

# **GR551x IAR User Manual**

Version: 1.3

Release Date: 2022-02-20

#### Copyright © 2022 Shenzhen Goodix Technology Co., Ltd. All rights reserved.

Any excerption, backup, modification, translation, transmission or commercial use of this document or any portion of this document, in any form or by any means, without the prior written consent of Shenzhen Goodix Technology Co., Ltd. is prohibited.

#### **Trademarks and Permissions**

**GODIX** and other Goodix trademarks are trademarks of Shenzhen Goodix Technology Co., Ltd. All other trademarks and trade names mentioned in this document are the property of their respective holders.

#### Disclaimer

Information contained in this document is intended for your convenience only and is subject to change without prior notice. It is your responsibility to ensure its application complies with technical specifications.

Shenzhen Goodix Technology Co., Ltd. (hereafter referred to as "Goodix") makes no representation or guarantee for this information, express or implied, oral or written, statutory or otherwise, including but not limited to representation or guarantee for its application, quality, performance, merchantability or fitness for a particular purpose. Goodix shall assume no responsibility for this information and relevant consequences arising out of the use of such information.

Without written consent of Goodix, it is prohibited to use Goodix products as critical components in any life support system. Under the protection of Goodix intellectual property rights, no license may be transferred implicitly or by any other means.

#### Shenzhen Goodix Technology Co., Ltd.

Headquarters: 2F. & 13F., Tower B, Tengfei Industrial Building, Futian Free Trade Zone, Shenzhen, China

TEL: +86-755-33338828 FAX: +86-755-33338099

Website: www.goodix.com



### **Preface**

### **Purpose**

This document introduces methods to develop and to debug GR551x Bluetooth Low Energy (Bluetooth LE) applications based on IAR Embedded Workbench IDE, to help users quickly get started with secondary development of GR551x Software Development Kit (SDK) applications.

#### **Audience**

This document is intended for:

- GR551x user
- GR551x developer
- GR551x tester
- Hobbyist developer
- Technical writer

#### **Release Notes**

This document is the fourth release of *GR551x IAR User Manual*, corresponding to GR551x System-on-Chip (SoC) series.

#### **Revision History**

Version	Date	Description
1.0	2021-03-08	Initial release
1.1	2021-06-26	<ul> <li>Updated descriptions in "Configuring an Application Project".</li> <li>Updated SoC model descriptions.</li> <li>Optimized "Installation Instructions".</li> </ul>
1.2	2021-08-09	<ul><li>Updated the section "Installation Requirements".</li><li>Modified SoC model descriptions.</li></ul>
1.3	2022-02-20	Modified two sections: "Configuring an Application Project" and "Compiling and Downloading a Project".



# **Contents**

Preface	
1 Introduction	1
2 Installation Instructions	2
2.1 Preparation	2
2.2 Software Installation	
3 Configurations for SDK Applications	
3.1 Building an Application Project	
3.2 Configuring an Application Project	5
3.3 Compiling and Downloading a Project	
3.4 Project Debugging	11
4 FAQ	12
4.1 Why Does an "Undefined Variable" Error Occur During Project Compilation?	12



## 1 Introduction

IAR Embedded Workbench IDE for Arm (IAR EWARM, mentioned as IAR below) is an integrated development environment (IDE) built by IAR Systems, supporting 8-bit, 16-bit, and 32-bit microprocessors and microcontrollers, and providing more user-friendly experience among ARM IDEs. IAR features:

- Highly optimized C/C++ Compiler
- IAR XLINK Linker supporting common applications
- Powerful support for project management
- C-SPY Debugger, a fully integrated debugger with performance analysis, power visualization, and RTOS plugins

This document serves as a roadmap for users building GR551x application development environment with IAR on Windows PCs, so as to create, compile, download, and debug Bluetooth Low Energy (Bluetooth LE) applications, with an example provided as reference.

Before getting started, you can refer to the following documents.

Table 1-1 Reference documents

Name	Description
GR551x Developer Guide	Introduces GR551x Software Development Kit (SDK) and how to develop and debug applications based on the SDK.
J-Link/J-Trace User Guide	Provides J-Link operational instructions.  Available at <a href="http://www.segger.com/downloads/jlink/UM08001">http://www.segger.com/downloads/jlink/UM08001</a> JLink.pdf.
Bluetooth Core Spec	Offers official Bluetooth standards and core specification from Bluetooth SIG.
Bluetooth GATT Spec	Provides details about Bluetooth profiles and services.  Available at <a href="https://www.bluetooth.com/specifications/gatt">https://www.bluetooth.com/specifications/gatt</a> .
IAR Embedded Workbench IDE User Guides	Provides various information about IAR IDE.  Available at: <a href="https://www.iar.com/support/user-guides/user-guide-iar-embedded-workbench-for-arm/">https://www.iar.com/support/user-guides/user-guide-iar-embedded-workbench-for-arm/</a> .



### 2 Installation Instructions

This chapter describes the requirements and process for installing IAR.

## 2.1 Preparation

#### • Hardware preparation

Table 2-1 Hardware preparation

Name	Description
Development board	GR5515 Starter Kit Board (SK Board)
Connection cable	Micro USB 2.0 cable

#### Software preparation

Table 2-2 Software preparation

Name	Description
Windows	Windows 7/Windows 10
IAR Embedded Workbench IDE for Arm (IAR EWARM)	IAR EWARM 8.2.22 or later versions
J-Link driver	A J-Link driver. Available at www.segger.com/downloads/jlink/.
Connection cable	Micro USB 2.0 cable

### 2.2 Software Installation

Download the IAR installation file (IAR for ARM 8.2.22 or later versions) on IAR Systems official website.

Double-click the .exe installation file. Choose **Install IAR Embedded Workbench** \* **for Arm**, and follow the prompts to finish installation.

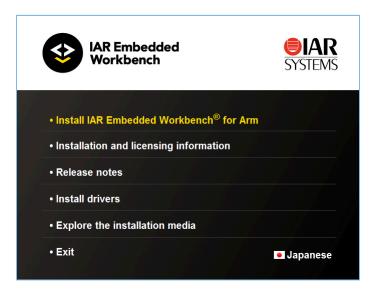


Figure 2-1 IAR installation page



IAR will be launched automatically after successful installation. The main user interface is shown as below:

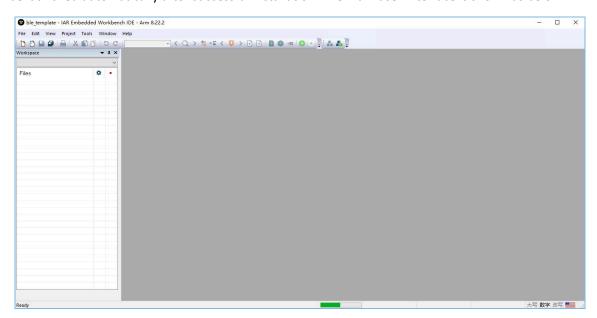


Figure 2-2 IAR main user interface

Users can choose to install the necessary software as prompted after installing IAR.



# **3 Configurations for SDK Applications**

This chapter elaborates on how to configure an IAR project, to help users quickly get familiar with and use IAR projects with GR551x SDK.

### Note:

- SDK Folder is the root directory of GR551x SDK.
- IAR\_Folder is the root directory of IAR installation.

## 3.1 Building an Application Project

This section introduces how to generate an IAR application project as needed based on an existing project (ble\_app\_template as an example, available in SDK\_Folder\projects\ble\ble\_peripheral\ble\_app\_t emplate\IAR).

- 1. Open SDK\_Folder\projects\ble\ble\_peripheral\, and copy ble\_app\_template to the directory.

  Rename the project to ble\_app\_example. In ble\_app\_example\IAR, change the file names of the .ewp file, the .ewt file, and the .eww file to ble\_app\_example.
- 2. Open *ble\_app\_example.eww* in text format, and replace ble\_app\_template (as shown in the figure below) with ble\_app\_example.

Figure 3-1 Identifying and replacing ble\_app\_template

3. After the modification completes, double-click ble\_app\_example.eww to enter the IAR main user interface.



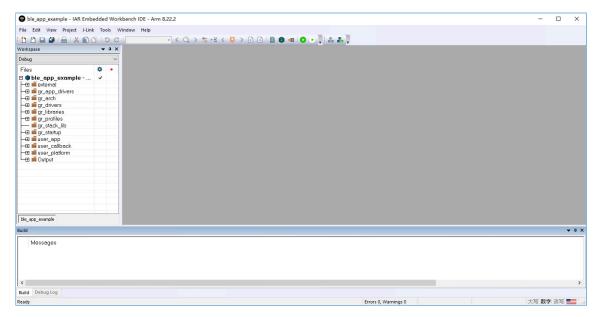


Figure 3-2 Main user interface when an example project opened in IAR

### 3.2 Configuring an Application Project

Copy the IAR download algorithm files to an appropriate directory before configuring an application project.

1. Find the IAR download algorithm files (GR55xx\_IAR\_8M.board, GR55xx\_IAR\_8M.flash, and GR55xx\_IAR\_flashloader\_8M.out) in SDK\_Folder\build\binaries\download\_algorithm\IAR.

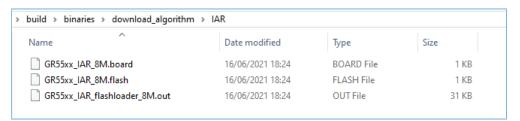


Figure 3-3 IAR download algorithm files

2. Create a new folder (such as Goodix) in IAR\_Folder\arm\config\flashloader, and copy the three algorithm files mentioned in Step 1 to the new directory, as shown in the figure below.

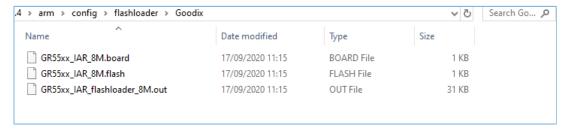


Figure 3-4 Copying the algorithm files

After copying the algorithm files to the target directory, users can start to configure the application project.

1. Right-click the project name in the **Workspace** pane and choose **Options** to start configuration.



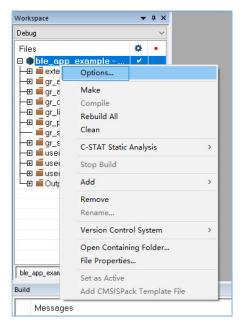


Figure 3-5 To configure the project

(1). Configure project compilation items.

In the C/C++ Compiler > Preprocessor pane, configure Additional include directories and Defined symbols in the pop-up window, as shown in the figure below.

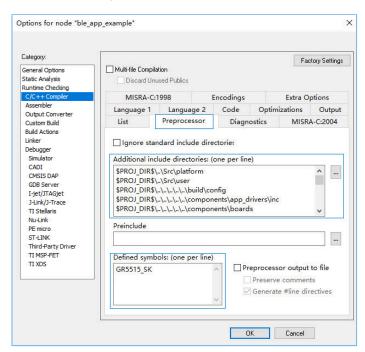


Figure 3-6 Project compilation

Note:

\$PROJ\_DIR\$ in the Additional include directories pane is the environment variable for IAR project file path.



#### (2). Configure Linker configuration file.

Configure Linker configuration file in the Linker > Config pane, as shown in the figure below. Choose the IAR linker configuration file provided in the SDK, available in SDK\_Folder\toolchain\gr551x\sour ce\iar\gr5515.icf. Click **OK** after the file is selected.

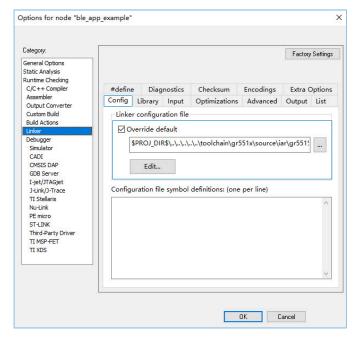


Figure 3-7 Choosing the linker configuration file

In the **Linker > Output** pane, modify the file names of the .out file and the .map file (generated when building the project) to ble\_app\_example. Click **OK** after modification completes.

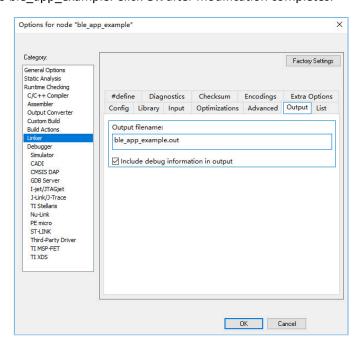


Figure 3-8 Modifying the name of a file generated when building the project



(3). Configure the output file.

Configure the format and the name of the file (generated when building the project) in the **Output Converter > Output** pane.

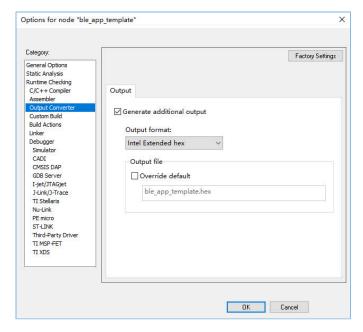


Figure 3-9 Modifying format and name of the output file

(4). Configure the download algorithm.

Enter the **Debugger > Download** pane, and choose the directory of IAR download algorithms mentioned in Step 2 (IAR\_Folder\arm\config\flashloader\Goodix\GR55xx\_IAR\_8M.board), to choose the SoC download algorithm.

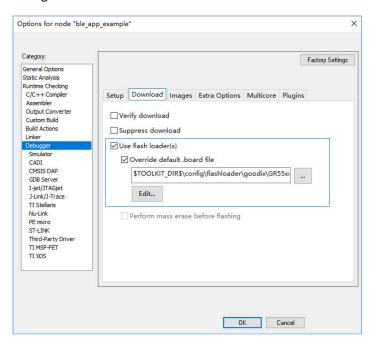


Figure 3-10 Configuring IAR download algorithm





**\$TOOLKIT\_DIR\$** in the **Use flash loader(s)** pane is the environment variable for IAR installation path.



# 3.3 Compiling and Downloading a Project

- Click Project on the menu bar, and choose Make or Rebuild All to compile firmware. Wait until the compilation completes.
- Choose Project > Download > Download active application, as shown in Figure 3-11 to complete a firmware file download.

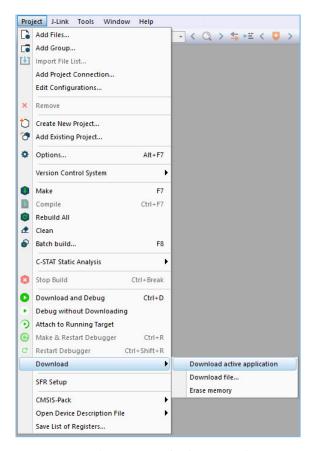


Figure 3-11 Choosing **Download active application** 

3. Choose the device to be connected to for the first download operation of firmware on the current project. When the dialog box shown below pops up, click **OK** to choose the device.

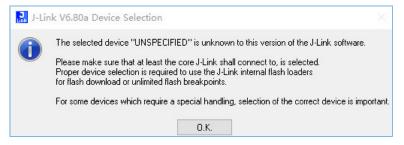


Figure 3-12 J-Link device selection window

4. Choose the device **Cortex-M4**, and click **OK** to start download.



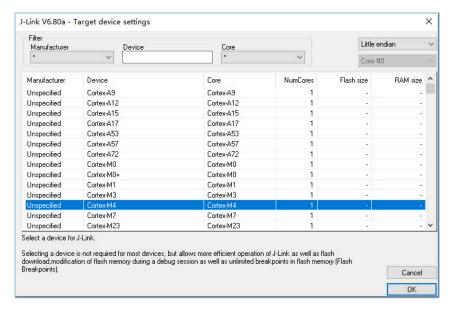


Figure 3-13 Selecting "Cortex-M4"

5. Press the **RESET** button on the SK board after download completes, to start to run the firmware on the SoC.

### 3.4 Project Debugging

Before project debugging, make sure the project firmware is programmed to the SoC by following the instructions above.

After the firmware is programmed to the SoC, click **Debug** on the menu bar to enter the debug mode. The program will run to the main() function line.

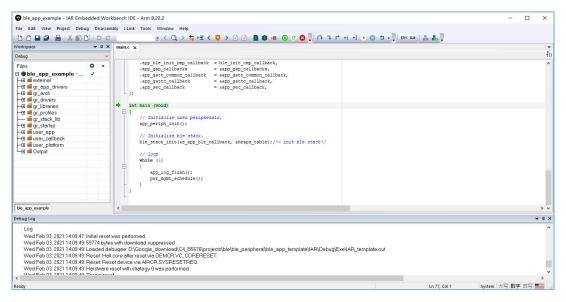


Figure 3-14 Entering the debug mode

In the debug mode, users can debug the project by breakpoints and single stepping.



### 4 FAQ

This chapter describes possible problems, reasons, and solutions when using an IAR example.

# 4.1 Why Does an "Undefined Variable" Error Occur During Project Compilation?

Description

When I try to compile the an IAR project, an "undefined variable" error occurs.

Analysis

This may be because some project macros (such as GR5515\_SK and APP\_DRIVER\_USE\_ENABLE) have not been defined prior to project compilation, and thus the corresponding contents in the source file are not compiled. Users can check whether there is any macro left undefined by comparing the project macros in Keil.

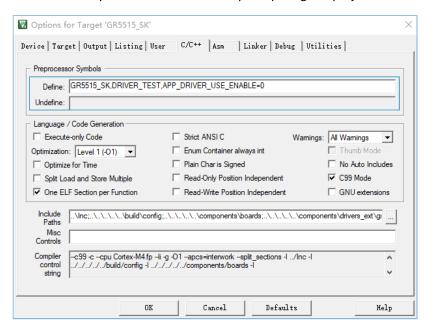


Figure 4-1 Project macro configuration in Keil

Solution

Add the undefined macro in the C/C++ Compiler > Preprocessor pane in IAR, as shown in the figure below.



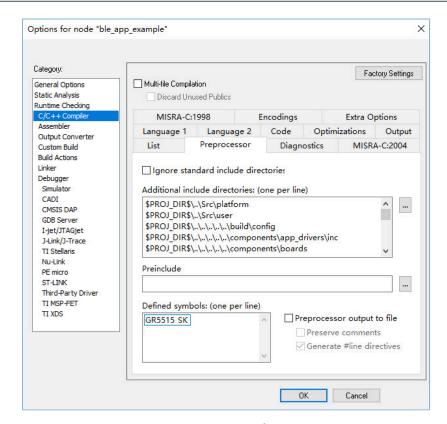


Figure 4-2 Adding an undefined macro