



## **GR5xx IAR User Manual**

**Version: 1.1**

**Release Date: 2024-08-23**

**Copyright © 2024 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**

**GOODiX** 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: Floor 13, Phase B, Tengfei Industrial Building, Futian Free Trade Zone, Shenzhen, China

TEL: +86-755-33338828      Zip Code: 518000

Website: [www.goodix.com](http://www.goodix.com)

# Preface

## Purpose

This document introduces methods to develop and to debug GR5xx Bluetooth Low Energy (Bluetooth LE) applications based on IAR Embedded Workbench IDE, to help users quickly get started with secondary development of GR5xx Software Development Kit (SDK) applications.

## Audience

This document is intended for:

- Device user
- Developer
- Test engineer
- Technical support engineer

## Release Notes

This document is the second release of *GR5xx IAR User Manual*, corresponding to Bluetooth LE GR5xx System-on-Chip (SoC) series.

## Revision History

Version	Date	Description
1.0	2023-12-27	Initial release
1.1	2024-08-23	Upgraded the supported version of IAR.

# Contents

<b>Preface</b> .....	<b>I</b>
<b>1 Introduction</b> .....	<b>1</b>
<b>2 Installation Instructions</b> .....	<b>2</b>
2.1 Preparation.....	2
2.2 Software Installation.....	2
<b>3 Compiling IAR Example Project in SDK</b> .....	<b>4</b>
3.1 Building an IAR Project.....	4
3.2 Compiling an IAR Project.....	4
3.3 Downloading Program.....	5
<b>4 FAQ</b> .....	<b>6</b>
4.1 Why Does an "Undefined Variable" Error Occur During Project Compilation?.....	6

# 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 GR5xx 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
Developer guide of the specific GR5xx SoC	Introduces GR5xx 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="https://www.segger.com/downloads/jlink/UM08001_JLink.pdf">https://www.segger.com/downloads/jlink/UM08001_JLink.pdf</a> .
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	Starter Kit Board (SK Board) of the corresponding SoC
Connection cable	USB Type-C cable (Micro USB 2.0 cable for GR551x SoCs)

- **Software preparation**

Table 2-2 Software preparation

Name	Description
Windows	Windows 7/Windows 10
IAR Embedded Workbench IDE for Arm (IAR EWARM)	IAR EWARM 9.40.1 or later versions
J-Link driver	A J-Link driver. Available at <a href="https://www.segger.com/downloads/jlink/">https://www.segger.com/downloads/jlink/</a> .
Python	Used to build an environment for executing scripts for GR5xx projects Required version: 3.0 or above Available at <a href="https://www.python.org/downloads/">https://www.python.org/downloads/</a> .

### 2.2 Software Installation

- IAR

1. Download the IAR installation file (IAR for ARM 9.40.1 or later versions) on [IAR Systems official website](#).
2. Double-click the .exe installation file. Choose **Install IAR Embedded Workbench® for Arm**, and follow the prompts to finish installation.
3. You can choose to install the necessary software as prompted after installing IAR.

- Python

1. Download and install Python from the [official website](#). Choose the appropriate version for the local operating system.
2. Follow the instructions in the installation wizard.
3. Set the environment variables as needed.

After installing Python and setting environment variables, users can check whether the environment variables are set successfully by entering `python` in **Command Prompt**. If Python version information is displayed, the environment variables are set successfully.

```
$ python
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> _
```

Figure 2-1 Python version information displayed

## 3 Compiling IAR Example Project in SDK

This chapter elaborates on how to generate, use, and compile an IAR project with an example project `ble_app_template`.

### Note:

`SDK_Folder` is the root directory of GR5xx SDK.

### 3.1 Building an IAR Project

By default, the application example projects in GR5xx SDK are compiled in Keil  $\mu$ Vision5 IDE. To compile application example projects with IAR Embedded Workbench, users can choose the script tool `keil2iarprj.py`, to convert Keil project files from `*.uvprojx` to IAR project files.

To generate an IAR project file with `keil2iarprj.py`:

1. Find the tool `keil2iarprj.py`, which is in `SDK_Folder\build\iar` by default.
2. Copy `keil2iarprj.py` to the `Keil_5` directory of the target application project. Copy the script file of the project (`ble_app_template` as the example) to the directory `SDK_Folder\projects\ble\ble_peripheral\ble_app_template\keil_5`.

### Note:

The script file `keil2iarprj.py` and the `*.uvprojx` file shall be under the same directory while in use, to ensure the paths of source file and header file that the IAR project refers to are correct.

3. Change the Start-in path in **Command Prompt** by running a command. A sample command is provided below for the example project `ble_app_template`.

```
python keil2iarprj.py ble_app_template.uvprojx
```



When the IAR project is generated successfully, the command-line interface prints as below.

```
$ python keil2iarprj.py ble_app_template.uvprojx
>>> Transfer project: ble_app_template.uvprojx
>>> OS type: Windows
>>> Generate IAR project successfully, locate at ../IAR/ble_app_template.eww
```

Figure 3-1 IAR project path displayed after successful generation

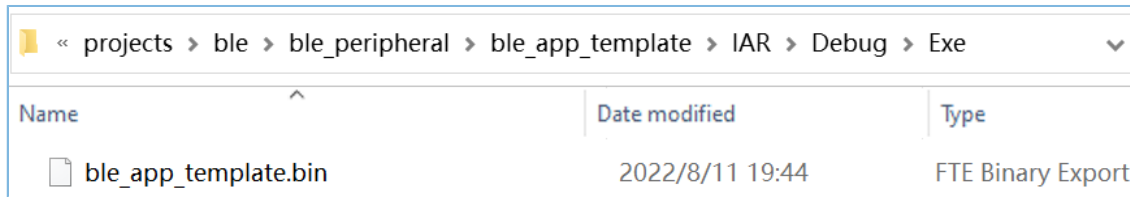
4. Put the newly generated IAR project at the same level and under the same directory with `Keil_5`.

### 3.2 Compiling an IAR Project

1. Open `SDK_Folder\projects\ble\ble_peripheral\ble_app_template\IAR`, and double-click the IAR project file `ble_app_template.eww`.
2. Click **Project** on the menu bar, and choose  (**Make**) or  (**Rebuild All**) to compile firmware. Wait until the compilation completes.



3. *ble\_app\_template.bin* will be generated after compilation completes. The .bin file will be available in SDK\_Folder\projects\ble\ble\_peripheral\ble\_app\_template\IAR\Debug\Exe.



Name	Date modified	Type
ble_app_template.bin	2022/8/11 19:44	FTE Binary Export

Figure 3-2 ble\_app\_template.bin generated from compilation

### 3.3 Downloading Program

Users can download programs with GProgrammer, a Goodix programming tool.

Follow *GProgrammer User Manual* to install and use the tool.

## 4 FAQ

This chapter describes possible problems, reasons, and solutions during application of an IAR example.

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

- Description

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

- Analysis

This may be because some project macros (such as 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.

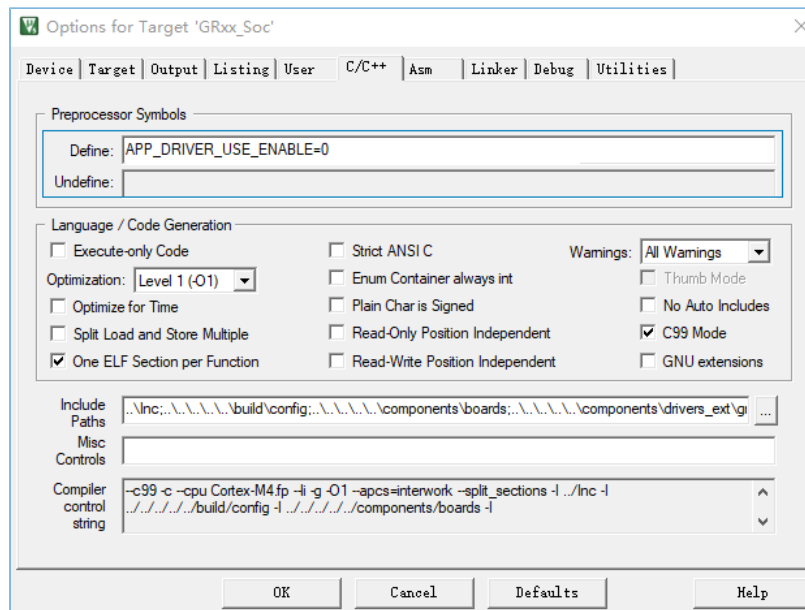


Figure 4-1 Project macro configuration in Keil

- Solution

Open **Options for node "ble\_app\_example"**. Add the undefined macro in the **C/C++ Compiler > Preprocessor** pane in IAR.