

GR5xx IAR User Manual

Version: 1.2

Release Date: 2025-07-11

Copyright © 2025 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: 26F, Goodix Headquarters, No.1 Meikang Road, Futian District, Shenzhen, China

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

Website: 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 third 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.
1.2	2025-07-11	Added "Firmware Download Using J-Link Commander" and "Firmware Download Using IAR IDE".



Contents

Preface					
1 Introduction	1				
2 Installation Instructions	2				
2.1 Preparation					
2.2 Software Installation	2				
3 Compiling IAR Example Project in SDK	2				
3.1 Building an IAR Project					
3.2 Compiling an IAR Project					
3.3 Downloading Firmware	5				
3.3.1 Firmware Download Using GProgrammer	5				
3.3.2 Firmware Download Using J-Link Commander	5				
3.3.3 Firmware Download Using IAR IDE	7				
4 FAQ	<u>9</u>				
4.1 Why Does an "Undefined Variable" Error Occur During Project Compilation?					



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	Introduces GR5xx Software Development Kit (SDK) and how to develop and debug applications based
GR5xx SoC	on the SDK.
J-Link/J-Trace User Guide	Provides J-Link operational instructions.
J-Lilik/J-liace Osel Guide	Available at https://www.segger.com/downloads/jlink/UM08001_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.
Bidetootii GAIT Spec	Available at https://www.bluetooth.com/specifications/gatt.
IAR Embedded Workbench IDE	Provides various information about IAR IDE.
User Guides	Available at https://www.iar.com/support/user-guides/user-guide-iar-embedded-workbench-for-
User Guides	arm/.



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/Windows 11
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 https://www.segger.com/downloads/jlink/ .
	Used to build an environment for executing scripts for GR5xx projects
Python	Required version: 3.0 or above
	Available at https://www.python.org/downloads/ .

2.2 Software Installation

- IAR
 - 1. Download the IAR installation file (IAR for ARM 9.40.1 or later versions) on <u>IAR Systems official website</u>.
 - 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
 - Download and install Python from the <u>official website</u>. 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 μ 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\bl e_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

- 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_Fold er\projects\ble\ble_peripheral\ble_app_template\IAR\Debug\Exe.

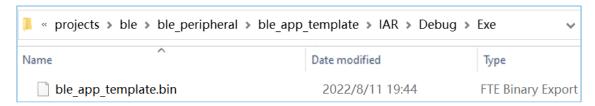


Figure 3-2 ble_app_template.bin generated from compilation

3.3 Downloading Firmware

3.3.1 Firmware Download Using GProgrammer

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

Follow GProgrammer User Manual to install and use the tool.

3.3.2 Firmware Download Using J-Link Commander

Users can download firmware using J-Link Commander (a tool available for Windows and Mac platforms) under the IAR directory.

Follow the steps below to download firmware using J-Link Commander:

1. In the J-Link installation directory, locate the JLinkDevices.xml file and append the following content to it.

```
<Device>
  <ChipInfo Vendor="Goodix" Name="GR551x" Core="JLINK CORE CORTEX M4"</pre>
WorkRAMAddr="0x30000000" WorkRAMSize="0x40000"/>
  <FlashBankInfo Name="Internal Flash" LoaderType="FLASH ALGO TYPE OPEN" Loader="Devices/</pre>
Goodix/GR5xxx 16MB Flash Jflash.FLM" MaxSize="0x100000" BaseAddr="0x200000"/>
 </Device>
 <Device>
  <ChipInfo Vendor="Goodix" Name="GR5526" Core="JLINK CORE CORTEX M4"</pre>
 WorkRAMAddr="0x20000000" WorkRAMSize="0x80000"/>
  <FlashBankInfo Name="Internal Flash" LoaderType="FLASH ALGO TYPE OPEN" Loader="Devices/</pre>
Goodix/GR5xxx 16MB Flash Jflash.FLM" MaxSize="0x100000" BaseAddr="0x200000"/>
 </Device>
 <Device>
  <ChipInfo Vendor="Goodix" Name="GR5x25" Core="JLINK CORE CORTEX M4"</pre>
 WorkRAMAddr="0x20000000" WorkRAMSize="0x40000"/>
  <FlashBankInfo Name="Internal Flash" LoaderType="FLASH ALGO TYPE OPEN" Loader="Devices/</pre>
Goodix/GR5xxx_16MB_Flash_Jflash.FLM" MaxSize="0x100000" BaseAddr="0x200000"/>
 </Device>
 <Device>
 <ChipInfo Vendor="Goodix" Name="GR533x" Core="JLINK CORE CORTEX M4"</pre>
 WorkRAMAddr="0x20000000" WorkRAMSize="0x18000"/>
  <FlashBankInfo Name="Internal Flash" LoaderType="FLASH ALGO TYPE OPEN" Loader="Devices/</pre>
Goodix/GR5xxx 16MB Flash Jflash.FLM" MaxSize="0x80000" BaseAddr="0x200000"/>
 </Device>
 <Device>
  <ChipInfo Vendor="Goodix" Name="GR5405" Core="JLINK CORE CORTEX M4"</pre>
 WorkRAMAddr="0x20000000" WorkRAMSize="0x18000"/>
  <FlashBankInfo Name="Internal Flash" LoaderType="FLASH ALGO TYPE OPEN" Loader="Devices/</pre>
Goodix/GR5xxx_16MB_Flash_Jflash.FLM" MaxSize="0x80000" BaseAddr="0x200000"/>
```



</Device>

- 2. Place the download algorithm file SDK_Folder\build\gcc\GR5xxx_16MB_Flash_Jflash.FLM in the JLink_Folder\JLink\Devices\Goodix directory.
- Launch the J-Link Commander tool, enter connect, and then type? to select the chip, as shown in Figure 3-3. For
 example, if the target chip is GR5405, select GR5405 in Target device settings, as depicted in Figure 3-4.

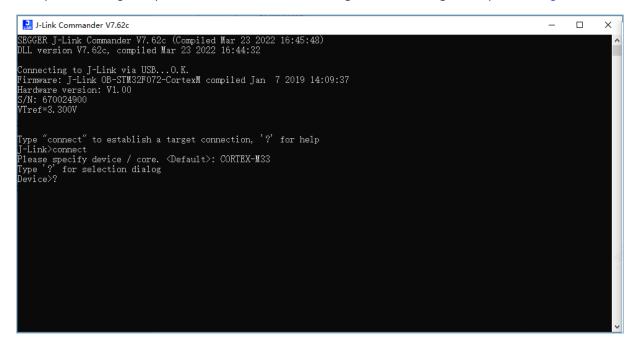


Figure 3-3 J-Link Commander GUI

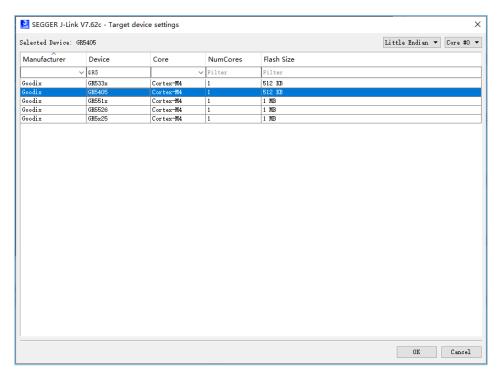


Figure 3-4 Chip selection interface



Use loadfile command to download firmware.

```
or
loadfile xxx.bin 0x00202000
```

Note:

- Use either of the above two commands to download firmware.
- 0x00202000 is the configured value of the APP_CODE_LOAD_ADDR macro in custom_config.h.

Figure 3-5 Firmware download using loadfile command

5. Run the firmware.

After firmware download, press the reset button on the board or enter the r and then g commands in the **J-Link Command** interface to reset the device and run the firmware.

```
J-Link>r
Reset delay: 0 ms
Reset type NORMAL: Resets core & peripherals via SYSRESETREQ & VECTRESET bit.
Reset: Halt core after reset via DEMCR. VC_CORERESET.
Reset: Reset device via AIRCR. SYSRESETREQ.
J-Link>
J-Link>
```

Figure 3-6 Running the firmware using the r and g commands

3.3.3 Firmware Download Using IAR IDE



Users can also download and debug firmware using IAR IDE, with the steps as follows:

- In the menu bar of IAR IDE, select Project > Options to open the configuration window. In the Category list on the left, select Debugger. Then, on the Setup page, select J-Link/J-Trace in the Driver field, check Use macro files(s), and enter \$PROJ_DIR\$\..\..\..\..\..\build\iar\GR5xxx.mac.
- 2. Copy the download algorithm files GR5xxx_IAR_16M.board, GR5xxx_IAR_16M.flash, and GR5xxx_IAR_flashloader_16M.out in SDK_Folder/build/iar to IAR_Install/arm/config/flashloader/Goodix ("IAR_Install" is the IAR installation directory, and "Goodix" is the new folder).
- 3. In the menu bar of IAR IDE, select **Project > Options > Debugger > Download** to set **Override default.board file** to *GR5xxx_IAR_16M.board* (in IAR_Install/arm/config/flashloader/Goodix).



If the target chip is GR5405, select **Project > Options > Debugger > J-Link/J-Trace > Setup**, and then select **Core** in the **Reset** field.

4. In the menu bar of IAR IDE, select **Project** > **Download** > **Download Active Application**, and then click **OK** in the pop-up dialog box, to start firmware download and enter debug mode.



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.

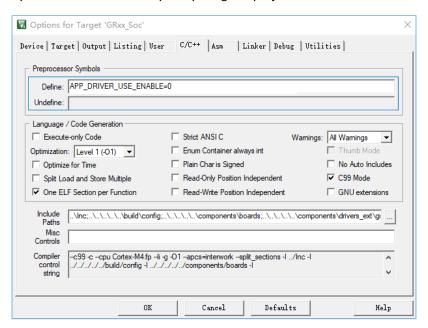


Figure 4-1 Project macro configuration in Keil

Solution

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