GR551x IAR User Manual

Version: 1.4

Release Date: 2022-09-30

Shenzhen Goodix Technology Co., Ltd.

Trademarks and Permissions

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 12-13, Phase B, Tengfei Industrial Building, Futian Free Trade Zone, Shenzhen, China

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

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 fifth 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	• Updated descriptions in “Configuring an Application Project”. • Updated SoC model descriptions. • Optimized “Installation Instructions”.
1.2	2021-08-09	• Updated the section "Installation Requirements". • Modified SoC model descriptions.
1.3	2022-02-20	Modified two sections: "Configuring an Application Project" and "Compiling and Downloading a Project".
1.4	2022-09-30	• Introduced Python download process in "Preparation" and "Software Installation". • Updated two chapters "Configurations for SDK" and "FAQ".

Contents

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 http://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. Available at https://www.bluetooth.com/specifications/gatt.
IAR Embedded Workbench IDE User Guides	Provides various information about IAR IDE. Available at:https://www.iar.com/support/user-guides/user-guide-iar-embedded-workbench-for-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	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/.
Python	Used to build an environment for executing scripts for GR551x projects Required version: 3.0 or above Available at https://www.python.org/downloads/.

2.2 Software Installation

- IAR

  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.

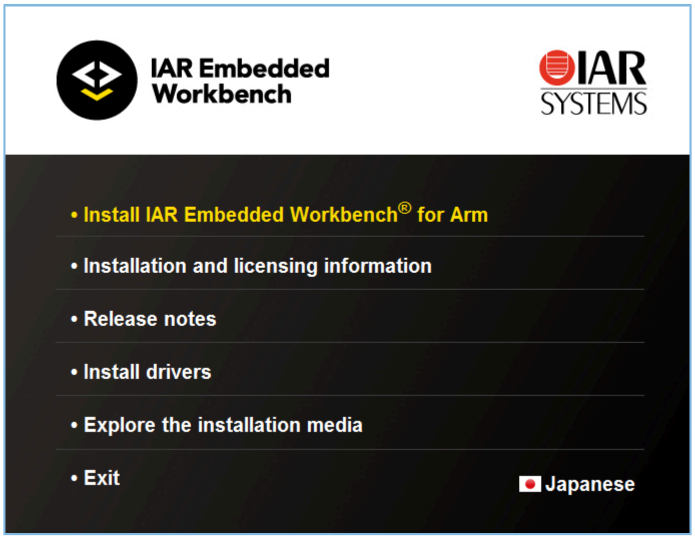

IAR Embedded Workbench
IAR SYSTEMS
• Install IAR Embedded Workbench® for Arm
• Installation and licensing information
• Release notes
• Install drivers
• Explore the installation media
• Exit
Japanese

Figure 2-1 IAR installation page

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

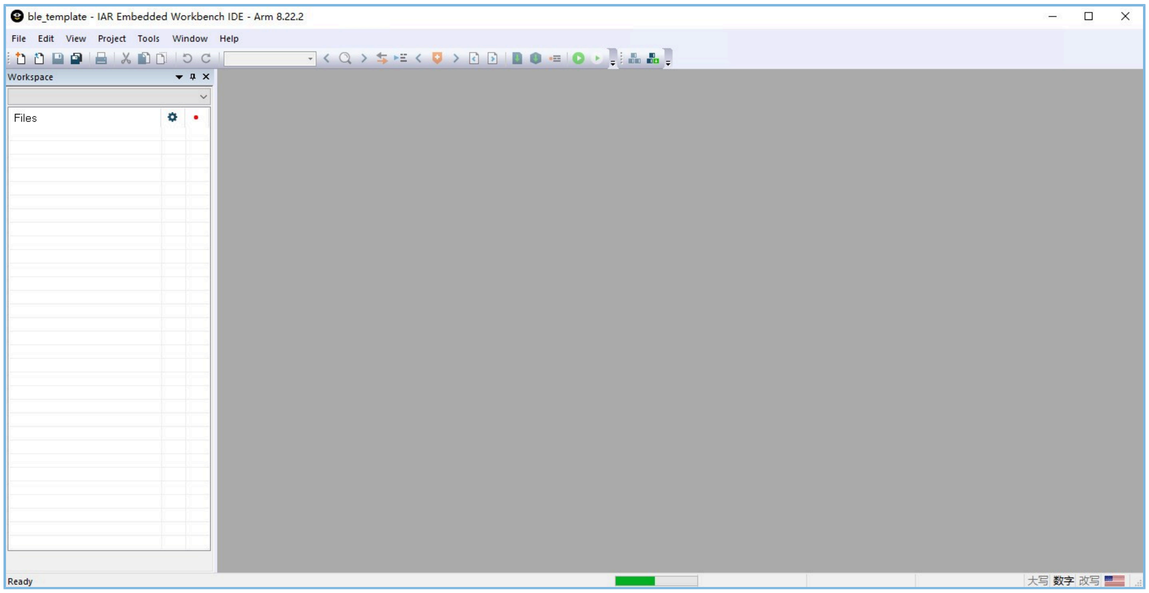

ble_template - IAR Embedded Workbench IDE - Arm 8.22.2
File Edit View Project Tools Window Help
Workspace
Files
Ready

Figure 2-2 IAR main user interface

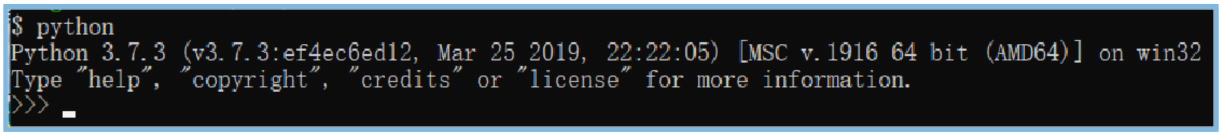
Users can choose to install the necessary software as prompted after installing IAR.

- 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 2-3 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 GR551x SDK.

3.1 Building an IAR Project

By default, the application example projects in GR551x SDK are compiled in Keil µVision5 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`.

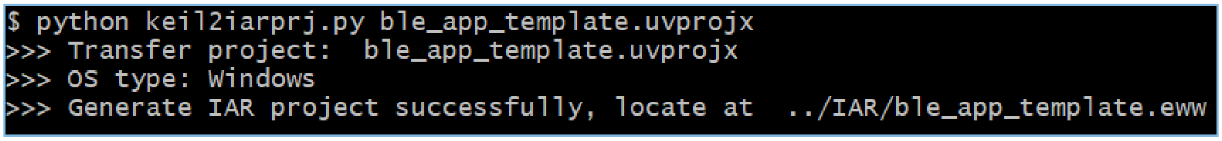
Note:

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
```

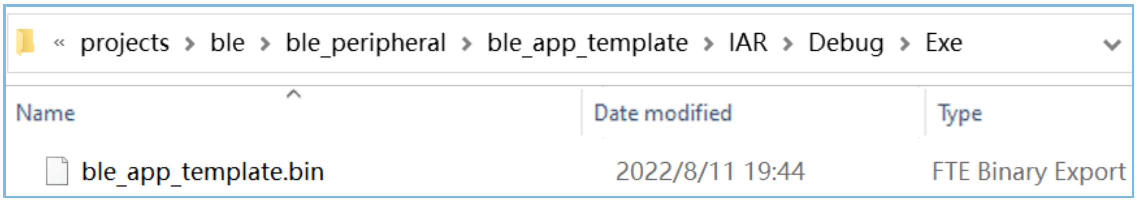
Figure 3-1 IAR project path displayed after successful generation

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`.


« 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 when using an IAR example.

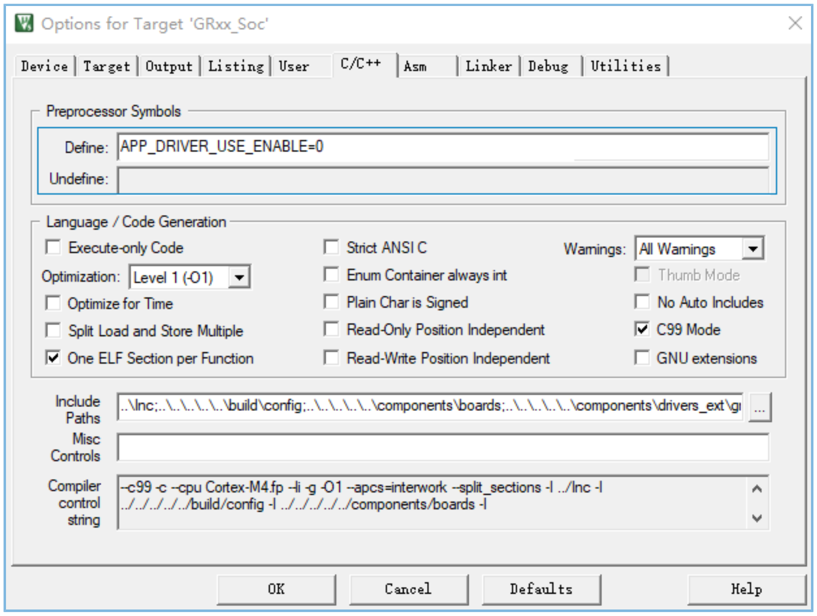
4.1 Why Does an “Undefined Variable” Error Occur During Project Compilation?

- 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 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.



Define: APP_DRIVER_USE_ENABLE=0

Undefine:

☐ Execute-only Code

Optimization: Level 1 (-O1)

☐ Optimize for Time

☐ Split Load and Store Multiple

☑ One ELF Section per Function

☐ Strict ANSI C

☐ Enum Container always int

☐ Plain Char is Signed

☐ Read-Only Position Independent

☐ Read-Write Position Independent

Warnings: All Warnings

☐ Thumb Mode

☐ No Auto Includes

☑ C99 Mode

☐ GNU extensions

Include Paths: ..\Inc;..\..\..\..\..\build\config;..\..\..\..\..\components\boards;..\..\..\..\..\components\drivers_ext\gi

Misc Controls:

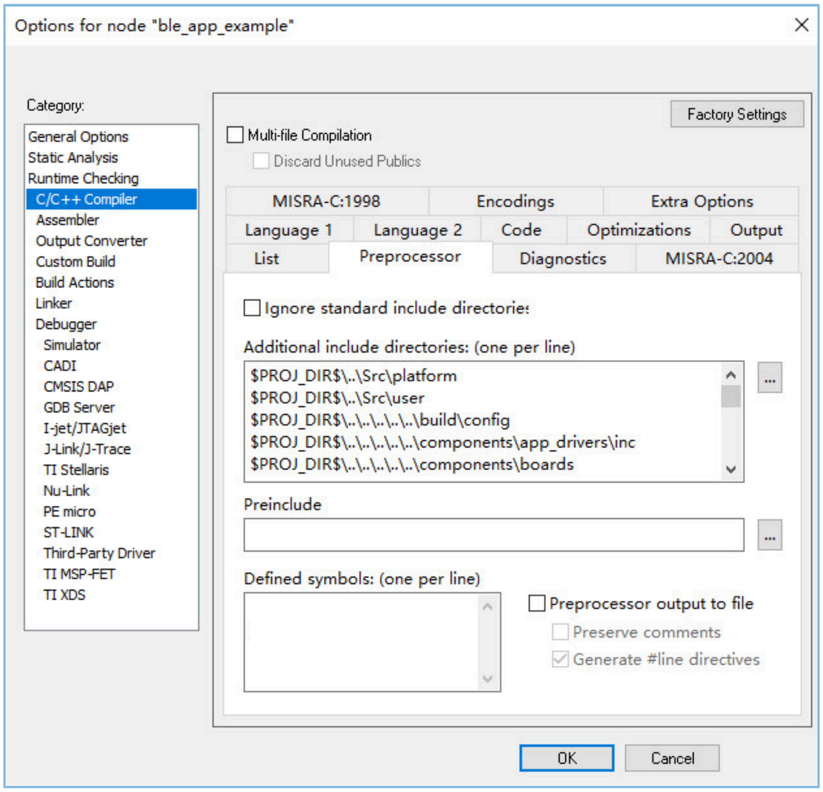
Compiler control string:

```
--c99 -c --cpu Cortex-M4.fp --li -g -O1 --apcs=interwork --split_sections -I ../Inc -I
../../../../../build/config -I ../../../../../components/boards -I
```

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, as shown in the figure below.



Options for node "ble_app_example"

Category:

□ Multi-file Compilation

□ Discard Unused Publics

□ Ignore standard include directorie:

Additional include directories: (one per line)

```
$PROJ_DIR$\..\Src\platform
$PROJ_DIR$\..\Src\user
$PROJ_DIR$\..\..\..\..\..\build\config
$PROJ_DIR$\..\..\..\..\..\components\app_drivers\inc
$PROJ_DIR$\..\..\..\..\..\components\boards
```

Preinclude

Defined symbols: (one per line)

□ Preprocessor output to file

□ Preserve comments

☑ Generate #line directives

Figure 4-2 Adding an undefined macro