

Embedded Studio IDE QuickStart

For Nuclei Processor Core

Release 1.1.0

Copyright © 2018–2020 Nuclei System Technology All rights reserved.

Copyright Notice

Copyright © 2018–2020 Nuclei System Technology. All rights reserved.

Nuclei[™] are trademarks owned by Nuclei System Technology. All other trademarks used herein are the property of their respective owners.

The product described herein is subject to continuous development and improvement; information herein is given by Nuclei in good faith but without warranties.

This document is intended only to assist the reader in the use of the product. Nuclei System Technology shall not be liable for any loss or damage arising from the use of any information in this document, or any incorrect use of the product.

Contact Information

Should you have any problems with the information contained herein or any suggestions, please contact Nuclei System Technology by email support@nucleisys.com, or visit "Nuclei User Center" website http://user.nucleisys.com for supports or online discussion.

Revision History

Rev	Revision Date	Revised Section	Revised Content
1.0.0	2020/02/15	N/A	1. First version as the full English
1.1.0	2020/08/04	ALL	 Note that path for software and project must be in English Add compile options instructions Modify the compilation option to macro when creating a new project When using j-link, directly modify the debug mode, and do not use GDB connection Update some old figures

Table of Contents

COPYRIGHT NOTICE	0
CONTACT INFORMATION	
REVISION HISTORY	
TABLE OF CONTENTS	
LIST OF FIGURES	
1. OVERVIEW	
1.1. INTRODUCTION OF EMBEDDED STUDIO	
1.2. INSTALLATION OF EMBEDDED STUDIO AND SETUP FOR NU	CLEI
1.3. INTRODUCTION OF HUMMINGBIRD EVALUATION KIT	
1.4. INTRODUCTION OF HUMMINGBIRD DEBUGGER KIT	
2. CREATE PROJECT	
2.1. OVERVIEW	
2.2. IMPORT PROJECT DIRECTLY FROM EXISTING PROJECT	
2.3. MANUALLY CREATE PROJECT WITHOUT TEMPLATE	
2.3.1. Manually Create Project	
2.3.2. Add the Source Codes for Project	
2.3.3. Set the Compile and Link Options for Project	
2.3.4. Set the Include File for Project	
3. COMPILE PROJECT	
3.1. Set Compile Options	
3.2. Compile Project demo_eclic in SES	
4. DOWNLOAD AND RUN PROJECT	
4.1. Set GDB Server according to Debugger Type	
4.1.1. Debug with Hummingbird Debugger Kit	
4.1.2. Debug with J-Link	
4.2. Set Printout Mode for printf according to Debugg	er Type
4.2.1. Printout through Serial Port	
4.2.2. Printout through RTT	
4.3. DOWNLOAD PROGRAM TO BOARD AND RUN	
5. DEBUG PROJECT	

List of Figures

FIGURE 1-1 HUMMINGBIRD EVALUATION KIT	
FIGURE 1-2 HUMMINGBIRD DEBUGGER KIT	7
FIGURE 2-1 DOWNLOAD PROJECT PACKAGE ON GITHUB	9
FIGURE 2-2 THE CONTENTS OF PACKAGE	9
FIGURE 2-3 THE CONTENTS OF THE HBIRD_EVAL_EXAMPLES FOLDER	9
FIGURE 2-4 FILE STRUCTURE OF THE EXISTING PROJECT	
FIGURE 2-5 SWITCH THE COMPILING AND DOWNLOADING MODES	11
FIGURE 2-6 CREATE NEW PROJECT	
FIGURE 2-7 PROJECT NAME AND LOCATION	
FIGURE 2-8 CHOOSE TARGET DEVICE	
FIGURE 2-9 ADD DEFAULT FILES	
FIGURE 2-10 ADD FOLDER TO THE PROJECT	
FIGURE 2-11 ADD FILE TO THE PROJECT	
FIGURE 2-12 THE STRUCTURE OF THE NEW PROJECT IN SES	
FIGURE 2-13 OPEN PROJECT CONFIGURATION	
FIGURE 2-14 CHANGE PROJECT TYPE	
FIGURE 2-15 CHANGE TOOL CHAIN DIRECTORY	
FIGURE 2-16 CHANGE COMPILATION OPTIONS	
FIGURE 2-17 CHANGE PROJECT MACROS	
FIGURE 2-18 SET USE MANUAL LINKER SCRIPT OPTION TO YES	
FIGURE 2-19 SET LINKER SCRIPT DIRECTORY	
FIGURE 2-20 PROJECT'S HEADER FILE PATH	
FIGURE 3-1 OPEN PROJECT OPTIONS	
FIGURE 3-2 OPEN SOLUTION OPTIONS	
FIGURE 3-2 VIEW MACROS DEFINED IN SOLUTION	
FIGURE 3-3 CHANGE PROJECT MACROS	
FIGURE 3-4 COMPILE THROUGH BUILD OPTION IN MENU BAR	
FIGURE 3-5 COMPILED SUCCESSFULLY	
FIGURE 4-1 HARDWARE CONNECTION OF HUMMINGBIRD DEBUGGER KIT	
FIGURE 4-2 OPEN PROJECT CONFIGURATION	
FIGURE 4-3 SET TO USE GDB SERVER	
FIGURE 4-4 USE OPENOCD TO CONNECT	
FIGURE 4-5 THE PIN DIAGRAM OF J-LINK	
FIGURE 4-6 HARDWARE CONNECTION OF J-LINK	
FIGURE 4-7 MODIFY PROJECT SETTINGS TO USE J-LINK	
FIGURE 4-8 OPEN SES SERIAL PORT TOOL	
FIGURE 4-9 OPEN THE SERIAL PORT SETTING POP-UP WINDOW	
FIGURE 4-10 SET BAUD RATE AND COM NUMBER	
FIGURE 4-11 OPEN SERIAL PORT	
FIGURE 4-12 THE OUTPUT OF PROJECT DEMO_ECLIC THROUGH SERIAL PORT	



FIGURE 4-13 CREATE AN EXTERNAL GNU USING PROJECT	41
FIGURE 4-14 ADDING SEGGER FOLDER	41
FIGURE 4-15 REDIRECT OUTPUT THROUGH RTT	
FIGURE 4-16 THE OUTPUT OF PROJECT DEMO_ECLIC THROUGH J-LINK	
FIGURE 4-17 CONNECT GDB	44
FIGURE 4-18 DOWNLOAD ELF FILE	44
FIGURE 4-19 DISCONNECT GDB SERVER	45
FIGURE 5-1 ENTER DEBUGGING MODE	46

1. Overview

This document will use Embedded Studio as IDE to develop embedded software on the FPGA evaluation board (called Hummingbird Evaluation Kit) with Debugger hardware (called Hummingbird Debugger Kit).

1.1. Introduction of Embedded Studio

SEGGER Embedded Studio (SES for short) developed by SEGGER Company, is a well-known Integrated Development Environment (IDE) for embedded software development.

SES has professional source code development and compilation user interface, powerful debugging features (equipped with famous J-Link).

SES is free for non-commercial usage, with stable cross platform compatibility and flexible configurations.

1.2. Installation of Embedded Studio and Setup for Nuclei

For the detailed installation steps of SES and how to set up the tool chain for Nuclei, please refer to the "Download" page of Nuclei website (http://www.nucleus.com/download.php) to download <Nuclei_SES_IDE_Installation.pdf>.

Note: Make sure the software installation path must be in English.

1.3. Introduction of Hummingbird Evaluation Kit

Nuclei have customized a FPGA evaluation board, called Hummingbird Evaluation Kit as shown in Figure 1-1. This FPGA board can be used as the SoC prototype board directly:

If the FPGA have been pre-burned (programmed) with "Nuclei evaluation SoC", this board can be worked as a SoC prototype directly. Since the board has been designed with buttons and extended ports names in line with the SoC GPIO pin name, the

embedded software engineers can directly use this board without knowing any FPGA hardware knowledge.

- To easy the writing, the "Nuclei evaluation SoC" will be shorted as "the SoC" in this document thereby.
- For the detailed introduction of the Hummingbird Evaluation Kit, please refer to the "Development Boards" page of Nuclei website (http://www.nucleisys.com/developboard.php) to download
 <Nuclei_FPGA_DebugKit_Intro.pdf>.

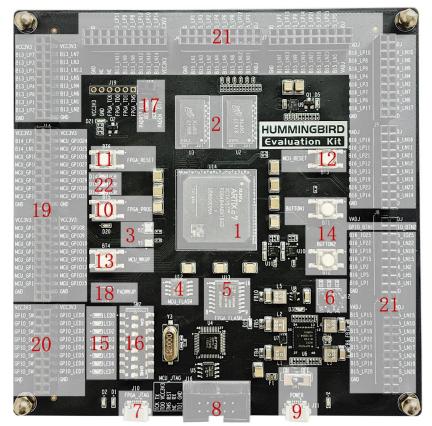


Figure 1-1 Hummingbird Evaluation Kit

1.4. Introduction of Hummingbird Debugger Kit

Nuclei have customized a Debugger hardware (called Hummingbird Debugger Kit), as shown in

Figure 1-2, which can be used to debug the RISC-V core in FPGA prototype or in real chip.

 For the detailed introduction of the Hummingbird Debugger Kit, please refer to the "Development Boards" page of Nuclei website (http://www.nucleisys.com/developboard.php) to download
 <Nuclei_FPGA_DebugKit_Intro.pdf>.



Figure 1-2 Hummingbird Debugger Kit

2. Create Project

2.1. Overview

There are two ways to create a new project in SES:

- Import project directly from existing project:
 - This is the most common way to create a new project. For example, user A can directly package an existing project; thereafter it can be easily shared and spread. User B can simply import the project on another computer, so as to use it, develop it or create a new project based on it.
- Create project manually without template:
 - This is the most tedious way to create a new project. It requires manual setting of options and paths. Because this way is inconvenient, it is seldom used in practical works. However, by explanations of this way, users can learn how to set various options and paths in details.

These two ways are described at follows.

2.2. Import Project Directly from Existing Project

This section will introduce how to use SES to create a new project by directly importing from an existing project.

This document takes demo-eclic as an example. The project package can be downloaded from Github (https://github.com/riscv-mcu/ses_nuclei_sdk_projects) as depicted in Figure 2-1. Note: this project is for demonstration, for more other examples you can contact Nuclei.



📮 riscv-mcu / ses_nuclei_sdk_j	projects		O Watch → 0	★ Star 0 % Fork 0
<>Code ① Issues 0 ℜ Pull r	equests 0 O Actions	III Projects 0 🗉 Wiki	Cecurity 🔟 Insights	
No description, website, or topics p	provided.			
2 commits	₽1 branch	🗇 0 packages	♡ 0 releases	a contributor
Branch: master - New pull request			Create new file Upload files	Find file Clone or download -
WsHelloWorld update			La	itest commit 30303ed 1 hour ago
hbird_eval_examples		update		1 hour ago
🖿 nuclei-sdk		update		1 hour ago
rvstar_demos		update		1 hour ago
rvstar_quick_start		update		1 hour ago
README.md		Initial comm	nit	1 hour ago

Figure 2-1 Download Project Package on Github

After decompressing the package, the contents are shown in Figure 2-2.

📜 hbird eval	examples
📕 nuclei-sdk	

Figure 2-2 The Contents of Package

Open the hbird_eval_examples folder, the contents are as shown in Figure 2-3. Please double click the file which marked in red-box to open SES and import project. If the operation is correct, the interface in Figure 2-4 will appears. The file structure of the project is shown in the red-box.

Note: Make sure the project path must be in English.

hbird_eval_examples.emProject
hbird_eval_examples.emSession

Figure 2-3 The Contents of the hbird_eval_examples Folder

bird_eval_examples - SEGGER Embedded Studio File Edit View Search Navigate Project			٥
Idemo_eclic 🔹 😨 🍟 🎽 💷 💷 🗿		u ← ¹ ¹ ² ² ^{2²} ²	
Citical Coluit Image: Citical Column Operative Citical Column Image: Citical Column Solution hbird_eval_examples' Project Column Project Citical Column Project Citical Column Project Citical Column Image: Citical Column Project Citical Column Image: Citical Column Project Citical Column Image: Citical Column Image: Citical Column Image: Citical Column	Code Data		
		Cutpus Show: Transcript	Q
		Mapping project ucosii_demo Mapping project whitstone Preparing spoilet "coremark' Preparing project "invrsome" Preparing project "freeros_demo' Preparing project "freeros_demo' Preparing project "timer_test' Preparing project "islowold' Preparing project "whetstone" Restoring state from previous session Preparing to load data Loading session data from file	

Figure 2-4 File Structure of the Existing Project

The Nuclei Evaluation SoC supports four "Compiling and Downloading" modes, including:

- ILM (Compile to run program from ILM);
- FLASH (Compile to run program from FLASH);
- FLASHXIP (Compile to upload program from FLASH and run from ILM).
- DDR(Compile to run program from DDR)

Please refer to document <Nuclei_Eval_SoC_Intro.pdf> for more information of the Nuclei Evaluation SoC.

The imported project is already set up in advance. As shown in Figure 2-5, the red-box indicates the location where user can switch the "Compiling and Downloading" modes.



Explorer			_	Source Navigator
Code Data H Code Data HXIP Code Data Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Configurations Image: Selid Config		Search Symbols A General Symbols A General Symbols A General Symbols A General Symbols A General Symbol Sym		
		Yesterday Trystar_demos Constant_demos	~	ECLIC_GetInfoVer() ECLIC_GetLevelIRQ(IRQn_Type) ECLIC_GetMth()
	Output Show: Transcript Tasks Image: Transcript Tasks Tasks Tasks	C 9	● M × ↓	
	Completed	7000 files	s/s	 _ECLIC_GetVector(IRQn_Type) _ECLIC_SetCfgNlbits(uint32_t)
	Mapping project information Dompleted	10 projec 10000 pr		 _ECLIC_SetCtrlIRQ(IRQn_Type, uint8_t) _ECLIC_SetLevelIRQ(IRQn_Type, uint8_t)
	Preparing solution 'hbird_eval_examples' Completed	10 projec 323 proje		ECLIC_SetMth(uint8_t) ECLIC_SetPendingIRQ(IRQn_Type)
	Restoring state from previous session Completed			ECLIC_SetPriorityIRQ(IRQn_Type, uint8_t) ECLIC_SetShvIRQ(IRQn_Type, uint32_t)
	 Completed SEGGER Embedded Studio is ready to use Completed 	10 projec 12 projec		ECLIC_SetTrigIRQ(IRQn_Type, uint32_t) ECLIC_SetVector(IRQn_Type, rv_csr_t) e_enable_all_counter() e_enable_irq()

Figure 2-5 Switch the Compiling and Downloading Modes

2.3. Manually Create Project without Template

This section describes how to manually create a project in SES.

As described in Section 2.1, this method is very tedious, which needs to manually set various options and paths. So, it is rarely used in practical work.

This document explains this method in details only to help users understand how to set up options and paths. For the user just want to quick start, this section can be skipped.

The detailed steps are as follows.

2.3.1. Manually Create Project

■ Select "File --> New Project" in the menu bar, as shown in Figure 2-6.



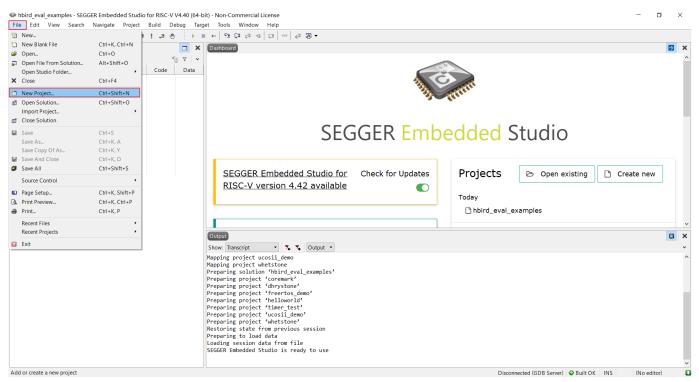


Figure 2-6 Create New Project

 Click "New Project" to pop up the window as shown in Figure 2-7, select the first option, enter the project name and select the folder location, then click next.



Don't see your device or board? Use the <u>Package Manager</u> to install packages		
scription	Manufacturer	Board
TTTTTTT		
A C/C++ executable for a RISC-V processor executing from FLASH memory.	Generic	Generic RISC-V
A C/C++ executable for a RISC-V processor executing from RAM memory.	Generic	Generic RISC-V
An externally built executable for a RISC-V processor.	Generic	Generic RISC-V Generic RISC-V
A C/C++ executable for a RISC-V processor executing from FLASH memory (internal tools and ext		
A C/C++ executable for a RISC-V processor executing from RAM memory (internal tools and exter		Generic RISC-V
An empty solution. A library project.	Generic Generic	(Standard Projects) (Standard Projects)
An object file project.	Generic	(Standard Projects)
A project for copying files to a target directory.	Generic	(Standard Projects)
A project for copying files to a target directory. A project for running a custom build when files have changed.	Generic	(Standard Projects)
A project for running a custom build when mes have changed.	Generic	(Standard Frojects)
ANDES		
ne: demo_eclic		

Figure 2-7 Project Name and Location

Select n307 (as an example) as the target device, as shown in Figure 2-8, then click next.



Select Target Device			
Select larget Device			
earch	Andesteen		
A25			· · · · · · · · · · · · · · · · · · ·
	GigaDevice		
GD32VF103C4T6			
GD32VF103C6T6			
GD32VF103C8T6			
GD32VF103CBT6			
GD32VF103R4T6			
GD32VF103R6T6			
GD32VF103R8T6			
GD32VF103RBT6			
GD32VF103T4U6			
GD32VF103T6U6			
GD32VF103T8U6			
GD32VF103TBU6			
GD32VF103V8T6			
GD32VF103VBT6			
• • • • • •	Nuclei System		
N101			
N205			
N207			
№ N208 № N305			
N 305			_
№ N308 № N605			

Figure 2-8 Choose Target Device

- Set to add default files, as shown in Figure 2-9. In this step, uncheck all files, and then click next.
- The new page does not need to be modified. Click "Finish" to finish.



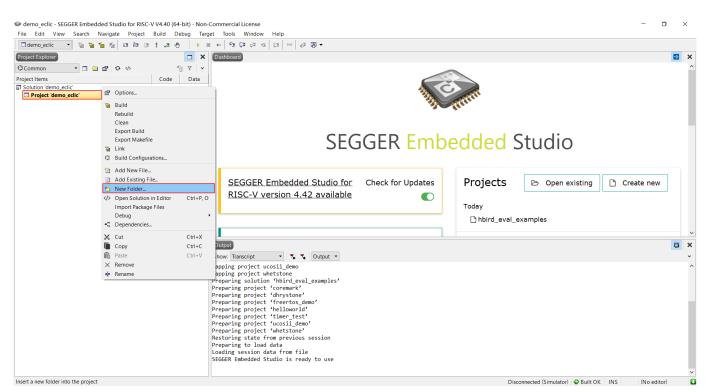
SEGGER Embedded Studio for RISC-V V4.40 - New Project					
Gelect files to add to project					
Files:					
E SEGGER					
 ▷ □ □ Setup ▷ □ □ Source 					
□ □ System					
Import all files and package files					
	Back	Next	Cancel		
	DACK	INCAL	Cancer		

Figure 2-9 Add Default Files

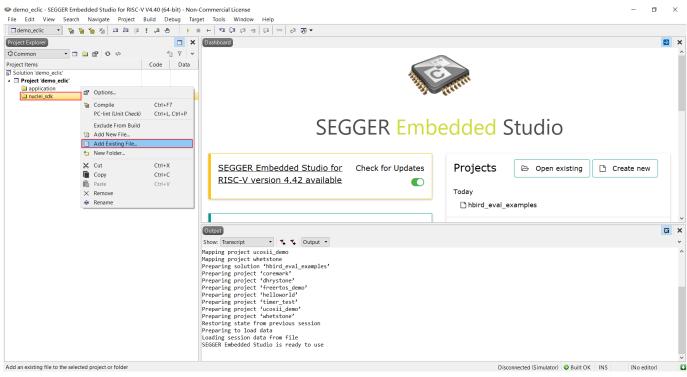
2.3.2. Add the Source Codes for Project

- Please create new folders and add files according to the actual file structure of project package downloaded in Section 2.2.
- Right click the new project created in the previous section to open the right-click menu, and select "New Folder", as shown in Figure 2-10.
 - As an example, please create two new folders and name them with "application" and "nuclei_sdk", where "application" folder is used to save application codes, and "nuclei_sdk" folder is used to save Nuclei-SDK.
- Right click the newly created folder and select "Add Existing File" to add files to the project, as shown in Figure 2-11.
 - Note: Please add only the contents under the hbird folder in SoC directory. The final project directory is shown in Figure 2-12.

A 芯来科技 NUCLEI











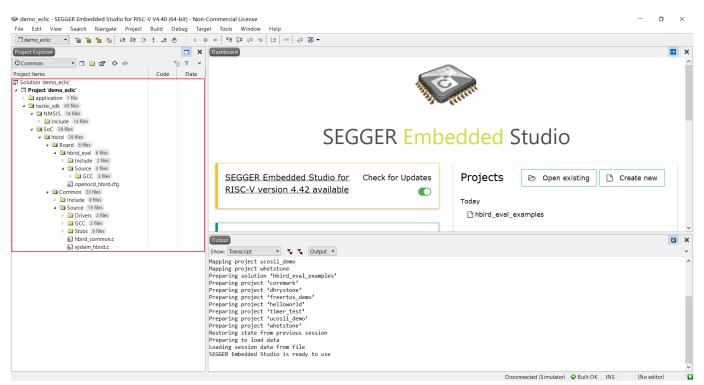


Figure 2-12 The Structure of the New Project in SES

2.3.3. Set the Compile and Link Options for Project

The Nuclei Evaluation SoC supports four "Compiling and Downloading" modes, including:

- ILM (Compile to run program from ILM);
- FLASH (Compile to run program from FLASH);
- FLASHXIP (Compile to upload program from FLASH and run from ILM).
- DDR(Compile to run program from DDR)

Please refer to document <Nuclei_Eval_SoC_Intro.pdf> for more information of the Nuclei Evaluation SoC.

This document takes ILM mode as an example, the setting steps are as follows.

 Select "Project --> Options" in the menu bar to open the project settings pop-up window, as shown in Figure 2-13. Modify the "Project Type" to "External Built Executable" in the "Build" column, as shown in Figure 2-14. Click the OK button in the pop-up window to save the settings.

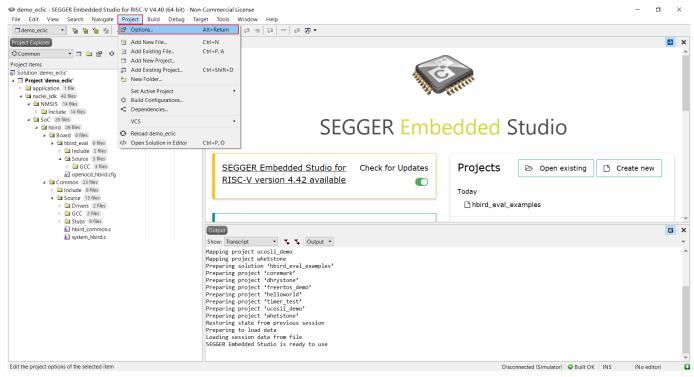


Figure 2-13 Open Project Configuration

SEGGER Embedded Studio for F Project 'demo_irqc' Optic			×
↑ ↓ 🗘 Debug 🕞	Search Options		Show Modified Options Only
↑ ↓ ♪ Debug • Code Assembler Build Code Generation Compiler External Build File Libraries Libraries Library Linker Preprocessor Printf/Scanf Runtime Memory Area Section Source Code User Build Step Usebugger GDB Server J-Link Loader Simulator Target Script	Search Options	Value No Yes None Yes Output/\$(ProjectName) \$(Config None FLASH1 RX 0x0000000 0x00100 Output/\$(Configuration)/Exe Default None modified Executable inherits None No C\Program Files\SEGER\SEGG No Yes	puration)/Obj 0000,RAM 1 RWX 0x20000000 0x00010000 [inherits SEGGER Embedded Studio for RISC-V V4.40 - Proper × Set Project Type Project demo_irqc Configuration: Debug Project Type: Externally Built Executable ▼
	Project Type Specifies the type of project to build. The options are Exe Inherits "Executable" from project in Common configuration	cutable, Library, Object file, Staging, Combining, Extern	OK Cancel Specifies the type of project to build. The options are Executable, Library, Object file, Staging, Combining, Externally Built Executable, Externally Built Library, Externally Built Object file.
			OK Cancel

Figure 2-14 Change Project Type

Select "Project --> Options" in the menu bar again to open the project settings pop-up window. Modify the "Tool Chain Directory" under the "Build" column to "\$(StudioDir)/Nuclei_Toolchain/gcc/bin", as shown in Figure 2-15.

SEGGER Embedded Studio f	or RISC-V V4.40 - Options	>
Project 'demo_irqc' Op	btions	
↑ ↓ ۞ Debug	Search Options	Show Modified Options Only
▲ Code	Option	Value
Assembler Build	A Build	
Compiler External Build File Linker Preprocessor Source Code User Build Step Debugger GDB Server J-Link Loader Simulator Target Script	Always Rebuild Advays Rebuild Advays Rebuild Advays Rebuild Advays Rebuild Advays Reprint Rep	No None No Output/\$(ProjectName) \$(Configuration)/Obj None FLASH1 RX 0x0000000 0x00100000 RAM1 RWX 0x2000000 0x00010000 inherits Output/\$(Configuration)/Exe Default None modified Kone modified None \$(StudioDir)/Nuclei_Toolchain/gcc/bin
		OK Cancel

Figure 2-15 Change Tool Chain Directory

 Modify the compilation options, as shown in Figure 2-16. Fill in the compilation instructions as follows.

```
Assemble Command: "$(ToolChainDir)/riscv-nuclei-elf-gcc" $(CORE_FLAGS)
$(COMMON_FLAGS) $(GC_CFLAGS) $(AsmOptions) $(Defines) $(Includes) -MD
-MF "$(RelDependencyPath)" -c -o "$(RelTargetPath)" "$(RelInputPath)"
```

```
C Compile Command: "$(ToolChainDir)/riscv-nuclei-elf-gcc" $(CORE_FLAGS)
$(COMMON_FLAGS) $(GC_CFLAGS) $(COptions) $(COnlyOptions) $(Defines)
$(Includes) -MD -MF "$(RelDependencyPath)" -c -o "$(RelTargetPath)"
```

"\$(RelInputPath)"

C++ Compile Command: "\$(ToolChainDir)/riscv-nuclei-elf-g++" \$(CORE_FLAGS) \$(COMMON_FLAGS) \$(GC_CFLAGS) \$(COptions) \$(CppOnlyOptions) \$(Defines) \$(Includes) -MD -MF "\$(RelDependencyPath)" -c -o "\$(RelTargetPath)" "\$(RelInputPath)"

Link Command: "\$(ToolChainDir)/riscv-nuclei-elf-gcc" \$(CORE_FLAGS) \$(COMMON_FLAGS) \$(GC_LDFLAGS) \$(NEWLIB_LDFLAGS) \$(EXTRA_LDFLAGS) --specs=nosys.specs -MMD -MT \$(ProjectName)\$(EXE) -MF \$(ProjectName)\$(EXE).d \$(Objects) -o "\$(OutDir)/\$(ProjectName)\$(EXE)" -T "\$(RelLinkerScriptPath)" -lstdc++ -nostartfiles -Wl,-M,-Map="\$(RelMapPath)" \$(LinkOptions)

SEGGER Embedded Stu	udio for RISC-V V4.52b - Options	×
Project 'helloworld	' Options	
↑↓ © FLASHXIP	✓ Search Options	Show Modified Options Only
▲ Code	Option	Value
Assembler		
Build	🕘 🗉 External Build	
Compiler	Assemble Command	"\$(ToolChainDir)/riscv-nuclei-elf-gcc" \$(CORE_FLAGS) \$(COMMON_FLAGS) \$(GC_CF
External Build	Build Command	None
File	C Compile Command	"\$(ToolChainDir)/riscv-nuclei-elf-gcc" \$(CORE_FLAGS) \$(COMMON_FLAGS) \$(GC_CF
Linker	C++ Compile Command	"\$(ToolChainDir)/riscv-nuclei-elf-g++" \$(CORE_FLAGS) \$(COMMON_FLAGS) \$(GC_C
Preprocessor	C++ Link Command	None
Source Code	Clean Command	None
User Build Step	Link Command	"\$(ToolChainDir)/riscv-nuclei-elf-gcc" \$(CORE_FLAGS) \$(COMMON_FLAGS) \$(GC_LC
▲ Debug	Objects File	None
Debugger		
GDB Server		
J-Link		
Loader		
Simulator	(No Property)	
Target Script		
		OK Cancel

Figure 2-16 Change compilation options

 Modify the Project Macros. As shown in Figure 2-17, Open Build options and modify the Project Macros option to the following. Note that each Macro should be separated by a new line.

CORE_FLAGS=-march=rv32imafc -mabi=ilp32f -mcmodel=medany

COMMON_FLAGS=-g -fno-common

-DDOWNLOAD_MODE=DOWNLOAD_MODE_FLASHXIP

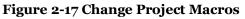
GC_CFLAGS=-ffunction-sections -fdata-sections

GC_LDFLAGS=-Wl,--gc-sections -Wl,--check-sections

NEWLIB_LDFLAGS=--specs=nano.specs

EXTRA_LDFLAGS=-u _isatty -u _write -u _sbrk -u _read -u _close -u _fstat -u _lseek

SEGGER Embedded Studi	io for RISC-V V4.52b - Options	×
Project 'helloworld' (Options	
↑ ↓ © FLASHXIP	Search Options	Show Modified Options Only
 Code Assembler Build Compiler External Build File Linker Preprocessor Source Code User Build Step Debug Debugger GDB Server J-Link Loader Simulator Target Script 	Option * Build Options Generic File Name • Exclude From Build • External Compiler • Intermediate Directory • Memory Map File • Memory Map File • Memory Segments • Output Directory • Project Can Build In Parallel • Project Directory • Project Directory • Project Directory • Project Type • Project Type • Project Macros Specifies macro values which are expanded in project properties a name=value and are seperated by ;. Inherits "CORE_FLAGS=-march=rv32imafc -mabi=ilp32f -mcmodel=medel	Value None No None Output/\$(ProjectName) \$(Configuration)/Obj None FLASH RX 0x08000000 0x00020000;RAM RWX 0x20000000 0x00008000 modif Output/\$(Configuration)/Exe Default Default None Mone Sterrally Built Executable modified None \$(StudioDir)/Nuclei_Toolchain/gcc/bin inherits V any ;COMMON_FLAGS=-g -fno-common - V
		OK Cancel



Modify the "Use Manual Linker Script" option under "Linker" column to "Yes", as shown in Figure 2-18. When the option is set to "Yes", a new option is added, as shown in Figure 2-19. Enter the path of link script in this option.



SEGGER Embedded Studio for RISC-	-V V4.40 - Options		×
Project 'demo_irqc' Options			
↑↓ Oebug - Sea	rrch Options		Show Modified Options Only
Code Assembler Build Compiler External Build File Linker Preprocessor Source Code User Build Step Debug Debugger GDB Server J-Link Loader Simulator Target Script	Option • Linker • Executable File Name • Link Dependent Projects • Use Manual Linker Script • Additional Linker Options • Backup Additional Linker Options	Value \$(OutDir)/\$(ProjectName)\$(EXE) Yes No None None	SEGGER Embedded Studio for RISC-V V4.40 - Proper × Set Use Manual Linker Script Project: demo_irqc Configuration: Debug Use Manual Linker Script: Yes OK Cancel Specifies whether to use a manual linker script.
			OK Cancel

Figure 2-18 Set Use Manual Linker Script Option to Yes

SEGGER Embedded Studio for RISC	-V V4.40 - Options		×
Project 'demo_irqc' Options			
	arch Options		
	arch Options		Show Modified Options Only
▲ Code	Option	Value	
Assembler			
Build	🖌 📕 Linker		
Compiler	Executable File Name	\$(OutDir)/\$(ProjectName)\$(EXE)	
External Build	Link Dependent Projects	Yes	
File Linker	Use Manual Linker Script	Yes modified	
Preprocessor	Linker Script File Additional Linker Options	\$(SolutionDir)//n100-sdk/bsp/core/env/link_ilm.lds	
Source Code	Backup Additional Linker Options	None	
User Build Step	backap / talitional critici o pitono	Hone	
▲ Debug			
Debugger			
GDB Server			
J-Link			
Loader			
Simulator			
Target Script			
	(No Property)		
			OK Cancel

Figure 2-19 Set Linker Script Directory

2.3.4. Set the Include File for Project

Select "Project --> Options" in the menu bar to open the project settings pop-up window, as shown in Figure 2-20. Under the "Preprocessor" option, enter the required header file path for the project, including nuclei_sdk header file path and application header file path.

SEGGER Embedded Studio for RISC-	-V V4.40 - Options	X
Project 'demo_irqc' Options		
↑ ↓ Oebug • Sea	arch Options	Show Modified Options Only
Code Assembler Build Compiler External Build File Linker Preprocessor Source Code User Build Step	Option Preprocessor Preprocessor Definitions System Include Directories User Include Directories User Include Directories	Value Inherits S(SolutionDir)/_/n100-sdk/bsp/core/drivers_\$(SolutionDir)/_/n100-sdk/bsp/core/stubs\;\$(SolutionDir)/_/n100-sdk/bsp/core***
Debug Debugger GDB Server J-Link Loader Simulator Target Script		
	(No Property)	
		OK Cancel

Figure 2-20 Project's Header File Path

3. Compile Project

3.1. Set Compile Options

Whether you are using an existing project or creating a project manually, please do not modify the content of code > external build unless necessary. This article takes the project imported from an existing project as an example to introduce how to modify the compilation options.

As shown in Figure 3-1, to open Project Options pop-out window, right click in the current project to open the right-click menu bar, then select Options. As shown in Figure 3-2, click the up arrow to enter the Solution Options. As shown in Figure 3-3, in "Code > Build > Project Macros" can find "CORE_FLAGS, COMMON_FLAGS, GC_CFLAGS, GC_LDFLAGS, NEWLIB_LDFLAGS and EXTRA_LDFLAGS". Each macro is explained separately below.

- CORE_FLAGS: Saves Core related options
- COMMON_FLAGS: Saves common options
- GC_CFLAGS: Compile options when using "gc sections"
- GC_LDFLAGS: Link options when using "gc sections"
- NEWLIB_LDFLAGS: Link options when using NEWLIB
- EXTRA_LDFLAGS: Extra Link options

If you want to modify the above compilation options, you can modify the project settings options. Open the project settings page, as shown in Figure 3-4, and modify the macro at the Project level. It should be noted that the Project level will overwrite the Solution level for the macro with the same name. Therefore, if you want to add compilation options in current macros, please make sure to copy the original macros first. As shown in Figure 3-5, here we change optimization level to "-O2". Because the Project level will overwrite the Solution level, we should copy the contents of "COMMON_FLAGS" first. Then we add "-O2" at front. Remember to add space between each option.



t Explorer		• •		8	×	Source Navigator		*	-
SHXIP 🔹 🖬 🖨	ο φ	1 V V			^	Search Symbols	3 .= 1 + (& ×	
Items		Code Data							
tion 'rvstar_example'	Options	2.8				 Functions 749 function _AMOADD_W(volation) 			
r Grapplation 106 i Grapplatio	Wild Rebuild Clean Export Build Export Makefile Wild Configurations Add New File Add Statisting File You Package Files Debug Debug Column of Laborations	Ctrl+P, O	SEGGER Embedded Check for Studio for RISC-V is up to Updates C date	edded Studio Projects Create new Today	 _AMOAND_Wivoli _AMOMAX_Wivoli _AMOMAX_Wivoli _AMOMIN_Wivola _AMOMIN_Wivola _AMOMIN_Wivola _AMOS Wivolatile _AMOS Wivolatile _CL2/win122,0 _disable_all.counter _disable_all.counter _disable_mryde_co _EGR4K0 _ECALL_ClearPenditi 	vunter() de_counter() tret_counter() PendingIRQ(IRQn_Type)			
	Cut Copy Paste Remove Rename Edit Linker Script	Ctrl+X Ctrl+C Ctrl+V	All packages are up to Check for date Packages	Yesterday hbird_eval_examples rvstar_demos		ECLIC_DisableIRQ(IRQn_Type) ECLIC_EnableIRQ(IRQn_Type) ECLIC_EnableIRQ(IRQn_Type) ECLIC_GetCrlRQIIRQn_Type) ECLIC_GetCrlRQ(IRQn_Type) ECLIC_GetInfoCtBhsj ECLIC_GetInfoCtBhsj ECLIC_GetInfoCtBhsj			
				Disaster dance	 _ECLIC_GetLevelIF 				
			Output Show: Transcript • 7. 7. Tasks •	0 a a a	n X	ECLIC_GetMth() ECLIC_GetPending ECLIC_GetPriority ECLIC_GetShvIRQ	RQ(IRQn_Type)		
			Completed Completed Completed	2 files in 0.0s 2000 files/s		 _ECLIC_GetTrigIRQ _ECLIC_GetVector(I _ECLIC_SetCfgNlbi _ECLIC_SetCtrlIRQ 	(IRQn_Type) RQn_Type) ts(uint32_t) (IRQn_Type, uint8_t)		
			Preparing solution 'vstar_example' Completed			ECLIC_SetLevelIRe ECLIC_SetMth(uin ECLIC_SetPending ECLIC_SetPriorityI	t8_t) IRQ(IRQn_Type)		
			SEGGER Embedded Studio is ready to use Completed			ECLIC_SetShvIRQ ECLIC_SetTrigIRQ ECLIC_SetVector(I enable_all_counter	(IRQn_Type, uint32_t) (IRQn_Type, uint32_t) RQn_Type, rv_csr_t)	t)	

Figure 3–1 Open Project Options

SEGGER Embedded Studio	for RISC-V V4.52b - Options		X
Project 'helloworld' O	ptions		
↑↓ ODR	Search Options		Show Modified Options Only
▲ Code	Option	Value	^
Assembler Build	₄ ■ Assembler		
Compiler	 Additional Assembler Options 		
External Build	 Backup Additional Assembler Options 	None	
File Linker	≁ ■ Build		
Preprocessor	 Always Rebuild 	No	
Source Code	 Batch Build Configurations 		
User Build Step	 Build Options Generic File Name 	None	
▲ Debug	Exclude From Build	No	
Debugger	External Compiler	None	
GDB Server	Intermediate Directory	Output/\$(ProjectName) \$(Configuration)/Obj	
J-Link	Memory Map File Memory Map Macros	None	~ ·
Loader			¥
Simulator			
Target Script			
	Open Solution Options		
			OK Cancel

Figure 3–2 Open Solution Options



SEGGER Embedded St	udio for RISC-V V4.52b - Options		SEGGER Embedded Studio for RISC-V V4.52b - Property Editor		\times
Solution 'rvstar_exa	ample' Options		Set Project Macros		
 ↓ CELASHXIP Code Assembler Build Compiler External Build File General Linker Preprocessor Source Code User Build Step Debugger GDB Server J-Link Loader Simulator Target Script 	Search Options Option Build Configurations Build Options Generic File Name Intermediate Directory Memory Map File Memory Map File Memory Segments Output Directory Project Macros Property Groups File Tool Chain Directory Project Macros Specifies macro values which are expanded in project properties and are separated by ;.	CORE None \$(Stud	Project Macros: CORE_FLAGS=-march=rv32imac -mabi=ilp32 -mcmodel=medany COMMON_FLAGS=-g -fno-common -DDOWNLOAD_MODE=DOWNLOAD_MODE_FLASHXIP GC_CFLAGS=-flunction-sections -fdata-sections GC_LDFLAGS=-wigc-sections -Wicheck-sections NEWLIB_LDFLAGS=-visatty -uwrite -u_sbrk -u_read -u_close -u_fstat -u_lseek L Macros:		۲
			Specifies macro values which are expanded in project properties and for file names in Common config defined as name=value and are seperated by ;	OK Car guration only. Each macro	



roject 'helloworld	•	
• ↓ ॒ FLASHXIP	Search Options Option	Value Show Modified Options On
Assembler	Always Rebuild	No
Build	Batch Build Configurations	
Compiler	Build Options Generic File Name	None
External Build	Exclude From Build	No
File	External Compiler	None
Linker	Intermediate Directory	Output/\$(ProjectName) \$(Configuration)/Obj
	 Memory Map File 	None
Preprocessor	 Memory Map Macros 	
Source Code	Memory Segments	FLASH RX 0x08000000 0x00020000;RAM RWX 0x20000000 0x00008000 modif
User Build Step	Output Directory	Output/\$(Configuration)/Exe
Debug	 Project Can Build In Parallel 	Default
Debugger	 Project Dependencies 	
GDB Server	Project Directory	None modified
J-Link	 Project Macros 	inherits
Loader	Project Type	Externally Built Executable modified
Simulator	name=value and are seperated by ;.	
Target Script	Inherits	
	DDOWNLOAD_MODE=DOWNLOAD_MODE_FLAS	model=medany ;COMMON_FLAGS=-g -fno-common - HXIP;GC_CFLAGS=-ffunction-sections -fdata-sections;GC_LDFLAGS=-Wl,gc-sections -Wl, :cs;EXTRA_LDFLAGS=-u _isatty -u _write -u _sbrk -u _read -u _close -u _fstat -u _lseek" from





^o roject 'helloworld' (Options	Set Project Macros	
 ↓ ♥ FLASHXIP Assembler Build Compiler External Build File Linker Preprocessor Source Code User Build Step Debug Debug Debuger GD8 Server J-Link Loader Simulator Target Script 	Search Options Option Suild Options Generic File Name Exclude From Build External Compiler Intermediate Directory Memory Map Macros Memory Map Macros Memory Segments Output Directory Project Can Build In Parallel Project Dependencies Project Directory Project Macros Project Type Property Groups File Tool Chain Directory Project Type Specifies macro values which are expanded in project properties name=value and are seperated by ;. Inherits "CORE_FLAGS=-march=rv32imac -mabi=ilp32 -mcmodel=med.	Macros:	OK Cancel

Figure 3-5 Change "COMMON_FLAGS" In Project Level

If the compilation options to be added are not in the above categories, you can add options by yourselves. Make sure each command takes a line in the pop-out window.

Add assembler options in "Code > Assembler > Additional Assembler Options".

Add C compiler only options in "Code > Compiler > Additional C Compiler Only Options".

Add C/C++ compiler only options in "Code > Compiler > Additional C/C++ Compiler Option".

Add C++ compiler only options in "Code > Compiler > Additional C++ Compiler Only Options".

Add Link options in "Code > Linker > Additional Linker Options".

3.2. Compile Project demo_eclic in SES

This section takes the imported project demo_eclic (described in Section 2.2) as an example to introduce how to compile the project and generate Elf file in SES.

After the "compiling and downloading modes (DDR, ILM, FLASH or FLASHXIP)" is selected, to compile the current project, press F7 directly on the keyboard or click the first option under "Build" in the menu bar, as shown in Figure 3-6.

After the compilation is successful, the user interface is as shown in Figure 3-7.

🗆 demo_eclic 🔹 🖓 🍟 🎢 🚛 🖽		F7	r ¢≣ 5	9 -					
Project Explorer	Rebuild demo_eclic Clean demo_eclic	Alt+F7							
】Debug External ・ □ □ □ □ □ ● ゆ か roject Items I Solution 'hbird eval examples'	Build Solution Rebuild Solution	Shift+F Alt+Sh							
Project 'coremark' Project 'demo_eclic'	Clean Solution	Ctrl+F		Studio for	Check for Updates	Projects	🕞 Open existing	🗅 Create new]
Project 'dhrystone' Project 'freertos_demo' Project 'helloworld'	Build and Debug	Ctrl+T, Ctrl+T	5	available		Today		,	-
application 1 file application 4 file	Kancel Build	Ctrl+.				hbird_eval	_examples		
SEGGER 5 files, modified options Project 'timer_test' Project 'ucosii_demo' Project 'whetstone'	Build Configurations Set Active Build Confi Batch Build Parallel and Unity Buil	iguration	, re up i	o date	Check for Packages 🌑	Yesterday	:		
	Show Build Log	la Ctrl+B	trl+T			hbird_eval	_examples		
						Last Seven Day			
						Prystar eva	mnles		
		Output							Ţ,
	, p , p , p , p , p , p , p , p , p , p	Show: Transcript lapping project Preparing proje Preparing proje Preparing proje Preparing proje Preparing proje Preparing proje Preparing proje Preparing proje Restoring state Lestoring state SEGER Embedded	<pre>n 'hbird_eval_('dhrystone' 'whetstone' 'helloworld' 'timer_test' 'freertos_demm 'ucosii_demo' 'coremark' 'Executable_1: rom previous se data ta from file</pre>	xamples'					

Figure 3-6 Compile Through Build Option in Menu Bar

A 芯来科技 NUCLEI

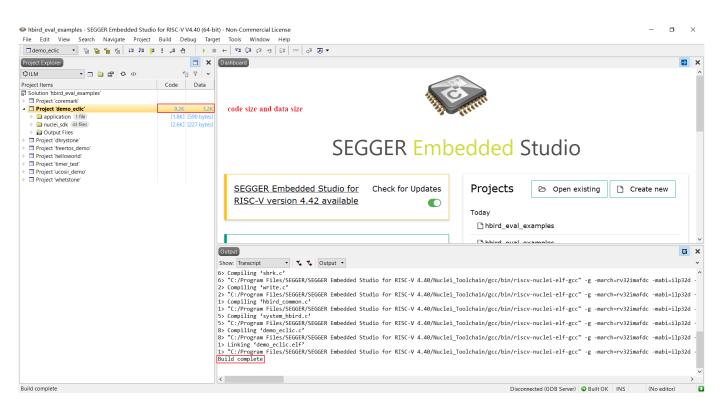


Figure 3-7 Compiled Successfully

4. Download and Run Project

4.1. Set GDB Server according to Debugger Type

SES supports two Debugger types for Nuclei:

- Debug with its native J-Link debugger.
- Debug with Hummingbird Debugger Kit.

Before debugging or downloading the program, user needs to set up GDB Server according to the debugger type.

The steps to set up GDB Server (according to the Debugger type) are detailed as follows.

4.1.1. Debug with Hummingbird Debugger Kit

For the details of Hummingbird Debugger Kit, please refer to Section 1.4.

The Hummingbird Debugger Kit is connected to the host PC through USB, therefore, USB driver needs to be installed. About how to connect Hummingbird Debugger Kit, and how to install its driver, please refer to the document <Nuclei_FPGA_DebugKit_Intro.pdf> which can be downloaded from "Development Boards" page of Nuclei website (http://www.nucleisys.com/developboard.php).

The final hardware connection is shown in Figure 4-1. After the correct connection, turn on the power switch to connect board.



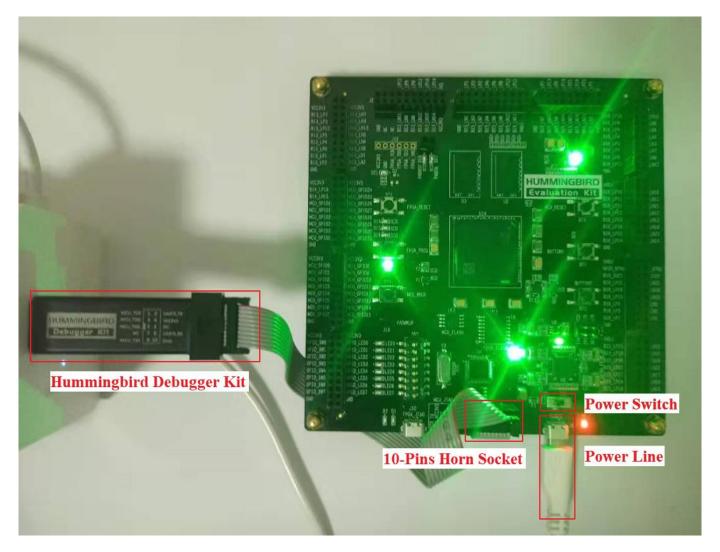
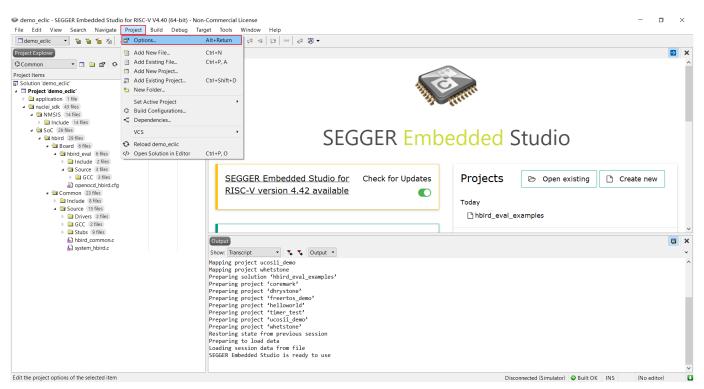


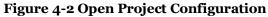
Figure 4-1 Hardware Connection of Hummingbird Debugger Kit

The steps to set up GDB Server for Hummingbird Debugger Kit are as follows:

- Click "Options" under "Project" option in the menu bar, as shown in Figure 4-2.
- Change the "Target Connection" option to "GDB Server" under the "Debugger" column, as shown in Figure 4-3.

A 芯来科技 NUCLEI





COLIC EINDEdded Stadio I	or RISC-V V4.40 - Options		
oject 'demo_irqc' Op	otions		
🔶 🤤 Debug	Search Options		Show Modified Options
Code	Option	Value	
Assembler	option	value	
Build			
Compiler	Debugger		
External Build	Target Connection	GDB Server (modified)	
File	Run To Control	Always	
	* Run To	main	
Linker	 Startup Completion Point 	main	
Preprocessor	Start From Entry Point Symbol	Yes	
Source Code	 Leave Target Running 	No	
User Build Step	 Register Definition File 	None	
Debug	 Debug Terminal Log File 	None	
Debugger	 Threads Script File 	None	
GDB Server	Thread Maximum	25	
J-Link	Working Directory	\$(ProjectDir)	
Loader	 Command Arguments 	\$(ProjectName)\$(EXE)	
	 Entry Point Symbol 	None	
Simulator	 Load Additional Projects 		
Target Script	PULP Extensions Debug	Yes	
	 RTT Control Block Address 	_SEGGER_RTT	
	RTT Enable	Yes	
	 Starting Stack Pointer Value 	None	
	Target Device	N101 inherits	
	 Debug Symbols File[0] 	None	
	 Debug Symbols Load Address[0] 	None	
	 Debug Symbols File[1] 	None	
	 Debug Symbols Load Address[1] 	None	
	 Debug Symbols File[2] 	None	
	 Debug Symbols Load Address[2] 	None	
	 Debug Symbols File[3] 	None	
	Debug Symbols Load Address[3]	None	
	Target Connection		
	Specifies the target to connect to for debugging actions.		
			ОК Саг



- As shown in Figure 4-4, There are three settings under the "GDB Server" option should be changed:
 - Change the type to OpenOCD.
 - Change the "GDB Server Command Line" to
 "\$(studiodir)/Nuclei_Toolchain/OpenOCD/bin/OpenOCD" -f, and add the path of
 OpenOCD's configuration file. Here you should select the configuration file
 corresponding to the Downloading mode (DDR, ILM, FlASH, or FLASHXIP).
 - Modify "Auto Start GDB Server" to "Yes".
- After the above modifications, click OK to save the project settings.

SEGGER Embedded Studio for RISC-V V4.40 - Options								
Project 'demo_irqc' Options								
↑ ↓ 🗘 Debug	Search Options	Show Modified Options Only						
 Code Assembler Build Compiler Extemal Build File Linker Preprocessor Source Code User Build Step Debug Debug Debug Debug Debug J-Link Loader Simulator Target Script 	Option • Host • Type • GDB Server Command Line • Auto Start GDB Server • Port • Reset and Stop Command • Ignore Checksum Errors • Allow Memory Access During Execution • Register Access • Log File • Target XML File • Connect Timeout • Read Timeout • Write Timeout	Value OpenOCD imodified "SStudioDir/Nuclei_Toolchain/openocd" -f "\$(SolutionDir)/_/n100-sdk/bsp/core/env/openocd_hbird_ilm.cfg" Yes imodified "set halt imodified Ves imodified "General and Individual imodified Nore Nore Seconds 60 seconds						
	(No Property)	OK Cancel						



4.1.2. Debug with J-Link

SES certainly supports the J-Link Debugger Probe. Note: when using J-Link for debugging, only

ILM mode can be supported. For more detailed reason about this problem, please contact Nuclei.

The key points for connecting of J-Link are as below:

- Prepare several jumper wires.
- The pin diagram of J-Link is as shown in Figure 4-5, with the red-box marked as the pin to be connected.
- The pin diagram of Hummingbird Evaluation Kit FPGA board is next to the 10-Pins Horn Socket as shown in Figure 4-6.
- Connect other pins first, and then lastly connect the "VTref" pin to VCC₃V₃ on Hummingbird Evaluation Kit.
- The hardware connection of J-Link is as shown in Figure 4-6. If the connection is correct, turn on the power switch.

VTref	1 💿	• 2	NC
nTRST	3 •	• 4	GND
TDI	5 •	• 6	GND
TMS	7 •	• 8	GND
тск	9 •	• 10	GND
RTCK	11 •	• 12	GND
TDO	13 •	• 14	*
RESET	15 •	• 16	*
DBGRQ	17 •	• 18	*
5V-Supply	19 •	• 20	*

Figure 4-5 The Pin Diagram of J-Link



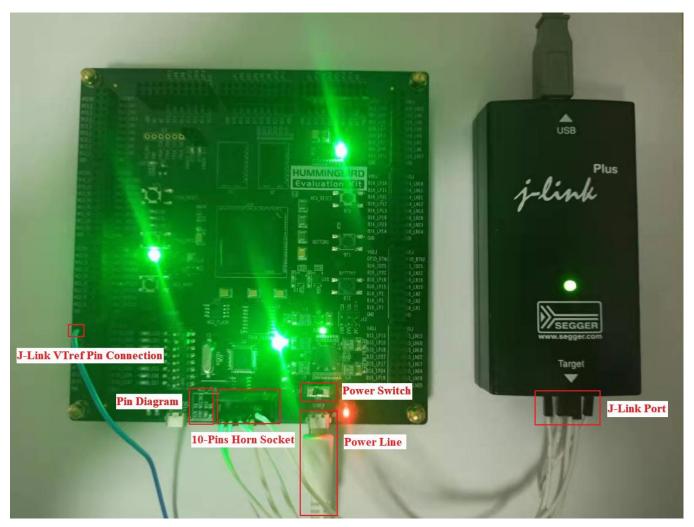


Figure 4-6 Hardware Connection of J-LINK

The steps to set up GDB Server for J-Link are as follows:

- Click "Options" under "Project" option in the menu bar, as shown in Figure 4-2.
- Switch the "Target Connection" option to "J-Link" under the debugger column, as shown in Figure 4-7.



SEGGER Embedded Stu	dio for RISC-V V4.52b - Options		X
Project 'helloworld'	Options		
↑ ↓ 🗘 FLASHXIP	Search Options		Show Modified Options Only
⊿ Code	Option	Value	^
Assembler Build	₄ ■ Debugger		
Compiler	Target Connection	J-Link (modified)	-
External Build	Run To Control	Always	
File	Run To	main	
Linker	 Startup Completion Point 	main	
Preprocessor	 Start From Entry Point Symbol 	Yes	
Source Code	 Leave Target Running 	No	
User Build Step	Register Definition File	None	
▲ Debug	Debug Terminal Log File	None	
Debugger	Threads Script File	None	
GDB Server	Thread Maximum	25	
J-Link	Working Directory	\$(ProjectDir)	
Loader	 Command Arguments 	\$(ProjectName)\$(EXE)	
Simulator			
Target Script	Target Connection		
larger Script	Specifies the target to connect to for debugging action	ons.	
	Inherits		
	"GDB Server" from solution in FLASHXIP configurati	on	
			OK Cancel

Figure 4-7 Modify Project Settings to Use J-Link

4.2. Set Printout Mode for printf according to Debugger Type

Using printf function can help developers to confirm the running status of programs efficiently. Because embedded system usually does not equip with display screen, it is necessary to redirect the printf function to the host PC.

SES supports two Printout modes for Nuclei:

- Printout through UART Serial Port from Hummingbird Debugger Kit.
- Printout through RTT functions from native J-link debugger.

The steps to set up Printout modes (according to the Debugger type) are detailed as follows.

4.2.1. Printout through Serial Port

In embedded system, UART port of SoC is commonly used to connect COM port of host PC (or USB port of host PC after UART is converted to USB) for debugging, so that printf function in

embedded system can be redirected and printed to display screen of host PC.

The Hummingbird Debugger Kit has a FT2232 chip inside it to convert the UART to USB, and then connect to host PC, which will be recognized as COM port at host PC (if the driver is installed correctly). About how to connect Hummingbird Debugger Kit, and how to install its driver, please refer to the document <Nuclei_FPGA_DebugKit_Intro.pdf> which can be downloaded from "Development Boards" page of Nuclei website (http://www.nucleisys.com/developboard.php).

The setting steps at SES to grab this COM port are as follows:

- After connecting the Hummingbird Debugger Kit, as shown in Figure 4-8, select "Tools

 > Terminal Emulator --> Terminal Emulator" in the menu bar to open the Serial Port tool.
- As shown in Figure 4-9, select "Tools > Terminal emulator > Properties" in the menu bar to open the Serial Port setting pop-up window.
- The connected COM port can be found (after the driver is installed correctly). As shown in Figure 4-10, set "Baud Rate" to 115200, then double-click "Port" option, select Serial Port number (as recognized by PC), and complete Serial Port selection.
- As shown in Figure 4-11, click the "old type phone" icon to connect the Serial Port.

After the above settings, take demo_eclic as an example, when the program is downloaded to the FPGA evaluation board, if the printf function is used in the running program, you can see the printf output on the SES Serial Port as shown in Figure 4-12.



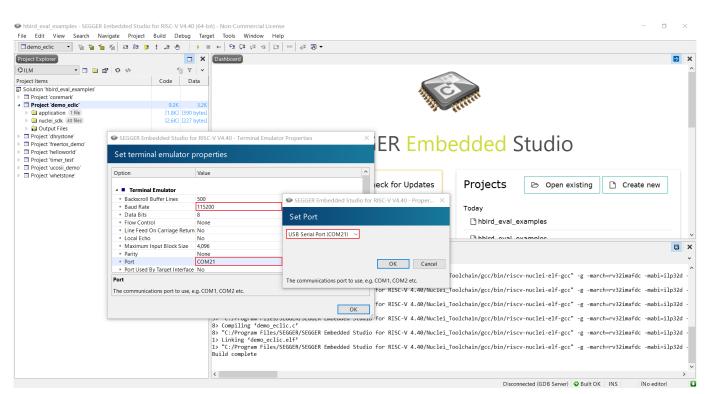
hbird_eval_examples - SEGGER Embedded Studio File Edit View Search Navigate Project	for RISC-V V		Non-Commercial License Tools Window Help	- 🛛 ×
I demo_eclic - 🖓 🍇 🍎 🖓 💵 📭 🍺			Options Alt+,	
Project Explorer			License Manager	X
GILM Project Items	Code	⊽ ✓ Data	Package Manager § Show Installed Packages Manually Install Packages	^
Solution 'hbid_eval_examples' Project 'demo_eclic' Project 'demo_eclic' Project 'demo_eclic' Project 'demo_eclic' Project (demo_eclic' Project (demo_eclic' Project 'demo_eclic' Project 'demo_eclic' Project 'demo_eclic'		3.2K 590 bytes] 227 bytes]	New File Comparison Ctrl+K, F New Binary File Comparison Browser Terminal Emulator Ctrl+Alt+M	ed Studio
 □ Project 'helloworld' □ Project time_test' □ Project cosi demo' □ Project volsi demo' ▷ Project 'whetstone' 			SEGGER Embedded Stud RISC-V version 4.42 ava Baud Rate Parity Data Bits Stop Bits Bronetier	
			Гэры	v v
			Upput Compiling 'sbrk.c' 'C:/Program Files/SEGGER/SEGGER Embedded Studio for RISC-V 4.40/Nuclei_Toolchain/gc Compiling 'write.c' 'C:/Program Files/SEGGER/SEGGER Embedded Studio for RISC-V 4.40/Nuclei_Toolchain/gc Compiling 'hbird_common.c' 'C:/Program Files/SEGGER/SEGGER Embedded Studio for RISC-V 4.40/Nuclei_Toolchain/gc Compiling 'system hbird.c' 'C:/Program Files/SEGGER/SEGGER Embedded Studio for RISC-V 4.40/Nuclei_Toolchain/gc Compiling 'system hbird.c' 'C:/Program Files/SEGGER/SEGGER Embedded Studio for RISC-V 4.40/Nuclei_Toolchain/gc Linking (demo_eclic.clf) 'C:/Program Files/SEGGER/SEGGER Embedded Studio for RISC-V 4.40/Nuclei_Toolchain/gc Linking (demo_eclic.clf)	<pre>c/bin/riscv-nuclei-elf-gcc" -g -march=rv32imafdc -mabi=ilp32d - c/bin/riscv-nuclei-elf-gcc" -g -march=rv32imafdc -mabi=ilp32d - c/bin/riscv-nuclei-elf-gcc" -g -march=rv32imafdc -mabi=ilp32d - c/bin/riscv-nuclei-elf-gcc" -g -march=rv32imafdc -mabi=ilp32d -</pre>
				>
Activates the Terminal Emulator window				Disconnected (GDB Server) 🔮 Built OK INS (No editor)



File Edit View Search Navigate Project	Build Debug Tar	pit) - Non-Commercial License get Tools Window Help	- ¤ ×
🗖 demo_eclic 🔹 🐐 🎦 🎽 🖌 💷 📭	:!, ⊒ ⊕ →	• Options Alt+,	
Project Explorer	🗆 X	License Manager	S X
् ्रीILM • 🗆 🖻 🗗 🗘 🚸	1 V V	Package Manager	
Project Items	Code Data	Show Installed Packages	
Solution 'hbird_eval_examples'	Couc Data	Manually Install Packages	
Project 'coremark'			
Project 'demo eclic'	9.2K 3.2F	% New File Comparison Ctrl+K, F	is write
application 1 file	[1.8K] [590 bytes	New Binary File Comparison	3-14-
Inuclei_sdk 43 files	[2.6K] [227 bytes	Browser	
Output Files		Terminal Emulator	
Project 'dhrystone'		Terminal Emulator	Terminal Emulator Ctrl+Alt+M
Project 'freertos_demo'		Admin •	© Connect COM21 (115200,N,8,1)
Project 'helloworld'			3 Disconnect
Project 'timer_test'			-
Project 'ucosii_demo'			Send Control
Project 'whetstone'			Baud Rate 'Durate atta
		SEGGER Embedded Stud	Parity , Projects
		RISC-V version 4.42 ava	Data Bits •
			Stop Bits ' Today
			Properties
			hbird_eval_examples
			B bbird oval ovamolos
		Output	
		Show: Transcript 🔹 🐾 🍾 Output	•
		6> Compiling 'sbrk.c'	
		<pre>2> Compiling 'write.c'</pre>	reares rearies to write a tradymeter contrary becard, the meter of the P material architecture monthly re
			wedded Studio for RISC-V 4.40/Nuclei_Toolchain/gcc/bin/riscv-nuclei-elf-gcc" -g -march=rv32imafdc -mabi=ilp32d -
		1> Compiling 'hbird_common.c'	
			bedded Studio for RISC-V 4.40/Nuclei_Toolchain/gcc/bin/riscv-nuclei-elf-gcc" -g -march=rv32imafdc -mabi=ilp32d -
		5> Compiling 'system_hbird.c'	bedded Studio for RISC-V 4.40/Nuclei Toolchain/gcc/bin/riscv-nuclei-elf-gcc" -g -march=rv32imafdc -mabi=ilp32d
		8> Compiling 'demo eclic.c'	
			wedded Studio for RISC-V 4.40/Nuclei_Toolchain/gcc/bin/riscv-nuclei-elf-gcc" -g -march=rv32imafdc -mabi=ilp32d -
		1> Linking 'demo_eclic.elf'	
			bedded Studio for RISC-V 4.40/Nuclei_Toolchain/gcc/bin/riscv-nuclei-elf-gcc" -g -march=rv32imafdc -mabi=ilp32d -
		Build complete	
		<	>

Figure 4-9 Open the Serial Port Setting Pop-up Window

A 芯来科技 NUCLEI





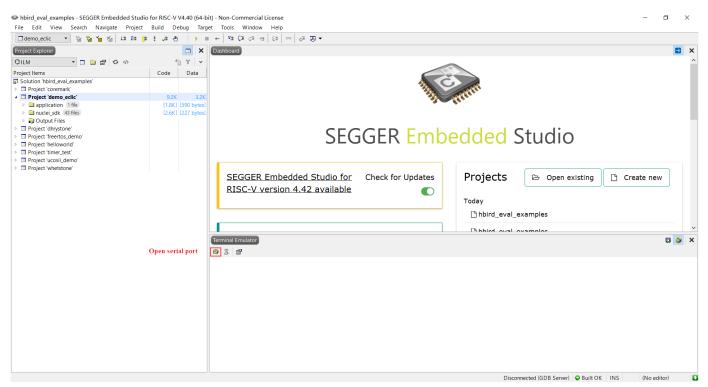


Figure 4-11 Open Serial Port

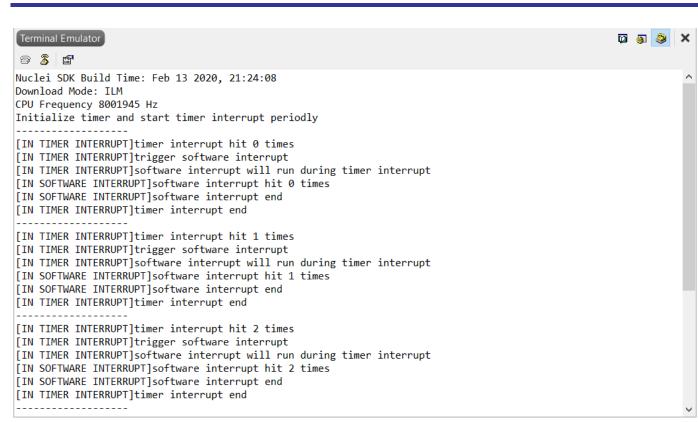


Figure 4-12 The Output of Project demo_eclic through Serial Port

4.2.2. Printout through RTT

If J-Link is used, the RTT (Real Time Terminal) of J-Link (does not need to occupy UART interface of SoC at all) can also be utilized to redirect printf output. The setting steps are as follows:

- Create an "external GNU using project" as shown in Figure 4-13. And when adding a default folder, select the SEGGER folder, as shown in Figure 4-14.
- Drag the SEGGER folder to the current project, right-click the red-box marked file in Figure 4-15, open the right-click menu, select "Excluded From Build" option.
- Recompile the project.

After that, user can redirect the output to RTT, and the debug terminal will be opened automatically at runtime. Take demo-eclic project output as an example, use J-Link RTT to output is as shown in Figure 4-16.

芯来科技 NUCLEI

C - I -		[Connels Ducinate			
Sele	ct new project template		Search Projects			
Don't see your device or board? Use the <u>Package Manager</u> to install packages						
escripti	on	Manufacturer	Board			
	E Tratter					
AC	C/C++ executable for a RISC-V processor executing from FLASH memory.	Generic	Generic RISC-V			
AC	C/C++ executable for a RISC-V processor executing from RAM memory.	Generic	Generic RISC-V			
	externally built executable for a RISC-V processor.	Generic	Generic RISC-V			
	C/C++ executable for a RISC-V processor executing from FLASH memory (internal tools and external G		Generic RISC-V			
	C/C++ executable for a RISC-V processor executing from RAM memory (internal tools and external GN		Generic RISC-V			
	empty solution.	Generic	(Standard Projects)			
	brary project.	Generic	(Standard Projects)			
	object file project.	Generic	(Standard Projects)			
	project for copying files to a target directory.	Generic	(Standard Projects)			
Ар	roject for running a custom build when files have changed.	Generic	(Standard Projects)			
A ame:	NDES Executable 1					
	-					
cation:	C:\Users\24954\Desktop\EmbeddedStdioProjectnuclei_eval\hbird_eval_examples		Brow			
		Back	Next Cance			

Figure 4-13 Create an External GNU Using Project

SEGGER Embedded Studio for RISC-V V4.40 - New Project	×
Generation Select files to add to project	
Files:	
System	
✓ Import all files and package files	
Back Next Can	cel

Figure 4-14 Adding SEGGER Folder

A 芯来科技 NUCLEI

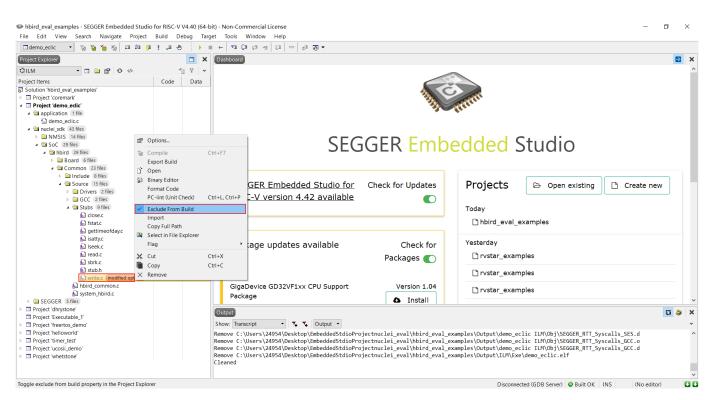


Figure 4-15 Redirect Output through RTT

Debug Terminal	Ø	()	٩	×
Nuclei SDK Build Time: Feb 13 2020, 21:24:08 Download Mode: ILM CPU Frequency 8003911 Hz Initialize timer and start timer interrupt periodly				^
[IN TIMER INTERRUPT]timer interrupt hit 0 times [IN TIMER INTERRUPT]trigger software interrupt [IN TIMER INTERRUPT]software interrupt will run during timer interrupt [IN SOFTWARE INTERRUPT]software interrupt hit 0 times [IN SOFTWARE INTERRUPT]software interrupt end [IN TIMER INTERRUPT]timer interrupt end				
[IN TIMER INTERRUPT]timer interrupt hit 1 times [IN TIMER INTERRUPT]trigger software interrupt [IN TIMER INTERRUPT]software interrupt will run during timer interrupt [IN SOFTWARE INTERRUPT]software interrupt hit 1 times [IN SOFTWARE INTERRUPT]software interrupt end [IN TIMER INTERRUPT]timer interrupt end				
[IN TIMER INTERRUPT]timer interrupt hit 2 times [IN TIMER INTERRUPT]trigger software interrupt [IN TIMER INTERRUPT]software interrupt will run during timer interrupt [IN SOFTWARE INTERRUPT]software interrupt hit 2 times [IN SOFTWARE INTERRUPT]software interrupt end [IN TIMER INTERRUPT]timer interrupt end				l
				\sim

Figure 4-16 The Output of Project demo_eclic through J-Link

4.3. Download Program to Board and Run

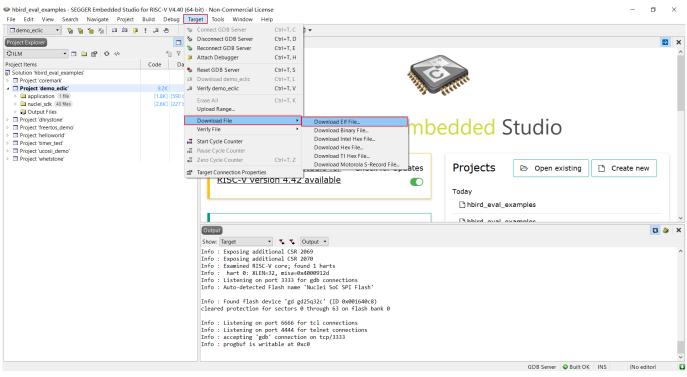
After the above settings are completed, you can download the program to the board and run. The steps are as follows:

- Select the "Connect GDB Server" option under "Target" option in the menu bar of SES, as shown in Figure 4-17.
- Select "Download File" option under "Target" option in the menu bar, then select "Download Elf File", as shown in Figure 4-18, find the Elf file generated by compilation (by default the Elf file is in the "Output Files" folder), and double-click to download it to the board.
 - After downloading the program to the board, you can enter the debugging mode to debug or run it. See Chapter o for more details of how to debug.
 - When the program running, if printf function is used, the output can be redirected to PC. Take demo-eclic as an example, you can see the output as shown in Figure 4-12 (Printout through Serial Port) or Figure 4-16 (Printout through RTT).
- If user don't want to debug or re-download it again, then user can just disconnect GDB Server after downloading, select "Target --> Disconnect GDB Server" in the menu bar, as shown in Figure 4-19.
 - Note: after disconnecting the GDB Server, press the MCU reset button on Hummingbird Evaluation Kit, and the Processor Core will start to execute again from the flash. This is because in the Nuclei Evaluation SoC, the reset address of Procesor Core is the starting address of QSPIo Flash. Please refer to document <Nuclei_Eval_SoC_Intro.pdf> for more information of the Nuclei Evaluation SoC.



		Target Tools Window	Ctrl+T, C			
	!~⊒~0	X. D.	Ctrl+T, D			
Project Explorer		Reconnect	Ctrl+T, E			•
ដុំILM 🔹 🗖 🖻 🗗 🗘	1 V	Attach Debugger	Ctrl+T, H			
Project Items	Code D			1	×	
Solution 'hbird_eval_examples'		🎭 Reset	Ctrl+T, S			
Project 'coremark'		↓3 Download demo_eclic	Ctrl+T, L	13	A CONTRACT OF	
Project 'demo_eclic'	9.2K	Verify demo_eclic	Ctrl+T, V		WILL	
application 1 file	[1.8K] [590		Ctrl+T, K	• -	*	
Inuclei_sdk 43 files	[2.6K] [227	Upload Range				
Gutput Files Groiect 'dhrystone'						
Project drivstone Project 'freertos_demo'		Download File	: SF	CCER Emb	edded Studio	
Project 'helloworld'		Verify File	JL			
Project 'timer_test'		📲 Start Cycle Counter				
Project 'ucosii_demo'		Pause Cycle Counter				
Project 'whetstone'		📓 Zero Cycle Counter	Ctrl+T, Z			
		Target Connection Proper	tier Studio fo	Check for Updates	Projects 🗁 Open existing 🗋 Crea	te new
			ersion 4.42 available		-	
					Tadau	
					Today	
					hbird_eval_examples	
				I	Phird avail avamples	
		Output				🔯 🦃
		Show: Target	- 🍾 🐔 Output -			









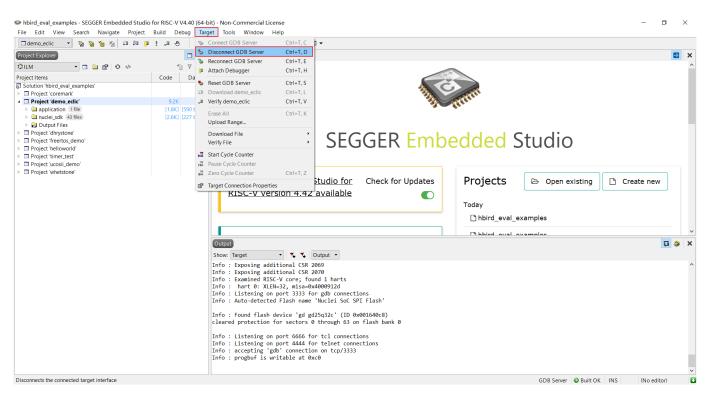


Figure 4-19 Disconnect GDB Server

5. Debug Project

There are many debugging features of SES. Please refer to the following website for detailed usage of debugging:

- SEGGER WIKI: https://wiki.segger.com/Embedded_Studio
- SES manual: https://www.segger.com/downloads/embedded-studio

To enter the debugging mode, press F5 directly or select the "GO" option of "Debug" option in the menu bar. The user interface after successfully entering the debugging mode is as shown in Figure 5-1.

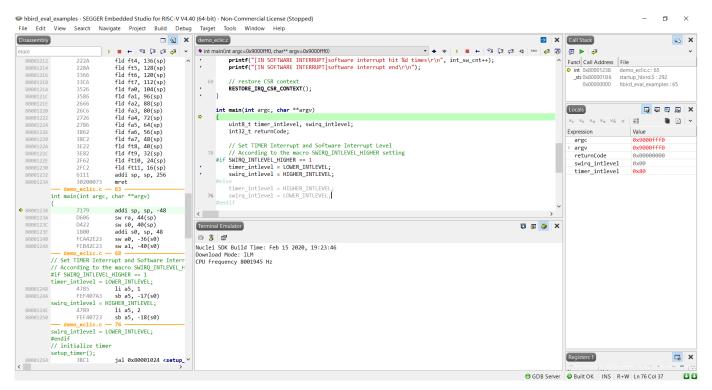


Figure 5-1 Enter Debugging Mode