



GR5xx IAR用户手册

版本： 1.2

发布日期： 2025-07-11

版权所有 © 2025 深圳市汇顶科技股份有限公司。保留一切权利。

非经本公司书面许可，任何单位和个人不得对本手册内的任何部分擅自摘抄、复制、修改、翻译、传播，或将其全部或部分用于商业用途。

商标声明

GOODiX 和其他汇顶商标均为深圳市汇顶科技股份有限公司的商标。本文档提及的其他所有商标或注册商标，由各自的所有人持有。

免责声明

本文档中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。

深圳市汇顶科技股份有限公司（以下简称“GOODiX”）对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。GOODiX对因这些信息及使用这些信息而引起的后果不承担任何责任。

未经GOODiX书面批准，不得将GOODiX的产品用作生命维持系统中的关键组件。在GOODiX知识产权保护下，不得暗或以其他方式转让任何许可证。

深圳市汇顶科技股份有限公司

总部地址：深圳市福田区梅康路1号汇顶科技总部大厦26楼

电话：+86-755-33338828 邮编：518000

网址：www.goodix.com

前言

编写目的

本文档介绍了基于IAR Embedded Workbench IDE，对GR5xx低功耗蓝牙（Bluetooth LE）应用进行开发与调试的方法，旨在帮助用户在IAR环境下快速进行对GR5xx SDK应用的二次开发。

读者对象

本文适用于以下读者：

- 芯片用户
- 开发人员
- 测试人员
- 技术支持工程师

版本说明

本手册为第3次发布，对应的产品为低功耗蓝牙GR5xx系列。

修订记录

版本	日期	修订内容
1.0	2023-12-27	首次发布
1.1	2024-08-23	升级IAR支持版本。
1.2	2025-07-11	新增“使用J-Link Commander下载固件”和“使用IAR IDE下载固件”。

目录

前言.....	1
1 简介.....	1
2 安装指南.....	2
2.1 准备工作.....	2
2.2 安装软件.....	2
3 编译SDK应用示例工程.....	4
3.1 生成IAR工程.....	4
3.2 IAR工程编译.....	4
3.3 下载固件.....	5
3.3.1 使用GProgmmmer下载固件.....	5
3.3.2 使用J-Link Commander下载固件.....	5
3.3.3 使用IAR IDE下载固件.....	8
4 常见问题.....	10
4.1 未定义工程宏.....	10

1 简介

IAR Embedded Workbench IDE for Arm（IAR EWARM，以下简称IAR）是由IAR Systems公司开发的集成开发环境（IDE），支持8位、16位以及32位的微处理器和微控制器，较其他的ARM开发环境更易入门和使用。

IAR的主要特性如下：

- 高度优化的C/C++ Compiler
- 一个通用的IAR XLINK Linker
- 功能强大的工程管理器
- TM IAR C-SPY调试器

本文将介绍在Windows系统下，使用IAR构建GR5xx应用开发环境的方法，帮助用户完成BLE应用的创建、编译、下载和调试，并为用户提供参考示例。

在进行操作前，可参考以下文档。

表 1-1 文档参考

名称	描述
对应芯片开发者指南	介绍GR5xx SDK以及基于SDK的应用开发和调试
J-Link用户指南	J-Link使用说明： https://www.segger.com/downloads/jlink/UM08001_JLink.pdf
Bluetooth Core Spec	Bluetooth官方标准核心规范
Bluetooth GATT Spec	Bluetooth Profile和Service的详细信息查看地址： https://www.bluetooth.com/specifications/gatt
IAR Embedded Workbench IDE User Guides	IAR IDE用户手册： https://www.iar.com/support/user-guides/user-guide-iar-embedded-workbench-for-arm/

2 安装指南

本章介绍IAR的安装要求与安装流程。

2.1 准备工作

- 硬件准备

表 2-1 硬件准备

名称	描述
开发板	对应芯片Starter Kit开发板（以下简称“开发板”）
连接线	USB Type-C（GR551x系列使用Micro USB 2.0连接线）

- 软件准备

表 2-2 软件准备

名称	描述
Windows	Windows 7/Windows 10/Windows 11操作系统
IAR Embedded Workbench IDE for Arm（EWARM）	IAR for ARM 9.40.1及以上版本
J-Link驱动	J-Link驱动程序，下载网址： https://www.segger.com/downloads/jlink/
Python	用于GR5xx应用工程的脚本执行环境的构建。版本：python 3.x + 下载网址： https://www.python.org/downloads/

2.2 安装软件

- IAR

1. 用户可从[IAR官网](#)下载IAR安装包，需使用IAR for ARM 9.40.1及以上版本。
2. 下载完成后，双击IAR.exe安装程序，点击“Install IAR Embedded Workbench® for Arm”，根据安装向导提示逐步完成安装。
3. IAR安装完成后，用户可根据弹窗提示安装所需软件。

- Python

1. 访问网址：<https://www.python.org/downloads/>，下载并安装Python3。要求下载的Python版本和用户使用的Windows系统兼容。
2. 根据安装向导进行安装。
3. 设置环境变量。

Python安装成功后在Windows命令窗口输入python命令，显示Python版本信息如下图所示，则说明环境变量设置成功。

```
$ 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.
>>> _
```

图 2-1 查看Python版本信息

3 编译SDK应用示例工程

本章通过应用示例工程ble_app_template，对IAR工程的生成、使用和编译等进行介绍。

说明:

SDK_Folder为GR5xx SDK的根目录。

3.1 生成IAR工程

GR5xx SDK开发包中的应用示例工程，默认使用Keil μ Vision5 IDE环境进行编译构建。如果用户希望使用IAR工具编译构建应用示例工程，可使用keil2iarprj.py脚本工具，将Keil的工程文件*.uvprojx转换为IAR工程文件。

keil2iarprj.py使用说明如下：

1. keil2iarprj.py工具文件默认位于SDK_Folder\build\iar目录下。
2. 将keil2iarprj.py脚本文件拷贝到目标应用工程的Keil_5目录下。以ble_app_template为例，将脚本拷贝至SDK_Folder\projects\ble\ble_peripheral\ble_app_template\keil_5。

说明:

为保证转换后的IAR工程引用的源文件和头文件路径正确，已约束keil2iarprj.py脚本文件在使用时需要与*.uvprojx文件位于同一个目录。

3. 从命令行切换到目标路径，执行如下命令。以ble_app_template为例，命令行和生成的IAR工程文件如下图所示：

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

```
$ 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
```

图 3-1 成功生成IAR工程文件

4. 转换成功后的IAR工程文件，放置于与Keil_5目录平行的IAR目录，用户可进入此目录进行查看。

3.2 IAR工程编译

1. 进入SDK_Folder\projects\ble\ble_peripheral\ble_app_template\IAR目录，双击ble_app_template.eww文件打开IAR工程文件。
2. 进入IAR工程后，点击菜单栏的“Project”，选择“Make”或“Rebuild All”编译固件，等待编译完成。
3. 编译完成后可在SDK_Folder\projects\ble\ble_peripheral\ble_app_template\IAR\Debug\Exe目录下查看生成的bin文件ble_app_template.bin。

« projects > ble > ble_peripheral > ble_app_template > IAR > Debug > Exe		
名称	修改日期	类型
ble_app_template.bin	2022/8/11 19:44	FTE Binary Export

图 3-2 成功生成ble_app_template.bin文件

3.3 下载固件

3.3.1 使用GProgrammer下载固件

用户可使用GProgrammer图形化界面烧写工具进行程序下载。

GProgrammer的安装与使用可参考文档《GProgrammer用户手册》。

3.3.2 使用J-Link Commander下载固件

在Windows和Mac平台上，可在IAR目录下使用J-Link Commander工具进行固件下载。

使用J-Link Commander下载固件的步骤如下：

1. 在J-Link的安装目录下，找到文件*JLinkDevices.xml*，并在该文件中添加以下内容：

```
<Device>
  <ChipInfo Vendor="Goodix" Name="GR551x" Core="JLINK_CORE_CORTEX_M4"
  WorkRAMAddr="0x30000000" WorkRAMSize="0x40000"/>
  <FlashBankInfo Name="Internal Flash" LoaderType="FLASH_ALGO_TYPE_OPEN" Loader="Devices/
  Goodix/GR5xxx_16MB_Flash_Jflash.FLM" MaxSize="0x100000" BaseAddr="0x200000"/>
</Device>
<Device>
  <ChipInfo Vendor="Goodix" Name="GR5526" Core="JLINK_CORE_CORTEX_M4"
  WorkRAMAddr="0x20000000" WorkRAMSize="0x80000"/>
  <FlashBankInfo Name="Internal Flash" LoaderType="FLASH_ALGO_TYPE_OPEN" Loader="Devices/
  Goodix/GR5xxx_16MB_Flash_Jflash.FLM" MaxSize="0x100000" BaseAddr="0x200000"/>
</Device>
<Device>
  <ChipInfo Vendor="Goodix" Name="GR5x25" Core="JLINK_CORE_CORTEX_M4"
  WorkRAMAddr="0x20000000" WorkRAMSize="0x40000"/>
  <FlashBankInfo Name="Internal Flash" LoaderType="FLASH_ALGO_TYPE_OPEN" Loader="Devices/
  Goodix/GR5xxx_16MB_Flash_Jflash.FLM" MaxSize="0x100000" BaseAddr="0x200000"/>
</Device>
<Device>
  <ChipInfo Vendor="Goodix" Name="GR533x" Core="JLINK_CORE_CORTEX_M4"
  WorkRAMAddr="0x20000000" WorkRAMSize="0x18000"/>
  <FlashBankInfo Name="Internal Flash" LoaderType="FLASH_ALGO_TYPE_OPEN" Loader="Devices/
  Goodix/GR5xxx_16MB_Flash_Jflash.FLM" MaxSize="0x80000" BaseAddr="0x200000"/>
</Device>
<Device>
```

```
<ChipInfo Vendor="Goodix" Name="GR5405" Core="JLINK_CORE_CORTEX_M4"  
WorkRAMAddr="0x20000000" WorkRAMSize="0x18000"/>  
<FlashBankInfo Name="Internal Flash" LoaderType="FLASH_ALGO_TYPE_OPEN" Loader="Devices/  
Goodix/GR5xxx_16MB_Flash_Jflash.FLM" MaxSize="0x80000" BaseAddr="0x200000"/>  
</Device>
```

2. 将下载算法文件SDK_Folder\build\gcc\GR5xxx_16MB_Flash_Jflash.FLM添加到JLink_Folder\JLink\Devices\Goodix目录。
3. 打开J-Link Commander工具，键入“connect”，然后输入“？”，进行芯片选择，如图 3-3所示。例如，若目标芯片为GR5405，则在“Target device settings”中选择“GR5405”，如图 3-4所示。

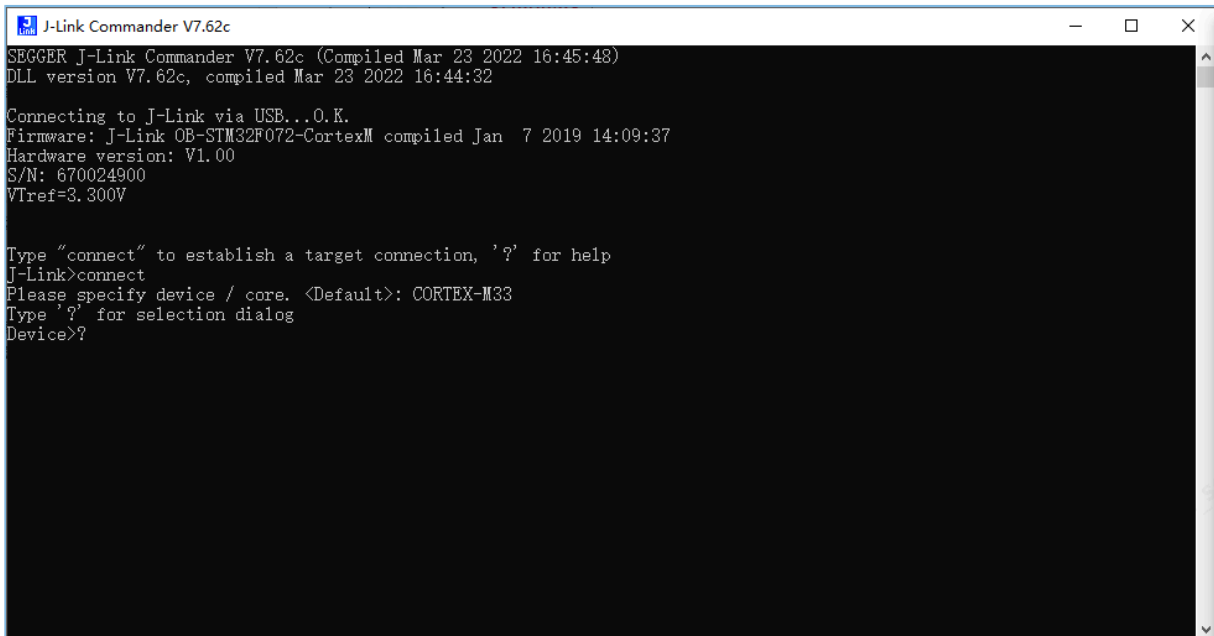


图 3-3 J-Link Commander界面

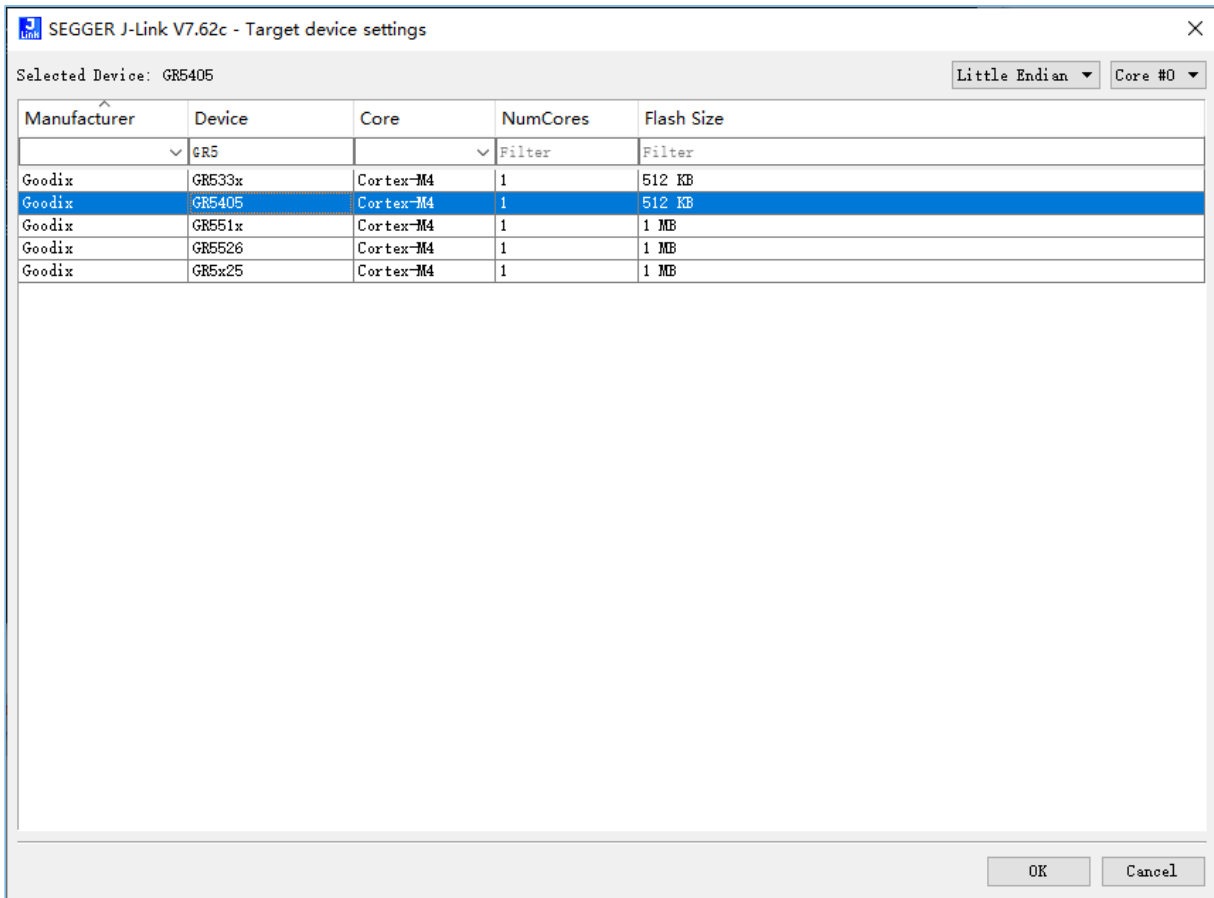


图 3-4 芯片选择界面

4. 使用loadfile命令下载固件。

```
loadfile xxx.hex
```


或

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

说明:

- 上述两种命令，任选其中一种使用。
- “0x00202000”为*custom_config.h*中配置的“APP_CODE_LOAD_ADDR”宏的值。

3. 在IAR IDE的菜单栏中，选择“Project > Options > Debugger > Download”，将“Override default.board file”设置为GR5xxx_IAR_16M.board文件（位于IAR_Install/arm/config/flashloader/Goodix）。
-

 提示:

若目标芯片为GR5405，还需选择“Project > Options > Debugger > J-Link/J-Trace > Setup”，然后在“Reset”栏选择“Core”。

4. 在IAR IDE的菜单栏中，选择“Project > Download > Download Active Application”，在弹出的对话框中点击“OK”按钮，即可开始下载固件，并进入调试模式。

4 常见问题

本章描述了在使用IAR工程时，可能出现的问题、原因及处理方法。

4.1 未定义工程宏

- 问题描述

编译IAR工程时，出现变量未定义的编译错误。

- 问题分析

可能是由于未定义某些工程宏（如APP_DRIVER_USE_ENABLE），导致源文件中的对应内容未被编译，用户可通过和Keil的工程宏进行对比，判断是否有缺漏。

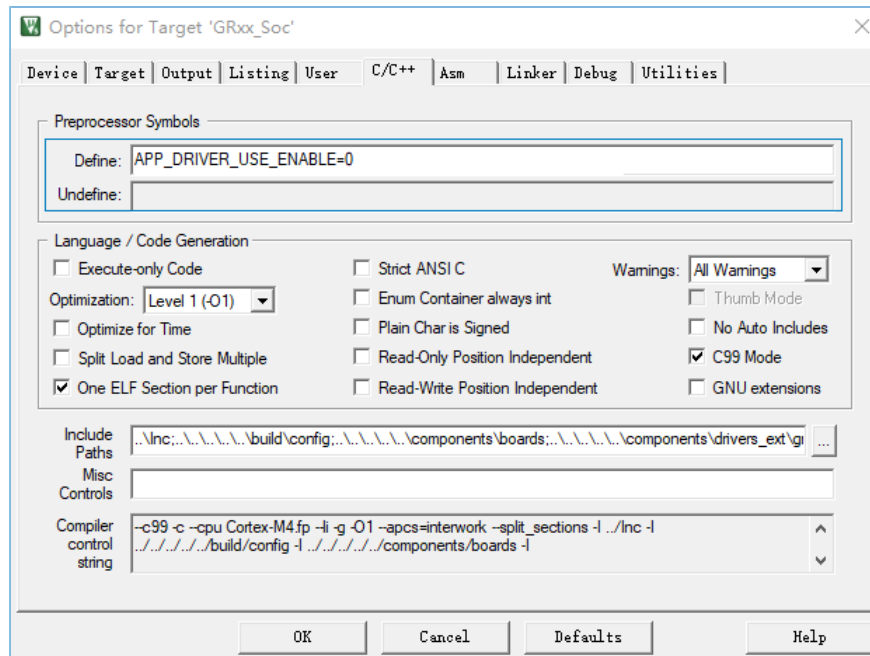


图 4-1 Keil工程宏配置界面

- 处理方法

在IAR工程“Options for node "ble_app_example"”的“C/C++ Compiler > Preprocessor”中添加缺少的宏定义。