

GR55xx IAR User Manual

Version: 1.1

Release Date: 2021-06-26

Shenzhen Goodix Technology Co., Ltd.

Copyright © 2021 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 GR55xx Bluetooth Low Energy (Bluetooth LE) applications based on IAR Embedded Workbench IDE, to help users quickly get started with secondary development of GR55xx SDK applications.

Audience

This document is intended for:

- GR55xx user
- GR55xx developer
- GR55xx tester
- Hobbyist developer
- Technical writer

Release Notes

This document is the second release of *GR55xx IAR User Manual*, corresponding to GR55xx SoC series.

Revision History

Version	Date	Description
1.0	2021-03-08	Initial release
1.1	2021-06-26	 Updated the descriptions in "Configuring an Application Project". Updated GR551x to GR55xx. Optimized "Installation Instructions".

Contents

Preface	I
1 Introduction	1
2 Installation Instructions	2
2.1 Installation Requirements	2
2.2 Software Installation	
3 Configurations for SDK Applications	4
3.1 Building an Application Project	4
3.2 Configuring an Application Project	5
3.3 Compiling and Downloading a Project	
3.4 Project Debugging	
4 FAQ	
4.1 Why does an "Undefined Variable" Error Occur During Project Compilation?	14

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 GR55xx application development environment with IAR on Windows PCs, so as to create, compile, download, and debug Bluetooth LE applications, with an example provided as reference.

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

Name Description GR55xx Developer Guide Introduces the software/hardware and quick start guide of GR55xx SoCs. Provides J-Link operational instructions. J-Link/J-Trace User Guide Available at http://www.segger.com/downloads/jlink/UM08001 JLink.pdf. Offers official Bluetooth standards and core specification (v5.1) from Bluetooth SIG. Bluetooth Core Spec v5.1 Available at https://www.bluetooth.com/specifications/bluetooth-core-specification/. Provides details about Bluetooth profiles and services. **Bluetooth GATT Spec** Available at https://www.bluetooth.com/specifications/gatt. Provides various information about IAR IDE. IAR Embedded Workbench IDE Available at:https://www.iar.com/support/user-guides/user-guide-iar-embedded-workbench-for-User Guides arm/.

Table 1-1 Reference documents

2 Installation Instructions

This chapter describes the requirements and process for installing IAR.

2.1 Installation Requirements

• Supported Development Platform

Table 2-1 Supported Development Platform

Hardware Platform	Development Board Model
GR551x development kit	GR5515-SK-BASIC

Hardware and Software preparation

Table 2-2 Hardware and 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 <u>www.segger.com/downloads/jlink/</u> .
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 GR55xx SDK.

🛄 Note:

- SDK_Folder is the root directory of the GR55xx SDK in use.
- 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).

- 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.

1	<pre><?xml version="1.0" encoding="UTF-8"?></pre>
2	
3	<project></project>
4	<pre><path>\$WS_DIR\$\ble_app_template.ewp</path></pre>
5	<pre>- </pre>
6	<batchbuild></batchbuild>
7	L
8	

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.



te dit Ver Pejeti Juhi Tools Window Help Ditation of the set of t									 			
Image:	Ble_app_example - IAR Emple - IAR Emplement	bedded	Vorkbench IDE -	Arm 8.22.2						-1		×
prispace • # x stage ibite ibite <td>File Edit View Project J-Li</td> <td>ink Tools</td> <td>Window Help</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>	File Edit View Project J-Li	ink Tools	Window Help									
prispace • # x stage ibite ibite <td>10 10 10 10 10 10 10</td> <td>CIDI</td> <td>C</td> <td>- < Q > t</td> <td>► E < 🖸 > 💽</td> <td>🕞 📗 🌒 🛥 🖸</td> <td>• 1: 1</td> <td></td> <td></td> <td></td> <td></td> <td></td>	10 10 10 10 10 10 10	CIDI	C	- < Q > t	► E < 🖸 > 💽	🕞 📗 🌒 🛥 🖸	• 1: 1					
iies	Norkspace	- 4	×									
• be axetand • gr., app., askangle • gr., app., divers • gr., app., di	Debug		~									
Image: State and State an	Files	٥	•									
Image: September 1 Image: September 2 Image: Se	🗆 🌒 ble_app_example											
The grach is a second s	- external											
Ale G chives Ale	He gr_app_drivers											
e g.c.,chianes g.c., g.c	H 🖬 🛑 gr_arch											
AG G_profiles	⊣⊞ 🖬 gr_drivers											
- G g_stack_lb G g_stack_lb G g_stack_lb G g_stack_lb G user_spp G user_platform - C Output ite_spp_example Ite_spp_exam	Here i gr_libraries											
en grupshanp en user, politicak. en user, politi	Here in the second seco											
He Guer, app Ge User, platform He User, platform He Segres Messages Messages	- gr_stack_lib											
ee euer, caliback. ee eue	🕀 🔳 gr_startup											
E Ser platform A Couput ie.esp.exmple id Messages A Debug top	- 🖽 📕 user_app											
EB C Output	He user_callback											
ie_sop_example id views of the source of th	He user_platform		_									
id V I Messages Ind Debug top	He Output											
id V I Messages Ind Debug top												
id V I Messages Ind Debug top			-									
id V I Messages Ind Debug top												
id V I Messages Ind Debug top												
id V I Messages Ind Debug top												
id V I Messages Ind Debug top												
id V I Messages Ind Debug top		-										
Messages and Debug tog	ble_app_example											
and Debug tog	Build											, a x
and Debug tog												
and Debug tog	wessages											
and Debug tog												
and Debug tog												
and Debug tog												
and Debug tog												
	<											>
xdy Errors 0, Warnings 0 大弓 数字 改写 👥	Build Debug Log											
	Ready							Errors 0, Warnings 0		大写数字	改写	

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.

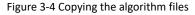
 Find the IAR download algorithm files (GR55xx_IAR_8M.board, GR55xx_IAR_8M.flash, and GR55xx_IAR_flashloader_8M.out) in SDK_Folder\build\binaries\xflash_flm_tools\IAR.

Name	Date modified	Туре	Size
GR55xx_IAR_8M.board	17/09/2020 11:15	BOARD File	1 KB
GR55xx_IAR_8M.flash	17/09/2020 11:15	FLASH File	1 KB
GR55xx_IAR_flashloader_8M.out	17/09/2020 11:15	OUT File	31 KB
IAR Download Algoritm Use Instruction.docx	21/09/2020 15:25	Microsoft Word D	204 KB

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.

lame	Date modified	Туре	Size	
GR55xx_IAR_8M.board	17/09/2020 11:15	BOARD File	1 KB	
] GR55xx_IAR_8M.flash	17/09/2020 11:15	FLASH File	1 KB	
GR55xx_IAR_flashloader_8M.out	17/09/2020 11:15	OUT File	31 KB	



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.

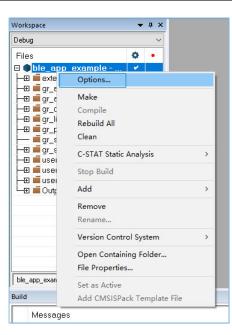


Figure 3-5 To configure the project

(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.

Category: General Options	 Multi-file Compil-	ation			Fac	tory Setting		
Static Analysis Runtime Checking		used Publics						
C/C++ Compiler	MISRA-C:	1998	Encodings	E	Extra Op	otions		
Assembler	Language 1	Language 2	Code	Optimiza	tions	Output		
Output Converter Custom Build	List	Preprocessor	Diagno			-C:2004		
Build Actions	List	reprocessor	Diagilo	sucs	WIGHT	-0.2004		
Linker								
Debugger	Ignore sta	ndard include dir	rectories					
Simulator	Additional include directories: (one per line)							
CADI			(one per nne)					
CMSIS DAP		\Src\platform				<u>^</u>		
GDB Server	\$PROJ_DIR\$							
I-jet/JTAGjet		\\\\build\c						
J-Link/J-Trace		\\\\\compo						
TI Stellaris	\$PROJ_DIR\$	\\\\compo	nents\boards			~		
Nu-Link								
	Preinclude							
Nu-Link PE micro ST-LINK	Preinclude							
Nu-Link PE micro ST-LINK Third-Party Driver	Preinclude							
Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET		ools: (one per line	e)					
Nu-Link PE micro ST-LINK Third-Party Driver	Defined symb	ools: (one per line	and the second second second second	processor	output			
Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET		ools: (one per line	∧ □ Pre			to file		
Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET	Defined symb	ools: (one per line	^ □Pre	Preserve co	omment	to file		
Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET	Defined symb	ools: (one per line	^ □Pre		omment	to file		

Figure 3-6 Project compilation

🛄 Note:

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

GODIX

(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.

General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custon Buld Build Actions Unker Debugger CADI CADI CADI CADI CADI CADI CADI CASIS DAP CGB Server I_Stet/ITAGget J_ink/J-Trace TI Stellaris Nu-Link PE micro ST-LINK TImd-Party Driver TI MSP-FET TI XDS	Category:						Factory	Setting
C/C++ Compiler Assembler Output Converter Custon Build Build Actions Uibrary Input Optimizations Advanced Output List Uinker configuration file I blogger Simulator CADI CMSIS DAP COB Server I-jet/TJAGjet J-Link/ TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver II MSP-FET I MSP-F	Static Analysis							
Output Converter Config Library Input Optimizations Advanced Output List Custom Build Build Actions Linker configuration file University		#define	Diag	nostics	Checksum	Encodings	Extra C	options
Custom Buld Buld Actons Unker Debugger Simulator CADI CMSIS DAP COB Server I-jet/JTAGjet 3-Link/>-Trace TI Stellaris Nu-Link PE micro ST-LINK Thid-Party Driver TI MSP-FET		Config	Library	Input	Optimizations	Advanced	Output	List
Unker Debugger Simulator CADI CMSIS DAP GDB Server 1-jet/TIAGjet J-Link/>-Trace TI Stellaris Nu-Link PE micro ST-LINK Ti MSP-FET		Linker	configura	ation file				
Debugger Simulator CADI CMSIS DAP COB Server 1-jet/JTASjet J-link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET				C 11				
Simulator CADI CMSIS DAP CMSIS DAP CMSIS DAP CGB Server 1-jet/JTAGjet JLink/J-Trace TI Stelaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET		⊻Ov	erride de	fault				
CADI CMSIS DAP GDB Server I-jet/TAGjet J-link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Thid-Party Driver TI MSP-FET		\$	PROJ_DIF	\$\\\	.\\toolchain\gr	51x\source\i	ar\gr551!	1.22
GDB Server I set/ITAGjet Julnk/J-Trace Ti Stellaris Nu-Link PE micro ST-LINK Third-Party Driver II MSP-PET			80				2001	10000
CGB Server I jet/JTAGjet Juhk/J-Trace TI Stelaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET	CMSIS DAP		Edit					
J-Link/J-Trace Configuration file symbol definitions: (one per line) T1 Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET	GDB Server							
TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-PET		0.0			1.0.10	P 5		
Nu Link PE micro ST LINK Tind Party Driver TI MSP-FET		Configu	ration file	symbol	definitions: (one	per line)		
PE micro ST-LINK Third-Party Driver TI MSP-FET								^
Third-Party Driver TI MSP-FET								
TI MSP-FET	ST-LINK							
TI XDS								
	TI XDS							
· · · · · · · · · · · · · · · · · · ·								~
								-

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.

General Options Static Analysis						Factory	Settings
Runtime Checking C/C++ Compiler	#define	Diag	nostics	Checksum	Encodings	Extra C	ptions
Assembler Output Converter	Config	Library	Input	Optimizations	Advanced	Output	List
Custom Build				1 13			
Build Actions	Output	filename:					
Linker	ble_ap	o_exampl	e.out				
Debugger Simulator							
CADI	Inclu	de debug	informa	tion in output			
CMSIS DAP							
GDB Server							
I-jet/JTAGjet							
J-Link/J-Trace TI Stellaris							
Nu-Link							
PE micro							
ST-LINK							
Third-Party Driver TI MSP-FET							
II MOM-FET							
TT XDS							
TI XDS							
TI XDS							
TI XDS							

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.

General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Output Converter Output Converter Output Converter Output Converter Output Converter Output Converter Simulator CADI CMSIS DAP GDB Server I jet/JTAGjet J-Link()-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS	Output Generate additional output Output format: Raw binary Output file Output file Ioad_app.bin	Settings
--	--	----------

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

(4). Configure Post-build command line.

Users shall also configure the post-build script (in the **Build Actions** > **Post-build command line** box) for IAR projects provided in GR55xx SDK (available in SDK_Folder\build\scripts\after_build_for_ IAR.bat), to generate a *.hex* file.

Category:	
General Options Static Analysis Runtime Checking	
C/C++ Compiler Assembler	Build Actions Configuration
Output Converter Custom Build	Pre-build command line:
Build Actions	
Linker	Post-build command line:
Debugger	
Simulator	\$PROJ_DIR\$\\\\build\scripts\after_build_for_IAR.bat
CADI	
CMSIS DAP	
GDB Server	
I-jet/JTAGjet	
J-Link/J-Trace	
TI Stellaris	
Nu-Link	
PE micro	
ST-LINK	
Third-Party Driver	
TI MSP-FET	
TI XDS	

Figure 3-10 Configuring the post-build script

(5). 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.

							Factory Settings
Seneral Options itatic Analysis tuntime Checking C/C ++ Compiler Assembler Output Converter Custom Build	Setup □Ve	Download rify downloa		Extra Options	Multicore	Plugins	
Build Actions Linker	Su	ppress down	load				
Debugger Simulator CADI CADI CMSIS DAP GDB Server I-jet/JTAGjet J-Ink/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS		Edit	fault .boa DIR\$\cor	rd file fig\flashloader pefore flashing	.goodix\GR!		

Figure 3-11 Configuring IAR download algorithm

🛄 Note:

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

3.3 Compiling and Downloading a Project

- 1. Click **Project** on the menu bar, and choose **Make** or **Rebuild All** to compile firmware. Wait until the compilation completes.
- 2. Choose Project > Download > Download file, as shown in Figure 3-12. Choose IAR\build\load_app.hex, as shown in Figure 3-13.

Pro	ject J-L	ink	Tools	Wind	ow	Help		~									
	Add File	es						-	<	Q	>	\$	►E	<	U	>	
	Add Gr	oup															
	Import	File Li	st														
	Add Pro	oject (onned	tion													
	Edit Co	nfigu	rations.														
	Remove																
	Create I	New F	roject														
	Add Exi	sting	Project														
	Option	s				Alt	+ F7										
	Version	Cont	rol Syst	em			•										
0	Make						F7										
	Compile	e				Ctrl	+F7										
	Rebuild	I AII															
	Clean																
	Batch b	uild					F8										
	C-STAT	Static	Analysi	is			•										
٥	Stop Bu	uild			C	Ctrl+Bi	reak										
0	Downlo	oad ar	nd Deb	ug		Ctr	I+D										
٠	Debug	witho	ut Dow	vnloadir	ng												
	Attach f	to Rui	nning T	arget													
G	Make 8	Rest	art Deb	ugger		Ctr	I+R										
C	Restart	Debu	gger		Ctr	l+Shif	t+R										
	Downlo	bad					-		Do	wnlo	ad a	ctive	app	licat	ion		
	SFR Set	up							1	wnlo						3	J
	CMSIS-	Pack					•		Erd	se m	emo	ry		-	_		1
	Open D	evice	Descrip	otion Fil	e		•										
	Save Lis	t of R	egister	s													
	Save Lis	t of R	egister	s													

Figure 3-12 Choosing Download file



Open							
÷ → • ↑	<pre></pre>	ripheral > ble_app_template >	IAR → build	ٽ ×	Search build		م
Irganize 🔻	New folder					•	•
·	Name	Date modified	Туре	Size			
1417410111	₩ load_app.bin	05/01/2021 11:14 05/01/2021 11:14	BIN File HEX File	80 KB 224 KB			
Y	File <u>n</u> ame:			~	All Files (*.*) Open	Cancel	N

Figure 3-13 Selecting load_app.hex

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.

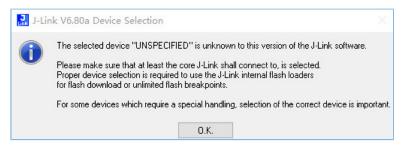


Figure 3-14 J-Link device selection window

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



Filter Manufacturer		Device	0		Little er	ndian	~
Manuracturer	~	Device	Core	~			
	Ť				Core #I	0	
Manufacturer	Device		Core	NumCores	Flash size	RAM size	^
Unspecified	Cortex-A9		Cortex-A9	1			
Unspecified	Cortex-A12		Cortex-A12	1			
Unspecified	Cortex-A15		Cortex-A15	1	-	2	
Unspecified	Cortex-A17		Cortex-A17	1	20	1	
Unspecified	Cortex-A53		Cortex-A53	1			
Unspecified	Cortex-A57		Cortex-A57	1		-	
Unspecified	Cortex-A72		Cortex-A72	1			
Unspecified	Cortex-M0		Cortex-M0	1			
Unspecified	Cortex-M0+		Cortex-M0	1	-8		
Unspecified	Cortex-M1		Cortex-M1	1	20	2	
Unspecified	Cortex-M3		Cortex-M3	1			
Unspecified	Cortex-M4		Cortex-M4	1		1	
Unspecified	Cortex-M7		Cortex-M7	1			
Unspecified	Cortex-M23		Cortex-M23	1	•		V

Figure 3-15 Selecting "Cortex-M4"

5. Press the **RESET** button on the GR55xx 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 b** on the menu bar to enter the debug mode. The program will run to the main() function line.

ble app example - IAR Embedded Work	here h IDE Ann 9 222		×
File Edit View Project Debug Disassemb		- 0	^
	· < Q, > \$ ⊨ < Q > 0 ▶ 1 ■ ● ⊨ O ⊂ Q ↓ A + A + 0 5 − ↓ Em >> [= #]		
	main.c x		÷
Debug ✓ Files ↔ He ble app_example ✓ He getend He geten	<pre> .epp_ble_init_cmp_callback = ble_init_cmp_callback, .epp_sup_callbacks = ispp_sup_callbacks, .epp_sup_callback = ispp_sup_callbacks, .epp_supt_callback = ispp_sup_callback, .epp_sup_callback = ispp_sup_callback, .epp_sup_callback = ispp_sup_callback, .epp_sup_init(); // Initialise user peripherals. epp_sup_init(); // Initialise user peripherals. ble_stack_init(is_spp_ble_callback, cheaps_table);//< init ble stack// // loop while (i) </pre>		fo
bie app. example	<pre></pre>		~ ~
	×		* 4 X
	wihi download suppressed urgee:D\Google_download)C4_B5676\projects\ble\ble_peripheral ble_app_template\\AR\Debug Exe\VAR_template.out ore after reset via DEMCR-VC_COREFESET. stelwith strategv): was performed		
Ready	Ln 77, Col 1 System 大写	数字 改写	

Figure 3-16 Entering the debug mode

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

GODIX

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.

Options for Target 'GR	-	C/C++ Asm Linker De	ebug Vtil:	ities	×
Preprocessor Symbols					
Define: GR5515_SK,D Undefine:	RIVER_TEST,APP_C	DRIVER_USE_ENABLE=0			
– Language / Code Generati	on				
Execute-only Code		Strict ANSI C	Warnings	All Warnings	-
Optimization: Level 1 (-01)	•	Enum Container always int		🗖 Thumb Mode	
Optimize for Time		Plain Char is Signed		🔲 No Auto Inclu	ides
Split Load and Store Mu	ltiple 🗌	Read-Only Position Independen	t	C99 Mode	
One ELF Section per Fu	inction	Read-Write Position Independer	nt	GNU extensio	ons
Include Paths Misc Controls	\build\config;\\	\\\components\boards;\\.	\\.\compon	ents\drivers_ext\g	
		apcs=interworksplit_sections /components/boards -I	-l/Inc -l		Ŷ
	OK	Cancel Defa	ults	н	elp

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.

Category: General Options Static Analysis	Multi-file Compila	ation used Publics			Fac	tory Settings:	
Runtime Checking C/C++ Compiler	MISRA-C:	1998	Encodings		Extra O	otions	
Assembler	Language 1	Language 2	Code	Optimiz	ations	Output	
Output Converter Custom Build	List	Preprocessor	Diagn	4.545.5		A-C:2004	
I-jet/JTAGjet J-Link/J-Trace TI Stellaris	\$PROJ_DIR\$\\\\\u00eduid\config \$PROJ_DIR\$\\\\\components\app_drivers\inc \$PROJ_DIR\$\\\\\components\boards						
Nu-Link PE micro ST-LINK	Preinclude						
	Defined symb	ools: (one per line)	^ Pr	eprocesso Preserve Generate	comment	ts	

Figure 4-2 Adding an undefined macro