stored directly under the BlinkLED folder, as well as in the driver\inc, src, and utility\board folders.
To configure the include paths:

1. Select [Compiler] > [Includes].
2. Click the [Add] button .
3. In the displayed [Add directory path] dialog, click the [Workspace] button.
4. In the [Folder selection] dialog, hold down the Ctrl key and select the following folders:
   BlinkLED
   driver > inc
   src
   utility > board
5. Click the [OK] button to confirm your selection.

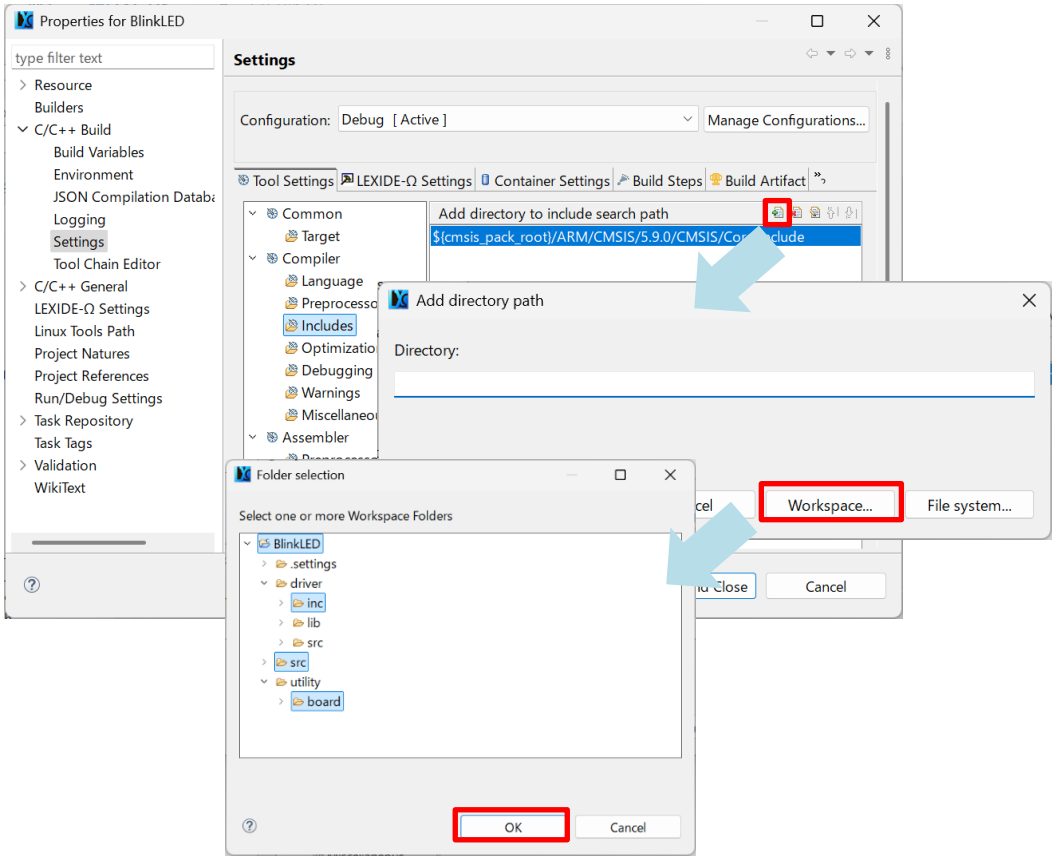

Figure 3-25. Include Path Settings

3.6.3 Optimization Option Settings

To configure optimization options, go to [Compiler] > [Optimization].
In this tutorial, none is selected to generate code that is best suited for debugging.
Once all settings are complete, click the [Apply and Close] button to apply the changes.

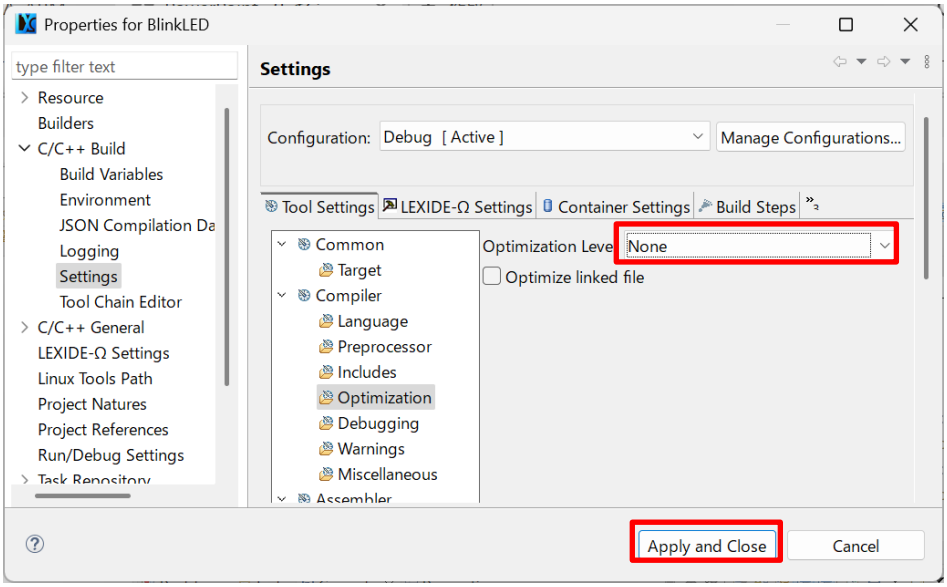

Figure 3-26. Optimization Option Settings

3.7 Build

The BlinkLED project is built using the following steps:

1. In the [Project Explorer], select the project (BlinkLED).
2. Click the Build button on the toolbar to start the build process.

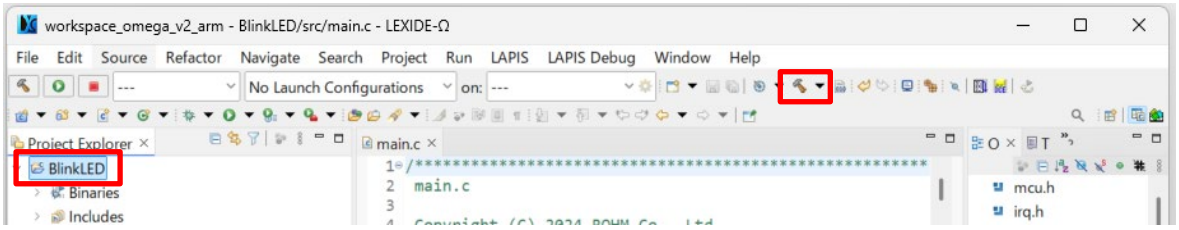

Figure 3-27. Build

When the building is complete, "Build Finished" will be displayed in the [Console] tab.

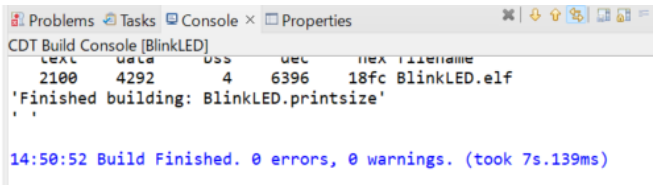

Figure 3-28. Message Displayed Upon Build Completion

3.8 Debugging

This section explains how to use the debugger in LEXIDE-Ω.
For more details about the debugger, please refer to Chapter 6, "Arm Debugging Features", in the LEXIDE-Ω for ARM Core Microcontrollers User's Manual.

3.8.1 Preparation

In this tutorial, the SEGGER J-Link is used as the debug adapter.
When using SEGGER J-Link, please make sure to install the J-Link Software and Documentation Pack version 7.62 or later in advance.
After installation, the flash loader for ROHM LSI must be recognized by J-Link.
The steps for this procedure are described below.

① Prepare the flash loader module file to be used. When you install the device information file as described in Section 2.3 Installing Device Information Files, the flash loader is typically stored in the following folder under your user account:
C:\Users\<UserName>\AppData\Local\Arm\Packs\<VendorName>\<ROHM_LSI_Name>\<Version>\Flash\<ROHM_LSI_FlashLoader_FileName>.FLM

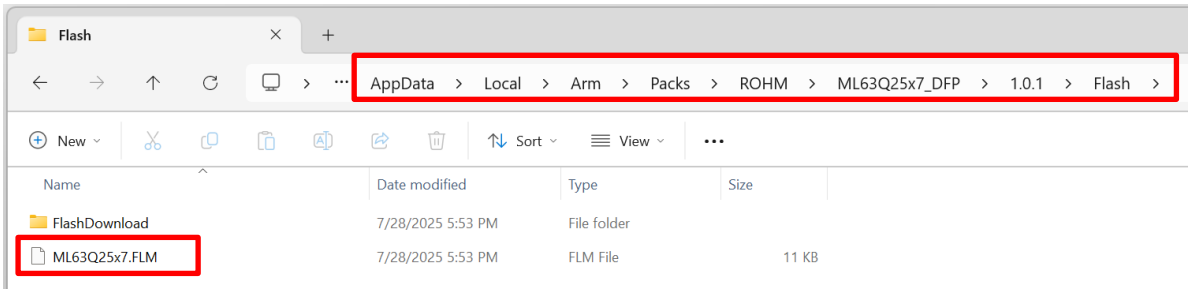

Figure 3-29. Storage Location of ROHM LSI (ML63Q25x7) Flash Loader

② Create the following folder and copy the flash loader module file from step ① into it:
C:\Users\<UserName>\AppData\Roaming\SEGGER\JlinkDevices\Rohm\ML63Q25x7\

③ In the same folder, create a file that allows J-Link to recognize the added device. In this tutorial, the file is named ML63Q25x7.xml.

```
\---JLinkDevices
    +---Rohm
        +---ML63Q25x7
        |       ML63Q25x7.FLM
        |       ML63Q25x7.xml
        …
```

④ In the XML file created in step ③, enter the following content. If you are using a device other than ML63Q25x7, please modify the content according to the specifications of your LSI.

Note: Lines enclosed in red boxes should be written as a single line without line breaks.

```
<DataBase>
  <!-- -->
  <!-- Rohm -->
  <!-- -->
  <Device>
  <ChipInfo Vendor="ROHM" Name="ML63Q25x7" Core="JLINK_CORE_CORTEX_M0"
    WorkRAMAddr="0x20000000" WorkRAMSize="0x02000"/>
  <FlashBankInfo Name="Internal Flash" BaseAddr="0x10000000" MaxSize="0x00040000"
    Loader="ML63Q25x7.FLM" LoaderType="FLASH_ALGO_TYPE_CMSIS" AlwaysPresent="1"/>
  </Device>
</DataBase>
```

3.8.2 Creating a Debug Configuration

For the debug configuration, select GDB SEGGER J-Link Debugging.
In the [Project Explorer], right-click on the project and select [Debug As] > [Debug Configurations…] from the popup menu to open the [Debug Configurations] dialog.

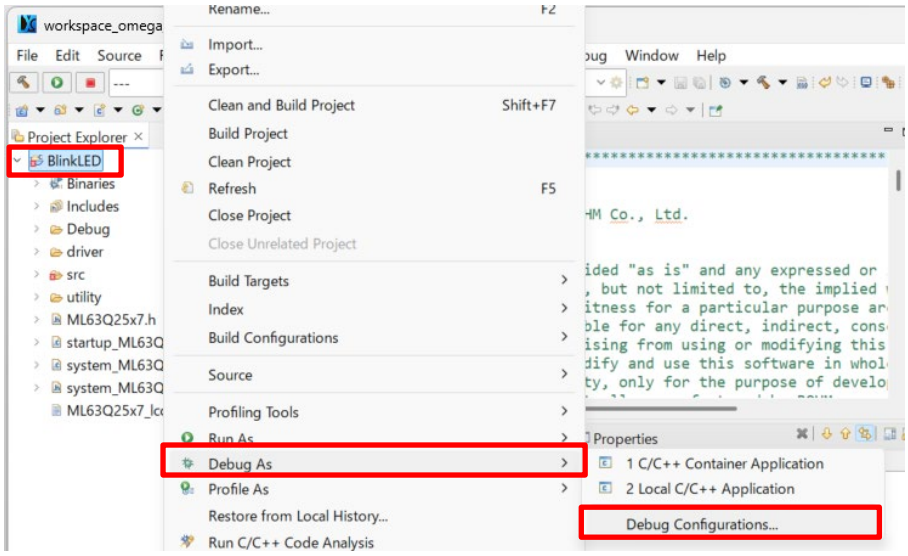

Figure 3-30. How to Display the [Debug Configuration] Dialog

In the displayed [Debug Configuration] dialog, select [GDB SEGGER J-Link Debugging] and click the [New launch configuration] button to create a new debug configuration.
By default, the name of the debug configuration will be added Debug to the project name, but you can change it if needed.
In this tutorial, to clearly indicate that the configuration is for J-Link, the name is set to BlinkLED Debug J-Link.

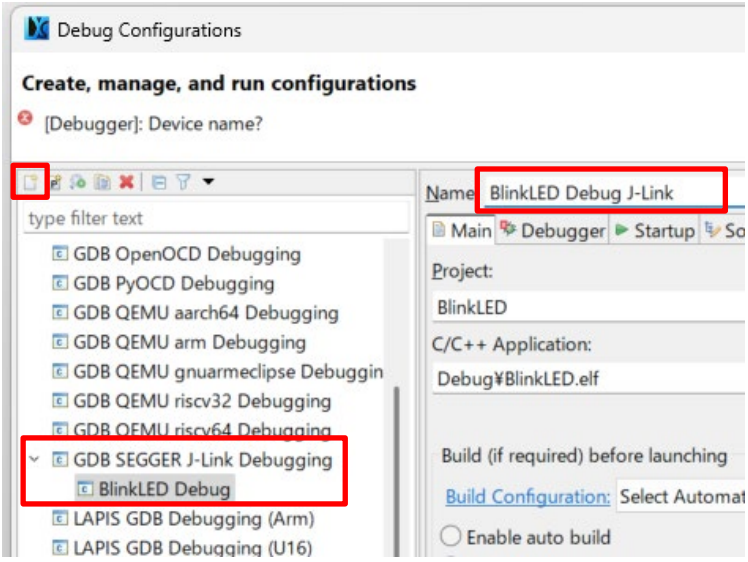

Figure 3-31. Creating a New Debug Configuration

After creating the debug configuration, additional settings need to be configured.
These settings are explained below.

In the [Debugger] tab, set the device name to ML63Q25x7.

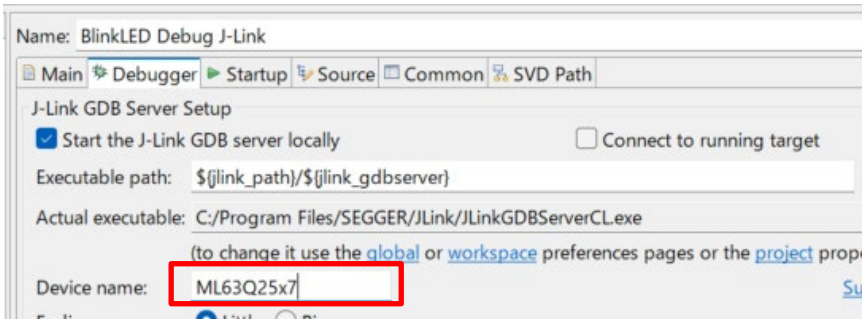

Figure 3-32. Setting the Device Name

In the [Debugger] tab, set the Executable name to the GDB client executable file.
Click the [Browse] button, and in the displayed [Select GDB Client Binary] dialog, navigate to:
LAPIS > LEXIDE > gdb
Then select arm-none-eabi-gdb.exe and click the [Open] button.

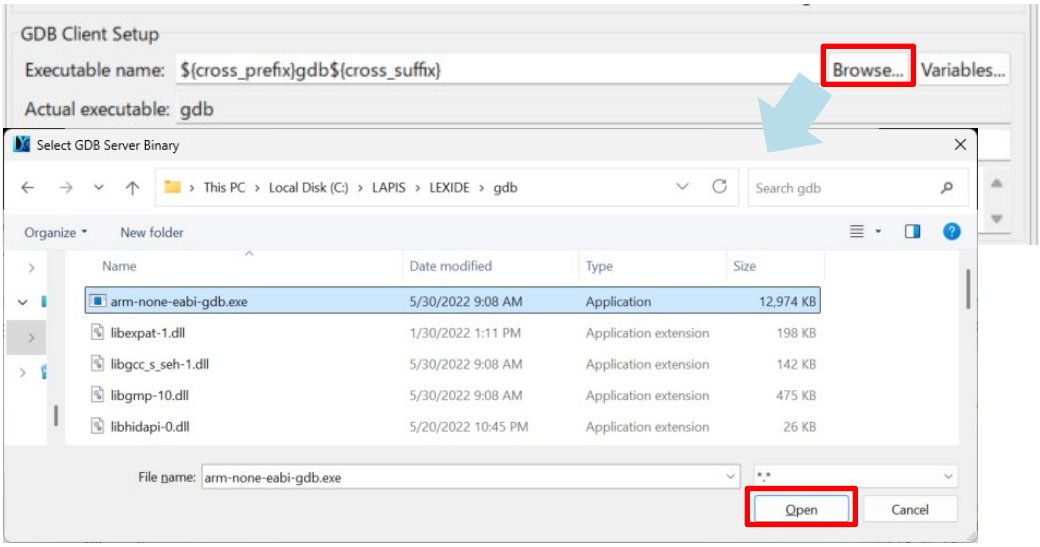

Figure 3-33. Setting the GDB Client Executable File Name

In the [SVD Path] tab, set the File path to the location of the SVD file.
By configuring the SVD file path, you will be able to view each peripheral register in the debugger.
Click the [Browse] button, and in the displayed [Select SVD File] dialog, navigate to the folder where the device information file is installed.
Then select the SVD file (in this tutorial, ML63Q25x7.svd) and click the [Open] button.

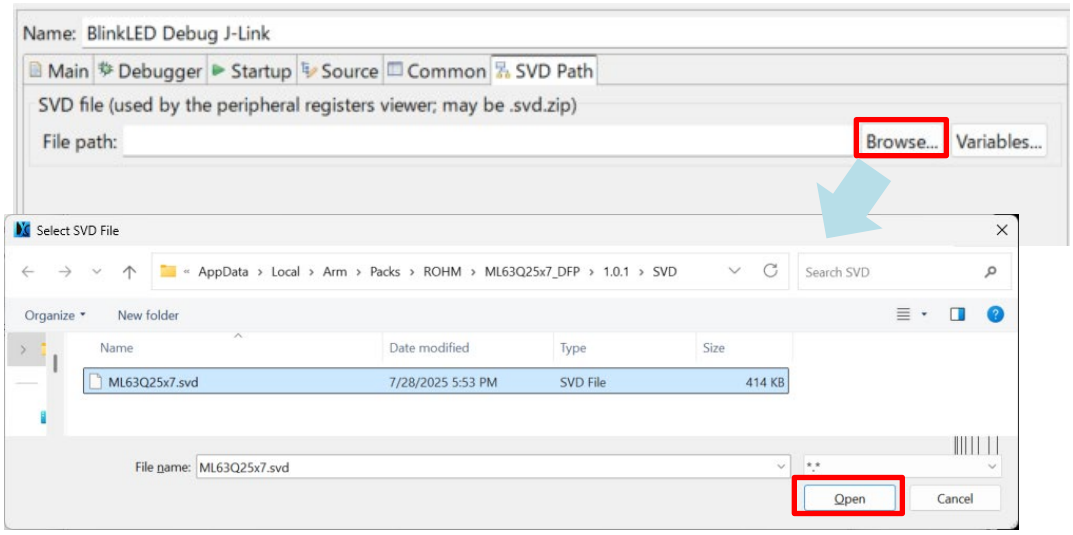

Figure 3-34. Setting the SVD File Path

Click the [Close] button to close the [Debug Configurations] dialog. The other settings can remain at their default values.

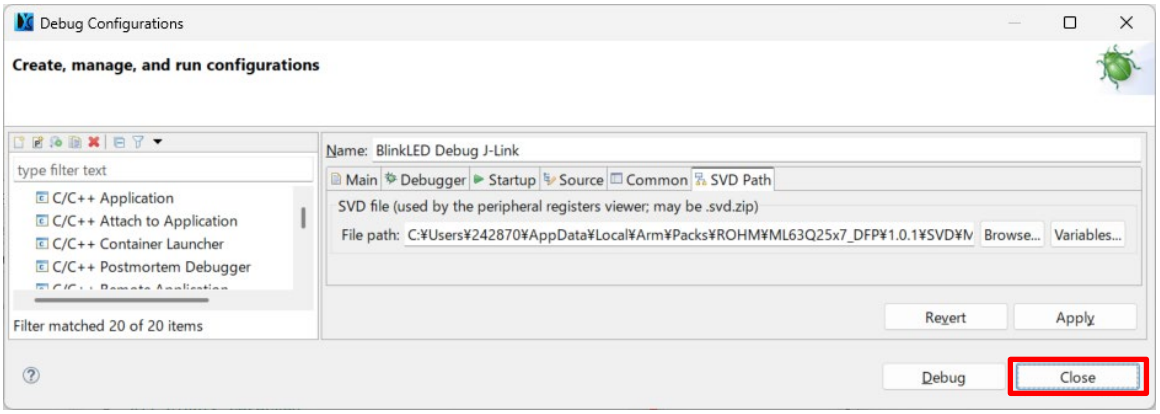

Figure 3-35. Closing the [Debug Configuration] Dialog

When the following dialog appears, click the [Save] button to save the settings.

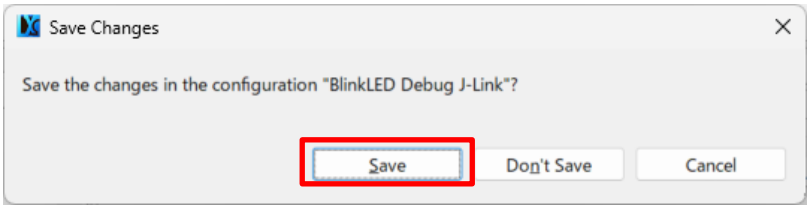

Figure 3-36. Confirmation Dialog When Closing [Debug Configuration]

3.8.3 Starting the Debugging Process

Connect the PC, J-Link, reference board, and USB to SPI conversion module as shown below.

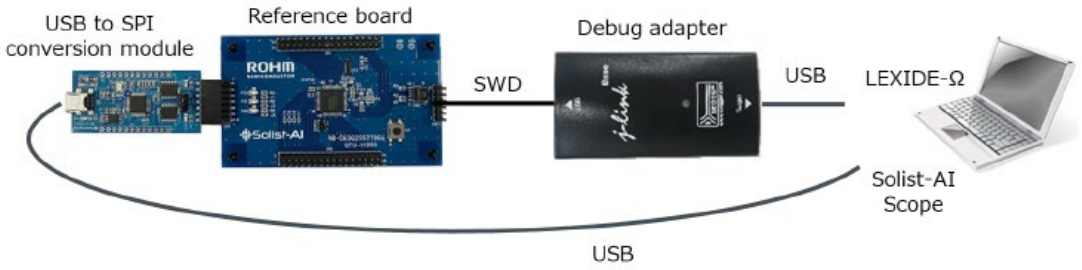

Figure 3-37. Debugging Connection Setup

To start debugging, select the debug configuration in [Launch Configuration], choose [Debug] in [Launch Mode], and click the [Launch in 'Debug' mode] button .

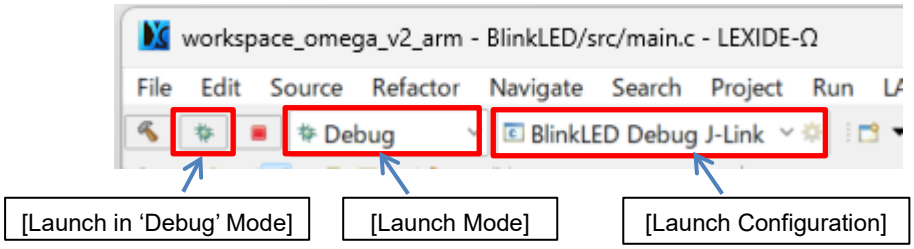

Figure 3-38. Starting the Debugging Process

When the dialog confirming the perspective switch appears, click the [Switch] button.

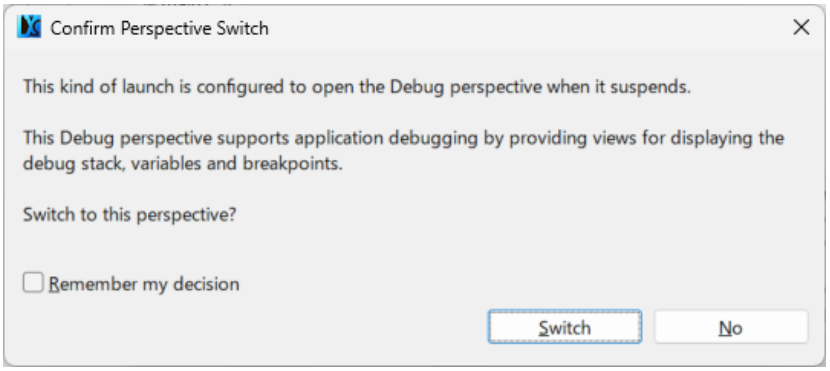

Figure 3-39. Perspective Switch Confirmation Dialog

The LEXIDE-Ω screen switches to the debugging perspective (hereafter referred to as the [Debug] perspective). The program breaks at the beginning of the main function after execution reaches that point.

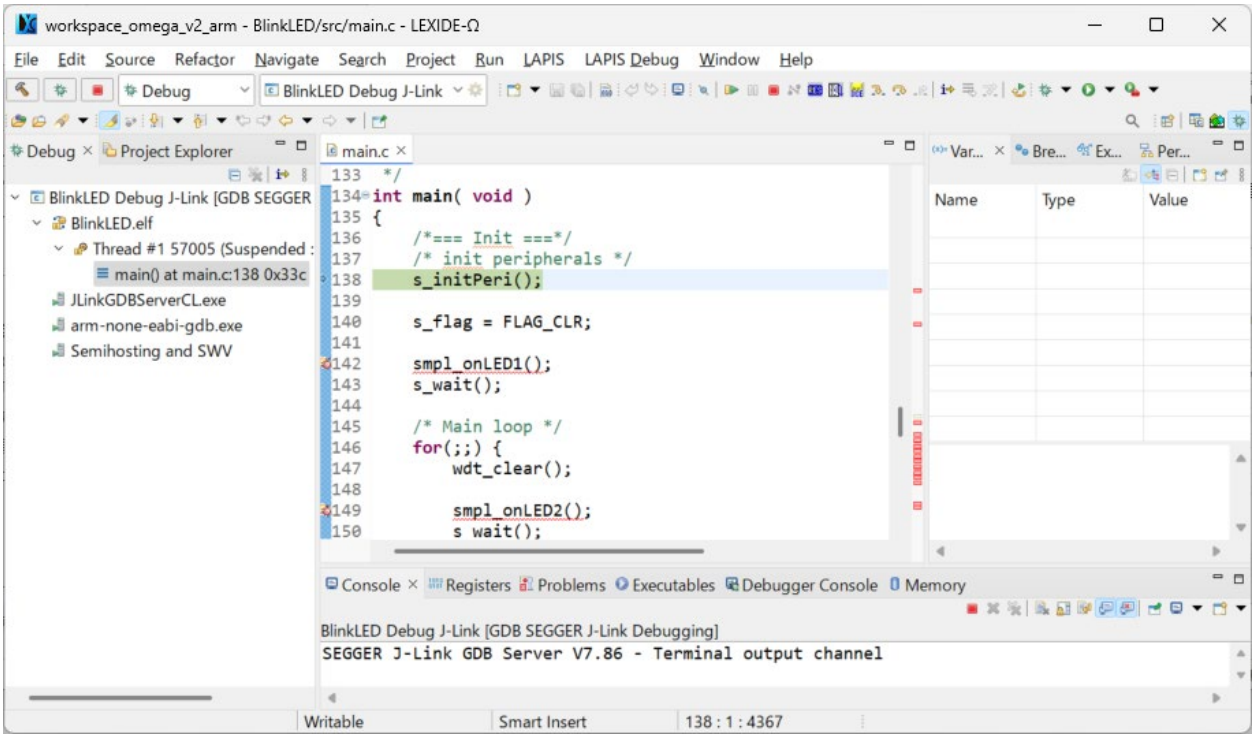

Figure 3-40. Debug Perspective View

3.8.4 Starting and Resuming Program Execution

Click the [Resume (F8)] button on the toolbar.
The program will start running, and you can observe the LED lighting pattern (P50, P51, P52) on the reference board changing.

Figure 3-41. Program Execution Start/Resume

3.8.5 Stopping Program Execution

Click the [Suspend] button on the toolbar to stop the program execution.

Figure 3-42. Suspending Program Execution

3.8.6 Setting Breakpoints

When you want to stop the program at a specific location, set a breakpoint.
You can toggle a breakpoint by double-clicking the left margin of a source line.
When a breakpoint is set, a small round icon with a check mark appears in the left margin of the source line.
In this example, a breakpoint is set in the TBC interrupt function smpl_procTbcInt() in main.c.
Each time you click the [Resume (F8)] button, the program will break at the line where the breakpoint is set, and you can confirm that the LED lighting pattern changes.

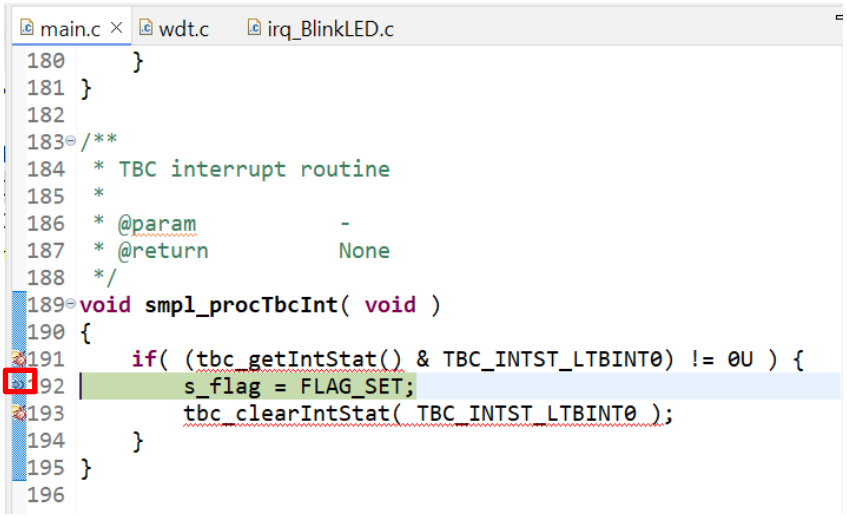

Figure 3-43. Setting Breakpoints

3.8.7 Ending the Debug Session

To end the debugging session, click the [Stop] button on the toolbar.

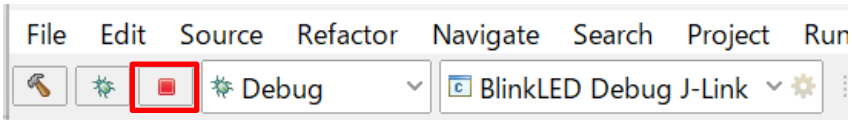

Figure 3-44. Ending the Debug Session

To return to the layout before debugging, click the [C/C++] perspective button on the right side of the toolbar to switch the display.

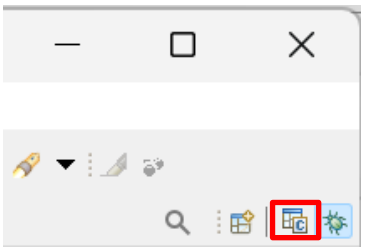

Figure 3-45. Switching to the [C/C++] Perspective

4. [Reference] About Firmware Development for Solist-AI™

Application Note “Anomaly Detection Using Accelerometers and AI Library,” Sample Program “AI Library for Anomaly Detection Using Accelerometers,” and “ML63Q2500 Series Reference Software” are available on the product pages for Solist-AI™ on the ROHM website (*1).
Please refer to these resources when developing firmware for Solist-AI™.

*1: They can be downloaded from the following page on the ROHM website:
https://www.rohm.com/products/micon/solist-ai/ml63q2500-group

Application Note

Title	Revision	Description	Last Updated
NEW Anomaly Detection Using Accelerometers and AI Library	1	This document describes how to use the AI library for anomaly detection using accelerometers.	2025/04/18

Sample Program

Title	Revision	Description	Last Updated
NEW ML63Q2500 Series Reference Software	1.2	This software includes peripheral drivers for the ML63Q2500 group and sample programs that run on the reference board. It can be used with LEXIDE-Ω.	2025/10/28
NEW Solist-AI™ Scope (Real-Time Waveform Display Tool)	1.2.11	This is a real-time waveform display tool for verifying the effects of Solist-AI™. To use it, you will need an adapter (such as the MM-FT232HC) to connect the Solist-AI™ microcontroller and the Solist-AI™ Scope. Please prepare the adapter yourself.	2025/10/28
NEW AI Libraries for Anomaly Detection Using Accelerometers	1	This is an AI library for anomaly detection using accelerometers.	2025/04/18

Figure 4-1. Excerpt of Application Note and Sample Program on Each Solist-AI™ Product Page

4.1 Control Flow of the Sample Program

The application note “Anomaly Detection Using an Accelerometers and AI Library” (hereinafter referred to as the application note) describes how to use the AI library for anomaly detection with an accelerometers.
The corresponding sample program for this application note is included in “Anomaly Detection Using an Accelerometer and AI Library” (hereinafter referred to as the AnomalyDetectionDemo sample program).
The control flow of the AnomalyDetectionDemo sample program is as follows.

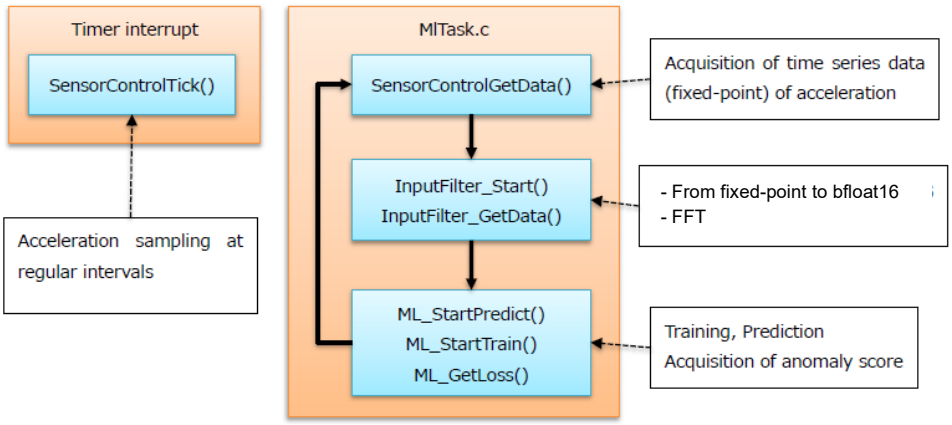

Figure 4-2. AnomalyDetectionDemo Sample Program Control Flow (Excerpt from the Application Note)

The AnomalyDetectionDemo sample program implements initial learning, prediction, and sequential learning, and the flow for each is as follows.

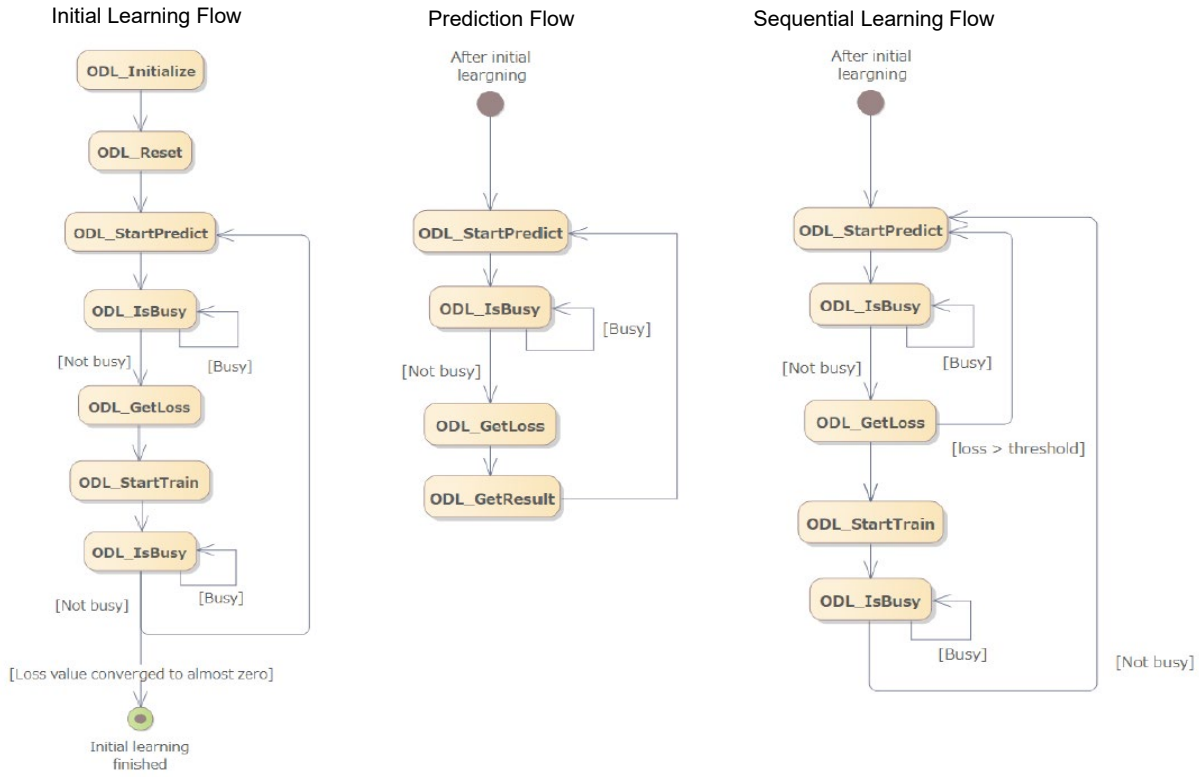

Figure 4-3. Control Flows for Initial Learning, Prediction, and Sequential Learning
(Excerpt from the Application Note)

For information on the APIs of the Solist-AI™ library used in the AnomalyDetectionDemo sample program, please refer to “3. Solist-AI™ Library” in the application note.
Additionally, refer to “2.1 API List” in the Startup Manual of the ML63Q2500 Series Reference Software (FEXT63Q2500_REFSOFT_STARTUP-xx.pdf) (hereinafter referred to as the Startup Manual), as well as FEX63Q2500_REFSOFT_DRIVER_API-xx.chm located in the Document folder of the Reference Software.

Please also refer to “3.2 Solist-AI Sample Program” in the Startup Manual and the source code located in the corresponding folder under SourceCode\Samples\Solist-AI.

4.2 How to Load (Import) the Sample Program

The sample programs in “AI Libraries for Anomaly Detection Using Accelerometers” and “ML63Q2500 Series Reference Software” include project files for LEXIDE-Ω, which can be imported into LEXIDE-Ω.
As an example, the following steps show how to import the sample program from “AI Library for Anomaly Detection Using Accelerometers”:

1. In LEXIDE-Ω, go to [File] > [Import], then in the displayed dialog, select [General] > [Existing Projects into Workspace] and click [Next].

2. Under [Select root directory], specify the folder containing the sample program’s project files (Solist-AI_FW_AnomalyDetectionSample\samples\Solist-AI\AnomalyDetectionDemo), then click [Finish] to load the project.

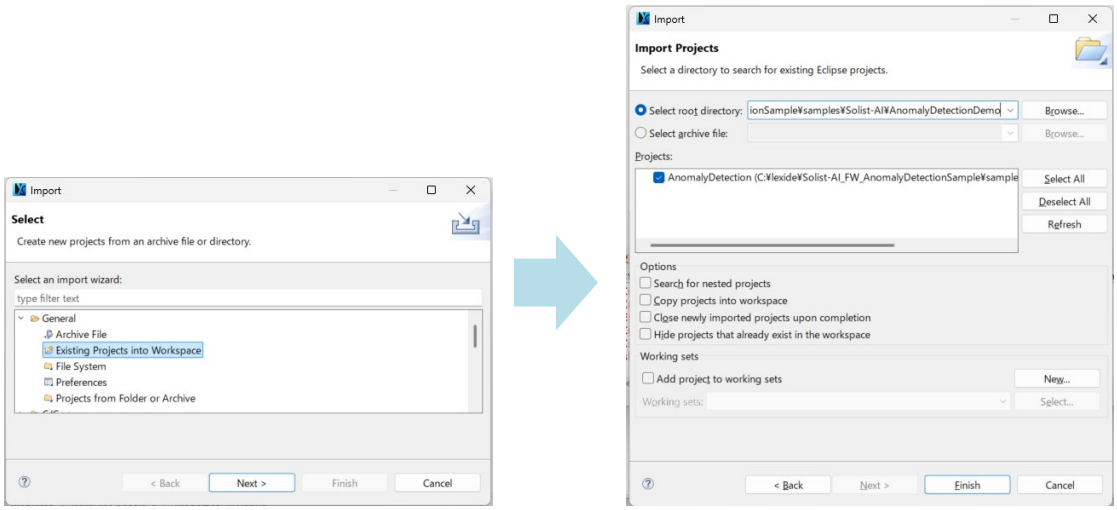

Figure 4-4. Importing the Sample Program Project

[Note]
The sample program is designed for the KXTJ3-1057 accelerometer.
If you are using an accelerometer other than KXTJ3-1057, or if no sensor is connected, Solist-AI™ Scope cannot be launched.
Please ensure that Solist-AI™ Scope is started only when data can be correctly received from the sensor.

When using an accelerometer other than KXTJ3-1057, you will need to modify the program to match your sensor.
Sensor-related source files are located in Solist-AI_FW_AnomalyDetectionSample\sensor. Please refer to these files for implementation.

The source file for KXTJ3-1057 is kxtj3.c.
The following is an overview of the functions defined in this source file.

Function Name	Description
int8_t Kxtj3Init(bool addr_pin)	Initialize the sensor device
int8_t Kxtj3Start(void)	Start the sensor device operation
int8_t Kxtj3Stop(void)	Stop the sensor device operation
int8_t Kxtj3RawGetVal(int16_t *raw)	Acquire data from the sensor
int8_t Kxtj3GetVal(int16_t *acc)	Scale the acquired data from the sensor
static int8_t InitCheckVal(void)	Read the WHO_AM_I register and verify the device ID
static int8_t InitSettingIc(uint8_t *checkval)	Configure the sensor device
static int8_t InitSettingGval(uint8_t checkval, int16_t*gval)	Set the acceleration range of the sensor device
static int8_t Write(uint8_t reg, uint8_t *data, uint16_t size)	Write to the sensor device
static int8_t Read(uint8_t reg, uint8_t *data, uint16_t size)	Read from the sensor device

4.3 Waveform Display Using LEXIDE Scope

When using a CMSIS-DAP compliant debug adapter, LEXIDE Scope allows you to read RAM and SFR values via SWD during program execution and display these values and graphs in real time. This enables you to use LEXIDE Scope as a an alternative tool to Solist-AI™ Scope.
Moreover, LEXIDE Scope can be used without requiring a USB-SPI conversion module such as the MM-FT232H.
The configuration of the development environment when using LEXIDE Scope is shown below.

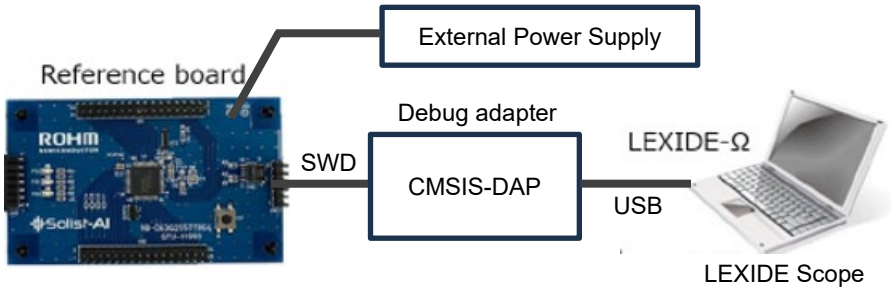

Figure 4-5. Development Environment Configuration When Using LEXIDE Scope

For details on how to use LEXIDE Scope, please refer to “6.9.1 LEXIDE Scope” in the LEXIDE-Ω for ARM Core Microcontrollers User’s Manual.

4.3.1 Debug Configuration

When using a CMSIS-DAP debug adapter, you need to select [LAPIS GDB Debugging (Arm)] in the Debug Configuration of LEXIDE-Ω.

In the [Debug Configuration] dialog, select [LAPIS GDB Debugging (Arm)] and click the [New launch configuration] button to create a new debug configuration.
By default, the name of the debug configuration will be <ProjectName> + Debug, but it can be changed.
In this tutorial, to clearly indicate that it is a debug configuration for CMSIS-DAP, we use AnomalyDetection Debug CMSIS-DAP.

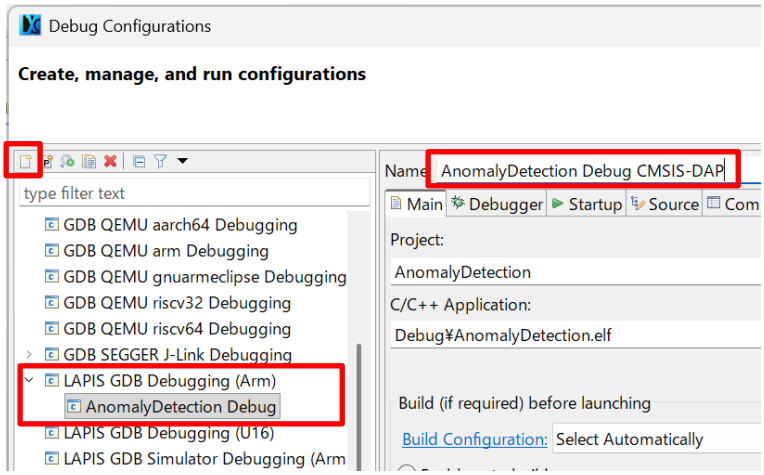

Figure 4-6. Creating a New Debug Configuration for CMSIS-DAP

In the [Debugger] tab, ensure that [Target ICE] is set to [CMSIS-DAP]. Then, under [LEXIDE Scope Setup], select the [Launch LEXIDE Scope] option.
All other settings can generally remain at their default values.

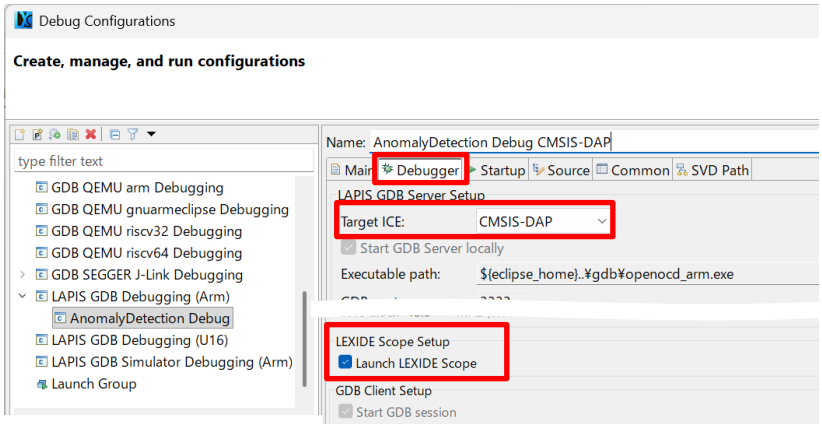

Figure 4-7. LEXIDE Scope Launch Settings

4.3.2 Launching LEXIDE Scope

LEXIDE Scope is launched from LEXIDE-Ω.
First, start the debugging process.
In [Launch Configuration], select the debug configuration (in this tutorial, AnomalyDetection Debug CMSIS-DAP), then choose [Debug] under [Launch Mode], and click the [Launch in 'Debug' mode] button .

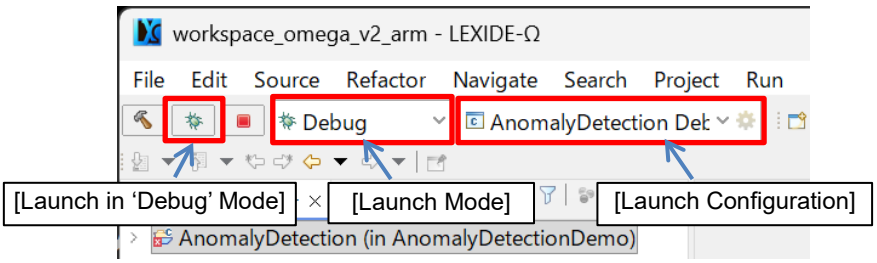

Figure 4-8. Start Debugging

Once debugging starts and hits a breakpoint, go to [LAPIS Debug] > [Launch LEXIDE Scope] in LEXIDE-Ω to launch LEXIDE Scope.

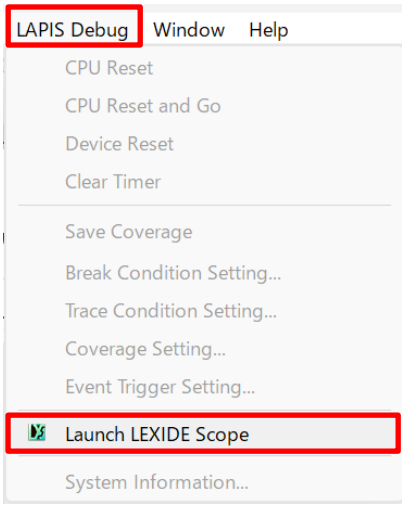

Figure 4-9. Launching LEXIDE Scope

Figure 4-10. LEXIDE Scope Startup Screen (Main Window)

4.3.3 How to Use LEXIDE Scope

The basic steps for using LEXIDE Scope are as follows:

Step1: Configure the Mode
When using CMSIS-DAP, go to [Settings] > [Mode Settings] in LEXIDE Scope and set the mode to [Universal Mode].

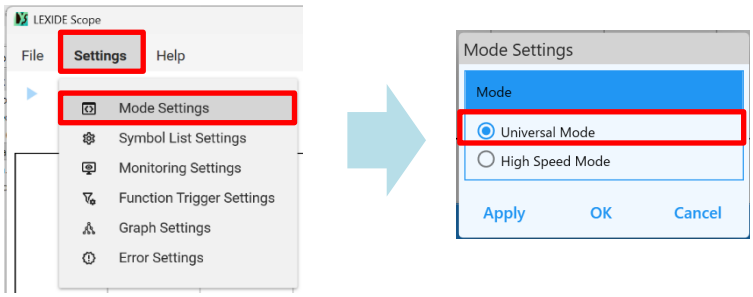

Figure 4-11. LEXIDE Scope Mode Settings

Step 2: Add Measurement Points
In the main window, click [Add] under [Measurement Point]. In the [Add Measurement Point] dialog that appears, select the desired measurement points and click [Add] to include them.
In this tutorial, add i2AccOutX_, i2AccOutY_, i2AccOutZ_, and output_loss.
Once all points have been added, click [Close] to close the dialog.

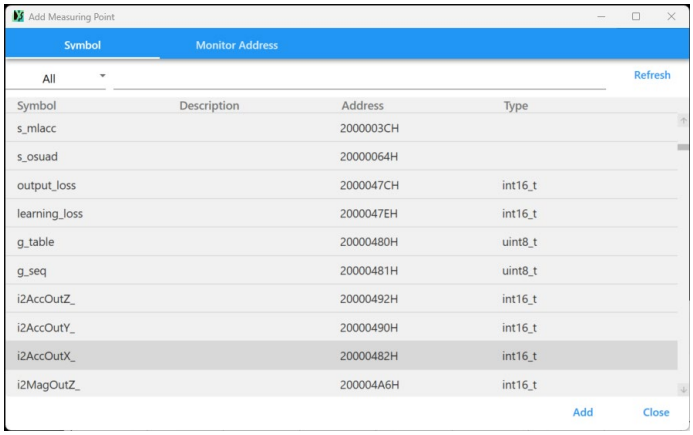

Figure 4-12. Adding Measurement Points

Step 3: Start Monitoring
Click the monitor start button ▶ in the main window to begin monitoring.
Next, click the [Resume (F8)] button ▶ on the LEXIDE-Ω toolbar to resume execution.
After completing these steps, you can observe real-time waveform changes.
Adjust [Y-Resolution] and [Y-Offset] under [Measurement Point] as needed.

Figure 4-13. Execution Result

Revision History

Document No.	Issue Date	Page		Description
		Previous Edition	Current Edition	
FEXT63Q2500_LEXIDE_TUTORIAL-01	Dec. 26, 2025	–	–	The first edition

Notice

1) The information contained in this document is intended to introduce ROHM Group (hereafter referred to asROHM) products. When using ROHM products, please verify the latest specifications or datasheets before use.

2) ROHM products are designed and manufactured for use in general electronic equipment and applications (such as Audio Visual equipment, Office Automation equipment, telecommunication equipment, home appliances, amusement devices, etc.) or specified in the datasheets. Therefore, please contact the ROHM sales representative before using ROHM products in equipment or devices requiring extremely high reliability and whose failure or malfunction may cause danger or injury to human life or body or other serious damage (such as medical equipment, transportation, traffic, aircraft, spacecraft, nuclear power controllers, fuel control, automotive equipment including car accessories, etc. hereafter referred to as Specific Applications). Unless otherwise agreed in writing by ROHM in advance, ROHM shall not be in any way responsible or liable for any damages, expenses, or losses incurred by you or third parties arising from the use of ROHM Products for Specific Applications.

3) Electronic components, including semiconductors, can fail or malfunction at a certain rate. Please be sure to implement, at your own responsibilities, adequate safety measures including but not limited to fail-safe design against physical injury, and damage to any property, which a failure or malfunction of products may cause.

4) The information contained in this document, including application circuit examples and their constants, is intended to explain the standard operation and usage of ROHM products, and is not intended to guarantee, either explicitly or implicitly, the operation of the product in the actual equipment it will be used. As a result, you are solely responsible for it, and you must exercise your own independent verification and judgment in the use of such information contained in this document. ROHM shall not be in any way responsible or liable for any damages, expenses, or losses incurred by you or third parties arising from the use of such information.

5) When exporting ROHM products or technologies described in this document to other countries, you must abide by the procedures and provisions stipulated in all applicable export laws and regulations, such as the Foreign Exchange and Foreign Trade Act and the US Export Administration Regulations, and follow the necessary procedures in accordance with these provisions.

6) The technical information and data described in this document, including typical application circuits, are examples only and are not intended to guarantee to be free from infringement of third parties intellectual property or other rights. ROHM does not grant any license, express or implied, to implement, use, or exploit any intellectual property or other rights owned or controlled by ROHM or any third parties with respect to the information contained herein.

7) No part of this document may be reprinted or reproduced in any form by any means without the prior written consent of ROHM.

8) All information contained in this document is current as of the date of publication and subject to change without notice. Before purchasing or using ROHM products, please confirm the latest information with the ROHM sales representative.

9) ROHM does not warrant that the information contained herein is error-free. ROHM shall not be in any way responsible or liable for any damages, expenses, or losses incurred by you or third parties resulting from errors contained in this document.

ROHM SEMICONDUCTOR

Thank you for your accessing to ROHM product informations.
More detail product informations and catalogs are available, please contact us.

ROHM Customer Support System

https://www.rohm.com/contactus