



Nuclei Studio

IDE 使用说明

Release 2021.02

版权声明

版权所有 © 2018–2021 芯来科技（Nuclei System Technology）有限公司。保留所有权利。

Nuclei™是芯来科技公司拥有的商标。本文件使用的所有其他商标为各持有公司所有。

本文件包含芯来科技公司的机密信息。使用此版权声明为预防作用，并不意味着公布或披露。未经芯来科技公司书面许可，不得以任何形式将本文的全部或部分信息进行复制、传播、转录、存储在检索系统中或翻译成任何语言。

本文文件描述的产品将不断发展和完善；此处的信息由芯来科技提供，但不做任何担保。

本文件仅用于帮助读者使用该产品。对于因采用本文件的任何信息或错误使用产品造成的任何损失或损害，芯来科技概不负责。

联系我们

若您有任何疑问，请通过电子邮件 support@nucleisys.com 联系芯来科技。

修订历史

版本号	修订日期	修订的章节	修订的内容
1.0	2019/10/19	N/A	1. 初始版本
1.1	2020/2/10	N/A	1. 修订了若干文字错误
1.2/ 2020.08	2020/08/06	ALL	<ul style="list-style-type: none"> 1. 全新的 Nuclei Studio IDE 开发工具 2. 采用 Project Wizard 方式全面支持 Nuclei SDK 3. 增加 Nuclei SDK 工程设置工具 4. 当前 Nuclei SDK 集成版本为 v0.2.5 5. 当前 Nuclei 工具链版本为 2020.07 6. Nuclei Studio 2020.08 版本正式发布 7. 集成 IDE Plugins 版本为 1.0.0
1.3/ 2020.09	2020/09/01	1、3、4、 5、6	<ul style="list-style-type: none"> 1. 增加 Linux 环境使用介绍。 2. 增加使用 J-Link 调试运行项目 3. IDE 不再需要单独安装 JDK 环境 4. 增加支持 HummingBird E203 模板工程，集成 Hbird SDK V0.1.0 5. 当前 Nuclei SDK 集成版本为 v0.2.6 6. 当前 Nuclei 工具链版本为 2020.08 7. Nuclei Studio 2020.09 版本正式发布 8. 集成 IDE Plugins 版本为 1.0.2
1.3/ 2020.09	2020/09/04	6.2.2	<ul style="list-style-type: none"> 1. 修订 J-Link 的 Device 选项描述
1.4/ 2021.02	2021/02/02	8、10	<ul style="list-style-type: none"> 1. 新增常见问题章节 2. 新增 QEMU 仿真调试章节 3. 增加支持 gd32vf103c_longan_nano 开发板模板工

版本号	修订日期	修订的章节	修订的内容
			<p>程</p> <ul style="list-style-type: none">4. 当前 OpenOCD 版本更新为 v0.10.0-155. 当前 Eclipse CDT 版本更新为 2020.096. 当前 Nuclei SDK 集成版本为 v0.3.07. 当前 Hbird SDK 集成版本为 v0.1.28. Nuclei Studio 2021.02 版本正式发布9. 集成 IDE Plugins 版本为 1.0.8

目 录

版权声明	0
联系我们	0
修订历史	1
目 录	3
图片清单	5
1. NUCLEI STUDIO IDE 简介与安装	9
1.1. NUCLEI STUDIO IDE 简介	9
1.2. NUCLEI STUDIO IDE 下载	10
1.3. NUCLEI STUDIO IDE 安装	11
2. 启动 NUCLEI STUDIO	12
3. NUCLEI STUDIO 功能介绍	14
4. 创建项目	21
4.1. 通过模板自动创建项目	22
4.2. 从已有项目直接导入创建新项目	32
4.3. 无模板手动创建项目	36
4.3.1. 手动创建项目	36
4.3.2. 配置项目的nuclei_sdk	41
4.3.3. 配置项目的编译和链接选项	44
4.3.4. 配置项目的包含路径和文件	51
4.4. 基于已有的MAKEFILE 创建项目	53
4.4.1. 手动新建项目	53
4.4.2. 设置Makefile 路径和Build 选项	55
5. 编译项目	56
5.1. NUCLEI SDK 工程设置工具	56
5.2. 在 NUCLEI STUDIO 中编译 HELLO WORLD 项目	60
6. 调试运行项目	62
6.1. 使用蜂鸟调试器结合OPENOCD 调试运行项目	62
6.1.1. 安装蜂鸟调试器驱动	62
6.1.1.1. 在 Windows 系统中安装驱动	63

6.1.1.2 在 Linux 系统中安装驱动	64
6.1.2. 新建并配置 <i>Debug Configuration</i>	66
6.1.3. 在原型开发板上调试程序	70
6.1.4. 下载运行程序	75
6.2. 使用 J-LINK 调试运行项目	77
6.2.1. 安装 <i>J-Link</i> 驱动并导入 <i>RTT</i> 文件	77
6.2.2. 新建并配置 <i>Debug Configuration</i>	82
6.2.3. 在原型开发板上调试程序	86
6.2.4. 下载运行程序	90
7. 导入旧版本 NUCLEI STUDIO 创建的工程	92
8. 常见问题	93
8.1. 打开/关闭 LAUNCH BAR	93
8.2. 使用 LAUNCH BAR	94
8.3. 不使用 LAUNCH BAR 进行运行/调试	95
8.4. DEBUG 页面查看寄存器	96
8.5. DEBUG 页面结束进程	96
8.6. NUCLEI STUDIO 工具栏中各按键功能	96
8.7. 显示其他窗口	98
8.8. 恢复默认窗口布局	98
8.9. 对比历史文件	98
8.10. 新建工程时可能出现报错	99
8.11. 新增 INCLUDE 路径出现缓存	99
8.12. 设置页面栏目找不到	99
8.13. 开发板下载速度很慢	100
8.14. LINUX 环境下多用户使用 NUCLEI STUDIO	100
9. NUCLEI STUDIO 升级	101
9.1. GCC/OPENOCD 等工具链的安装	101
9.2. IDE PLUGINS 升级	102
10. 使用 QEMU 仿真调试	104

图片清单

图 1-1 NUCLEI STUDIO IDE 软件包的下载界面	11
图 1-2 NUCLEI STUDIO IDE 压缩包文件内容	12
图 2-1 双击“NUCLEI STUDIO.EXE”启动 NUCLEI STUDIO	13
图 2-2 设置 NUCLEI STUDIO 的 WORKSPACE 目录	13
图 2-3 第一次启动 NUCLEI STUDIO 界面	14
图 3-1 使用菜单栏创建项目	15
图 3-2 在 WELCOME 界面中创建项目	15
图 3-3 选择建立项目类型	16
图 3-4 新建项目及模板选择	17
图 3-5 选择开发板，内核型号和下载模式	18
图 3-6 选择程序模板及相关设置项	19
图 3-7 打开 SDK 设置工具	20
图 3-8 修改工程设置	20
图 3-9 清理工程并重新编译	21
图 4-1 在 WELCOME 界面中创建项目	23
图 4-2 使用菜单栏创建项目	23
图 4-3 选择建立项目模型	24
图 4-4 新建项目及模板选择	25
图 4-5 选择开发板，内核型号和下载模式	26
图 4-6 选择程序模板及相关设置项	27
图 4-7 选择程序模板配置项	27
图 4-8 选择 GCC 工具链路径	28
图 4-9 完成模板创建项目	29
图 4-10 完成项目编译	30
图 4-11 开启调试配置面板	31
图 4-12 自动生成的 OPENOCD 配置	32
图 4-13 位于 GITHUB 上面的项目包	32
图 4-14 选择导入的项目	33
图 4-15 选择导入的方式	34
图 4-16 选择需要导入的项目	34
图 4-17 识别出要导入的项目	35

图 4-18 显示项目的文件结构.....	36
图 4-19 新建 C/C++ PROJECT	37
图 4-20 设置 C PROJECT 项目名和类型	38
图 4-21 设置 HELLO WORLD 项目的基本信息.....	38
图 4-22 设置项目的调试或者发布属性.....	39
图 4-23 创建完成 HELLO WORLD 项目的界面.....	39
图 4-24 打开新建文件夹.....	40
图 4-25 完成新建文件夹.....	40
图 4-26 打开工程设置选项页面.....	42
图 4-27 在弹出窗口进入 HELLO_WORLD 项目的文件夹位置	42
图 4-28 添加 NUCLEI_SDK 文件夹	43
图 4-29 刷新工程.....	43
图 4-30 新建项目目录下的文件夹.....	44
图 4-31 配置 TARGET PROCESSOR 选项	45
图 4-32 配置 OPTIMIZATION 选项	46
图 4-33 配置 DEBUGGING 选项.....	47
图 4-34 配置链接脚本.....	49
图 4-35 配置链接的 GENERAL 选项.....	50
图 4-36 配置链接的 MISCELLANEOUS 选项	51
图 4-37 设置 GNU RISC-V CROSS C ASSEMBLER 的 INCLUDES 栏目包含路径	52
图 4-38 设置 GNU RISC-V CROSS C COMPILER 的 INCLUDES 栏目包含路径	53
图 4-39 新建基于 MAKEFILE 的工程.....	54
图 4-40 设置工程选项.....	54
图 4-41 配置 MAKEFILE 路径	55
图 4-42 配置 BUILD 选项.....	56
图 5-1 打开修改编译选项弹窗 1	58
图 5-2 打开修改编译选项弹窗 2	58
图 5-3 打开修改编译选项弹窗 3	59
图 5-4 一键修改编译选项	59
图 5-5 清理工程并重新编译工程.....	60
图 5-6 对 HELLO_WORLD 项目单击右键选择“CLEAN PROJECT”.....	61
图 5-7 单击锤子图标对 HELLO_WORLD 项目进行编译	61

图 5-8 查看编译后的生成文件.....	62
图 6-1 在设备管理器中查询 COM 串口号	64
图 6-2 开发板连接至虚拟机中.....	66
图 6-3 单击“RUN CONFIGURATION”进行下载.....	67
图 6-4 添加新的 GDB OPENOCD DEBUGGING.....	68
图 6-5 GDB OPENOCD DEBUGGING 栏目下的 HELLO_WORLD 项目 DEBUG	68
图 6-6 配置 HELLO_WORLD_DEMO DEBUG 的参数 1	69
图 6-7 配置 HELLO_WORLD_DEMO DEBUG 的参数 2	69
图 6-8 通过 NUCLEI STUDIO 打开串口调试助手并配置参数	70
图 6-9 设置串口参数.....	71
图 6-10 打开 TERMINAL 终端.....	72
图 6-11 打开串口	72
图 6-12 单击“DEBUG CONFIGURATION”进行下载.....	73
图 6-13 切换至 DEBUG 模式.....	73
图 6-14 下载完成后进入调试界面	74
图 6-15 下载程序到开发板中运行	75
图 6-16 串口调试助手界面输出的 HELLO WORLD	76
图 6-17 下载完成单击红色按钮断开连接	76
图 6-18 打开工程所在目录.....	77
图 6-19 新建“SEGGER”文件夹	78
图 6-20 压缩包内文件内容.....	79
图 6-21 复制后 SEGGER 文件夹内容.....	79
图 6-22 注释不使用的头文件	80
图 6-23 添加 SEGGER 文件夹路径至 INCLUDE 中	81
图 6-24 打开 EXCLUDE FROM BUILD	82
图 6-25 移除 WRITE.C 文件	82
图 6-26 新建 J-LINK DEBUG CONFIGURATION	83
图 6-27 确保 PROJECT 和 C/C++ APPLICATION 与工程一致	84
图 6-28 设置 DEBUGGER 栏目内容	85
图 6-29 设置 STARTUP 栏目内容	86
图 6-30 J-LINK 引脚图.....	87

图 6-31 J-LINK 实物连接图	87
图 6-32 设置 RTT VIEWER	88
图 6-33 切换至下载模式	89
图 6-34 下载完成后进入调试页面	90
图 6-35 下载程序到开发板中运行	91
图 6-36 打印输出 HELLO WORLD	91
图 6-37 下载完成单击红色按钮断开连接	92
图 7-1 修改 NUCLEI STUDIO 工具链	93
图 8-1 关闭 LAUNCH BAR	94
图 8-2 LAUNCH BAR 功能介绍	94
图 8-3 不使用 LAUNCH BAR 首次进行调试	95
图 8-4 DEBUG 模式查看功能窗口	98
图 8-5 对比历史文件	99
图 8-6 显示隐藏栏目	100
图 9-1 IDE 工具链路径	102
图 9-2 打开安装软件弹窗	102
图 9-3 打开已安装界面	103
图 9-4 选择“NUCLEI STUDIO”更新	104
图 10-1 选择 QEMU 进行仿真调试	105
图 10-2 QEMU 仿真调试界面	106

1. Nuclei Studio IDE 简介与安装

1.1. Nuclei Studio IDE 简介

一款高效易用的集成开发环境（Integrated Development Environment, IDE）对于任何 MCU 都显得非常重要，软件开发人员需要借助 IDE 进行实际的项目开发与调试。ARM 的商业 IDE 软件 Keil，在中国大陆很多嵌入式软件工程师均对其非常熟悉。但是商业 IDE 软件（譬如 Keil）存在着授权以及收费的问题，各大 MCU 厂商也会推出自己的免费 IDE 供用户使用，譬如瑞萨的 e2studio 和 NXP 的 LPCXpresso 等，这些 IDE 均是基于开源的 Eclipse 框架，Eclipse 几乎成了开源免费 MCU IDE 的主流选择。Nuclei Studio IDE 正是芯来公司，基于 MCU Eclipse IDE 开发的一款针对芯来公司处理器核产品的集成开发环境工具。

Eclipse 平台采用开放式源代码模式运作，并提供公共许可证（提供免费源代码）以及全球发布权利。Eclipse 本身只是一个框架平台，除了 Eclipse 平台的运行时内核之外，其所有功能均位于不同的插件中。开发人员既可通过 Eclipse 项目的不同插件来扩展平台功能，也可利用其他开发人员提供的插件。一个插件可以插入另一个插件，从而实现最大程度的集成。

Eclipse IDE 平台具备以下几方面的优势。

■ 社区规模大

Eclipse 自 2001 年推出以来，已形成大规模社区，这为设计人员提供了许多资源，包括图书、教程和网站等，以帮助他们利用 Eclipse 平台与工具提高工作效率。Eclipse 平台和相关项目、插件等都能直接从 eclipse.org 网站下载获得。

■ 持续改进

Eclipse 的开放式源代码平台帮助开发人员持续充分发挥大规模资源的优势。Eclipse 在以下多个项目上不断改进。

- 平台项目——侧重于 Eclipse 本身。
- CDT 项目——侧重于 C/C++ 开发工具。

- PDE 项目——侧重于插件开发环境。

- 源码开源

设计人员始终能获得源代码，总能修正工具的错误，它能帮助设计人员节省时间，自主控制开发工作。

- 兼容性

Eclipse 平台采用 Java 语言编写，可在 Windows 与 Linux 等多种开发工作站上使用。开放式源代码工具支持多种语言、多种平台以及多种厂商环境。

- 可扩展性

Eclipse 采用开放式、可扩展架构，它能够与 ClearCase、SlickEdit、Rational Rose 以及其他统一建模语言（UML）套件等第三方扩展协同工作。此外，它还能与各种图形用户接口（GUI）编辑器协同工作，并支持各种插件。

Nuclei Studio 已经充分与 Nuclei SDK 整合，完全满足 SDK 的需要，可以方便快捷地新建模板工程，快速修改工程设置选项。内置 Nuclei SDK 源代码，可根据模板需求自动添加。

1.2. Nuclei Studio IDE 下载

为了方便用户快速上手使用，本文档推荐使用预先整理好的 Nuclei Studio IDE 软件压缩包。芯来公司已经将该软件压缩包上传至公司网站，具体地址为
<https://www.nucleisys.com/download.php>，如图 1-1 所示。

用户可以在芯来科技公司网站的“下载中心”，根据用户开发环境，下载对应 Windows 或 Linux 的 NucleiStudio 压缩包（注意：芯来科技公司网站的下载中心，其内容会不断更新，用户请自行选择使用最新版本或继续使用当前版本）。

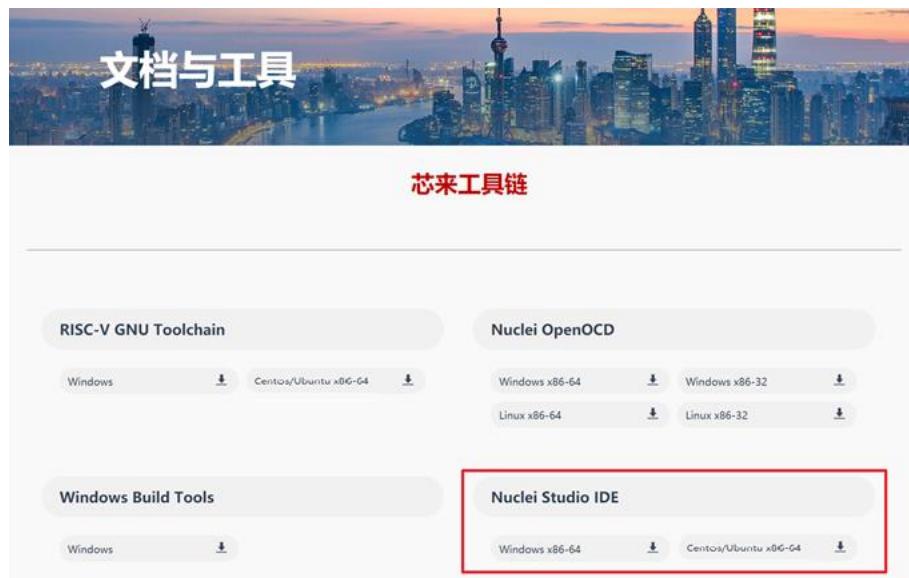


图 1-1 NucleiStudio IDE 软件包的下载界面

1.3. Nuclei Studio IDE 安装

当完成 NucleiStudio IDE 压缩软件包下载，解压后包含若干文件如图 1-2（解压路径中不可包含中文），分别介绍如下。

- NucleiStudio 软件包
 - 该软件包中包含了 Nuclei Studio IDE 的软件。注意：具体版本以及文件名可能会不断更新。
- HBird_Driver.exe（2021.02 版本起不再提供）
 - 仅 **Windows** 版提供，此文件为芯来蜂鸟调试器的 USB 驱动安装文件。
 - 当在 Windows 环境下，使用该调试器时，需要安装此驱动使该 USB 设备能够被系统识别。
 - 由于 2021.02 版本中更新的 openocd 引入了免驱功能。
- SerialDebugging_Tool（2021.02 版本起不再提供）

- 仅 **Windows** 版提供，此文件为“串口调试助手”软件。此软件可以用于后续软件示例调试时通过串口打印信息。

Data (D:) > NucleiStudio_IDE_202102-win64 >

名称	修改日期	类型	大小
NucleiStudio	2021/2/2 10:44	文件夹	

图 1-2 NucleiStudio IDE 压缩包文件内容

2. 启动 Nuclei Studio

启动 Nuclei Studio 的要点如下：

- 直接双击 NucleiStudio IDE 文件包中 NucleiStudio 文件夹下面的可执行文件，即可启动 Nuclei Studio，如图 2-1 所示。
- 第一次启动 Nuclei Studio 后，将会弹出对话框要求设置 Workspace 目录路径，该目录将用于存放后续创建的项目工程文件，如图 2-2 所示。
- 设置好 Workspace 目录之后，单击“Launch”按钮，将会启动 Nuclei Studio。第一次启动后的 Nuclei Studio 界面如图 2-3 所示。
- **2021.02** 版本 Nuclei Studio 默认关闭了 **Launch Bar**，请参照 **8.1** 开启 Nuclei Studio 中的 **Launch Bar** 功能，方便快速编译调试和下载。

Data (D:) > NucleiStudio_IDE_202102-win64 > NucleiStudio

名称	修改日期	类型	大小
configuration	2021/2/2 10:44	文件夹	
features	2021/2/2 10:44	文件夹	
jre	2021/2/2 10:44	文件夹	
p2	2021/2/2 10:44	文件夹	
plugins	2021/2/2 10:44	文件夹	
readme	2021/2/2 10:44	文件夹	
toolchain	2021/2/2 10:44	文件夹	
tools	2021/2/2 10:44	文件夹	
artifacts.xml	2021/2/2 10:35	XML 文件	213 KB
eclipsec.exe	2021/2/2 10:32	应用程序	127 KB
notice.html	2021/2/2 10:32	QQBrowser HTML ...	10 KB
Nuclei Studio User Guide.pdf	2021/2/2 10:37	Adobe Acrobat Do...	8,129 KB
NucleiStudio.exe	2021/2/2 10:32	应用程序	408 KB
NucleiStudio.ini	2021/2/2 10:32	配置设置	1 KB
Ver.2021-02.txt	2021/2/2 10:37	文本文档	0 KB

图 2-1 双击“NucleiStudio.exe”启动 Nuclei Studio

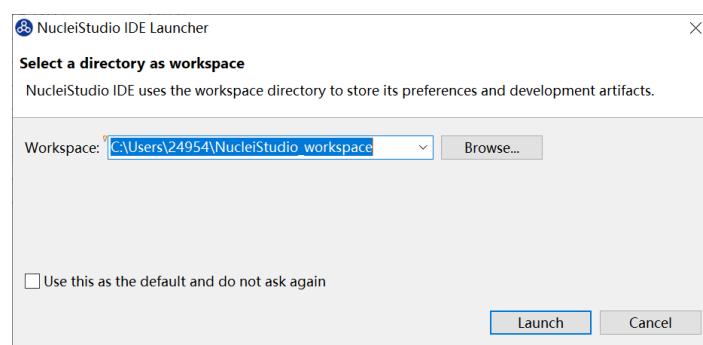


图 2-2 设置 Nuclei Studio 的 Workspace 目录

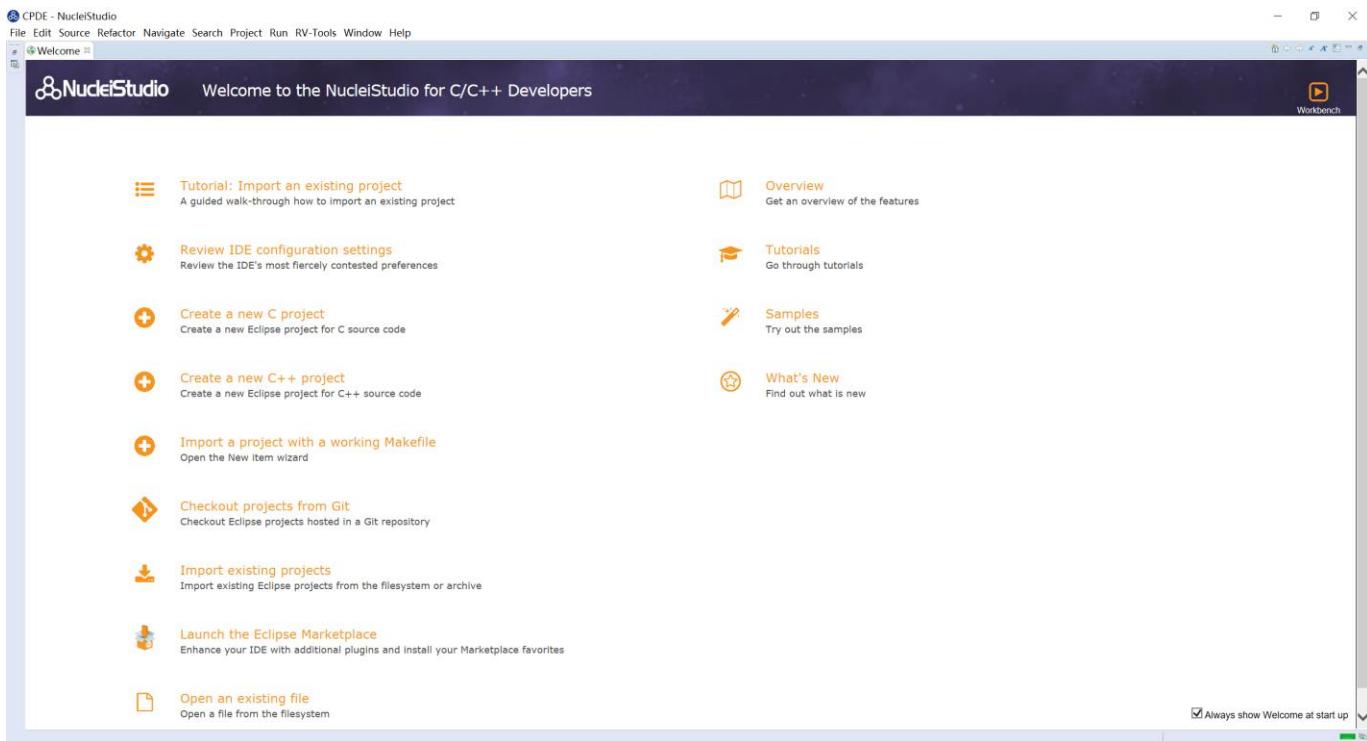


图 2-3 第一次启动 Nuclei Studio 界面

3. Nuclei Studio 功能介绍

Nuclei Studio 已经充分整合 Nuclei SDK，所以可根据不同的开发板快速新建不同的模板工程，并根据不同的模板添加需要的 SDK 源码，根据选项生成不同的编译链接选项设置。本章将在 RVSTAR 开发板上，以新建和修改 GD32VF103 的工程为例快速介绍 Nuclei Studio 功能，详细的流程请参考之后的章节。

- 新建一个工程，可以在菜单栏中，选择“File --> New --> C/C++ Project”，如图 3-1。也可以在“Welcome”界面中，选择“Create a new C project”，如图 3-2。如果不希望每次启动时都显示“Welcome”界面，可以取消页面右下角“Always show Welcome at start up”的选择。

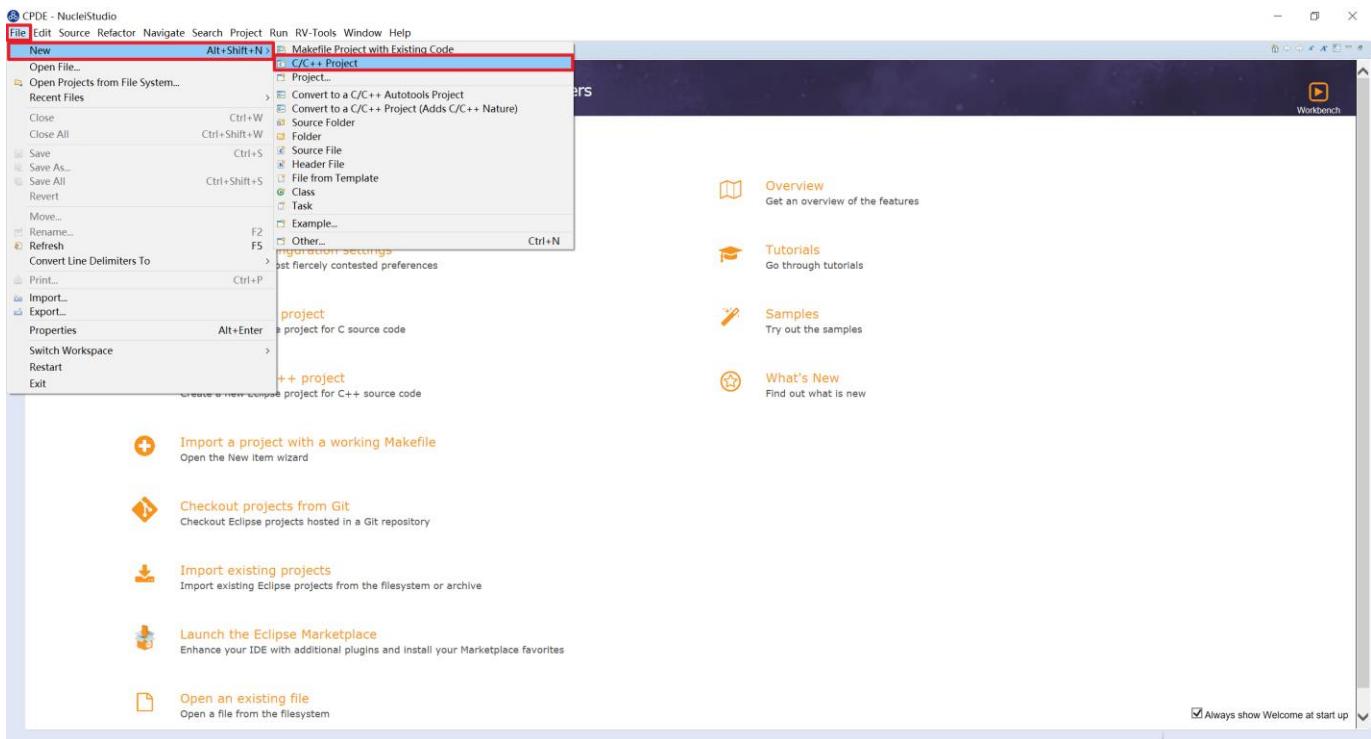


图 3-1 使用菜单栏创建项目

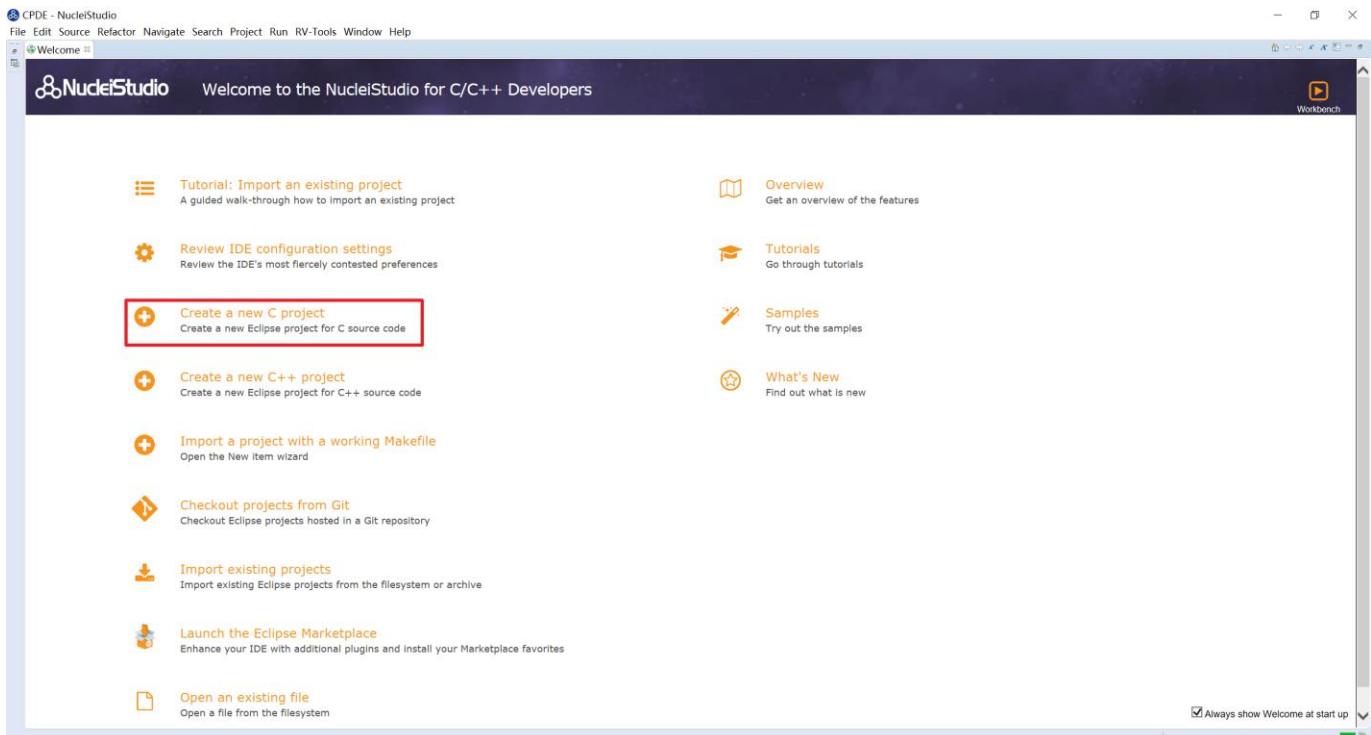


图 3-2 在 Welcome 界面中创建项目

- 如图 3-3 所示，在弹出的窗口中选择项目类型。

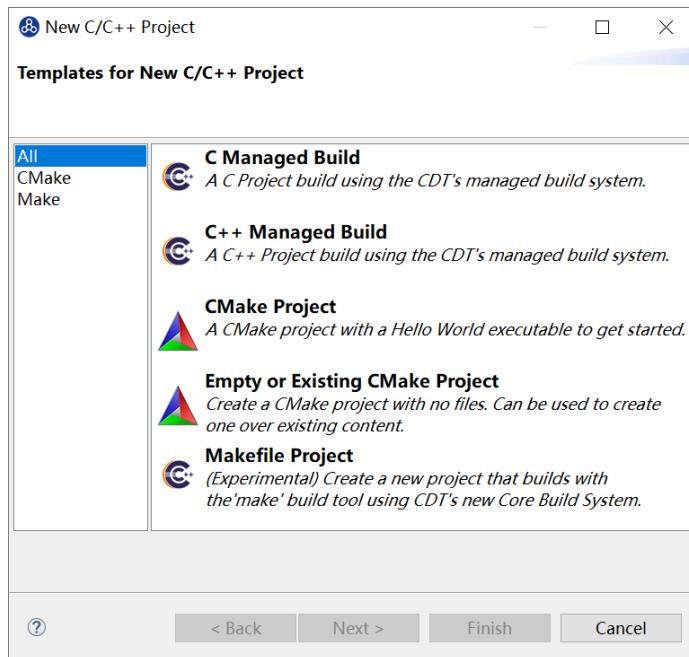


图 3-3 选择建立项目类型

- 如图 3-4 所示，可以根据使用的不同 SoC 选择不同的工程。Nuclei SDK 目前包含了两种 SoC，分别是 HummingBird SoC 和 GD32VF103 SoC。这里以 GD32VF103 SoC 为例，选择“Nuclei SDK Project For GD32VF103 SoC”和“RISC-V Cross GCC”

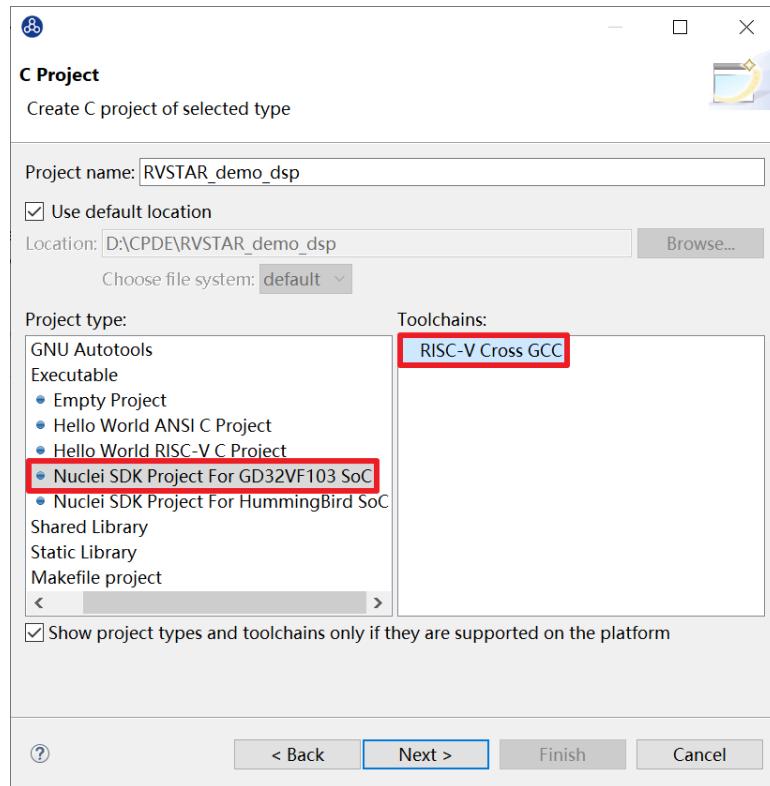


图 3-4 新建项目及模板选择

- 如图 3-5 所示，选择需要创建项目开发板类型。这里以 RVSTAR 开发板为例，所以这一项选择 gd32vf103_rvstar。因为 RVSTAR 的内核是固定的 N205，其对应的 arch 和 abi 分别是 rv32imac 和 ilp32，所以 Core 选项不能修改。同样，RVSTAR 开发板仅支持一种 FLASHXIP 下载模式，所以 DOWNLOAD 这一选项也不能修改。点击“Next”进入下一步。

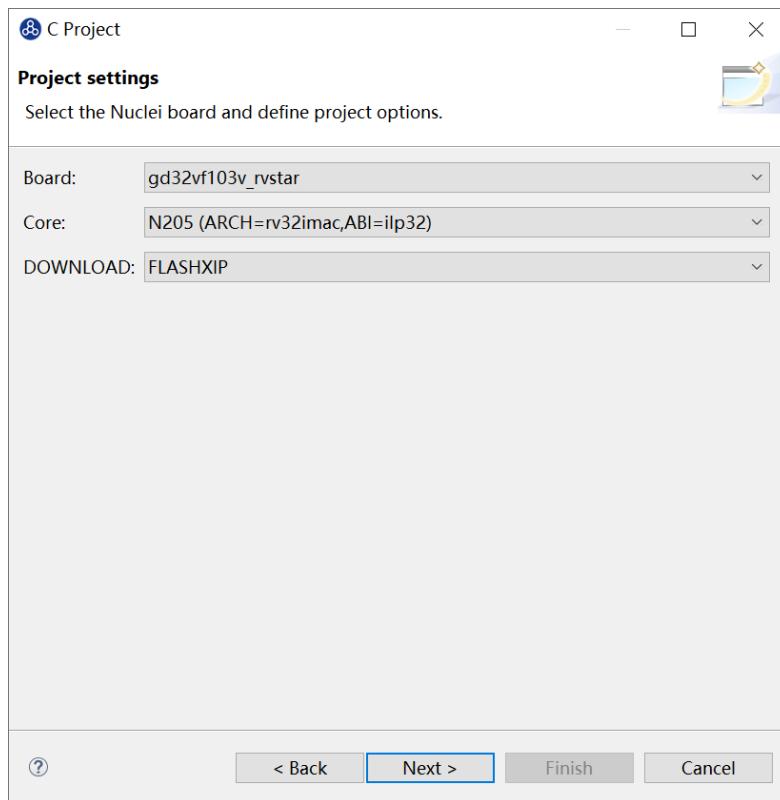


图 3-5 选择开发板，内核型号和下载模式

■ 如图 3-6 所示，选择程序模板及相关设置项。

- Nuclei Studio 可以根据不同的工程模板添加不同的 SDK 源码，例如 FreeRTOS 模板工程会添加对应的 OS 内容，Demo_dsp 模板工程可以添加 NMSIS 库文件。关于 NMSIS 详细信息请参考 (<https://doc.nucleisys.com/nmsis/index.html>)。这里以 Demo_dsp 为例，“Project Example” 选择 “baremetal_demo_dsp” 。因为使用 dsp 工程，需要添加 NMSIS 库，所以 “Libraries” 选择 “NMSIS DSP” 。
- Nuclei Studio 可以根据新建工程时的选项自动设置工程的选项。这里选择不使用浮点打印，所以 “NEWLIB” 选择 “newlib-nano without float” 。编译优化等级 “Optimization Level” 选择 “Optimize more (-O2) ” 。之后一直选择 “Next” 直到 “Finish” 。

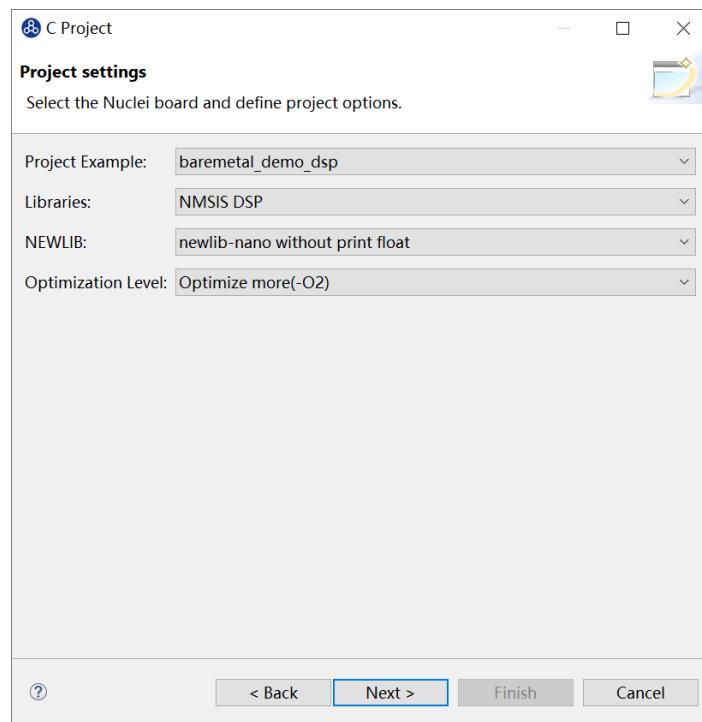


图 3-6 选择程序模板及相关设置项

- Nuclei Studio 可以快速修改工程的设置选项。新建好的工程如图 3-7 所示，单击要修改的工程名，右击打开右键菜单，选择“SDK Configuration Tools”打开设置选项工具。如图 3-8，如果要修改编译优化等级，修改“Optimization Level”为“None (-O0)”，点击“Save”修改选项。如图 3-9，修改成功后在修改后的工程处右击打开右键菜单，选择“clean”清除一下工程，再点击锤子图标编译工程。

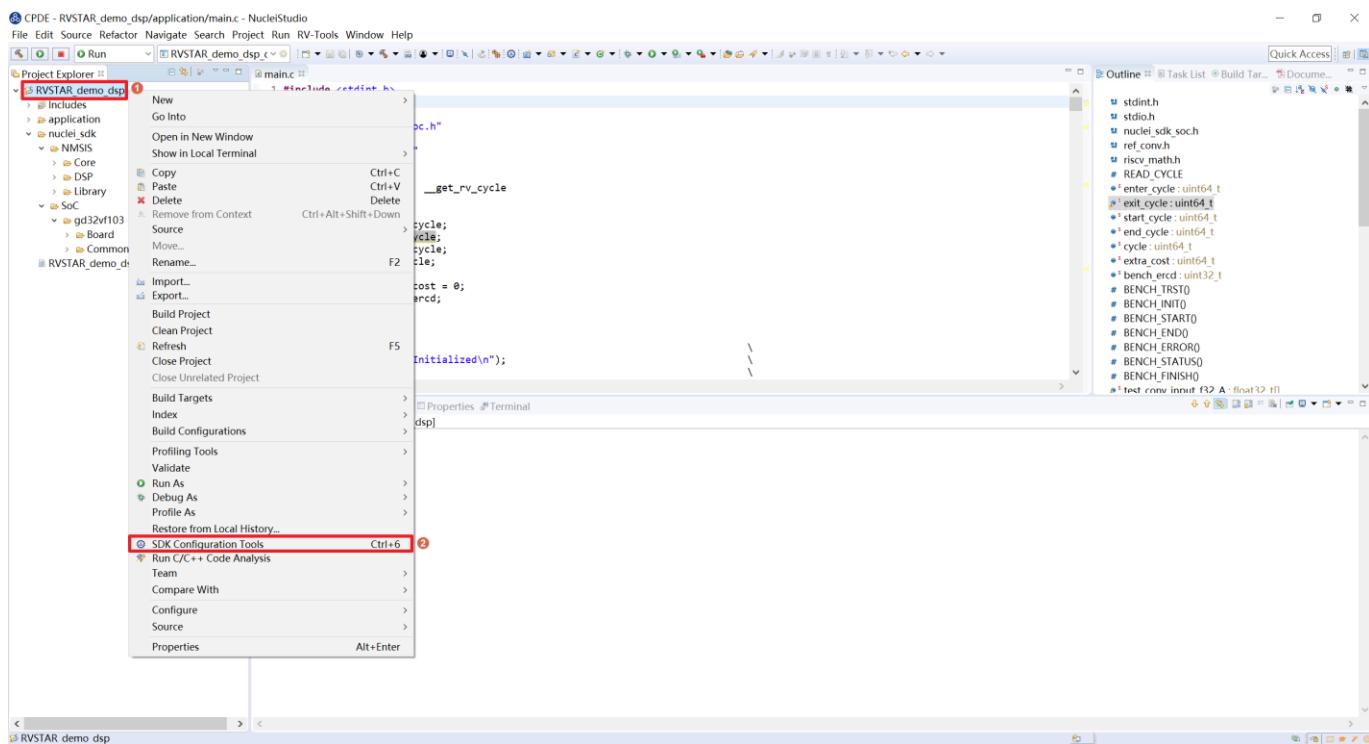


图 3-7 打开 SDK 设置工具

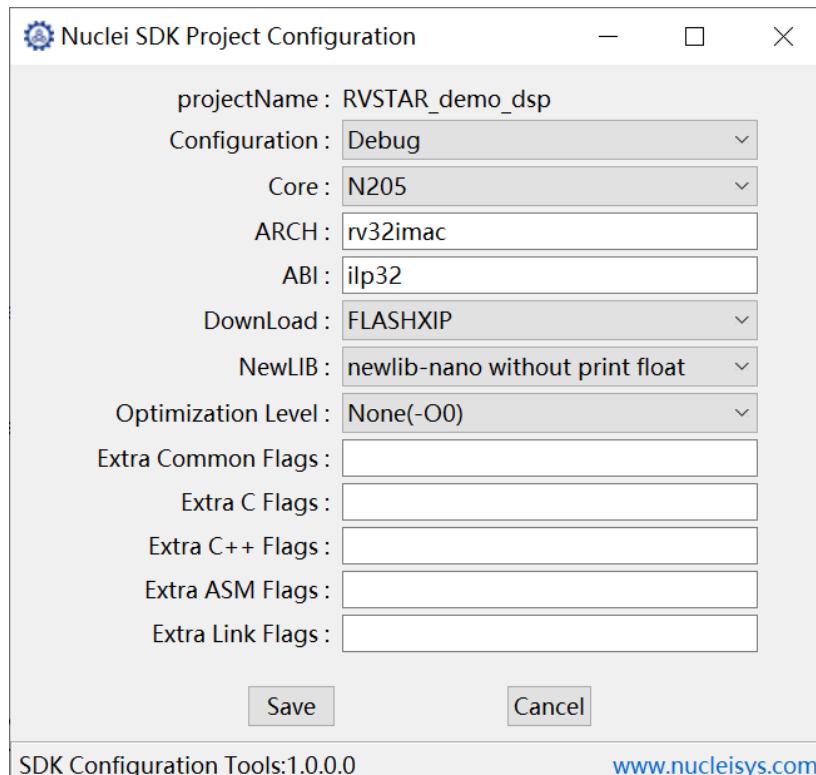


图 3-8 修改工程设置

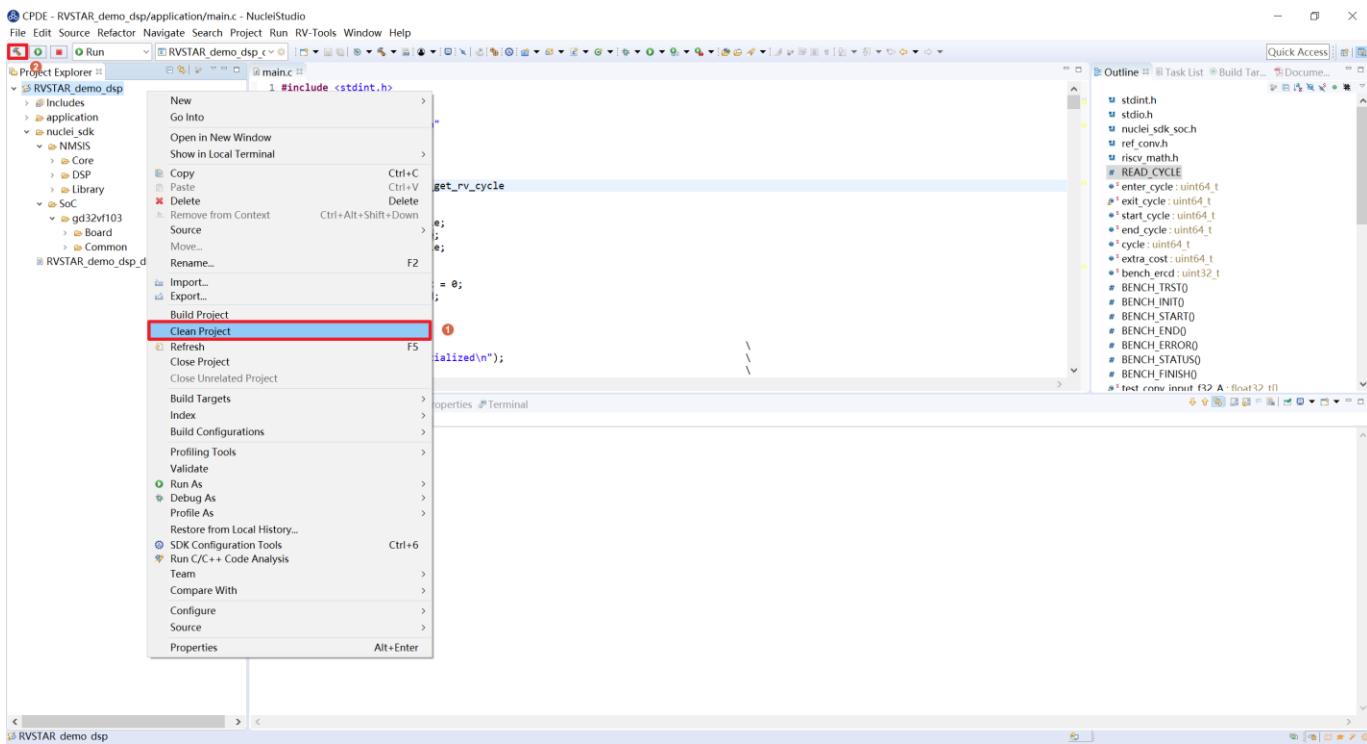


图 3-9 清理工程并重新编译

4. 创建项目

这里以开发板为 HummingBird Evaluation Board，评估处理器内核为 N307(rv32imafc)为例，详细介绍 Nuclei Studio 中创建项目的常见方式。

在 Nuclei Studio IDE 创建项目可以有以下几种常见方式：

■ 使用模板自动创建项目：

- 这是最简单快捷的方式，但前提是 IDE 里面已经预先做了模板项目，Nuclei Studio 已经充分整合 Nuclei SDK。如需对特定芯片产品构建项目模板，请与芯来科技联系。

■ 从已有项目直接导入创建新项目：

- 这是最常见的方法，譬如，用户 A 可以将已有项目的文件夹直接进行打包保存，然后进行分享传播，用户 B 可以在另外的电脑上直接导入该项目，从而以此为基础创建新

的项目，在此基础上直接使用或者开发修改。

■ 无模板手动创建项目：

- 这是最繁琐的方式，该方法除了创建项目之外，还需要手动设置各种选项和路径。由于该方式比较繁琐，所以在实际工作中较少使用，但是通过该方式的详细讲解，用户可以详细了解如何配置各中选项和路径。

■ 基于已有的 Makefile 创建项目：

- 这种方式比较适合于已经采用 Makefile 或者其他编译工具的项目，提供一种在 IDE 中编译工程，清理工程，调试工程的方式。在不修改编译系统的基础上，提供良好的 IDE 调试环境。

下文将对这几种方式分别进行介绍。

4.1. 通过模板自动创建项目

本节将介绍如何使用模板自动创建项目的方式，在 Nuclei Studio IDE 创建一个简单的 Hello World 项目，详细步骤如下。

■ 在“Welcome”界面中，选择“Create a new C project”。如图 4-1 所示。如果客户不希望每次启动时都显示“Welcome”界面，可以取消页面右下角“Always show Welcome at start up”的选择。

注意：也可以在菜单栏中，选择“File—> New —> C/C++ Project”。如图 4-2 所示。

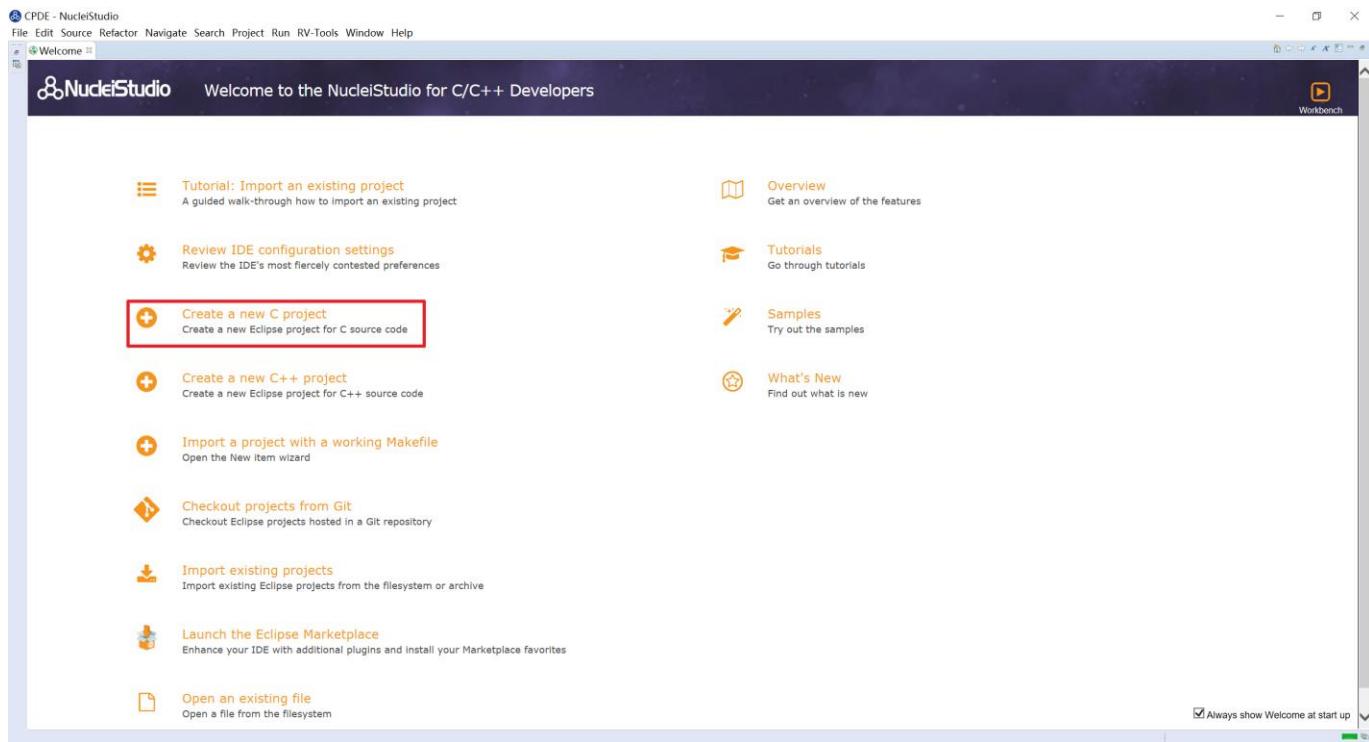


图 4-1 在 Welcome 界面中创建项目

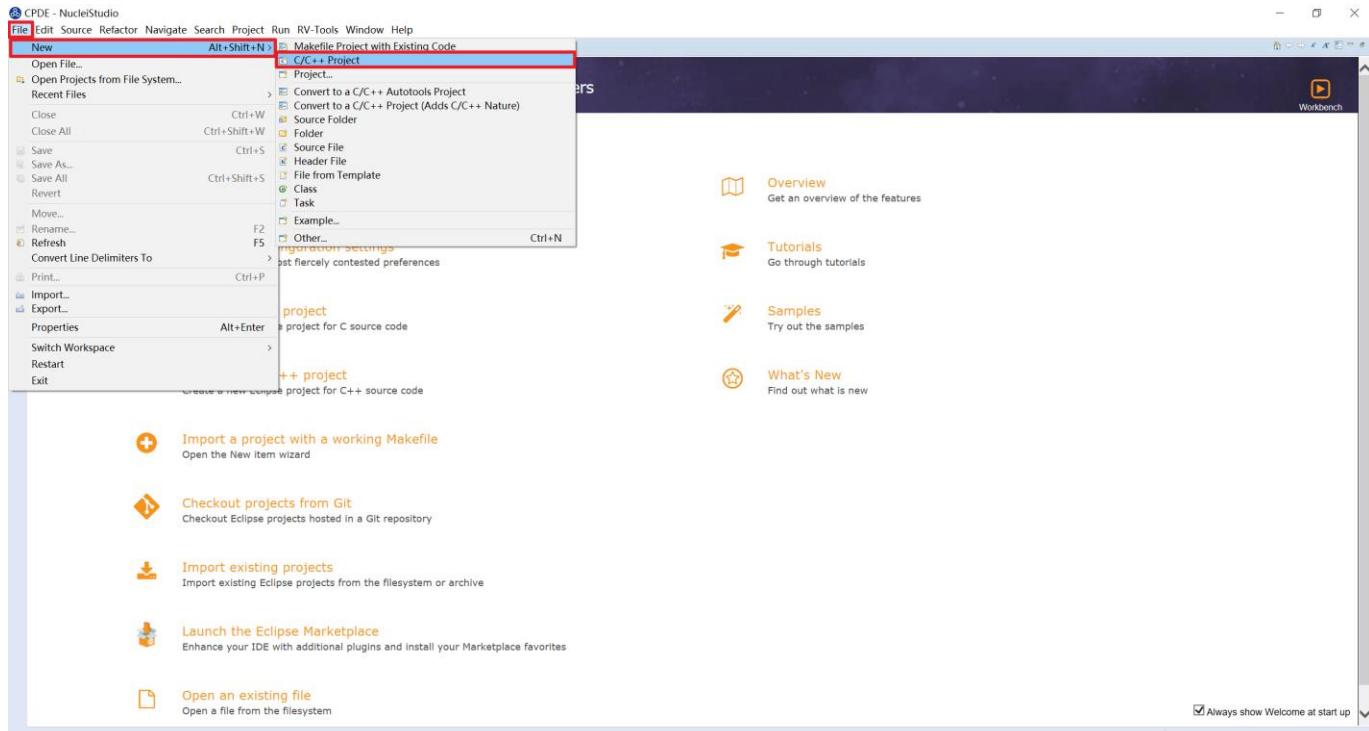


图 4-2 使用菜单栏创建项目

- 如图 4-3 所示，在弹出的窗口中选择项目类型。

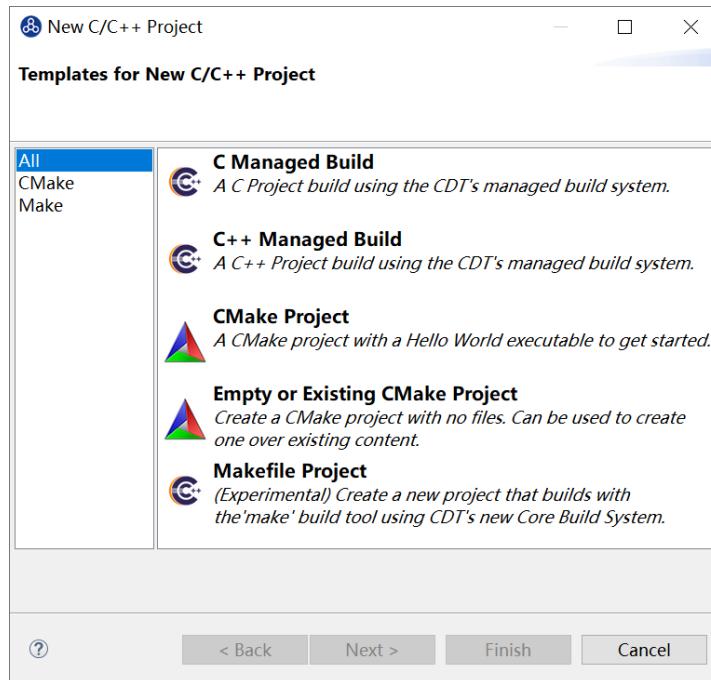


图 4-3 选择建立项目模型

- 如图 4-4 所示，在弹出的窗口中设定如下参数。

- **Project name:** 项目命名。这里设置为“1_helloworld”
- **Use default location:** 如果勾选了此选项，则会使用默认 Workspace 文件夹存放此项目文件。
- **Project type:** 根据使用的 SoC 来选择，这里我们 HummingBird Evaluation Board 使用的是 HummingBird SoC，所以选择“Nuclei SDK Project For HummingBird SoC”。
- **Toolchains:** 我们使用 GCC 工具链，所以这里选择“RISC-V Cross GCC”。
- 点击“Next”进入下一步。

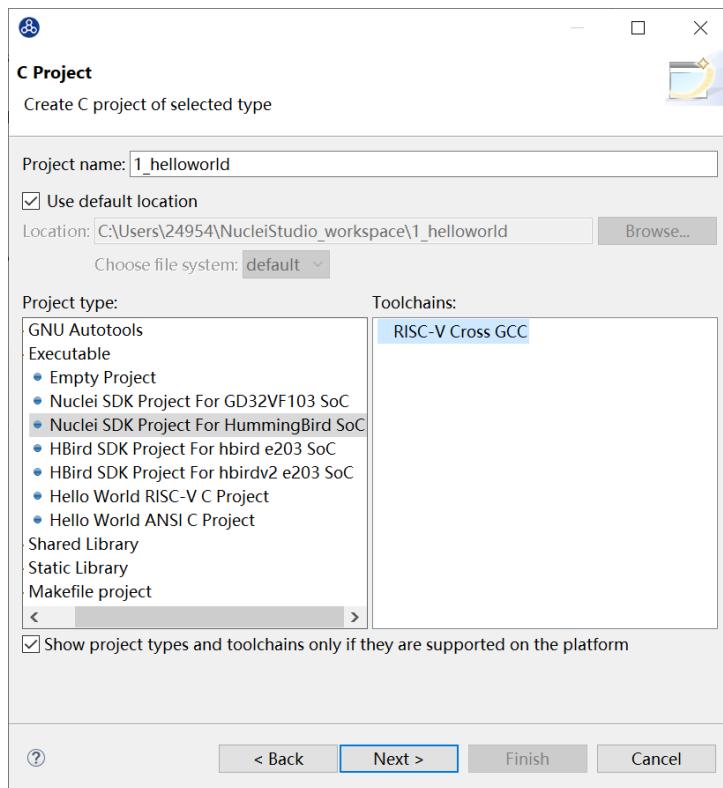


图 4-4 新建项目及模板选择

- 如图 4-5 所示，选择需要创建项目评估板类型。
 - 我们使用的板子是 HummingBird Evaluation Board，所以“Board”这里选择“hbird_eval”。
 - 我们的内核是 N307，所以“Core”选择“N307”。
 - 蜂鸟开发板支持三种下载模式，以下为每种下载模式的简介，这里我们选择 ILM 模式。
 - ILM 下载模式程序将被直接下载在 MCU 的 ILM 中，并从 ILM 开始执行。ILM 由 SRAM 组成，会掉电丢失。
 - FLASH 下载模式程序代码段的物理地址约束 Flash 区间，将代码段的逻辑地址约束在 ILM 的地址区间，意味着程序将被直接下载在 MCU 的 Flash 中，但是上电后要通过引导程序将代码段搬运到 ILM 中，然后从 ILM 中开始执行。程序被烧

写在 Flash 中，不会掉电丢失。

- FLASHXIP 下载模式程序代码段约束 Flash 区间，意味着程序将被直接下载在 MCU 的 Flash 中，并直接从 Flash 开始执行。程序被烧写在 Flash 中，不会掉电丢失。

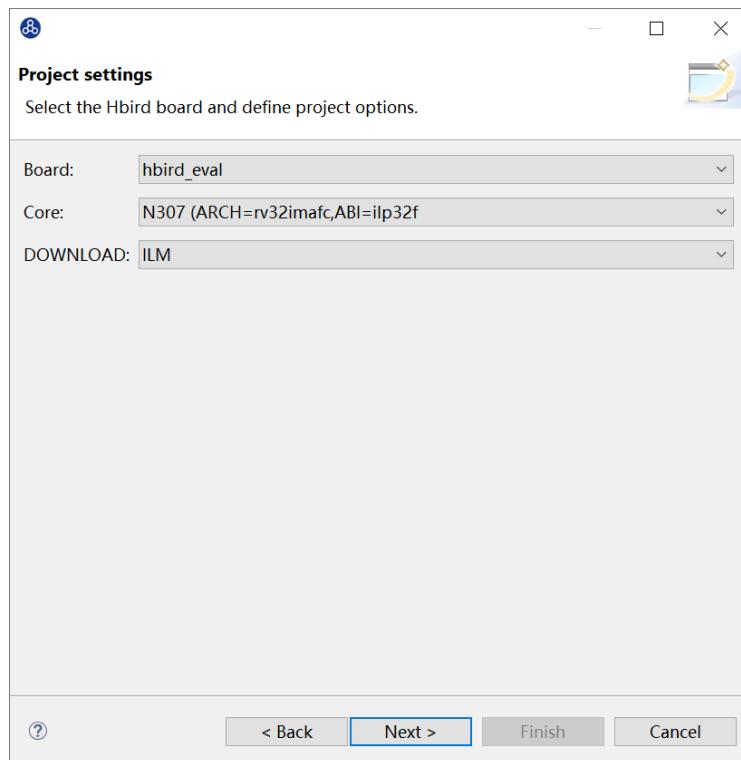


图 4-5 选择开发板，内核型号和下载模式

- 如图 4-6 所示，选择程序模板及相关设置项，这里以新建 helloworld 工程为例。选择工程模板，“project example”选择“baremetal_helloworld”；选择链接库，这里 helloworld 不使用 NMSIS 库，所以选择 None；选择是否使用 DSP 库的 dsp 加速模式，这里 helloworld 不使用 DSP 库，所以不勾选；选择使用 NEWLIB 打印，因为 helloworld 不用浮点打印，所以这里选择“newlib-nano without float”；选择编译优化等级，这里选 Optimize more (-O2)，某些 demo 因为其自身要求，新建时会自动设置优化等级，覆盖这里的选项。然后点击“Next”进入下一步。

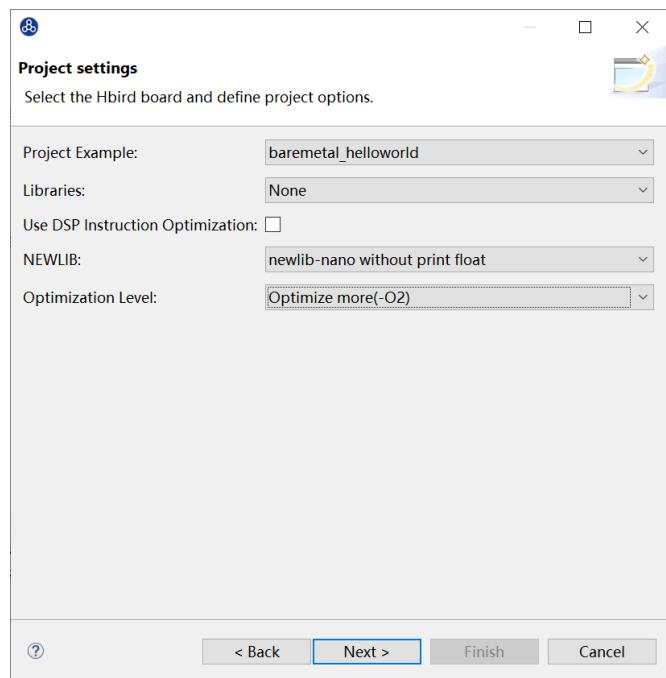


图 4-6 选择程序模板及相关设置项

■ 如图 4-7 所示，选择程序模板配置项。然后点击“Next”进入下一步。

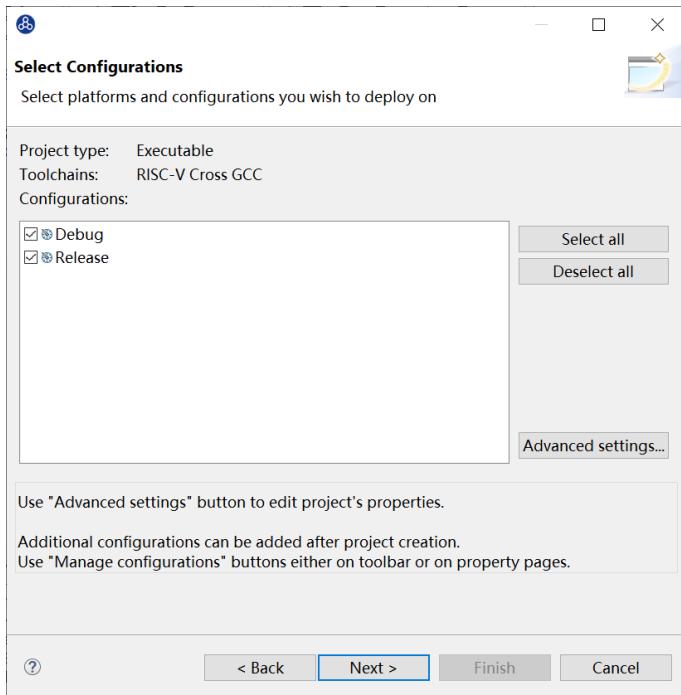


图 4-7 选择程序模板配置项

- 如图 4-8 所示，不需要对“Toolchain path”进行设置，直接点击“Finish”完成项目创建。

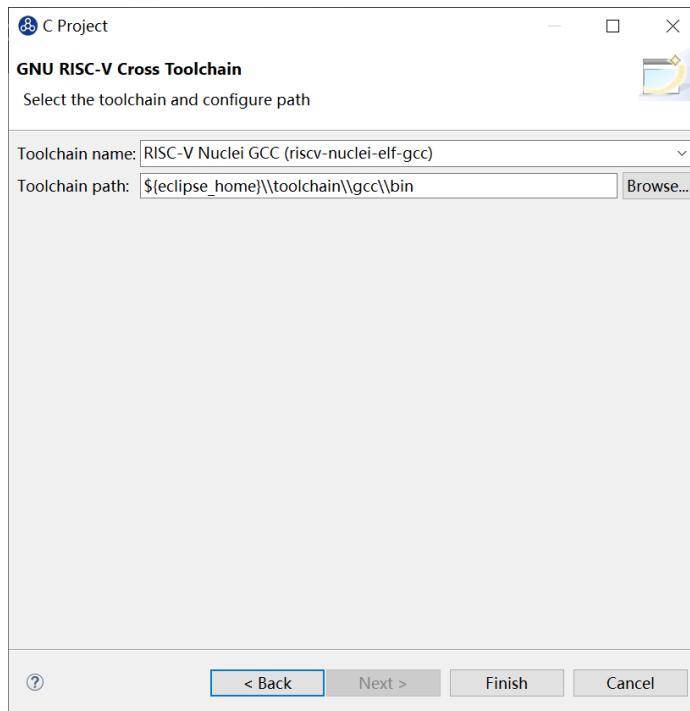


图 4-8 选择 GCC 工具链路径

- 如图 4-9 所示，使用模板自动创建 Hello World 项目已经完成。

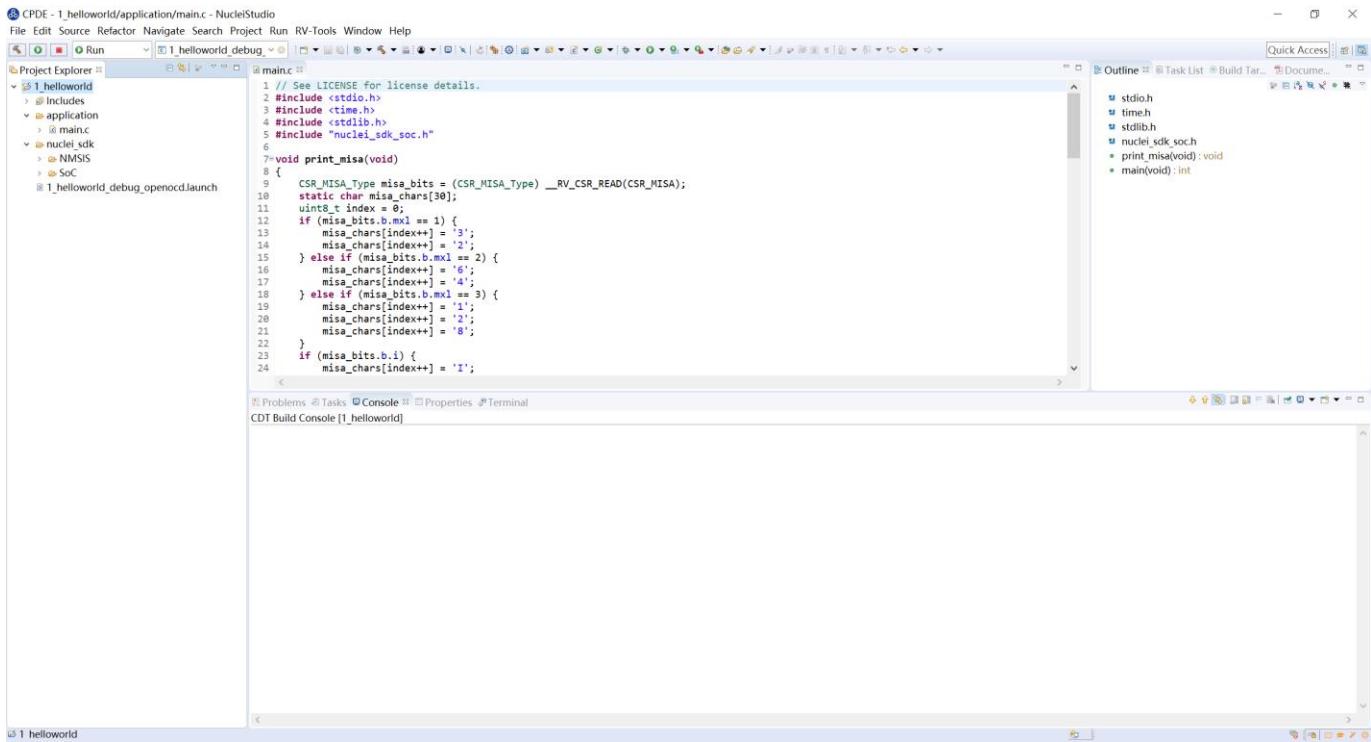
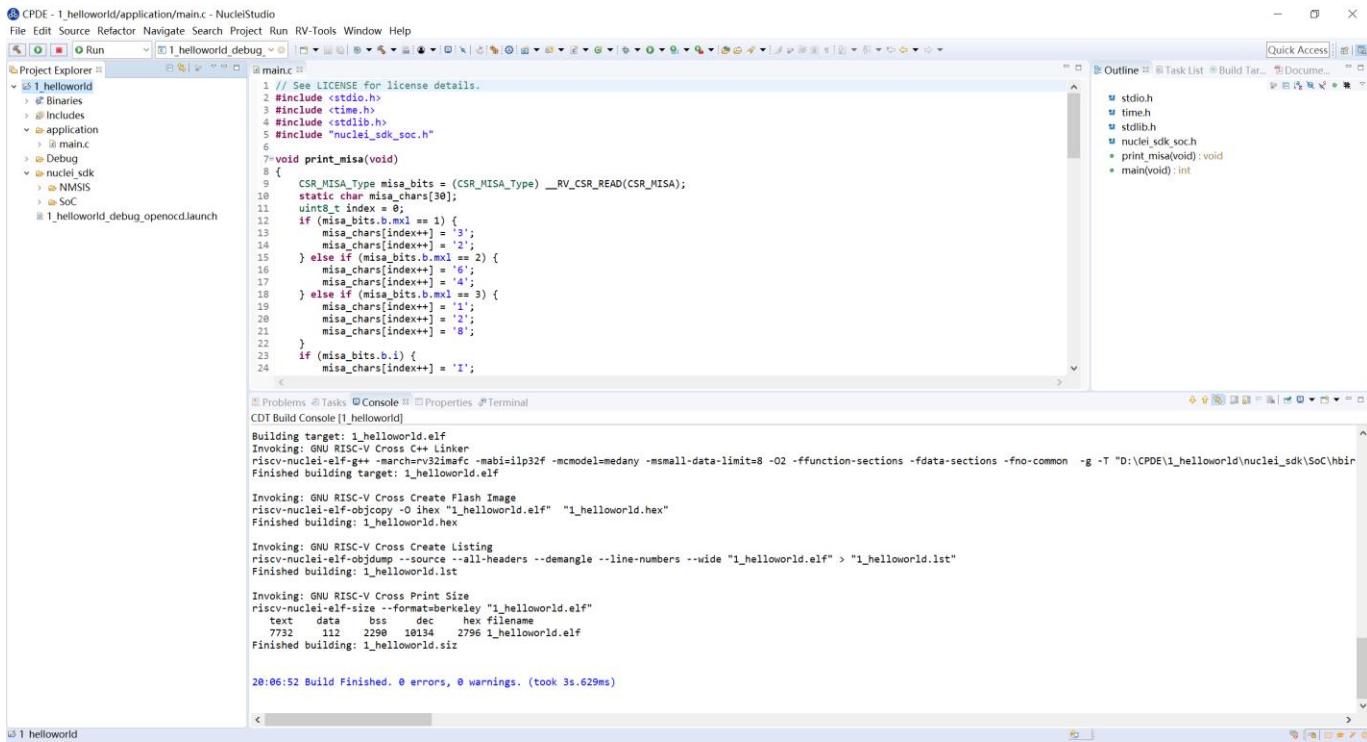


图 4-9 完成模板创建项目

- 用户可以直接使用菜单栏“Project—> Build Project”或“”按钮，来对该项目进行编译，编译后如图 4-10。



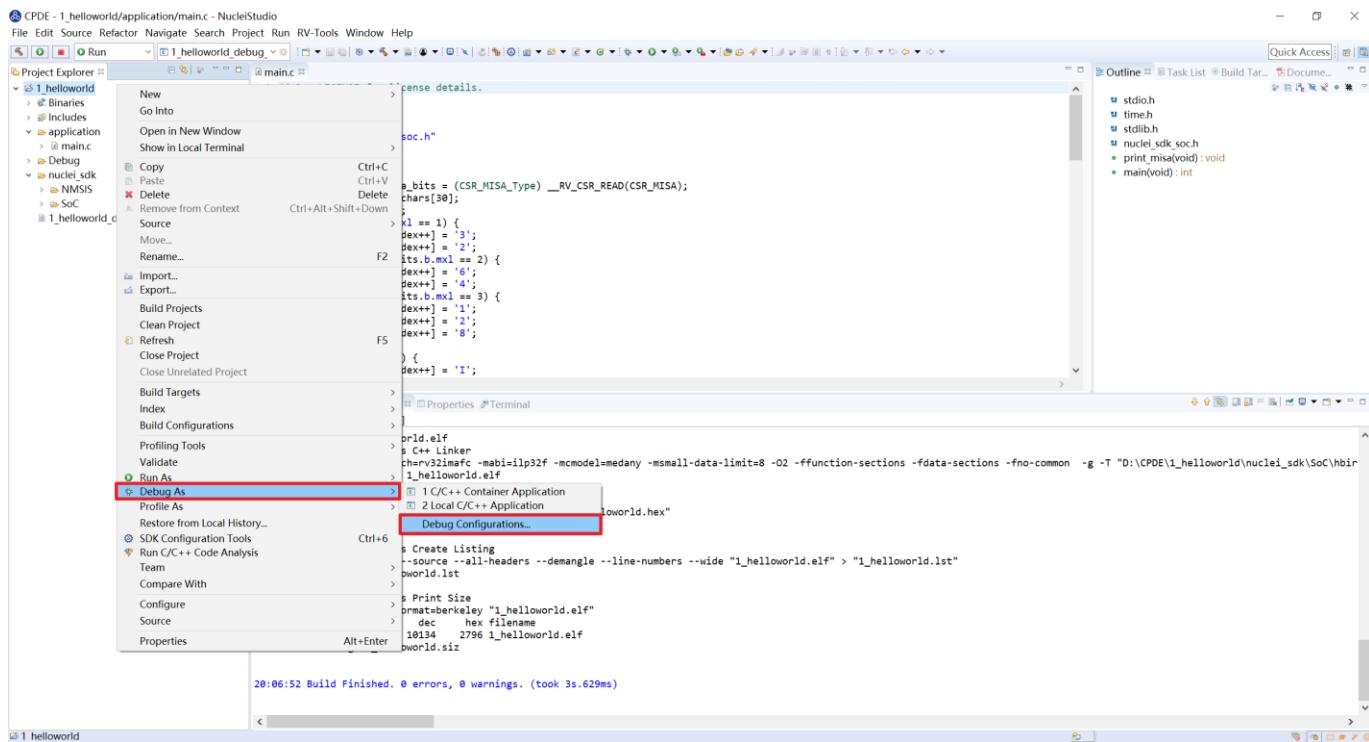


图 4-11 开启调试配置面板

- 如图 4-12 所示，用于调试使用的配置文件“Hello World_Debug_OpenOCD”已经自动生成。关于使用芯来蜂鸟调试器结合 OpenOCD 进行下载和调试的方法，可以查看第 6 章进行详细了解。

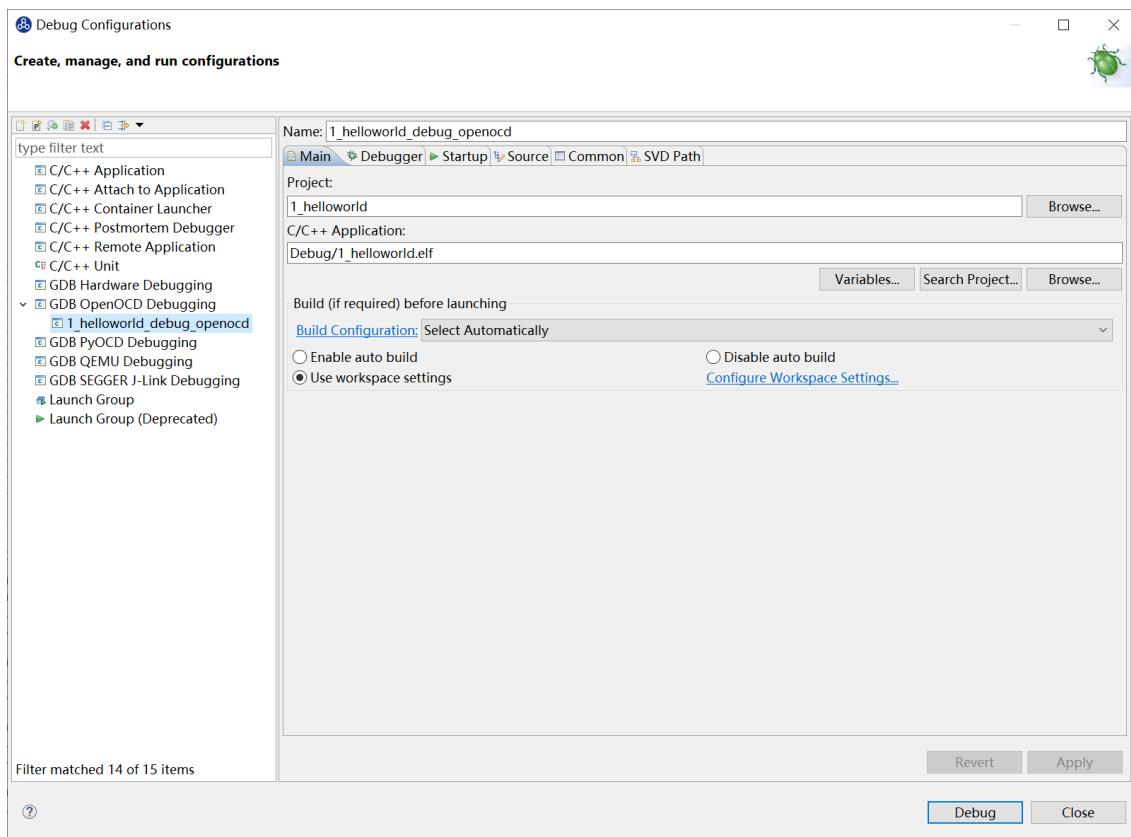


图 4-12 自动生成的 OpenOCD 配置

4.2. 从已有项目直接导入创建新项目

本节将介绍如何使用 IDE 从已有项目直接导入创建新项目，本文以 N307 的项目包为例进行导入，项目包存放在 (https://github.com/riscv-mcu/Nuclei-Studio_IDE-Project-Package)。如图 4-13 所示，如需其它项目包请与芯来科技联系。

在基于 Windows 的 Nuclei Studio IDE 开发环境中，如果用户使用“无模板手动创建工程”，也需要加载此项目包中的 nuclei-sdk 文件夹，相关内容会在下一节中具体介绍。

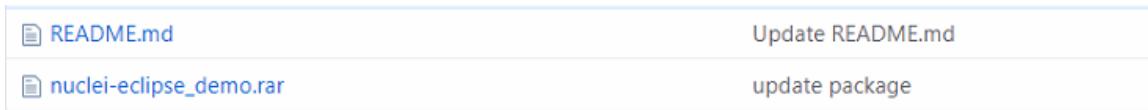


图 4-13 位于 github 上面的项目包

将 nuclei-eclipse_demo.rar 压缩包下载解压后，内容如图 4-14 所示，分别为：

名称	修改日期	类型
.settings	2020/8/5 17:46	文件夹
application	2020/8/5 17:46	文件夹
nuclei_sdk	2020/8/5 17:46	文件夹
.cproject	2020/8/5 17:46	CPROJECT 文件
.project	2020/8/5 17:46	PROJECT 文件
hello_world_debug_openocd.launch	2020/8/5 18:11	LAUNCH 文件

图 4-14 选择导入的项目

- 项目包的描述文件.setting, .project 和.cproject
- 项目包的 Debug 设置文件*.launch
- nuclei_sdk 文件夹

该文件夹下存放部分 SDK 源代码。

- application 文件夹

此文件夹包含 hello_world 样例程序的 main 函数源代码。

下一步导入下载好的项目包，导入步骤如下：

- 在菜单栏中选择“File—>import”。
- 如图 4-15 所示，选择“Existing Project into WorkSpace”后，点击“Next”。

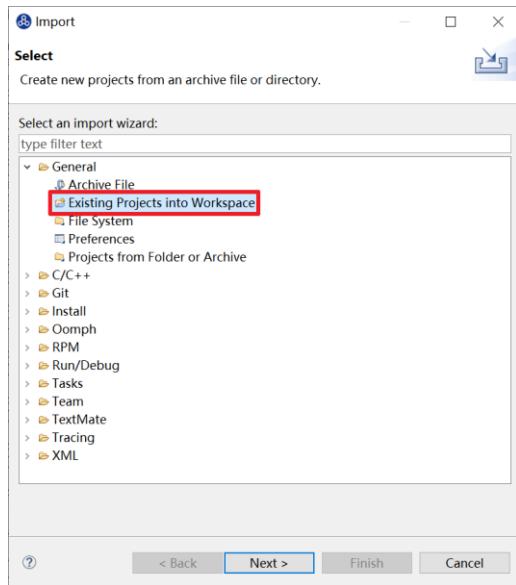


图 4-15 选择导入的方式

- 点击“Browse”，选择需要导入的项目路径，如图 4-16 所示，

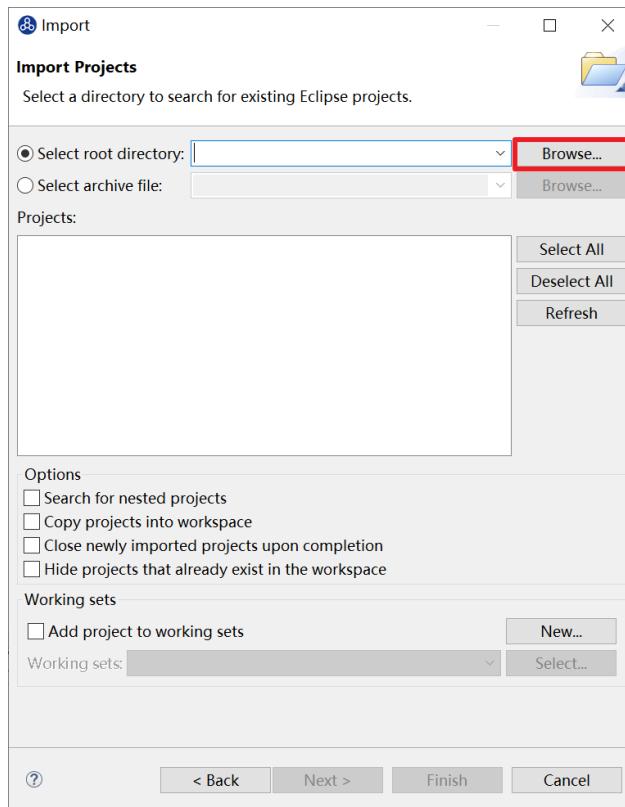


图 4-16 选择需要导入的项目

- 需要的导入的项目成功被 IDE 识别，点击“Finish”如图 4-17 所示

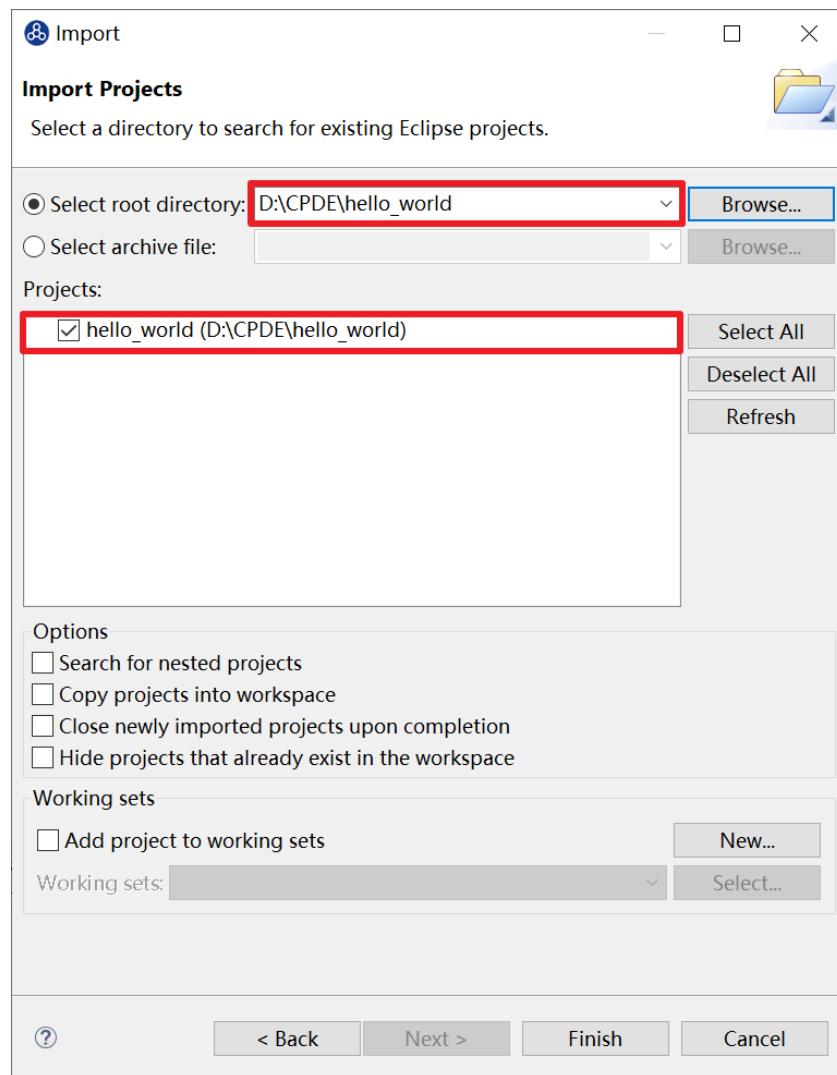


图 4-17 识别出要导入的项目

- 在 IDE 的项目资源管理器中显示导入项目的目录结构如图 4-18 所示。已有项目默认为 N307 的编译选项，Nuclei SDK 仅包含 helloworld 使用到的文件。需要更多的 Nuclei SDK 源码请访问 Github (<https://github.com/riscv-mcu/hbird-sdk>) 获取源码。

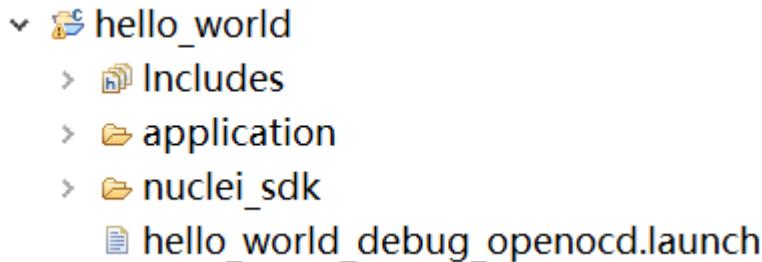


图 4-18 显示项目的文件结构

4.3. 无模板手动创建项目

本节将介绍如何使用手动方式在 Nuclei Studio IDE 创建一个用户自定义的 Hello World 项目。开发板为 HummingBird Evaluation Board，内核为 N307。该方法除了创建项目之外，还需要手动设置各种选项和路径，详细步骤如下。

4.3.1. 手动创建项目

- 在菜单栏中选择 “File—> New —> C/C++ Project” 。如图 4-19 所示。
- 如图 4-20 所示，在弹出的窗口中设定如下参数。
 - Project name: 项目命名。
 - Use default location: 如果勾选了此选项，则会使用默认 Workspace 文件夹存放此项目。
 - Project type: 选择 “Hello World RISC-V C Project” 。

然后点击 Next 进入下一步

- 如图 4-21 所示，在弹出的窗口中设置 Hello World 项目的基本信息。
 - 确保 “Source” 选项内容为空，直接单击 “Next” 进入下一步。

- 如图 4-22 所示，在弹出的窗口中设置项目的调试或者发布属性。

 - 该步骤可以使用默认信息不做任何修改，直接单击“Next”进入下一步。

- 在弹出的窗口中设置项目所使用的 RISC-V 工具链。

 - 此处不要配置，直接选择“Finish”，至此便完成了 Hello World 项目的创建。

- 创建完成的 Hello World 项目界面如图 4-23 所示。
- 新建一个 application 文件夹。如图 4-24，在工程处右击选择 New → Folder，输入 application，点击“Finish”完成新建工程，如图 4-25。将 main.c 拖入 application 文件夹完成文件分类。

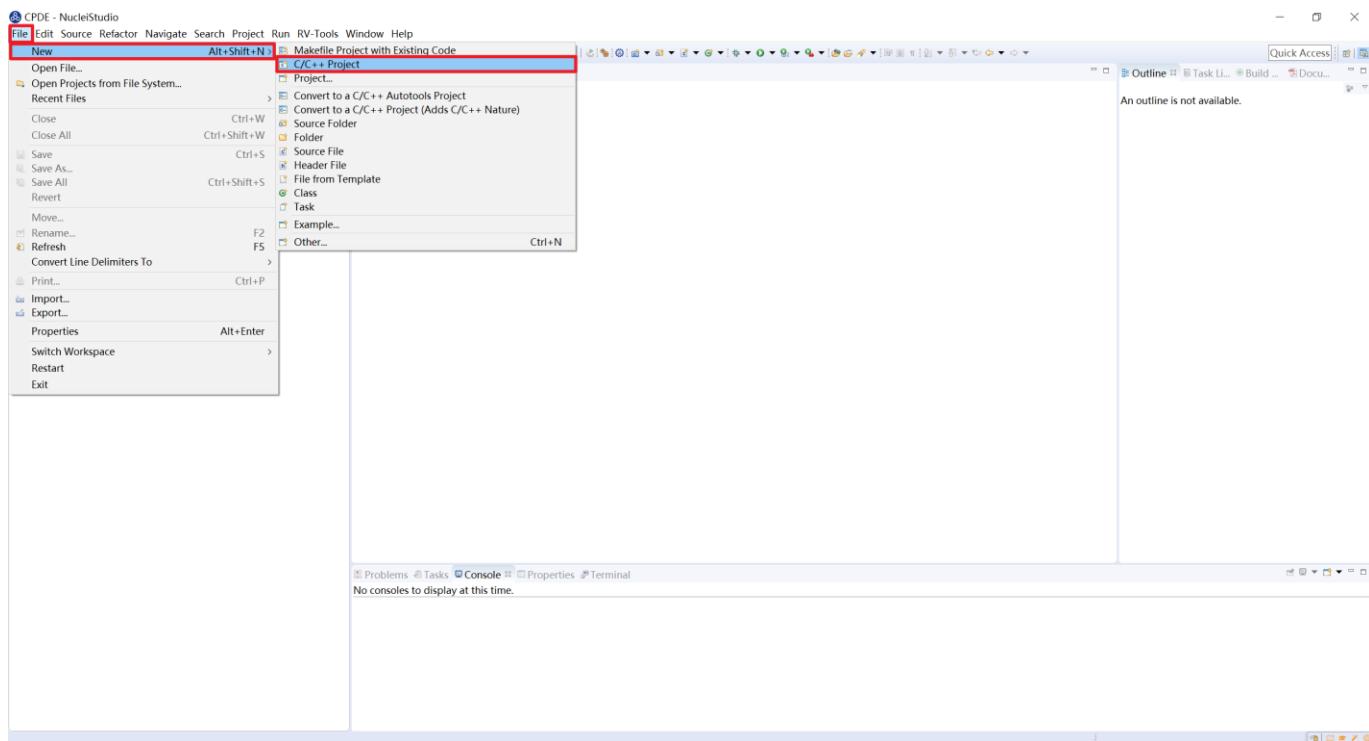


图 4-19 新建 C/C++ Project

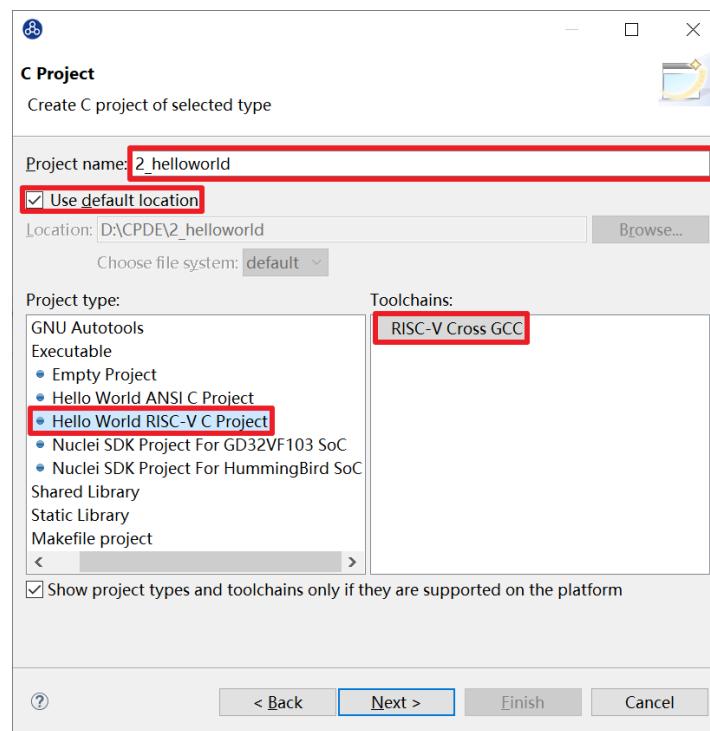


图 4-20 设置 C Project 项目名和类型

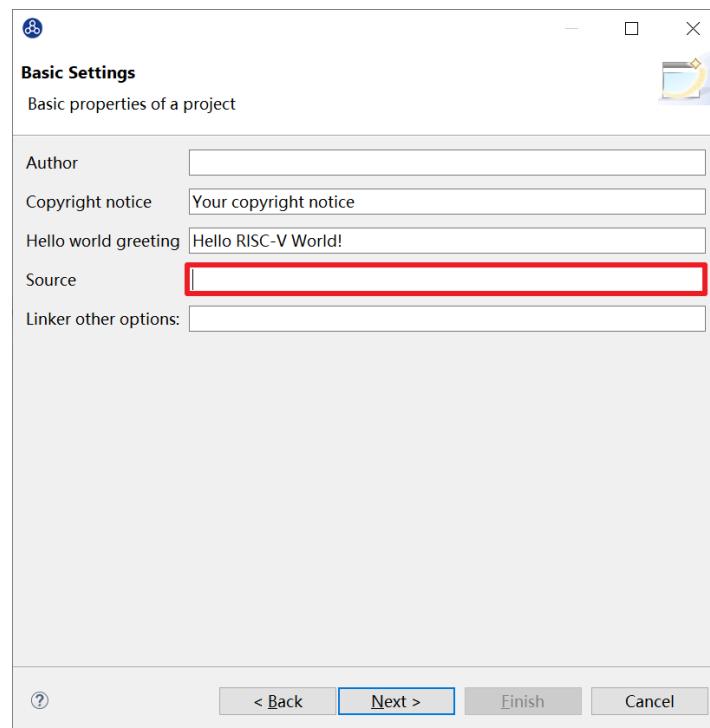


图 4-21 设置 Hello World 项目的基本信息

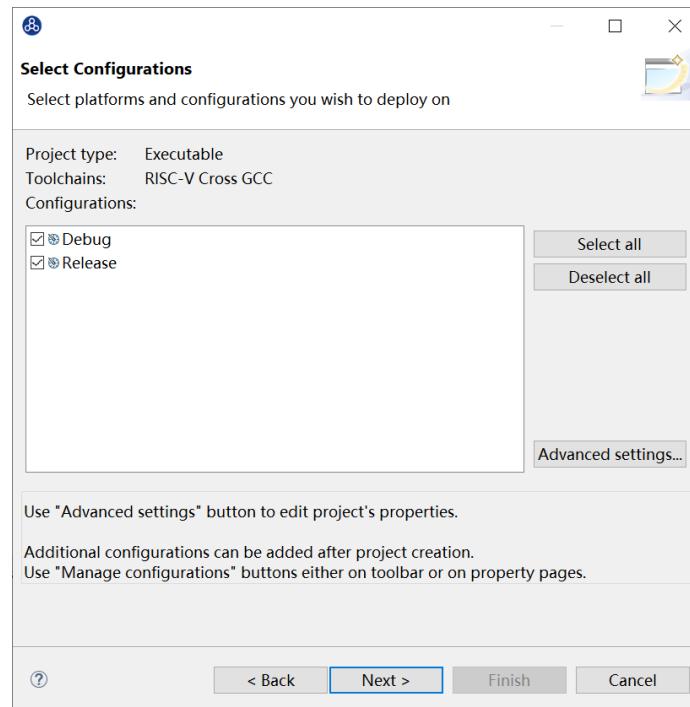


图 4-22 设置项目的调试或者发布属性

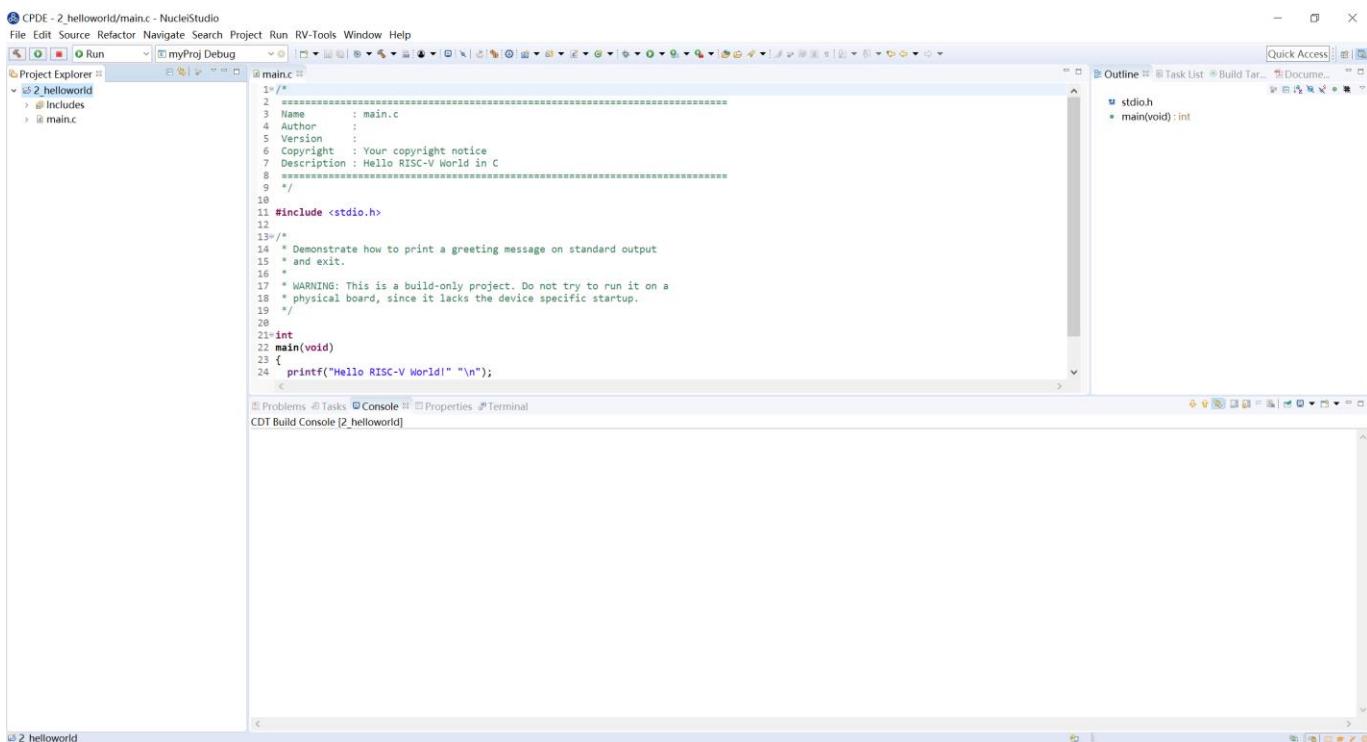


图 4-23 创建完成 Hello World 项目的界面

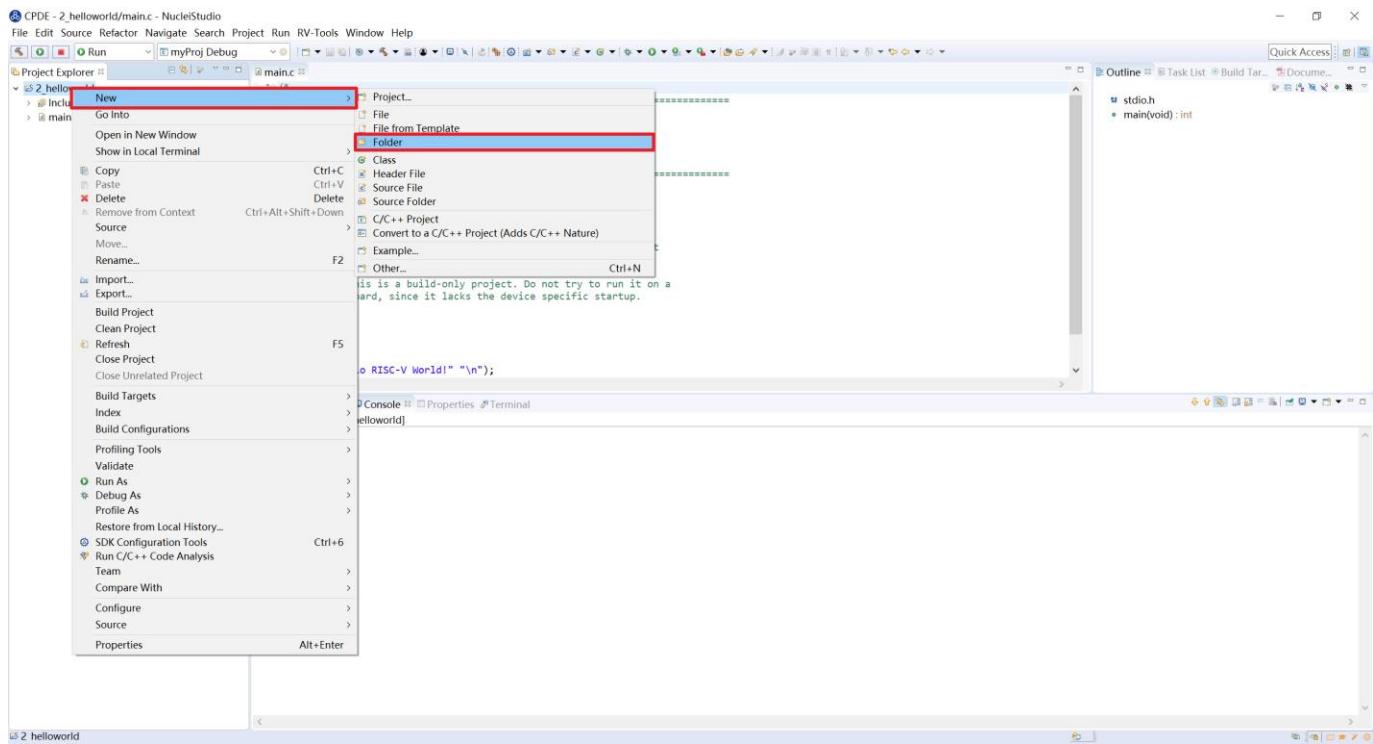


图 4-24 打开新建文件夹

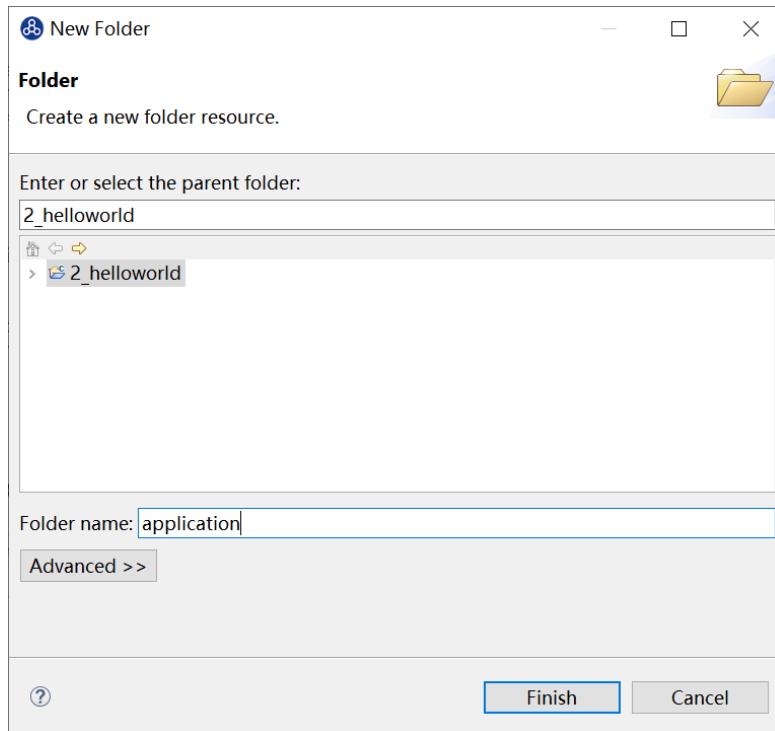


图 4-25 完成新建文件夹

4.3.2. 配置项目的 nuclei_sdk

本节介绍如何将 nuclei_sdk 加入到项目中，SDK 的具体内容本文不做详细介绍，可以参考 https://doc.nucleisys.com/nuclei_sdk/index.html。如果需要使用 SDK 的其他源文件，请到 Github 获取全部的 Nuclei SDK 源码，链接如下：<https://github.com/Nuclei-Software/nuclei-sdk>。本节仅介绍将 nuclei_sdk 中 helloworld 需要的文件加入到项目的步骤，具体步骤如下：

- 进入 Nuclei Studio 的 2_helloworld 项目，按照如下步骤添加 nuclei_sdk 源文件。
 - ◆ 如图 4-26 所示，在 Project Explorer 栏中选中 2_helloworld 项目，单击鼠标右键，选择“Properties”打开工程设置页面。
 - ◆ 如图 4-27 所示，在弹出的窗口中单击“Resource”，在右侧的 Location 栏目中单击其最右侧的箭头图标 ，则会弹出文件窗口进入 2_helloworld 项目的文件夹位置。
 - ◆ 将图 4-13 中介绍的 nuclei-eclipse_demo.rar 压缩包中的 nuclei_sdk 文件夹复制放于 2_helloworld 项目的目录下，如图 4-28 所示。
 - ◆ 回到 Nuclei Studio，在 Project Explorer 栏中选中 2_helloworld 项目，单击鼠标右键，选择“Refresh”，如图 4-29 所示。
 - ◆ Refresh 之后 2_helloworld 项目的下便可以看到 nuclei_sdk 文件夹，如图 4-30 所示，至此便完成了 nuclei_sdk 源文件的导入。

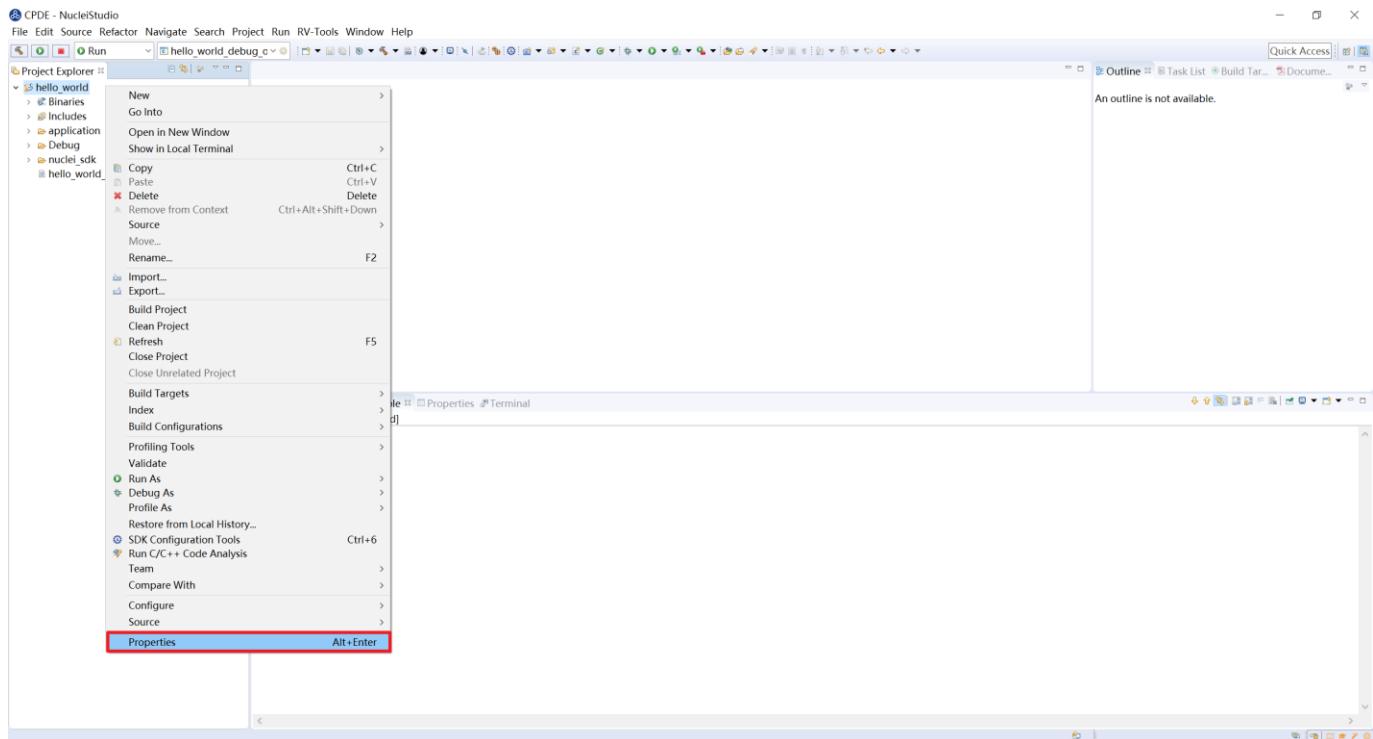


图 4-26 打开工程设置选项页面

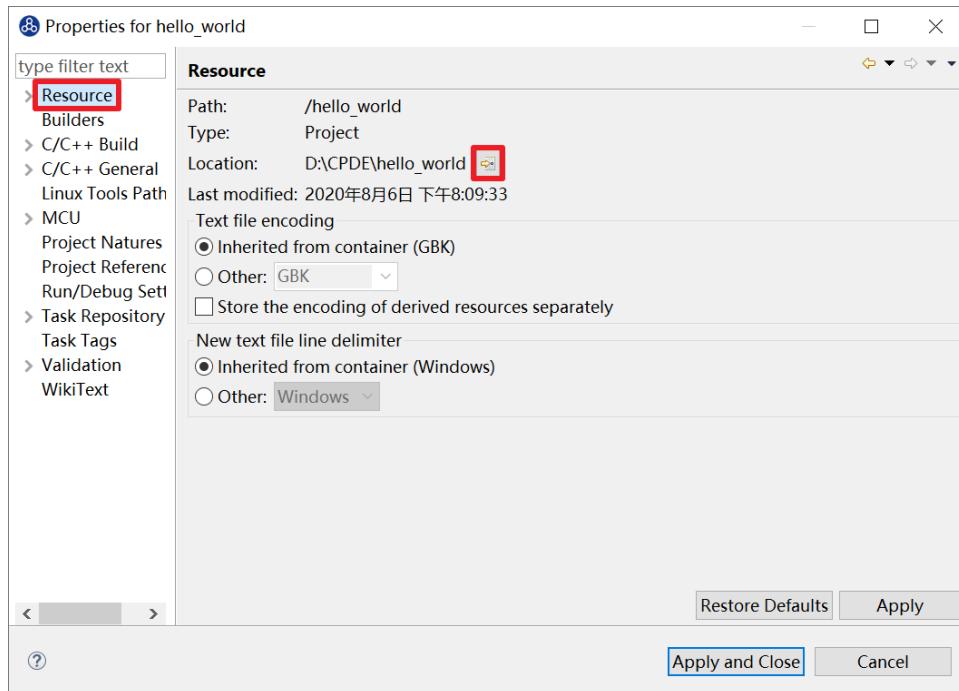


图 4-27 在弹出窗口进入 hello_world 项目的文件夹位置

.settings	2020/8/5 17:46	文件夹
application	2020/8/5 17:46	文件夹
nuclei_sdk	2020/8/5 17:46	文件夹
.cproject	2020/8/5 17:46	CPROJECT 文件
.project	2020/8/5 17:46	PROJECT 文件
hello_world_debug_openocd.launch	2020/8/5 18:11	LAUNCH 文件

图 4-28 添加 nuclei_sdk 文件夹

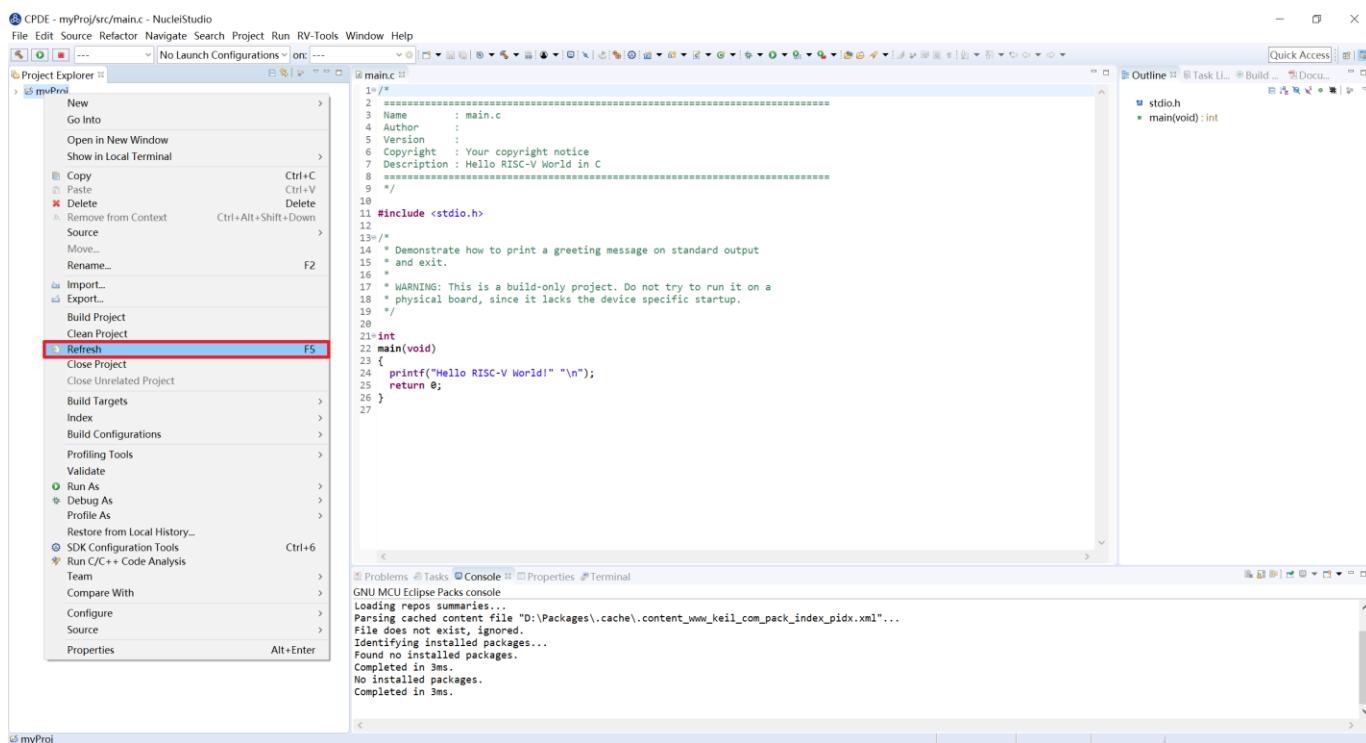


图 4-29 刷新工程

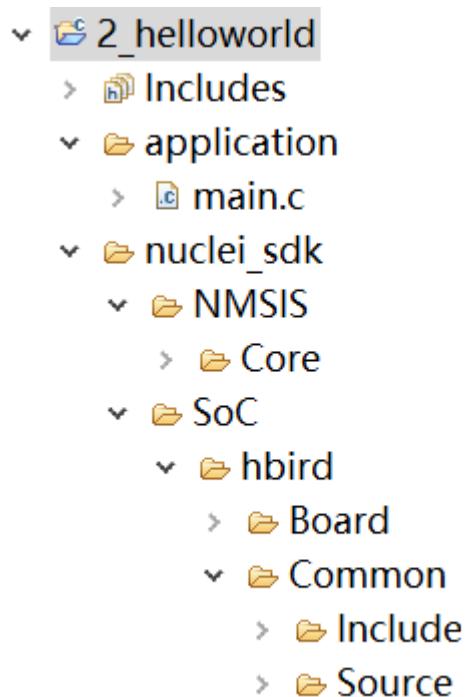


图 4-30 新建项目目录下的文件夹

4.3.3. 配置项目的编译和链接选项

为了使项目源代码能够被正确编译，需要配置编译和链接选项。

注意：本节中设置的编译与链接选项均为 GCC 工具链的常用选项，与在 Linux 环境中使用时的同名选项含义一致，本节在此不做赘述介绍。

配置编译与连接选项的步骤如下：

- 在 Project Explorer 栏中选中 hello_world 项目，单击鼠标右键，选择“Properties”。
- 在弹出的窗口中，展开 C/C++ Build 菜单，单击“Setting”，在右侧的 Tool Settings 栏目中进行设置。
- 如图 4-31 所示，选中 Target Processor，我们的内核是 N307，因此需要按照图所示勾选配置选项，分别如下。
 - **Architecture:** 选择 RV32I。

- Multiply extension (RVM) : 需勾选。
- Atomic extension (RVA) : 需勾选。
- Compressed extension (RVC) : 需勾选。
- Integer API: 选择“ILP32”。
- Floating Point ABI: 选择 single precision
- Code model: 选择“Medium Any”。
- 单击右下角的“Apply”按钮。

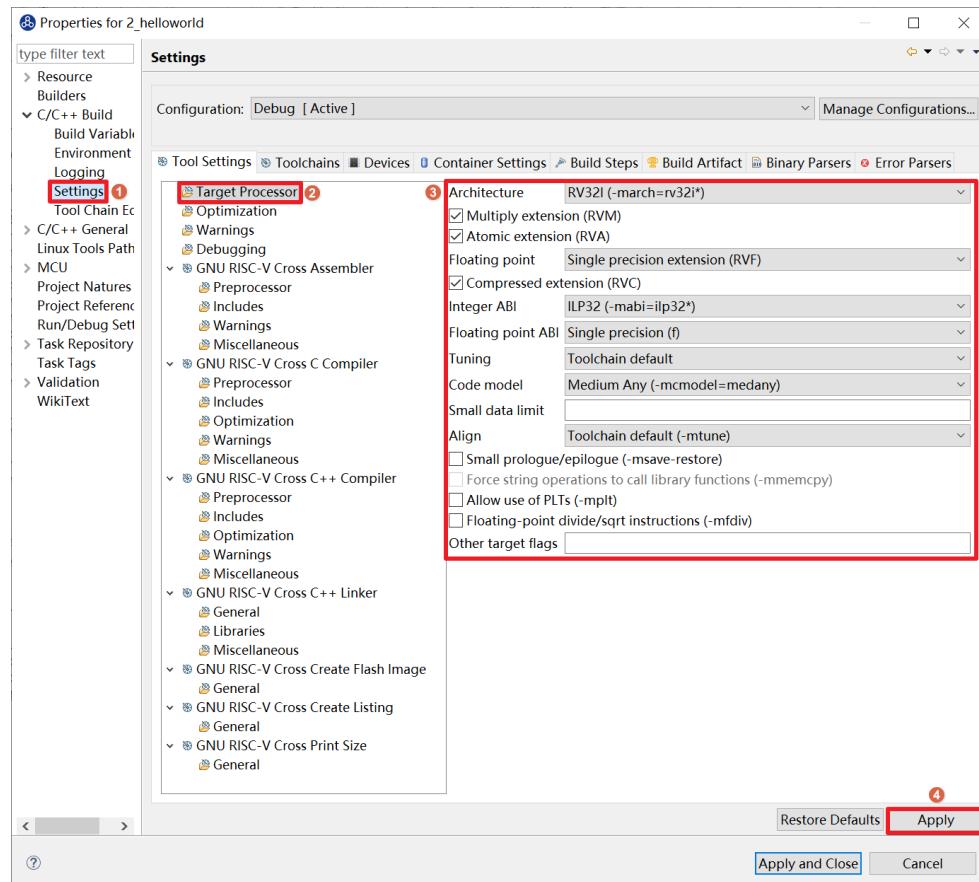


图 4-31 配置 Target Processor 选项

- 如图 4-32 所示，选中“Optimization”，按照图所示勾选配置选项。

- Optimization Level: 选择 Optimization Most (-O2)。
- 依次勾选:
 - ◆ Function Sections (-ffunction-sections)
 - ◆ Data Sections (-fdata-sections)
 - ◆ No common uninitialized (-fno-common)

注意：上述选项均为通用的 GCC 编译优化选项，请用户自行查阅 GCC 手册了解其含义。

- 单击右下角的“Apply”按钮。

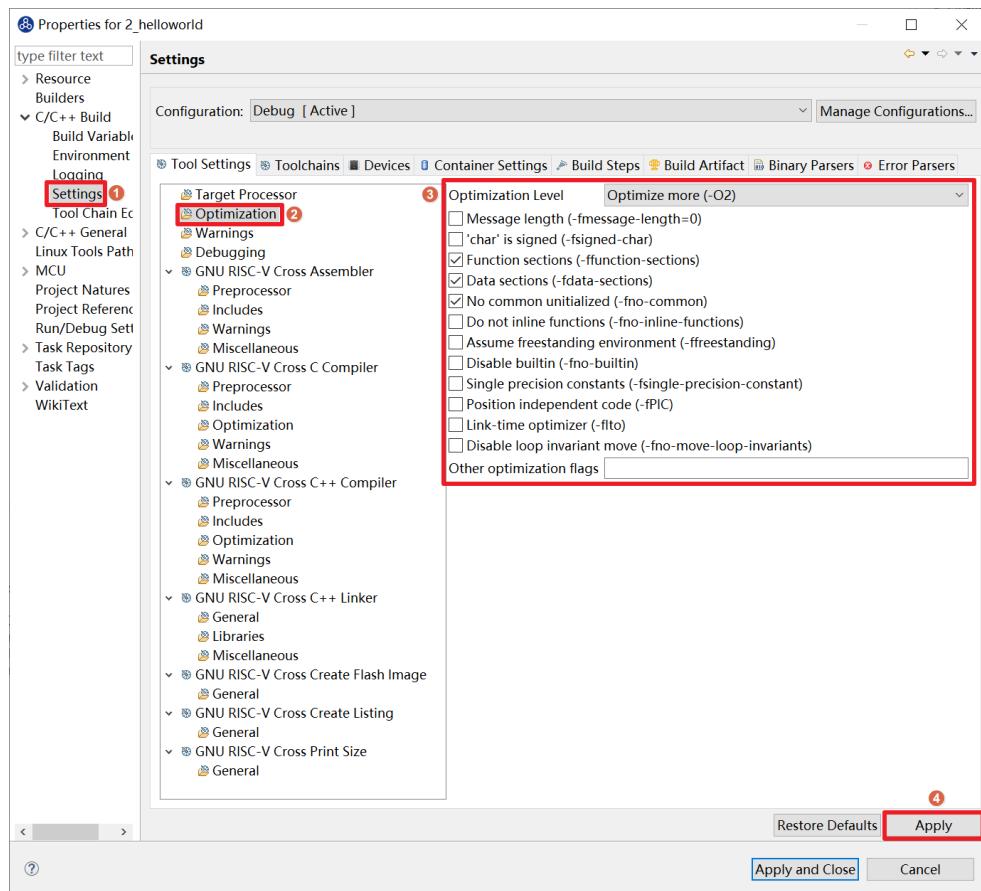


图 4-32 配置 Optimization 选项

- 如图 4-33 所示，选中 Debugging，按照图中所示勾选配置选项，分别为：

- Debug Level：选择 Default (-g)。
- 单击右下角的“Apply”按钮。

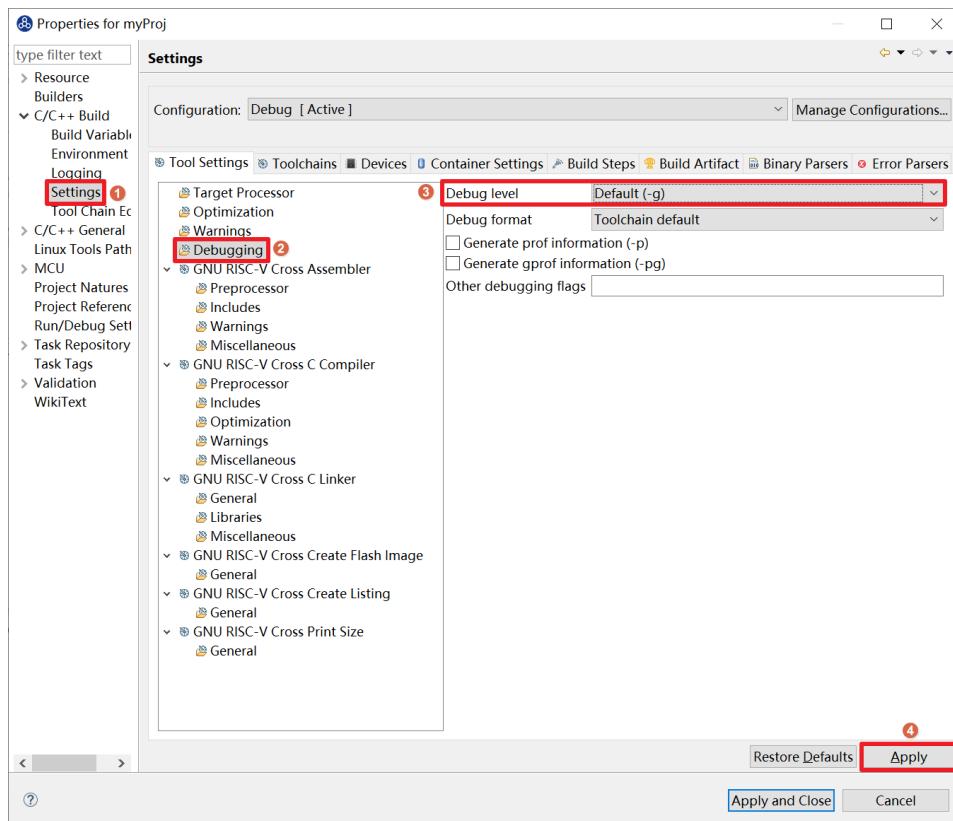


图 4-33 配置 Debugging 选项

- 选中 GNU RISC-V Cross C Linker 的 General。

- 如图 4-34 所示，按照如下步骤设置链接器的所需的链接脚本。
 - ◆ 选中右上角的加号按键。
 - ◆ 在弹出的窗口中单击“Workspace”按钮。
 - ◆ 这里我们使用 HummingBird 评估板，所以可以选择 ILM 下载模式对应的 `gcc_hbird_ilm.ld` 文件。在弹出的窗口中选择 NucleiStudio 文件包中的

nuclei_sdk/SoC/hbird/Board/hbird_eval/Source/GCC 文件夹下
gcc_hbird_ilm.ld 文件。其他下载模式切换此处文件，各文件详细介绍如下，可
根据自己的实际情况选择。

- **gcc_hbird_ilm.ld** 脚本将程序代码段约束在 ILM 的地址区间，意味着程序将被直接下载在 MCU 的 ILM 中，并从 ILM 开始执行。ILM 由 SRAM 组成，会掉电丢失。
 - **gcc_hbird_flash.ld** 脚本程序代码段的物理地址约束 Flash 区间，将代码段的逻辑地址约束在 ILM 的地址区间，意味着程序将被直接下载在 MCU 的 Flash 中，但是上电后要通过引导程序将代码段搬运到 ILM 中，然后从 ILM 中开始执行。
 - **gcc_hbird_flashxip.ld** 脚本程序代码段约束 Flash 区间，意味着程序将被直接下载在 MCU 的 Flash 中，并直接从 Flash 开始执行。程序被烧写在 Flash 中，不会掉电丢失。
 - 用户可以按照自己的需求选择合适的链接脚本。本节示例选择 **gcc_hbird_ilm.ld** 作为演示。
- ◆ 设置完毕请单击右下角的“Apply”按钮。

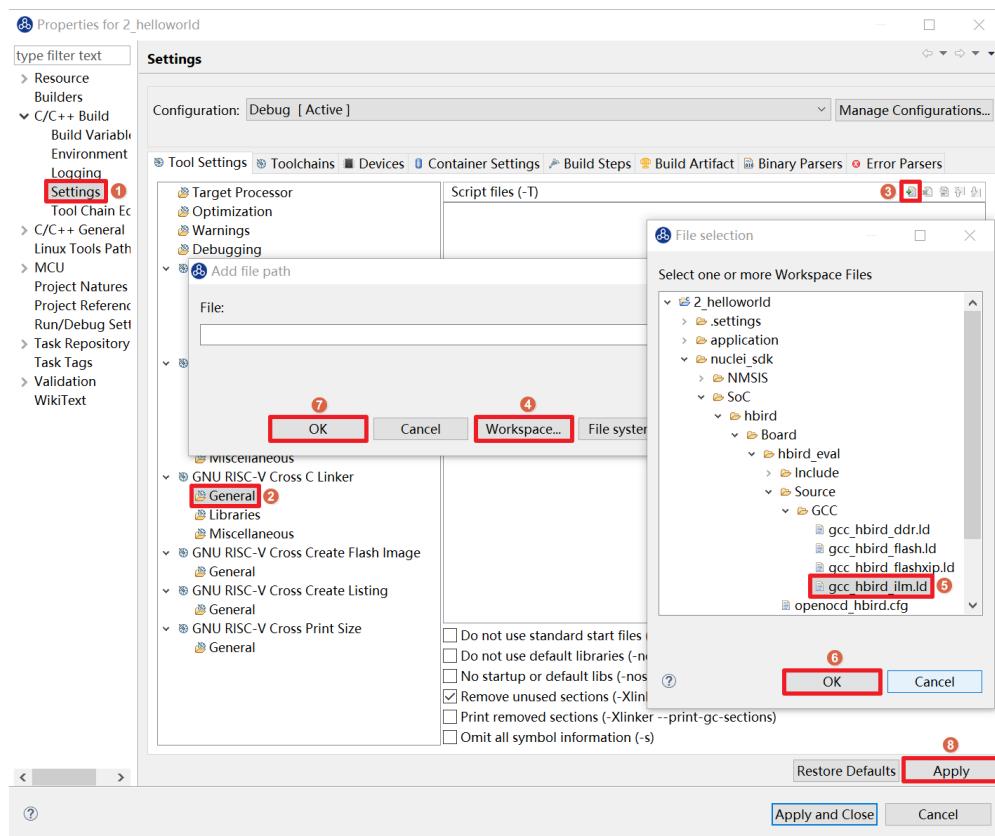


图 4-34 配置链接脚本

■ 如图 4-35 所示，按图所示勾选配置选项，分别如下。

- Do not use standard start files (-nostartfiles)。
- Remove unused sections (--gc-sections)。
- 单击右下角的“Apply”按钮。
- 注意：上述选项均为通用的 GCC 链接选项，请用户自行查阅 GCC 手册了解其含义。

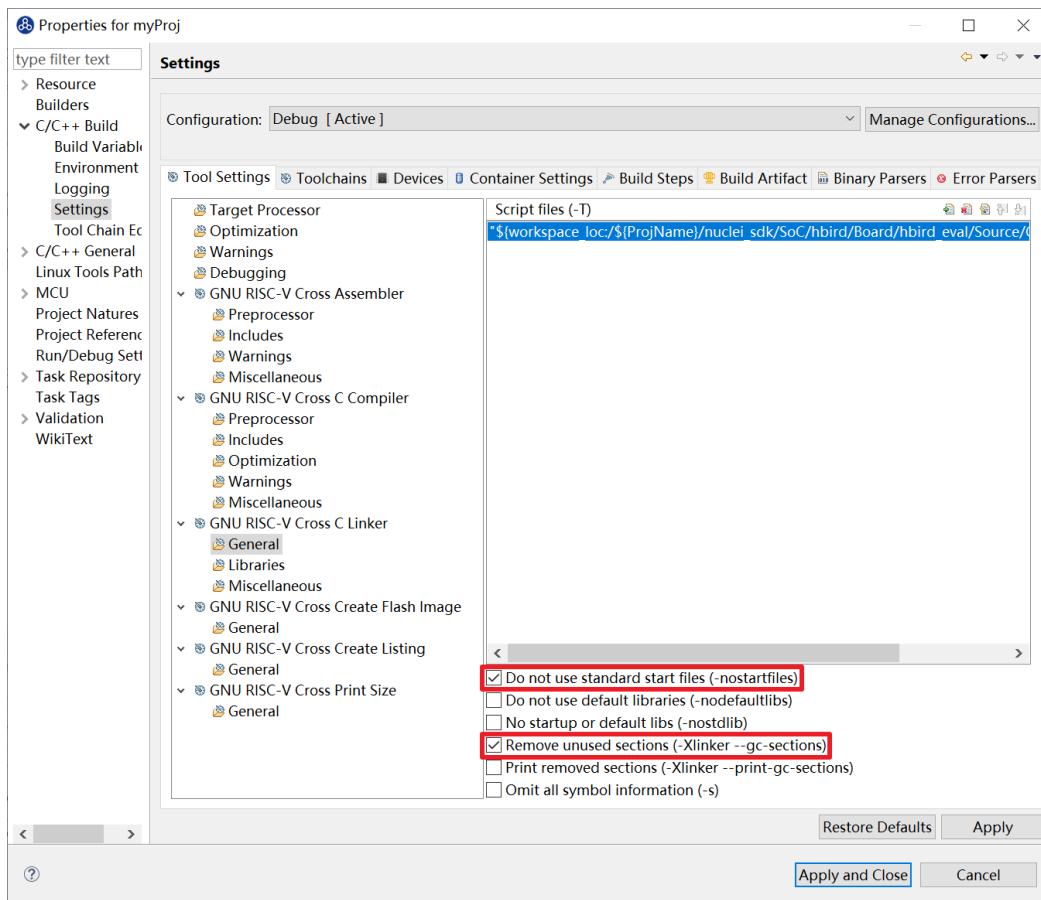


图 4-35 配置链接的 General 选项

- 如图 4-36 所示，选中 GNU RISC-V Cross C Linker 的 Miscellaneous，按照图所示勾选配置选项。
 - 勾选“Use newlib-nano”。
 - 因为 Hello World 程序的 Printf 不需要打印浮点数，所以不要勾选“Use float with nano printf”。
 - 单击右下角的“Apply”按钮。

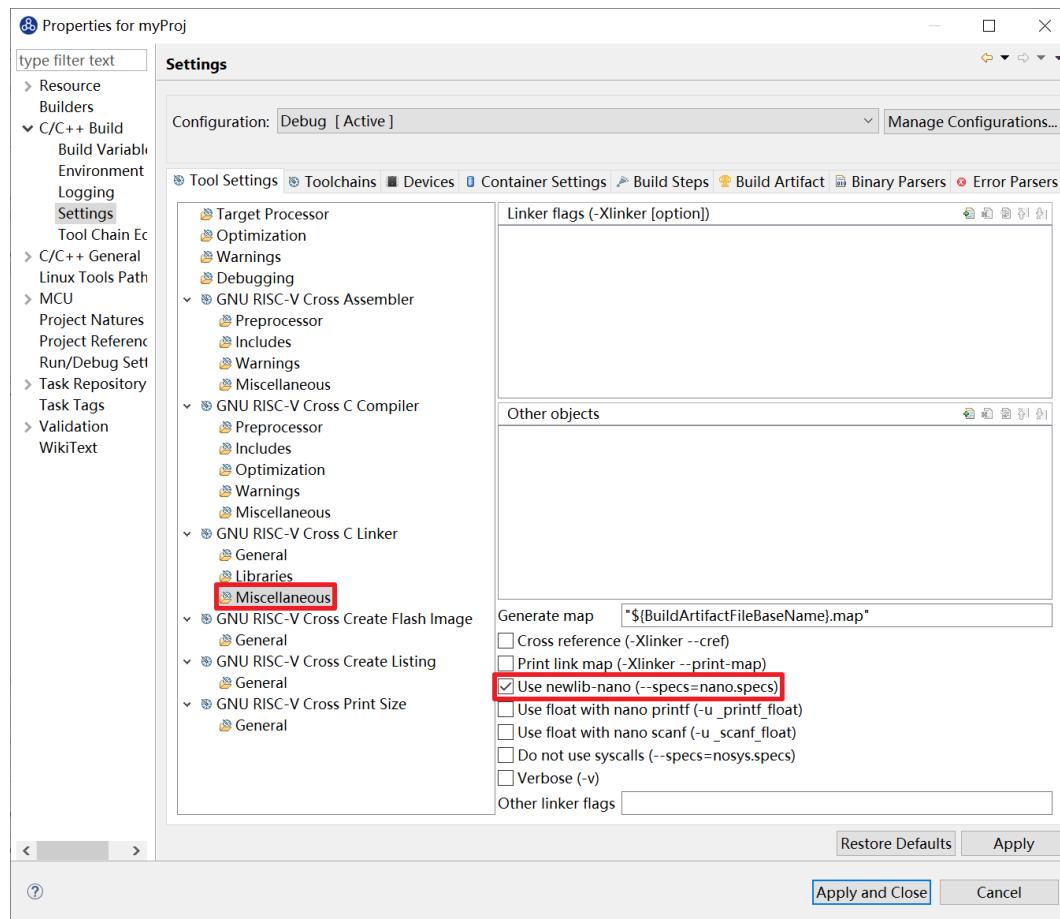


图 4-36 配置链接的 Miscellaneous 选项

4.3.4. 配置项目的包含路径和文件

为了能够正确编译 nuclei_sdk 文件夹中的源文件，需要按照如下步骤配置项目的包含路径和包含文件。

- 如图 4-26 所示，在 Project Explorer 栏中选中 hello_world 项目，点击鼠标右键，选择“Properties”。
- 在弹出的窗口中，展开 C/C++ Build 菜单，单击“Setting”，在右侧的 Tool Settings 栏目中进行设置。
- 如图 4-37 所示，选中 GNU RISC-V Cross C Assembler 的 Includes，按照图中所示配置包含文件，步骤如下。

- 在 **Include paths** 栏目单击加号键。
 - 在弹出的窗口中单击 “**Workspace**”，弹出 **Folder selection** 窗口。
 - 在 **Folder selection** 窗口中选择项目的 **nuclei_sdk** 目录下的 **NMSIS>Core>Include** 文件夹。
 - 在右下角单击 “**Apply**” 完成配置。
- 采用上述方法，依次添加 **nuclei_sdk** 目录下的 **SoC>hbird>Board>hbird_eval>Include**, **SoC>hbird>Common>Include** 和 **SoC>hbird>Common>Source>Stubs** 文件夹作为包含路径，并采用同样的方法为 **GNU RISC-V Cross C Compiler** 的 **Includes** 栏目设置包含路径。设置完成后的界面如图 4-38 所示。

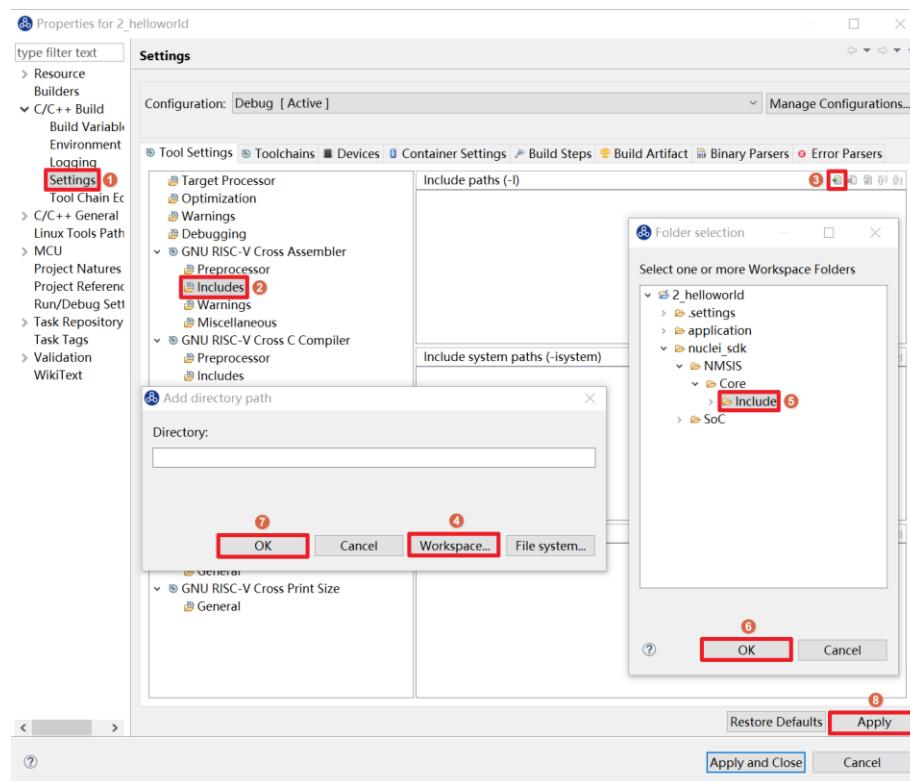


图 4-37 设置 GNU RISC-V Cross C Assembler 的 **Includes** 栏目包含路径

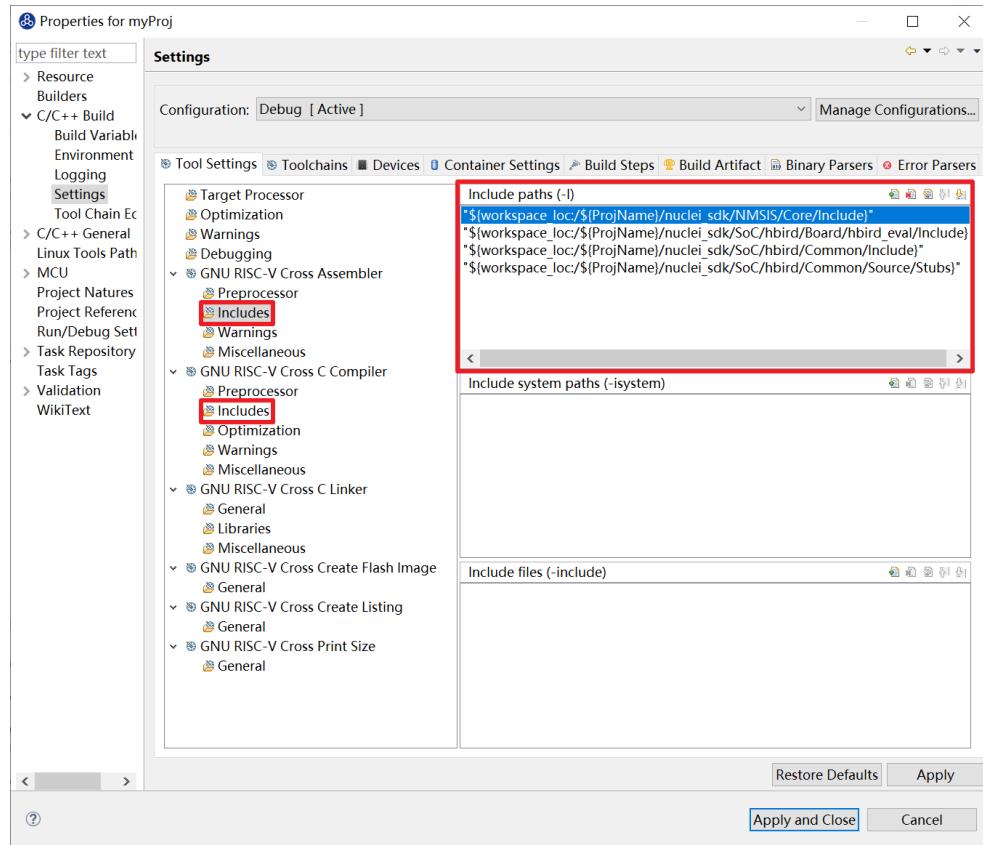


图 4-38 设置 GNU RISC-V Cross C Compiler 的 Includes 栏目包含路径

4.4. 基于已有的 Makefile 创建项目

本节将介绍如何使用已有的 Makefile 在 Nuclei Studio IDE 创建一个使用 Makefile 的 Hello World 项目。开发板为 HummingBird Evaluation Board，内核为 N307。请先下载 Nuclei SDK，Github 链接为：<https://github.com/Nuclei-Software/nuclei-sdk>。该方法除了创建项目之外，还需要手动设置各种选项和路径，这里以 helloworld 为例，详细步骤如下。

4.4.1. 手动新建项目

在菜单栏中选择“File—> New —> Makefile Project with Existing Code”。如图 4-39 所示。

如图 4-40 所示，在图标 1 处输入工程名，这里我们命名为 nuclei-sdk。在图标 2 处输入 SDK 的实际路径。在图标 3 处选择“RISC-V Cross GCC”。点击图标 4 完成新建项目。

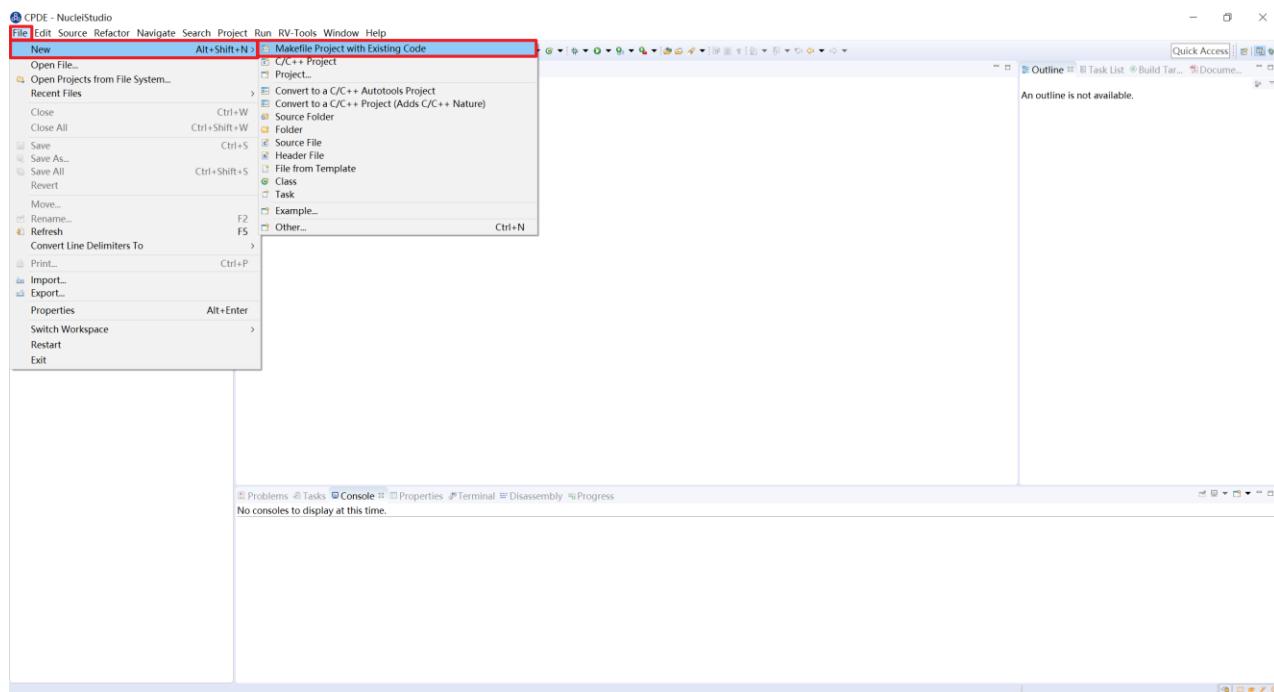


图 4-39 新建基于 Makefile 的工程

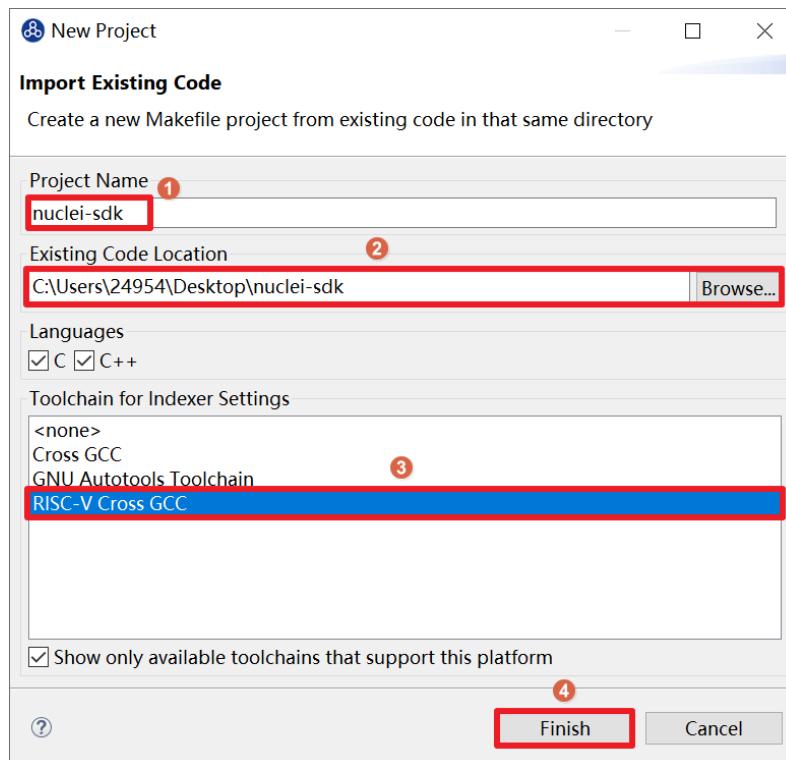


图 4-40 设置工程选项

4.4.2. 设置 Makefile 路径和 Build 选项

右击新建好的工程，选择“Properties”打开设置页面，如图 4-41，选择“C/C++ Build”，在“Build Location”中选择“Workspace”。在弹出的弹窗中选择“application -> baremetal -> helloworld”点击“OK”再点击“Apply”保存。

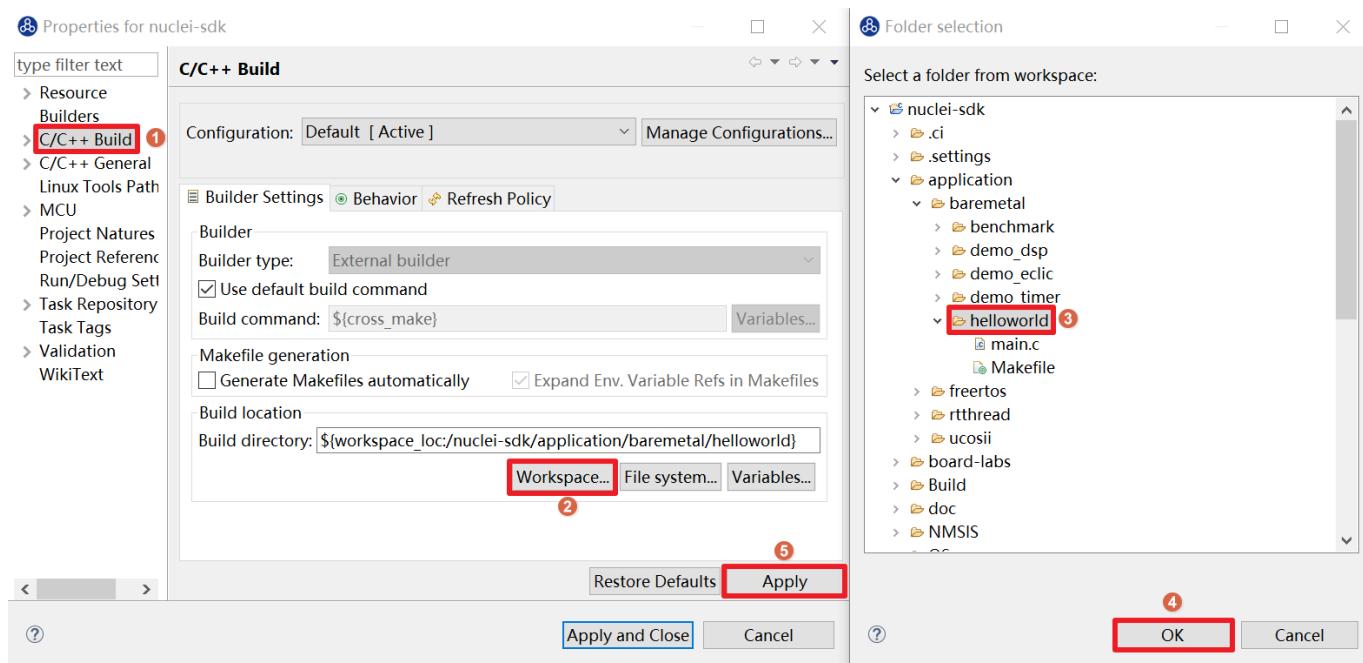


图 4-41 配置 Makefile 路径

如图 4-42，在“C/C++ Build”中选择“Behavior”栏目，确保勾选“Build (Incremental Build)”选项并输入“all CORE=n307 DOWNLOAD=ilm”。其中“CORE”选项根据实际的内核变化，这里以 n307 为例。“DOWNLOAD”选项可以修改不同的下载模式，详细请参考 4.1 节，这里以 ilm 模式为例。因为例程使用 HummingBird Evaluation Board，所以 SoC 和 Board 都不必修改，如果使用其他开发板，以 RVSTAR 为例，请在此处设置增加“SOC=gd32vf103 BOARD=gd32vf103v_rvstar”，并且由于 RVSTAR 仅支持 FLASHXIP 模式，需要将“DOWNLOAD”设置为“flashxip”，同时“CORE”修改为“n205”。完成后点击“Apply”保存修改。

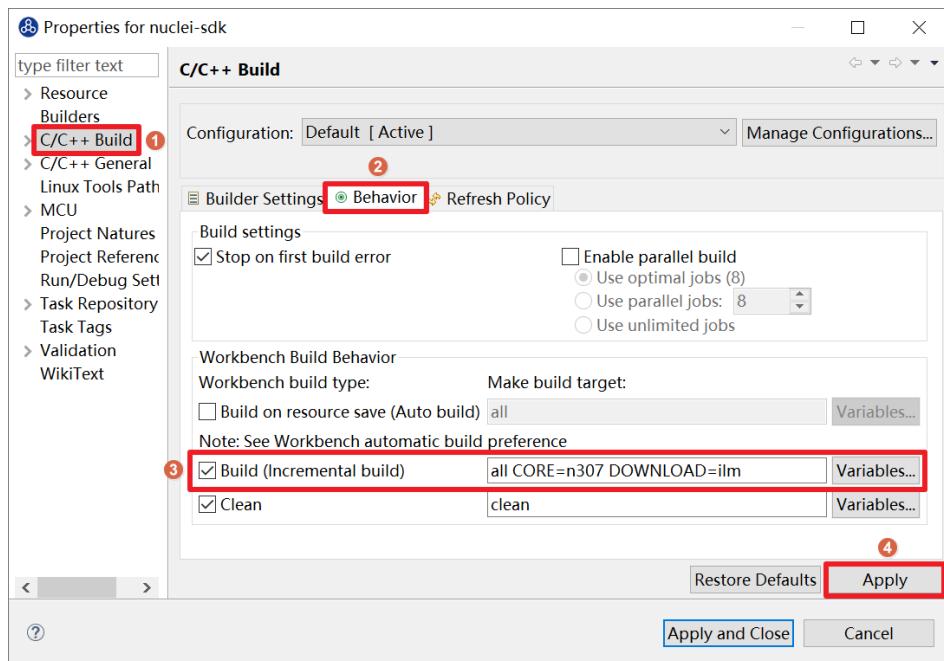


图 4-42 配置 Build 选项

编译运行以及下载的详细内容请参考 5、6 章。

5. 编译项目

5.1. Nuclei SDK 工程设置工具

在 Nuclei Studio 中可以通过 Nuclei SDK 工程设置工具一键修改通过 Nuclei SDK Project Wizard 创建的工程的编译链接选项。本工具不支持无模板手动创建的项目和基于 Makefile 创建的项目。详细使用步骤如下：

单击选中需要修改的工程，之后如图 5-1 打开 RV-Tools>SDK Configuration Tools，可以打开修改编译选项的弹窗。也可以单击要修改的工程后，如图 5-2，点击工具栏的工程设置工具图标。或者单击要修改的工程后，键盘按下 **ctrl+6**。也可以单击要修改的工程后，如图 5-3，右击打开右键菜单，选择 SDK Configuration Tools。如图 5-4 为工程设置工具弹窗，各选项详细功能如下：

- **projectName** 为当前选中的工程名。
- **Configuration Core** 为当前工程对应的内核。由于工具根据 **ARCH** 和 **ABI** 选项反推出对应的内核，而不同的内核可能有相同的 **ARCH** 和 **ABI** 选项，所以显示上可能会有所偏差，只要 **ARCH** 与 **ABI** 为正确的选项即可。此选项为方便快速切换内核选项使用。
- **ARCH** 对应的当前工程的 **arch** 选项。
- **ABI** 对应的当前工程的 **abi** 选项。
- **Download** 对应当前工程的下载模式，可以切换选择不同的下载模式，目前仅蜂鸟开发板支持切换下载模式，RVSTAR 仅有 **FLASHXIP** 模式。
- **NewLIB** 对应的使用 **NEWLIB** 库。可以选择是否使用 **nano** 库，是否使用浮点打印。
- **Optimization Level** 对应编译的优化等级。
- **Extra Common Flags** 对应的是额外的通用编译选项。可以添加额外的通用编译选项。
- **Extra C Flags** 对应的是额外的 **C** 编译选项。可以添加额外的 **C** 编译选项。
- **Extra C++ Flags** 对应的是额外的 **C++** 编译选项。可以添加额外的 **C++** 编译选项。
- **Extra ASM Flags** 对应的是额外的汇编编译选项。可以添加额外的汇编编译选项。
- **Extra Link Flags** 对应的是额外的链接选项。如果此选项已经有默认选项并且需要增加编译选项，可以在编译选项开头或结尾处相隔一个空格字符再增加编译选项。

根据需要修改以上的选项，这里我们修改优化等级为**-Os** 优化生成可执行文件大小。点击 **save** 一键修改编译选项，之后如图 5-5，右击修改后的工程打开右键菜单，选择“**Clean Project**”清理一下工程，再点击锤子图标即可完成修改编译选项后重新编译工程。

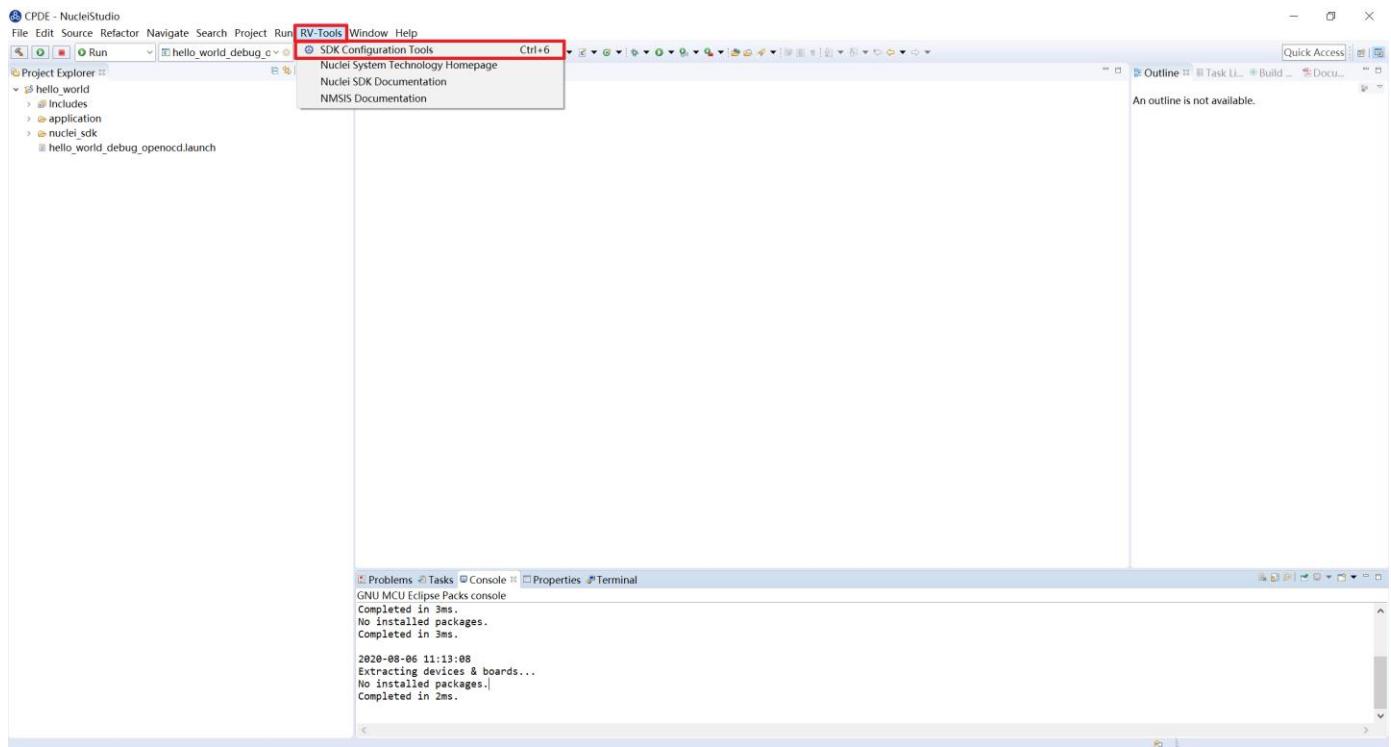


图 5-1 打开修改编译选项弹窗 1

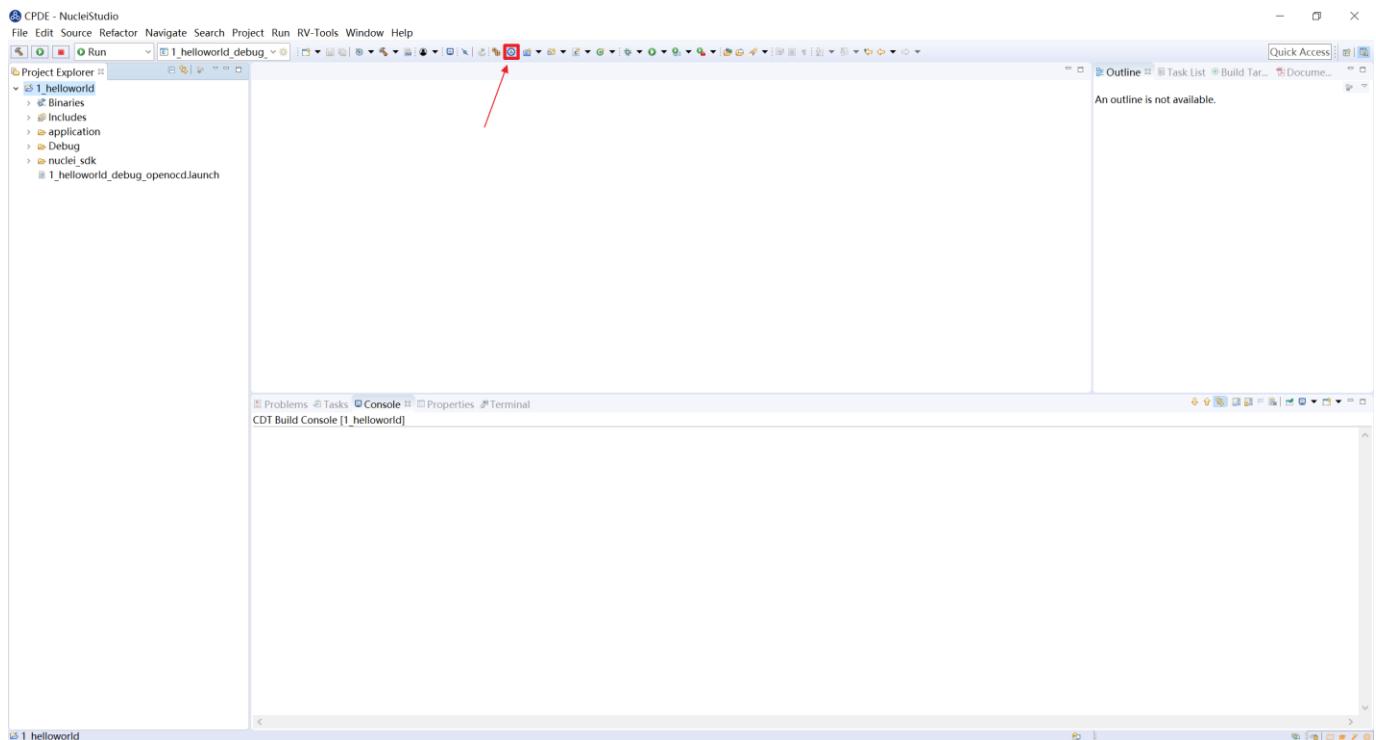


图 5-2 打开修改编译选项弹窗 2

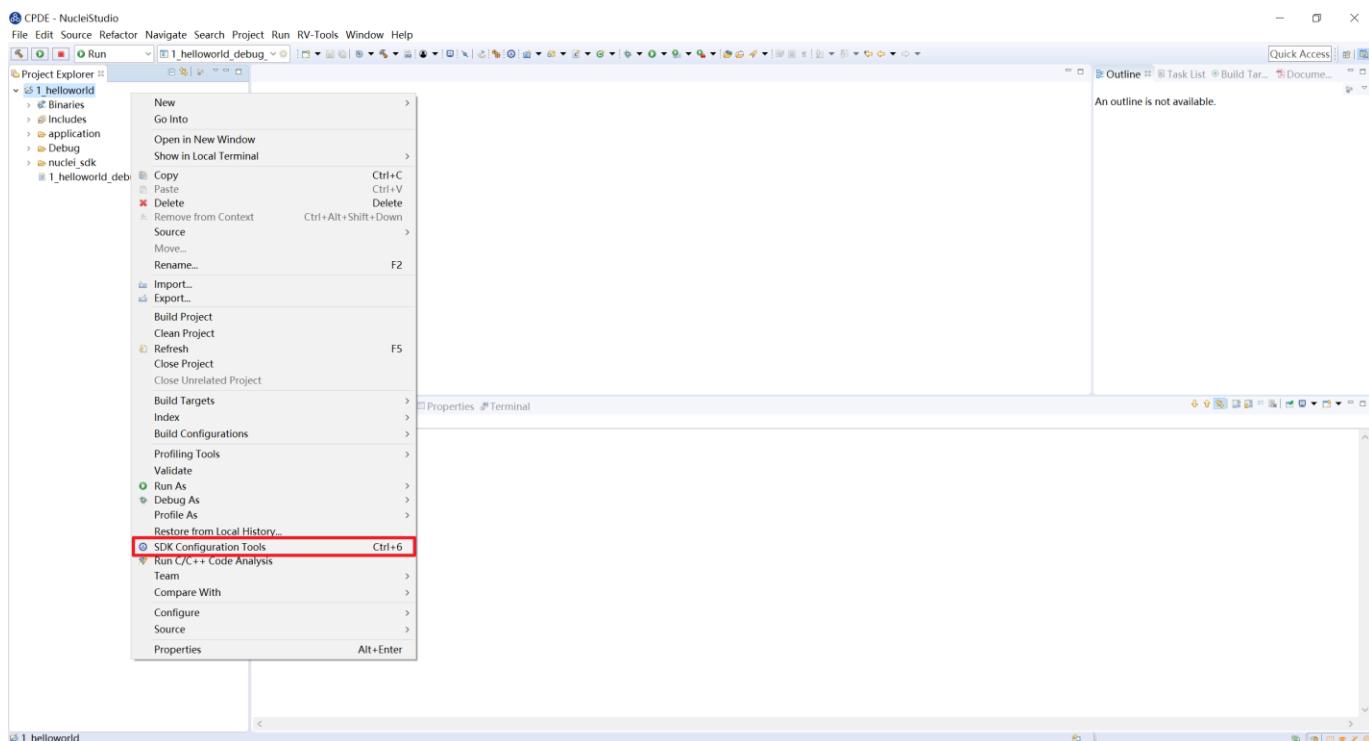


图 5-3 打开修改编译选项弹窗 3

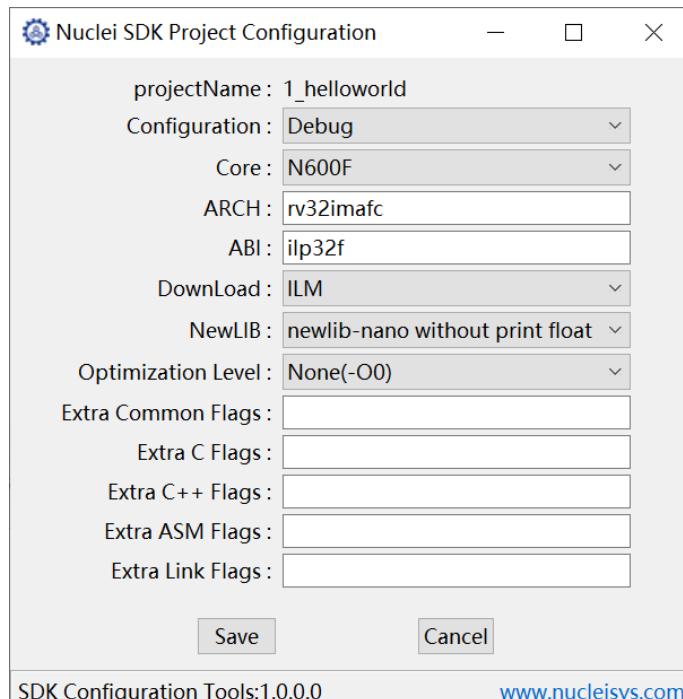


图 5-4 一键修改编译选项

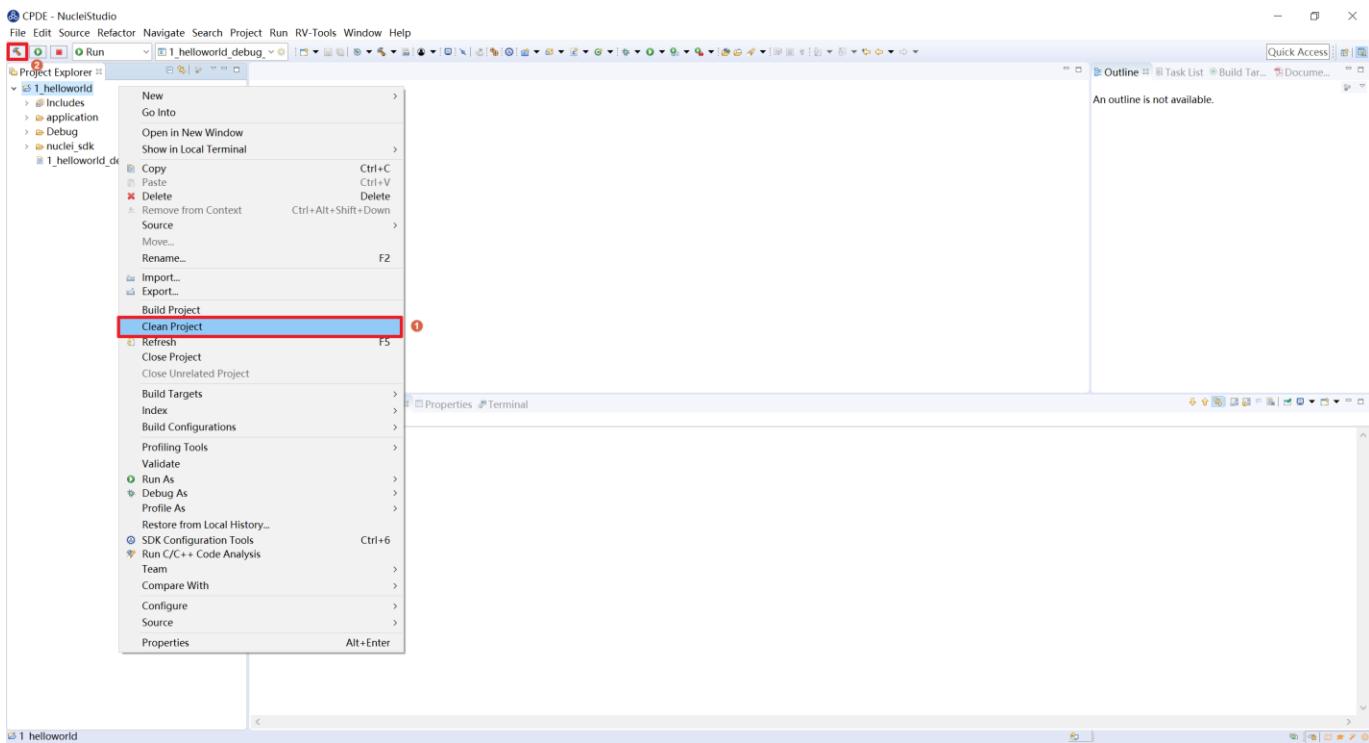


图 5-5 清理工程并重新编译工程

需要注意的是如果要切换工程从 32 位变为 64 位，需要打开调试设置页面，如图 6-6，修改“command”中“set arch riscv:rv32”为“set arch riscv:rv64”，从 64 位切换回 32 位也应当修改这里的参数为对应的数值。

5.2. 在 Nuclei Studio 中编译 Hello World 项目

在 Nuclei Studio 中编译的步骤如下。

- 为了保险起见，建议先将项目清理一下。如图 5-6 所示，在 Project Explorer 栏中选中 hello_world 项目，单击鼠标右键，选择“Clean Project”。
- 如图 5-7 所示，单击菜单上的锤子按钮，开始对项目进行编译。如果编译成功，则显示如图 5-7 所示，能够看到生成可执行文件的代码体积大小，包括 text 段、data 段和 bss 段，以及总大小的十进制和十六进制数值。使用 Makefile 方式新建的工程需要在右键菜单中选择“Build Project”进行编译。

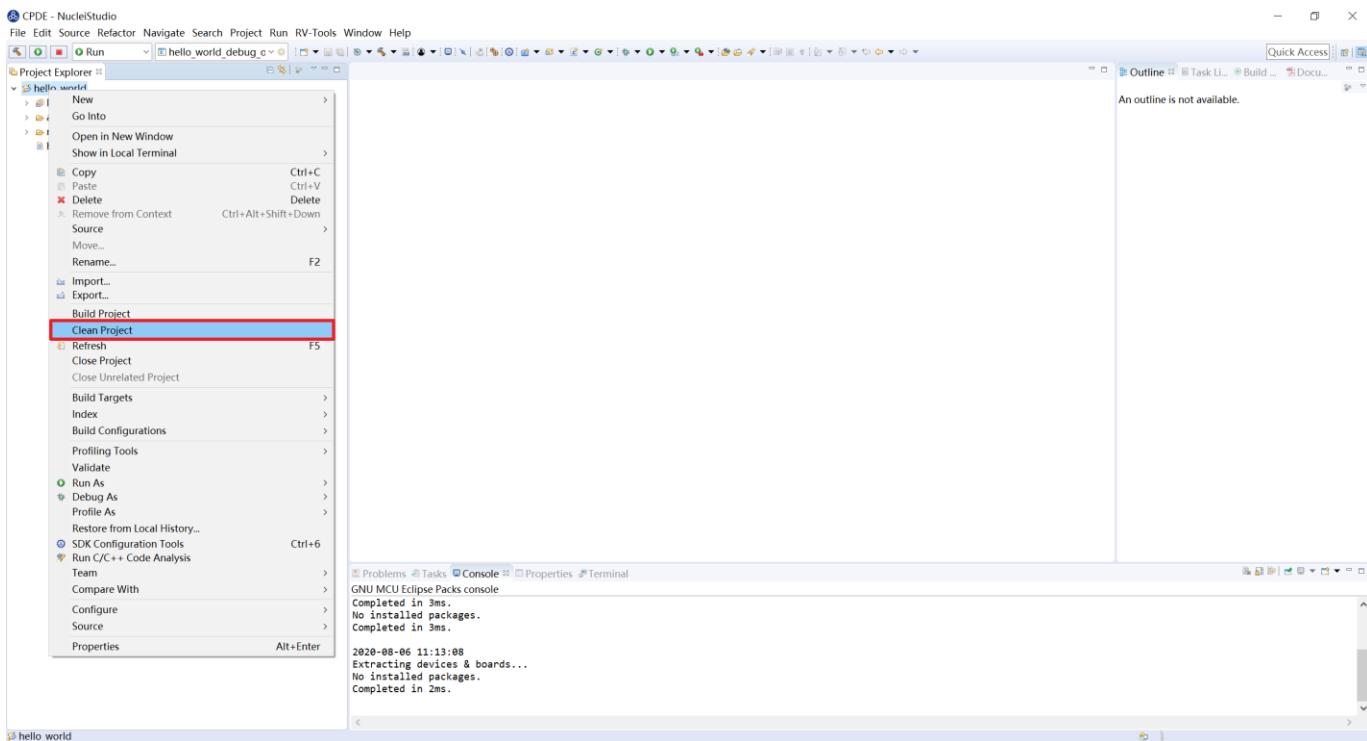


图 5–6 对 hello_world 项目单击右键选择“Clean Project”

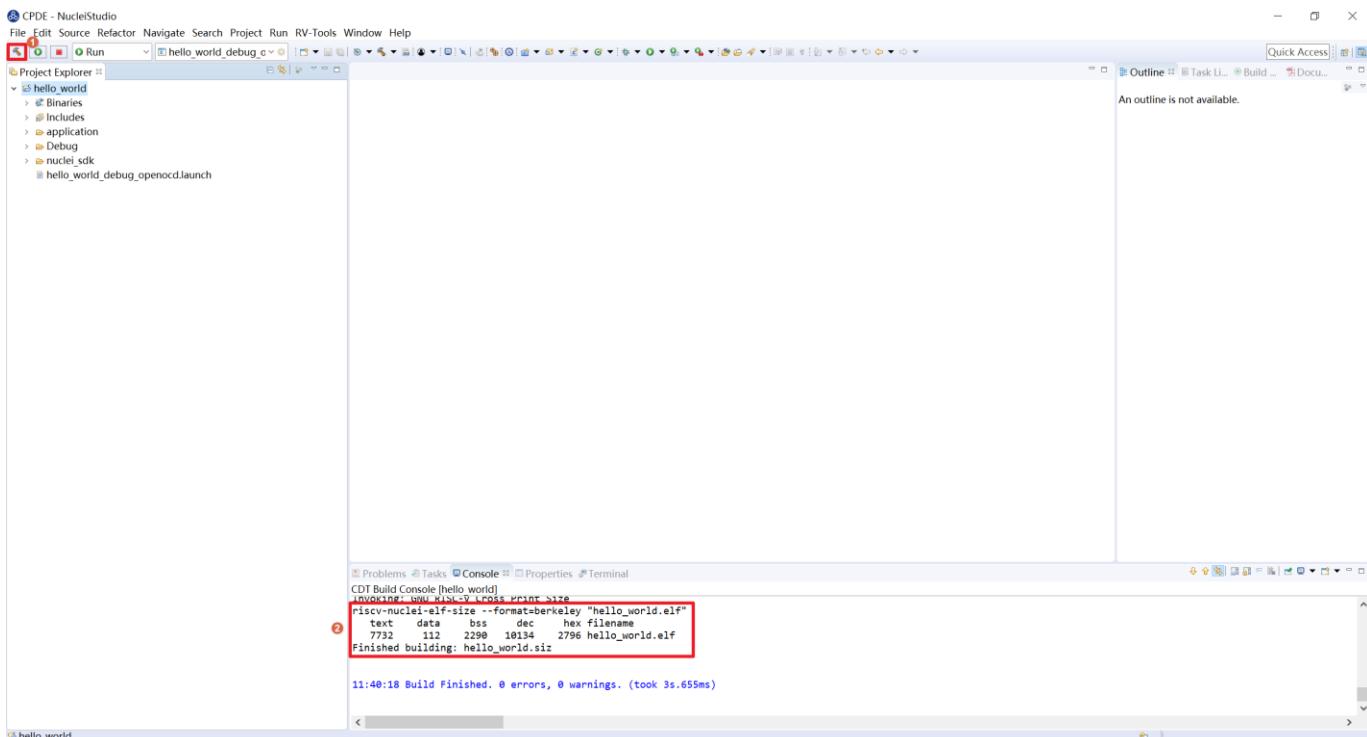
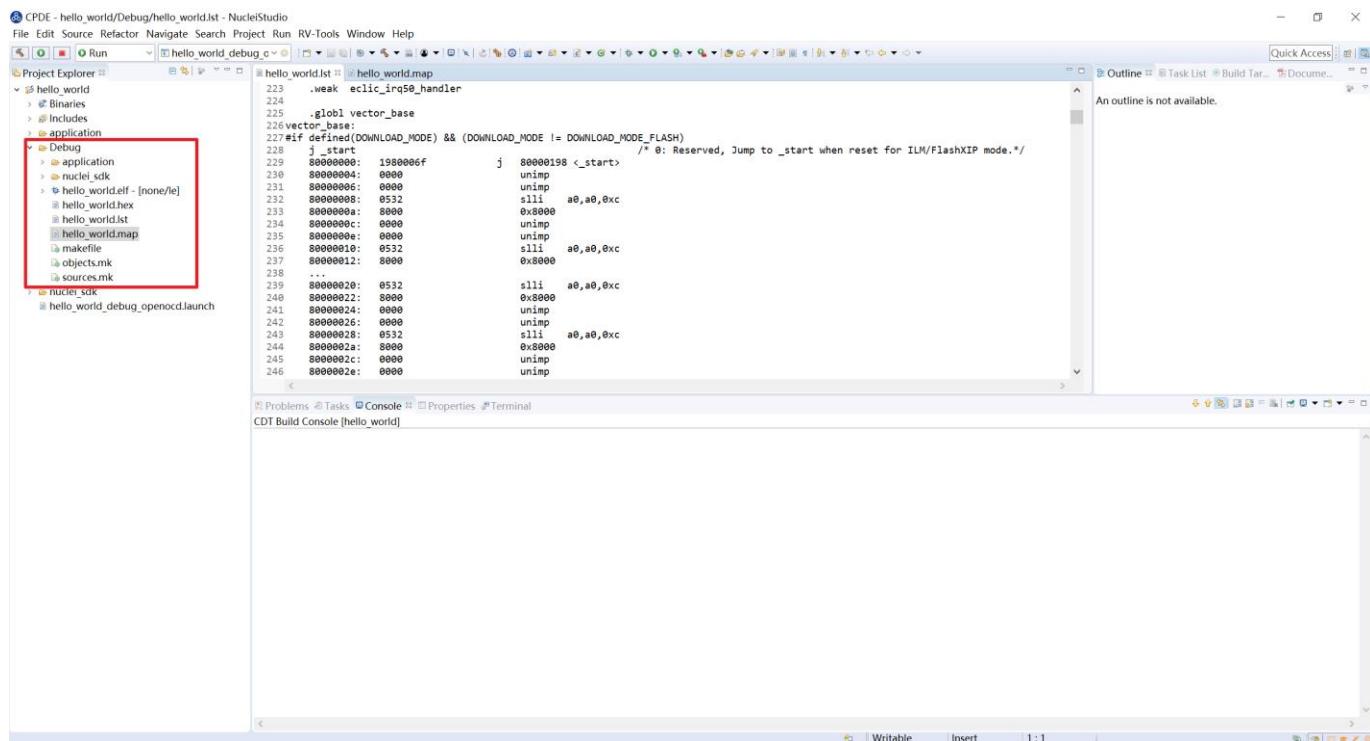


图 5–7 单击锤子图标对 hello_world 项目进行编译

编译成功后可以看到增加了 Debug 文件夹，如图 5-8，各文件作用如下：

- hello_world.elf 是生成的可执行文件。
- hello_world.hex 是生成的 Hex 文件。
- hello_world.lst 是生成的 list 文件，可以看到反汇编和简单的代码分部信息。
- hello_world.map 是生成的 map 文件，可以详细的看到生成的代码分布情况。



```

CPDE - hello_world/Debug/hello_world.lst - NucleiStudio
File Edit Source Refactor Search Project Run RV-Tools Window Help
Run hello_world_debug.c
Project Explorer hello_world.lst hello_world.map
- hello_world
  - Binaries
  - Includes
  - application
  - Debug
    - application
    - nucel_sdk
    - hello_world.elf - [none/le]
    - hello_world.hex
    - hello_world.lst
    - hello_world.map
  - makefile
  - objects.mk
  - sources.mk
nucel_sdk
hello_world_debug_OpenOCD.launch
Outline Task List Build Target Document
An outline is not available.

223 .weak _clic_irq50_handler
224
225 .globl vector_base
226 vector_base:
227 #ifndef DOWNLOAD_MODE && (DOWNLOAD_MODE != DOWNLOAD_MODE_FLASH)
228 j _start /* 0: Reserved, Jump to _start when reset for ILM/FlashZIP mode.*/
229 80000000: 1980006f      j 80000198 <_start>
230 80000004: 0000      unimp
231 80000006: 0000      unimp
232 80000008: 0532      slli a0,a0,0xc
233 8000000a: 8000      0x8000
234 8000000c: 0000      unimp
235 8000000e: 0000      unimp
236 80000010: 0532      slli a0,a0,0xc
237 80000012: 8000      0x8000
238 ...
239 80000020: 0532      slli a0,a0,0xc
240 80000022: 8000      0x8000
241 80000024: 0000      unimp
242 80000026: 0000      unimp
243 80000028: 0532      slli a0,a0,0xc
244 8000002a: 8000      0x8000
245 8000002c: 0000      unimp
246 8000002e: 0000      unimp

```

图 5-8 查看编译后的生成文件

6. 调试运行项目

6.1. 使用蜂鸟调试器结合 OpenOCD 调试运行项目

6.1.1. 安装蜂鸟调试器驱动

Nuclei Studio 自 2021.02 版本起 openocd 实现 windows 免驱功能，使用此版本及以上 windows 环境的用户可跳过此节，Linux 环境仍需配置驱动。低于此版本的用户需按照以下方法

安装驱动。

6.1.1.1 在 Windows 系统中安装驱动

程序编译成功后，便可以将程序下载到 FPGA 原型开发板运行。首先将原型开发板与主机 PC 进行连接，步骤如下。

- 将蜂鸟调试器的一端插入主机 PC 的 USB 接口，另一端与原型开发板连接。注意：如果是第一次使用蜂鸟调试器，为了使得主机 PC 的 Windows 系统能够识别蜂鸟调试器的 USB，需要安装驱动，双击如图 1-2 所示的 NucleiStudio_IDE 文件包中的 HBird-Driver.exe 即可完成此驱动的安装。
- 由于蜂鸟调试器还包含了“将原型开发板输出的 UART 转换成 USB”的功能，因此如果蜂鸟调试器被主机 PC 识别成功（且驱动安装成功），那么将能够被主机识别成为一个 COM 串口。
 - 如图 6-1 所示，在主机 PC 的设备管理器中的“端口（COM 和 LPT）栏目”中可以查询到该 COM 的串口号（譬如 COM11）。
 - 此串口在后续的程序运行过程中将充当原型开发板运行程序的 printf 输出显示接口。

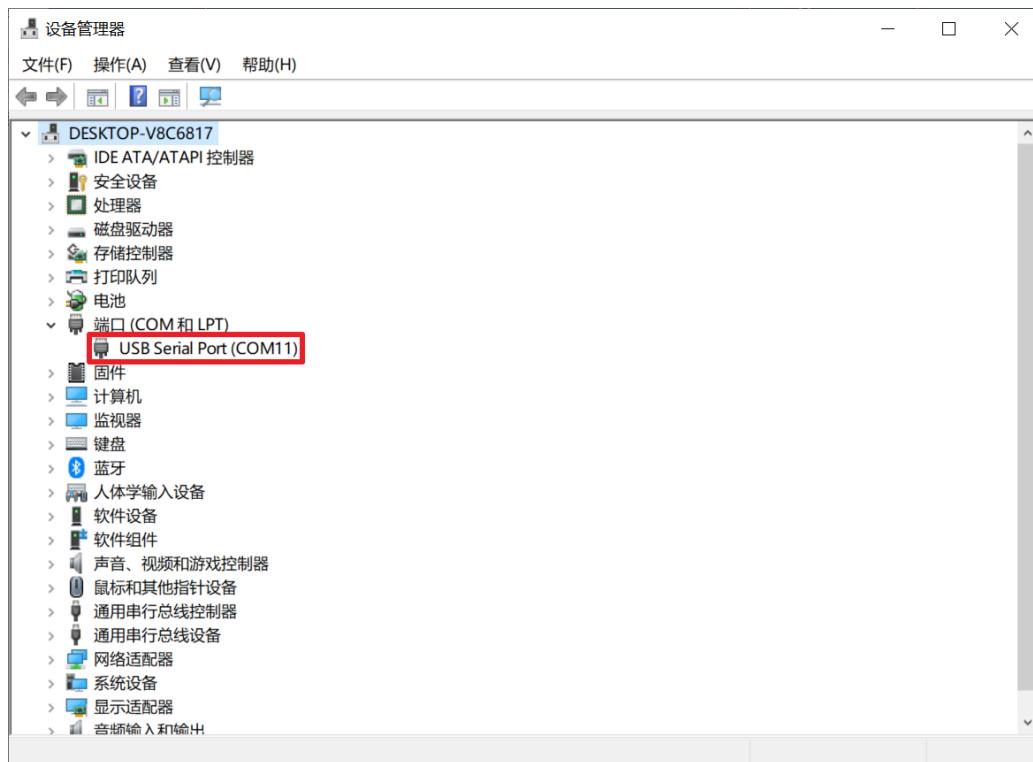


图 6-1 在设备管理器中查询 COM 串口号

注意：在通过“蜂鸟调试器”下载之前，需要注意“蜂鸟调试器”被 Windows 正确识别，检验的标准即为本节中所述正确地安装了 HBird-Driver.exe 的驱动，且能够在设备管理器中查询到 COM 的串口号。

6.1.1.2 在 Linux 系统中安装驱动

在 Linux 环境下安装驱动步骤如下：

- 1：连接开发板到 Linux 中，确保 USB 被 Linux 识别出来。如果使用虚拟机，如图 6-2，确保开发板连到了虚拟机当中。
- 2：在控制台中使用 lsusb 指令查看信息，参考的打印信息如下：

```
Bus 001 Device 010: ID 0403:6010 Future Technology Devices International, Ltd
FT2232XXXX
```

- 3: 控制台中输入 `sudo vi /etc/udev/rules.d/99-openocd.rules` 指令打开 `99-openocd.rules` 文件，输入如下内容，保存退出。

```
SUBSYSTEM=="usb", ATTR{idVendor}=="0403",  
ATTR{idProduct}=="6010", MODE="664", GROUP="plugdev"  
  
SUBSYSTEM=="tty", ATTRS{idVendor}=="0403",  
ATTRS{idProduct}=="6010", MODE="664", GROUP="plugdev"
```

- 4: 断开调试器再重新连接到 Linux 系统中。
- 5: 使用 `ls /dev/ttyUSB*` 命令查看 `ttyUSB` 信息，参考输出如下：`/dev/ttyUSB0 /dev/ttyUSB1`
- 6: 使用 `ls -l /dev/ttyUSB1` 命令查看分组信息，参考输出如下：`crw-rw-r-- 1 root plugdev 188, 1 Nov 28 12:53 /dev/ttyUSB1`

可以看到 `ttyUSB1` 已经加入 `plugdev` 组，接下来我们要将自己添加到 `plugdev` 组（不同环境可能名字不同，请根据实际情况修改）。使用 `whoami` 命令查看当前用户名，我们将其记录为`<your_user_name>`。

- 7: 使用 `sudo usermod -a -G plugdev <your_user_name>` 命令将自己添加进 `plugdev` 组。
- 8: 再次确认当前用户名已属于 `plugdev` 组，使用 `groups` 命令，可以看到打印信息中有 `plugdev` 即成功将当前用户添加至 `plugdev` 组。如果没有可以尝试重启。

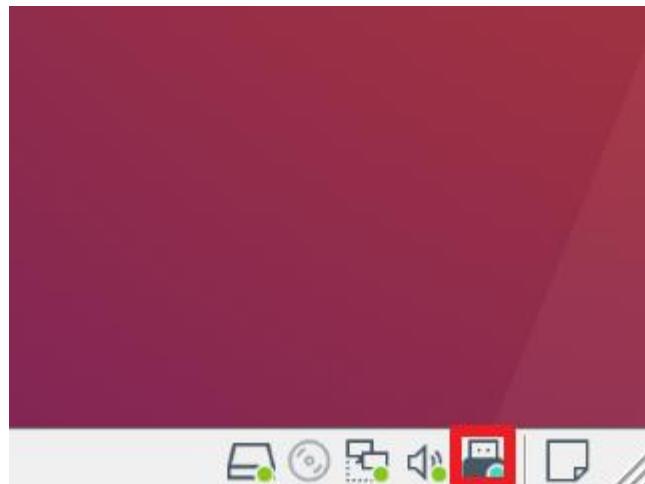


图 6-2 开发板连接至虚拟机中

6.1.2. 新建并配置 Debug Configuration

通过 Nuclei Studio 新建并配置 Debug Configuration 内容的步骤如下。

- 如图 6-3 所示, 在菜单栏中选择 “Run—>Debug Configurations” 。
- 如图 6-4 所示, 在弹出的窗口中, 如果没有当前工程的调试设置内容, 右键单击 “GDB OpenOCD Debugging” , 选择 “New” , 将会为本项目新建出一个调试项目 “hello_world_demo Debug” , 如图 6-5 所示。确保 “Project” 是当前需要调试的工程, “C/C++ Application” 中选择了正确的需要调试的 ELF 文件。
- 如图 6-6 所示, 选择调试项目 “hello_world_demo Debug” 的 Debugger 菜单, 在 Config options 栏目中填入-f
`"nuclei_sdk/SoC/hbird/Board/hbird_eval/openocd_hbird.cfg"`, 以确保 OpenOCD 使用正确的配置文件。这里的配置文件
`(nuclei_sdk/SoC/hbird/Board/hbird_eval/openocd_hbird.cfg)`根据实际工程中 openocd 的配置文件路径而定。

如果当前内核是 RISC-V 32 位内核, 请确保 Commands 内容包含 `set arch riscv:rv32`

如果当前内核为 64 位, 应确保替换为 `set arch riscv:rv64`

- 如图 6-7 所示，选择调试项目“hello_world_demo Debug”的 Startup 菜单，确保“Debug in RAM”，“Pre-run/Restart reset”，“Set Breakpoint at Main”和“Continue”被勾选。
- 如图 6-7，完成后点击右下方“Apply”保存设置。

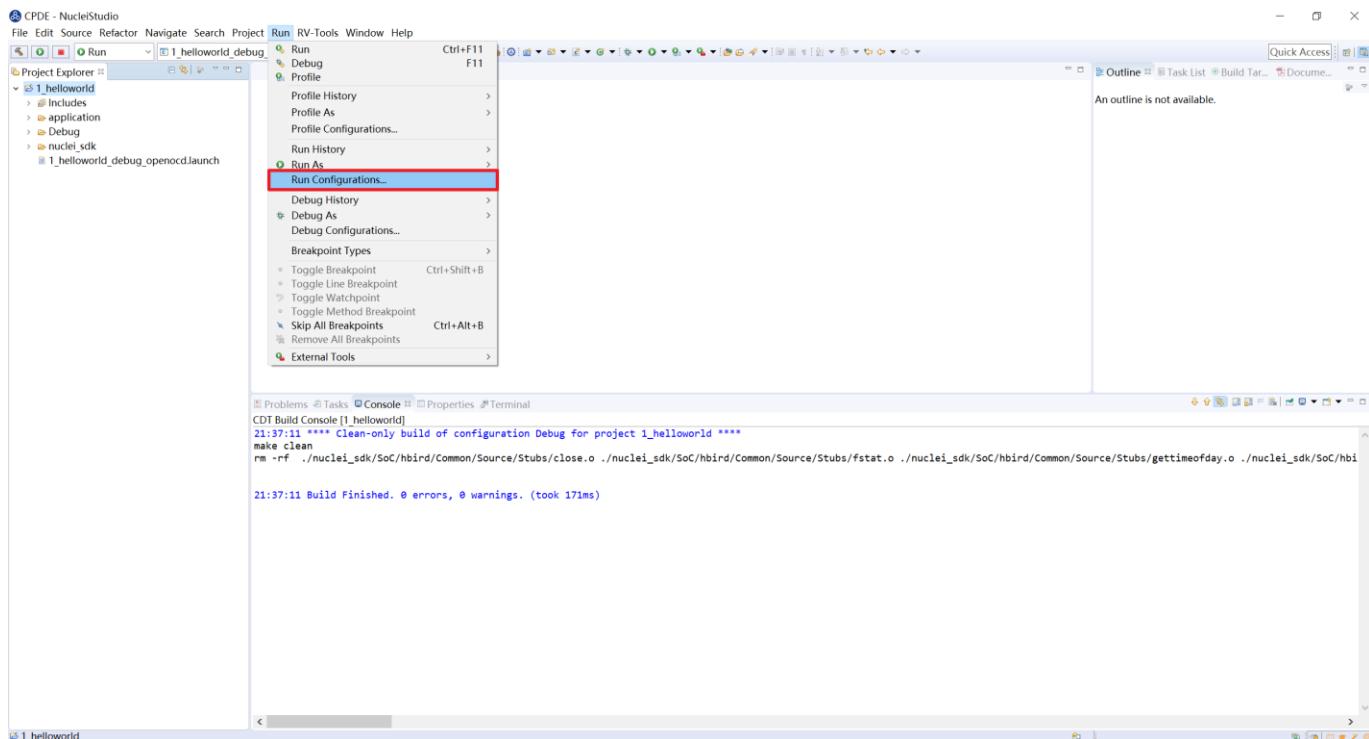


图 6-3 单击“Run Configuration”进行下载

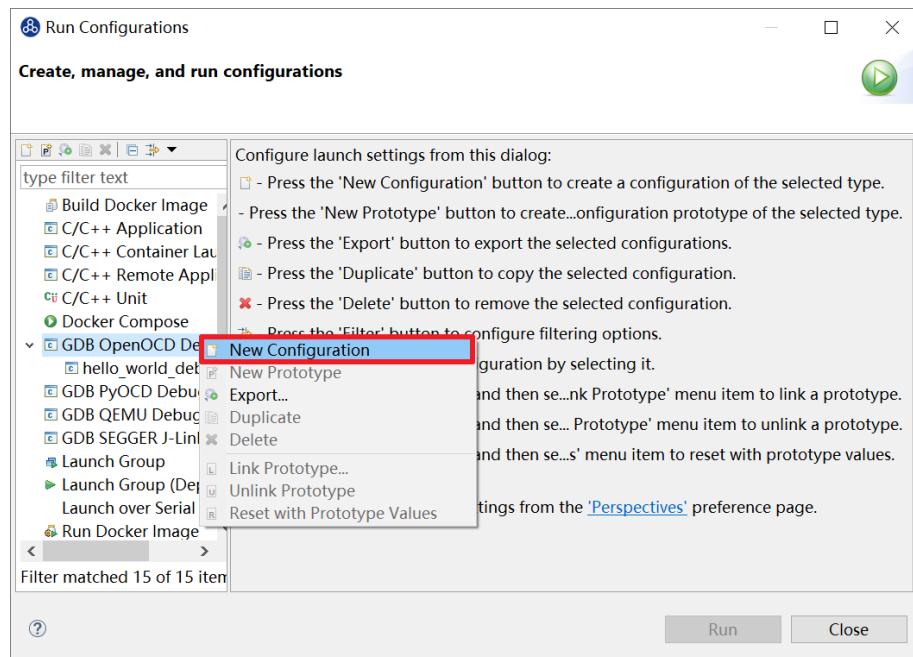


图 6-4 添加新的 GDB OpenOCD Debugging

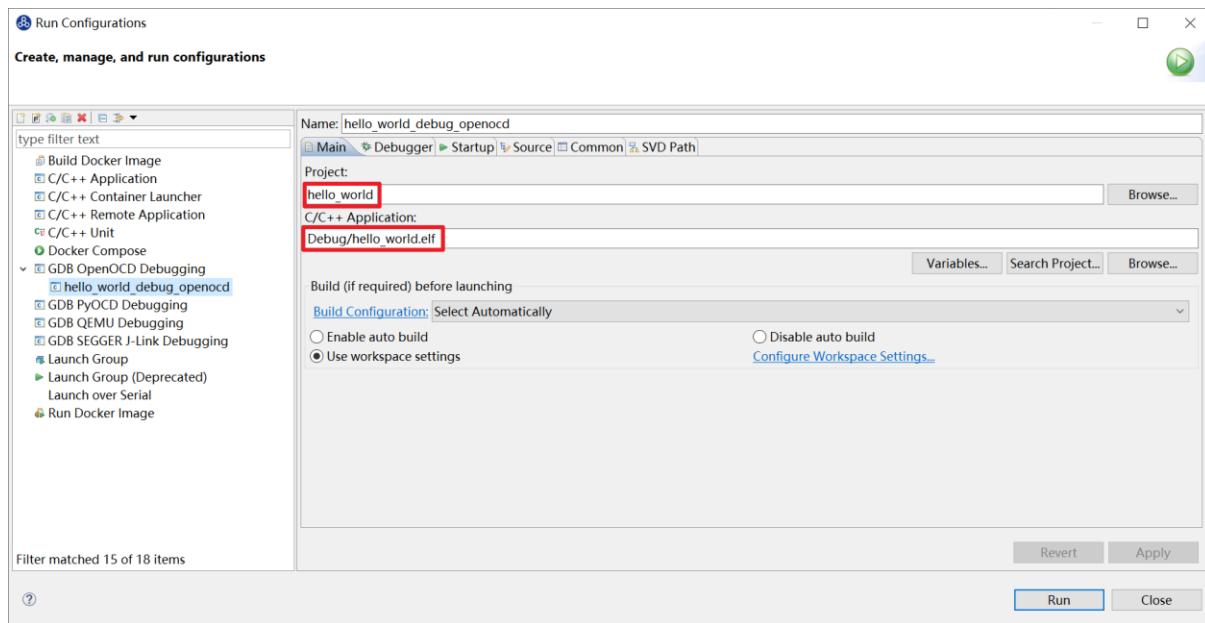


图 6-5 GDB OpenOCD Debugging 栏目下的 hello_world 项目 Debug

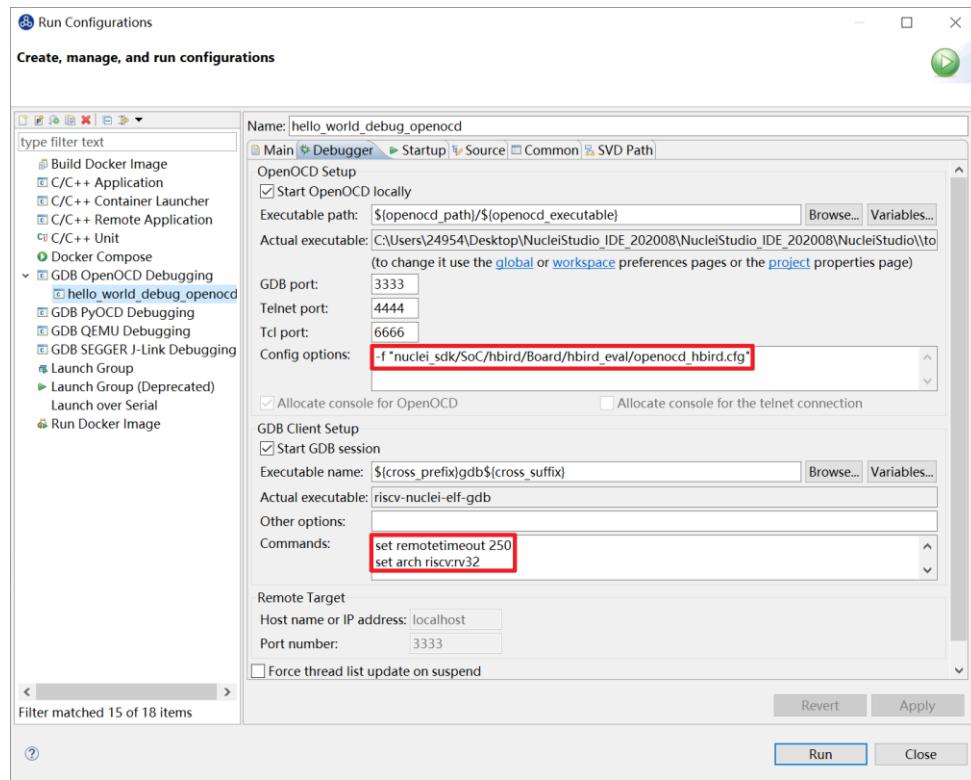


图 6-6 配置 hello_world_demo Debug 的参数 1

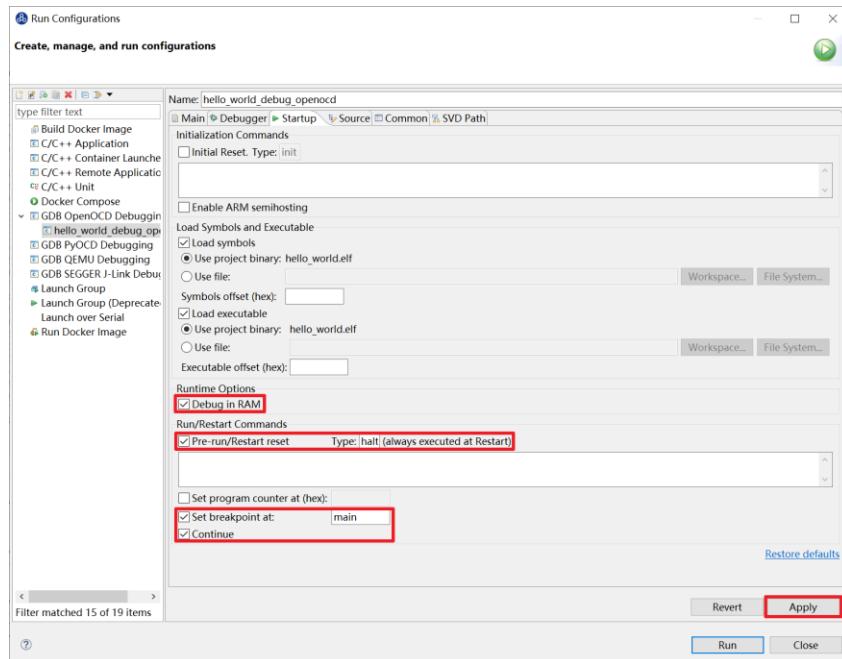


图 6-7 配置 hello_world_demo Debug 的参数 2

6.1.3. 在原型开发板上调试程序

在开发板上调试之前，需要打开串口以便观察 `Printf` 函数打印信息。

使用 Windows 系统打开串口的方法如下：

- 打开 Nuclei Studio 自带的串口打印通道，选择 Window>Show View>Terminal，如图 6-8 所示，点击显示器图标打开串口设置选项。如图 6-9 所示，在其窗口中设置 Choose terminal（选择串口，即 Serial Terminal）、“串口号”（这里以 COM11 为例）、“波特率（设置为 115200）”等参数后，单击“OK”按钮。

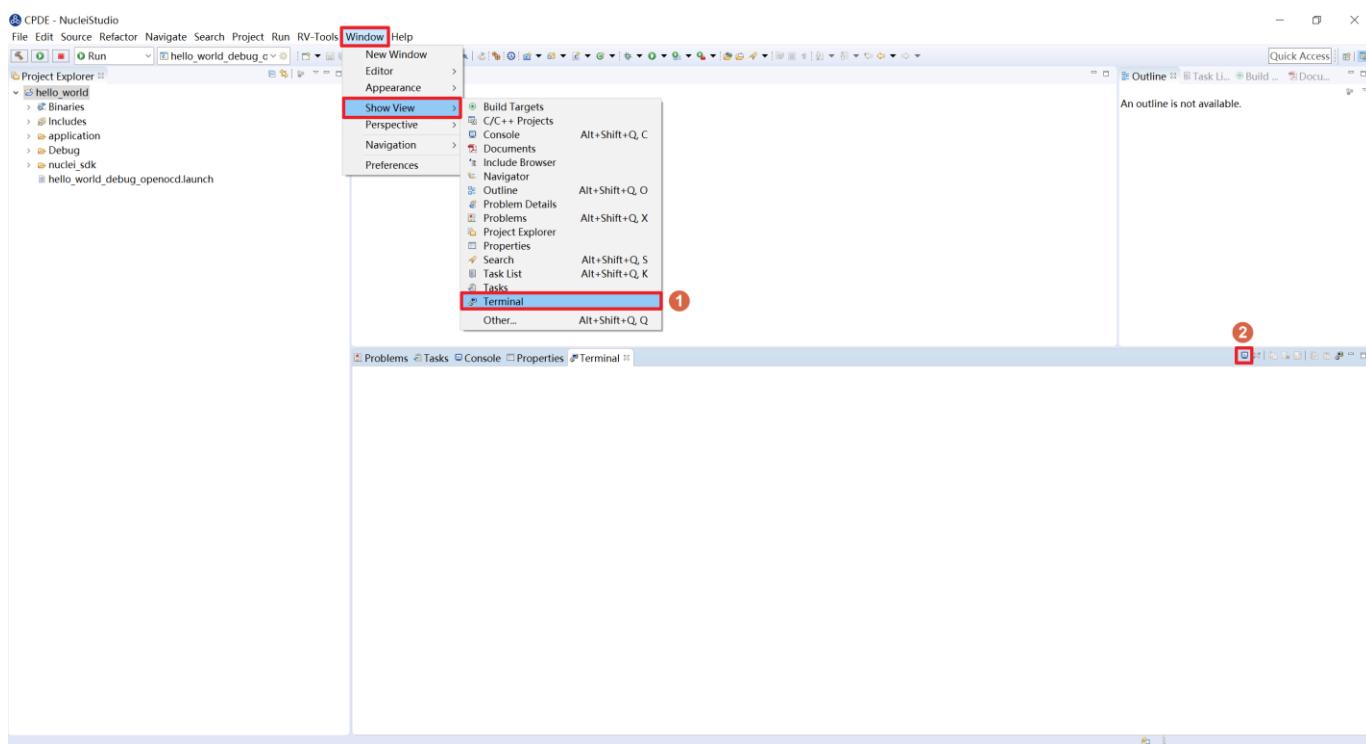


图 6-8 通过 Nuclei Studio 打开串口调试助手并配置参数

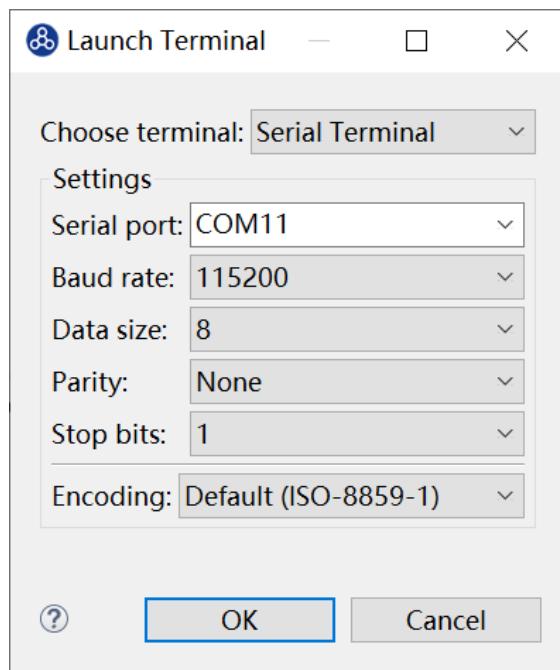


图 6-9 设置串口参数

使用 Linux 系统打开串口方法如下：

- 打开 Nuclei Studio 自带的 Terminal 终端，选择 Window>Show View>Terminal，如图 6-8 所示，点击显示器图标打开串口设置选项。如图 6-10 所示，choose terminal 选择 Local Terminal，点击 OK 打开 Terminal 终端。如图 6-11 所示，输入“minicom /dev/ttyUSB1 115200”打开串口，即可在 Nuclei Studio 中查看串口打印信息。

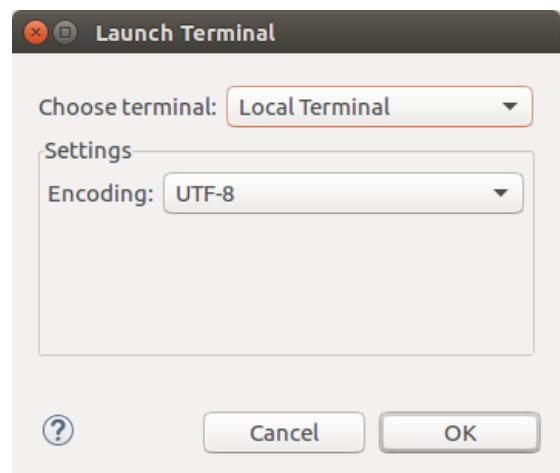


图 6-10 打开 Terminal 终端



```
hbird (ubuntu) ~
Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Nov 15 2018, 20:18:47.
Port /dev/ttyUSB1, 14:38:05

Press CTRL-A Z for help on special keys

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7 | VT102 | Offline | ttyUSB1
```

图 6-11 打开串口

如果程序员希望能够调试运行于原型开发板中程序，可以使用 Nuclei Studio IDE 进行调试。由于 IDE 运行于主机 PC 端，而程序运行于原型开发板上，因此这种调试也称为“在线调试”或者“远程调试”。

这里以 1_helloworld 为例，使用 Nuclei Studio IDE 对 HummingBird SoC 原型开发板进行在线调试的步骤如下。

- 确保 6.2 节中的 Debug 设置内容正确，可以打开 Debug 设置选项确认。如图 6-12 所示，在 1_helloworld 工程处右击，选择“Debug As ->Debug Configuration”打开 Debug 设置页面选择之前新建的设置进行检查。
- 确定设置无误后，在图 6-13 的下拉框选中 Debug，之后左侧图标会变为甲虫图标，单击即可进入调试模式并下载程序进入开发板中。

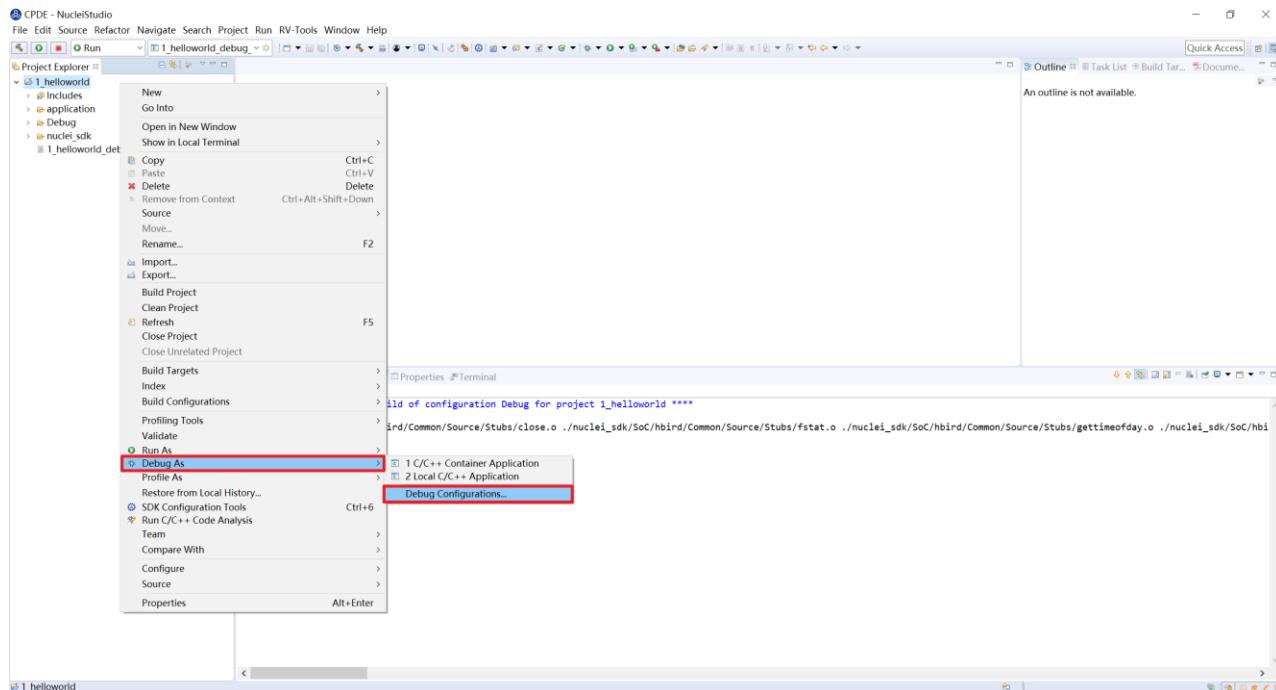


图 6-12 单击“Debug Configuration”进行下载

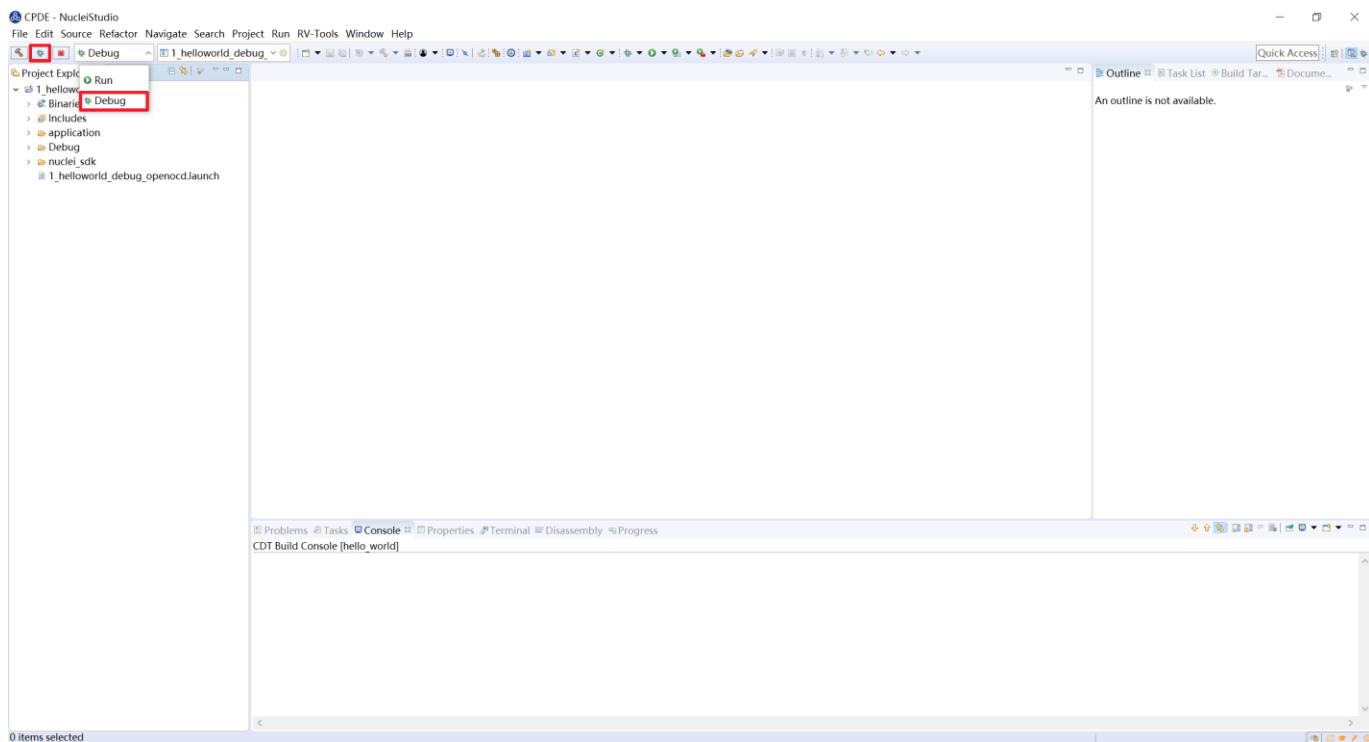


图 6-13 切换至 Debug 模式

- 如果下载成功，则如图 6-14 所示，并且会启动调试界面。

- 如图 6-14 的 1 号标注位置，这里功能包括单步，运行，汇编级调试等。
- 如图 6-14 的 2 号标注位置，这个箭头表示当前程序运行位置。
- 如图 6-14 的 3 号标注位置，在代码的左侧双击即可在该行设置断点，再次双击可以取消断点。
- 如图 6-14 的 4 号标注位置，这里可以切换编辑模式和调试模式。
- 如图 6-14 的 5 号标注位置，这里是函数内变量显示的位置。
- 如图 6-14 的 6 号标注位置，这里是查看寄存器数值的位置。图中显示的是 PC 寄存器当前的数值。
- 如图 6-14 的 7 号标注位置，点击这里红色按钮可以退出调试模式。
- 如图 6-14 的 8 号标注位置的下方，这里可以使用 GDB 控制台指令进行调试。

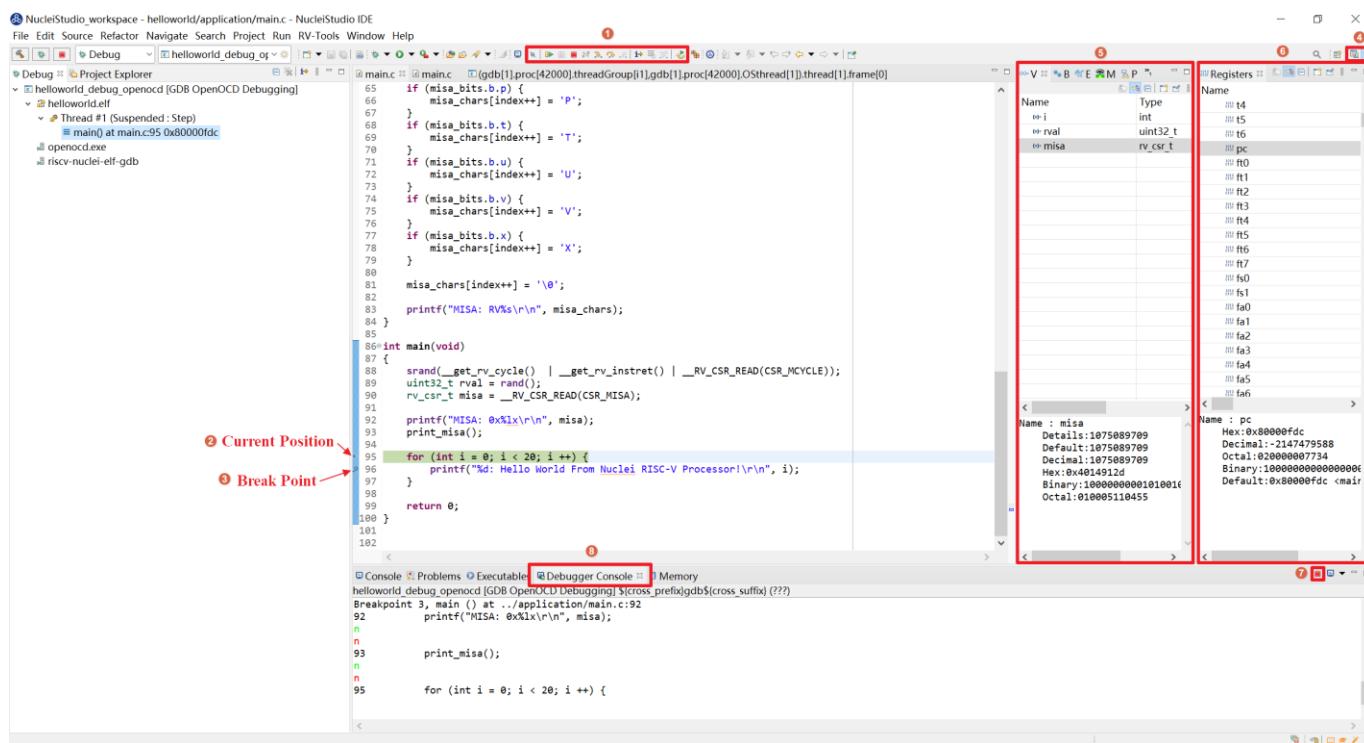


图 6-14 下载完成后进入调试界面

6.1.4. 下载运行程序

调试程序没有出现问题后，可以将程序下载进开发板。如图 6-15，点击下拉框切换至运行模式，此时左侧图标会切换为绿色运行按键，单击即可将程序下载至开发板并运行。由于调试和下载运行使用相同的设置文件，所以不需要再次设置。如图 6-16，可以看到串口正确打印出 helloworld 等信息。需要断开连接，如图 6-17，在 console 栏目下点击红色按钮断开连接。

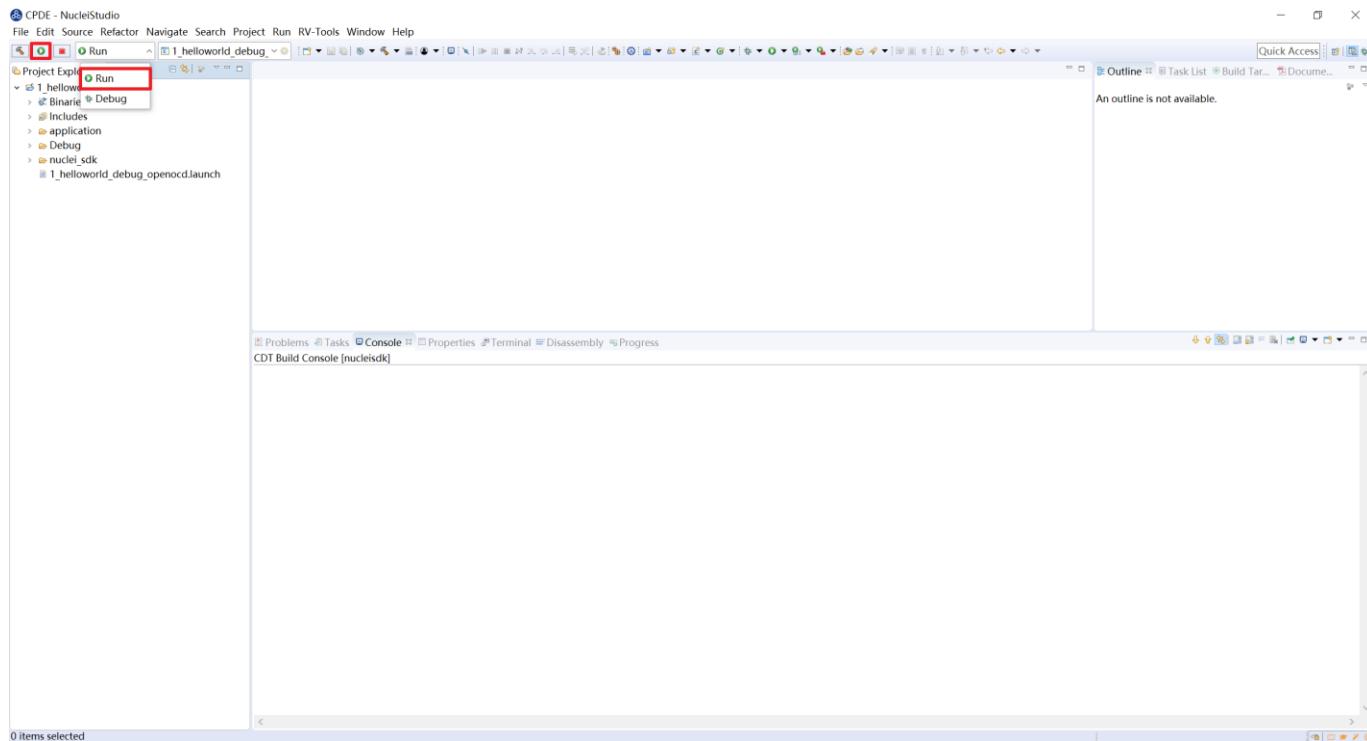


图 6-15 下载程序到开发板中运行

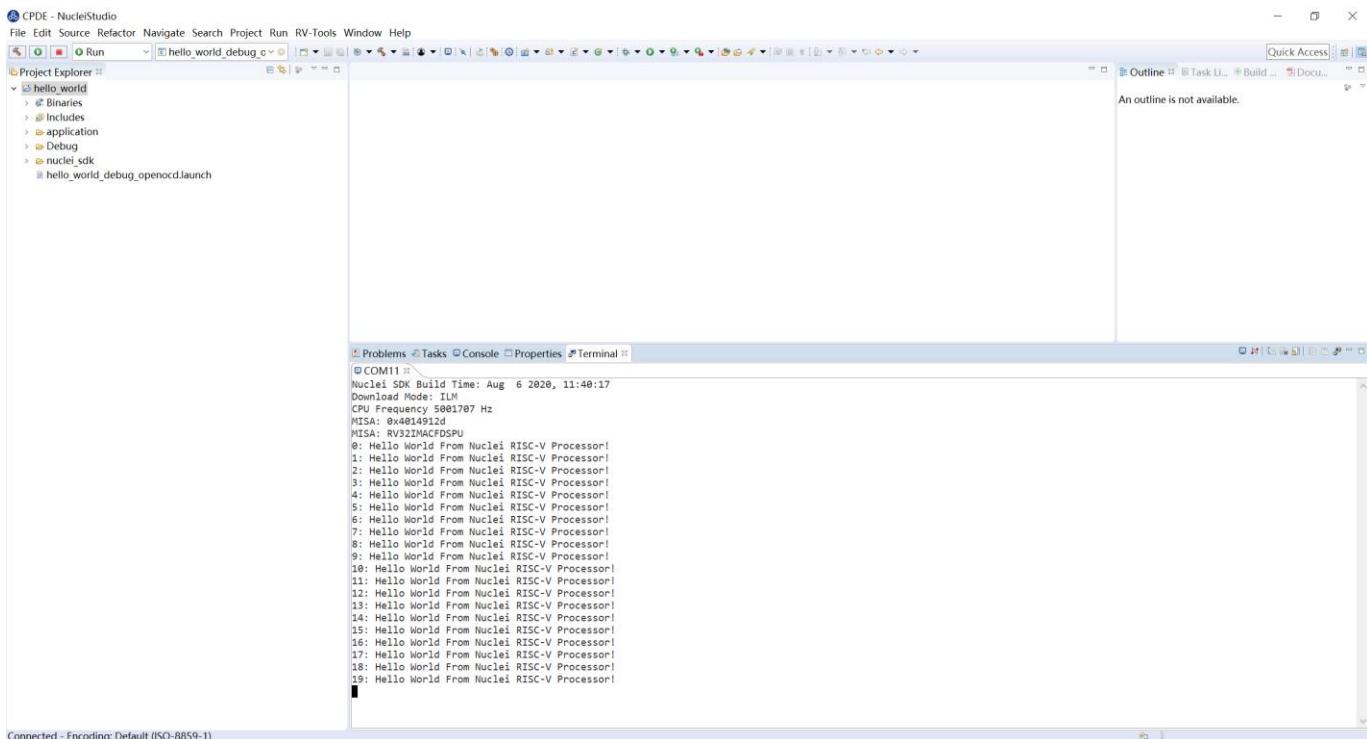


图 6-16 串口调试助手界面输出的 Hello World

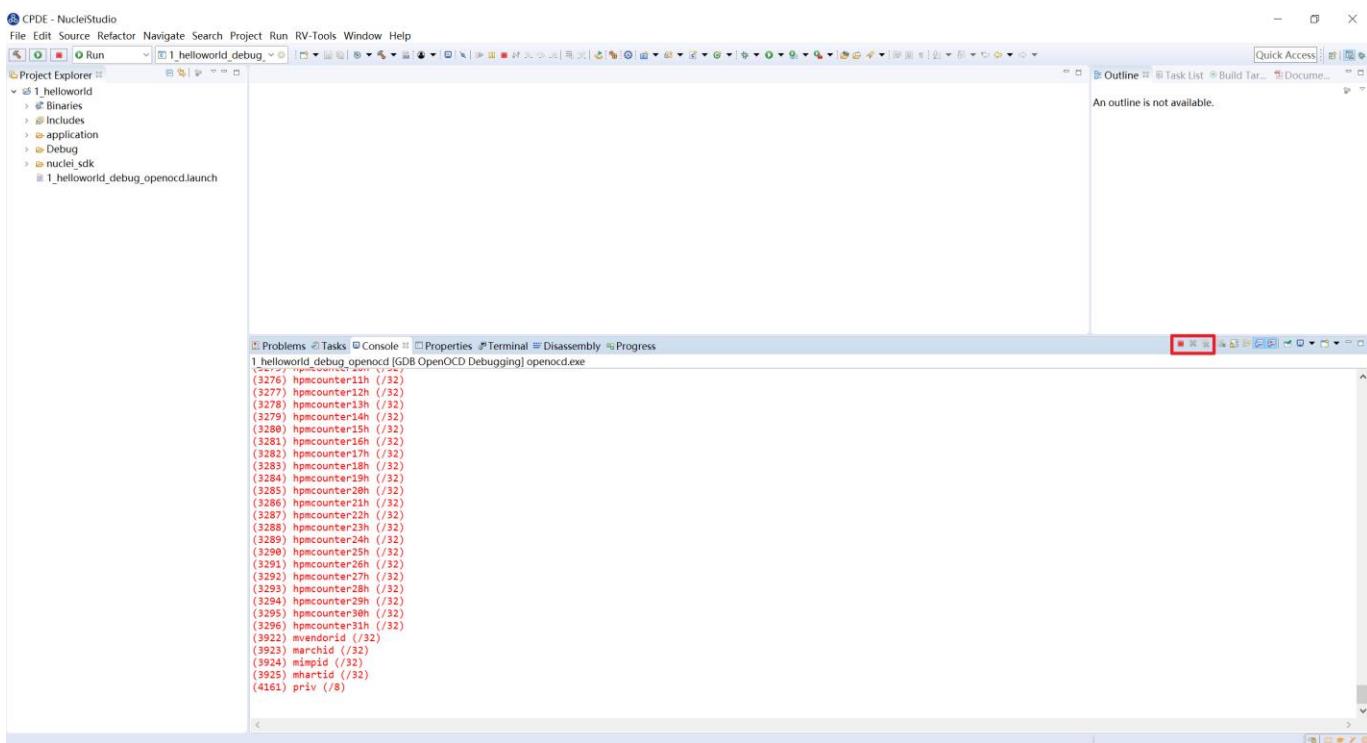


图 6-17 下载完成单击红色按钮断开连接

6.2. 使用 J-Link 调试运行项目

6.2.1. 安装 J-Link 驱动并导入 RTT 文件

HummingBird Evaluation Board 也支持使用 J-Link 调试。前往 SEGGER 官网 J-Link 页面 (<https://www.segger.com/downloads/jlink/#J-LinkSoftwareAndDocumentationPack>)，根据自己的操作系统下载最新的 J-Link 驱动并安装。注意，J-Link 的版本必须高于 v6.62 版本。

如果使用串口进行打印输出，则可以略过本节后续内容。如果想使用 J-Link 的 RTT 打印输出，请按照以下步骤配置。

打开当前工程的设置页面，如图 6-18，在“Resource”选项点击红框标注的图标快速打开工程所在的目录。如图 6-19，在“nuclei_sdk/SoC/hbird/Common/Source/Stubs”路径下新建一个“SEGGER”文件夹，此文件夹用来存放 RTT 相关文件。

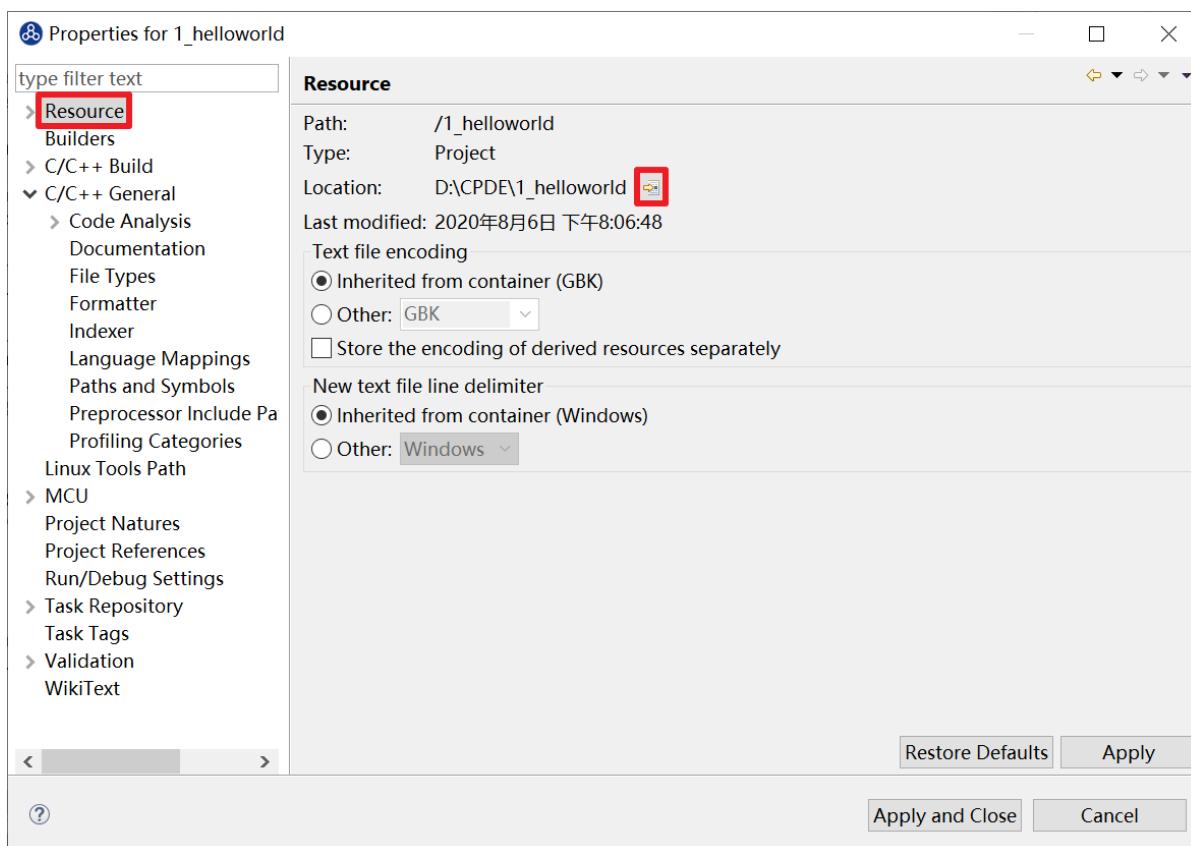


图 6-18 打开工程所在目录

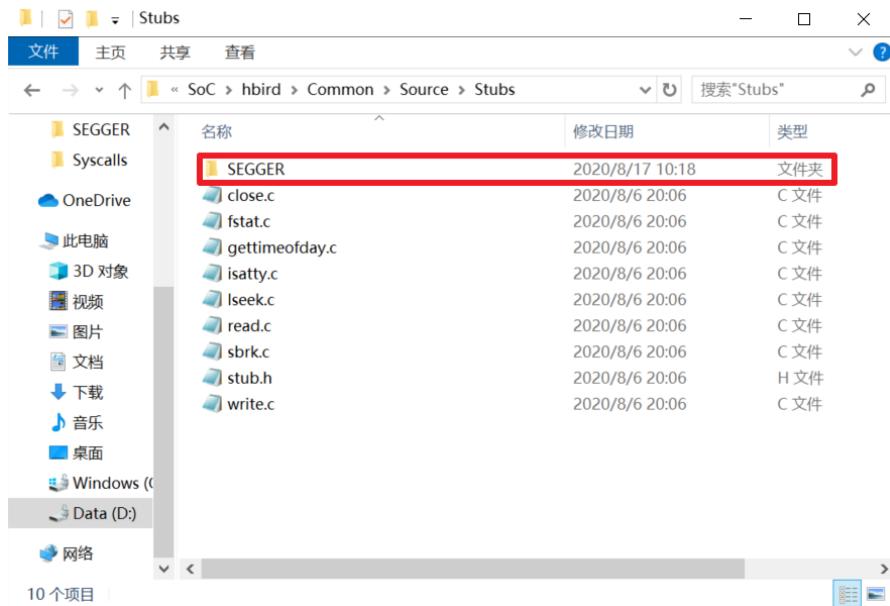


图 6-19 新建“SEGGER”文件夹

安装完成后打开 J-Link 驱动的根目录，将“Samples -> RTT”路径下的“SEGGER_RTT_V68od.zip”解压缩（具体压缩包名可能因版本不同而变化）。解压缩后文件内容如图 6-20，将 RTT 文件夹下的“SEGGER_RTT.c”，“SEGGER_RTT.h”和“SEGGER_RTT_Conf.h”三个文件以及 Syscalls 文件夹下的“SEGGER_RTT_Syscalls_GCC.c”这些文件复制到之前新建的 SEGGER 文件夹中，最后如图 6-21 所示。在 IDE 中打开“SEGGER_RTT_Syscalls_GCC.c”，如图 6-22，注释“#include <reent.h>”所在的这一行。

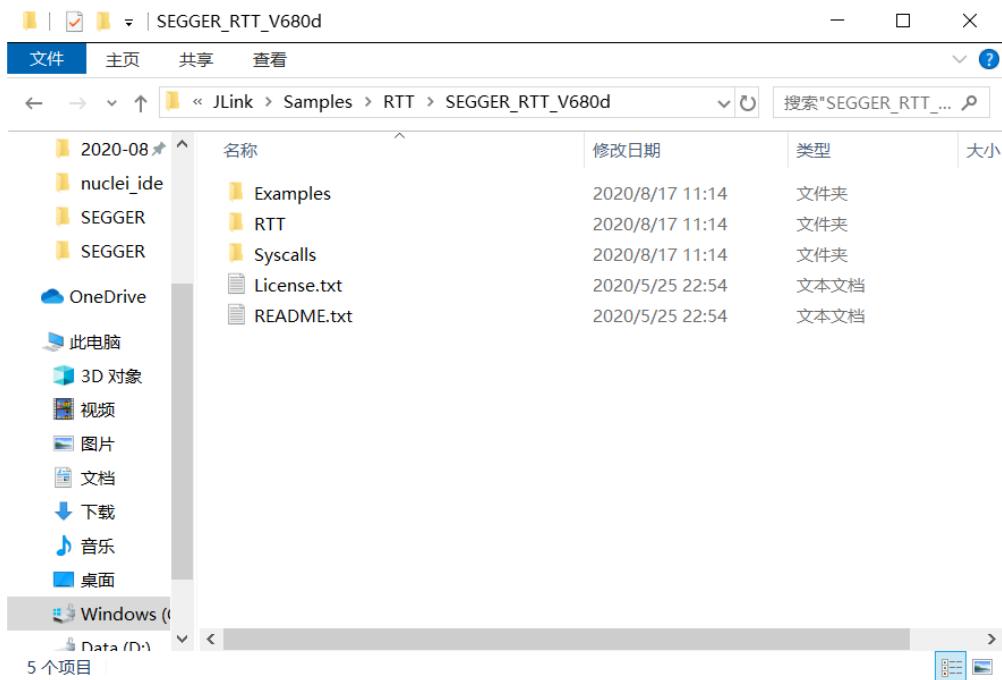


图 6-20 压缩包内文件内容

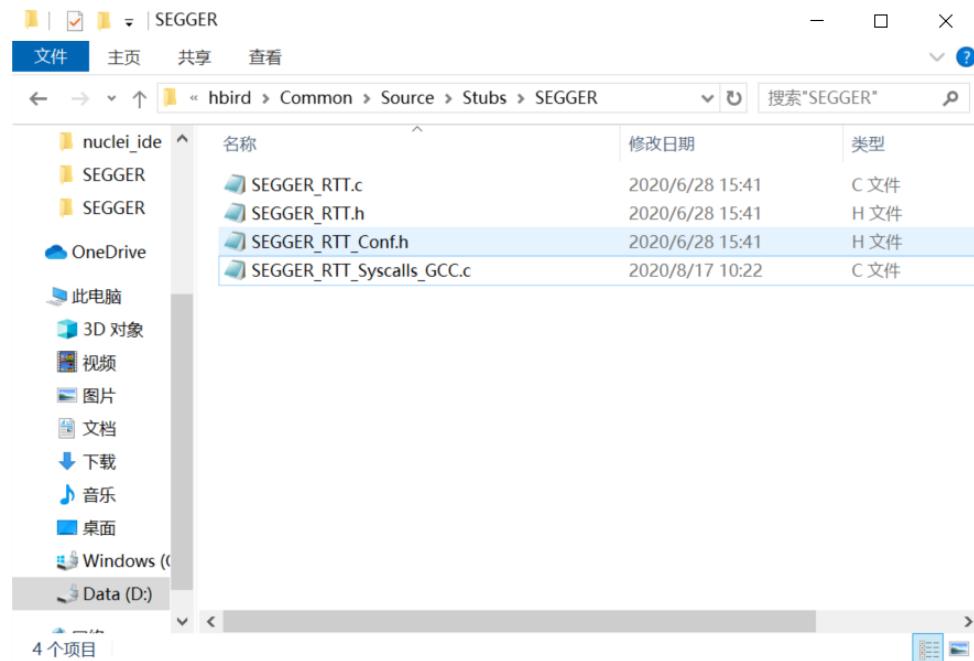
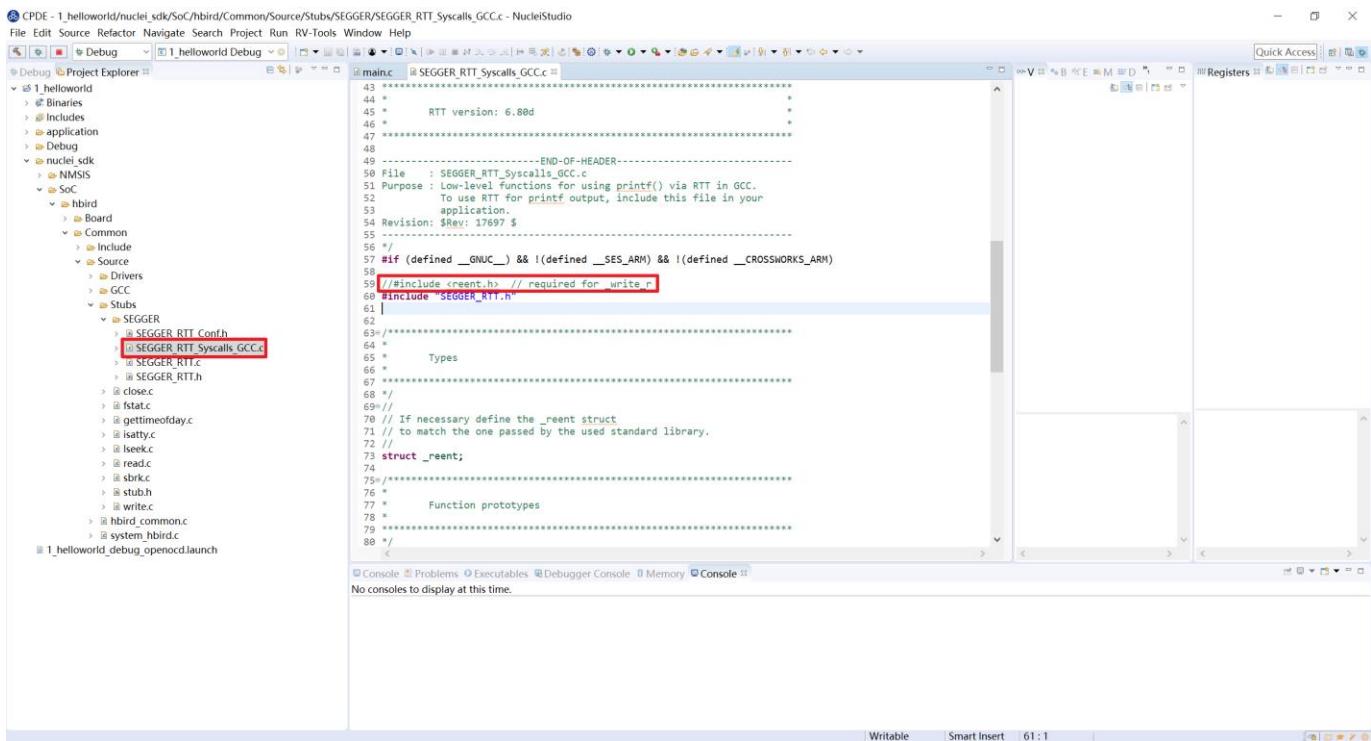


图 6-21 复制后 SEGGER 文件夹内容



```

43 ****SEGGER RTT Syscalls GCC.c ****
44 *
45 *      RTT version 6.88d
46 *
47 ****END-OF-HEADER****
48
49 File    : SEGGER_RTT_Syscalls_GCC.c
50 Purpose : Low-level functions for using printf() via RTT in GCC.
51 To use RTT for printf output, include this file in your
52 application.
53 Revision: $Rev: 17697 $
54
55 ****
56 /*
57 #if (defined __GNUC__) && !(defined __SES_ARM) && !(defined __CROSSWORKS_ARM)
58 //#include <reent.h> // required for write_r
59 #include "SEGGER_RTT.h"
60 */
61
62
63 */
64 *
65 *      Types
66 *
67 ****
68 */
69 /**
70 * If necessary define the _reent struct
71 * to match the one passed by the used standard library.
72 */
73 struct _reent;
74
75 ****
76 *
77 *      Function prototypes
78 *
79 ****
80 */

```

图 6-22 注释不使用的头文件

文件添加完成后添加 SEGGER 文件夹路径至 include，如图 6-23，打开当前工程的设置页面，添加 SEGGER 文件夹路径至 include 中。

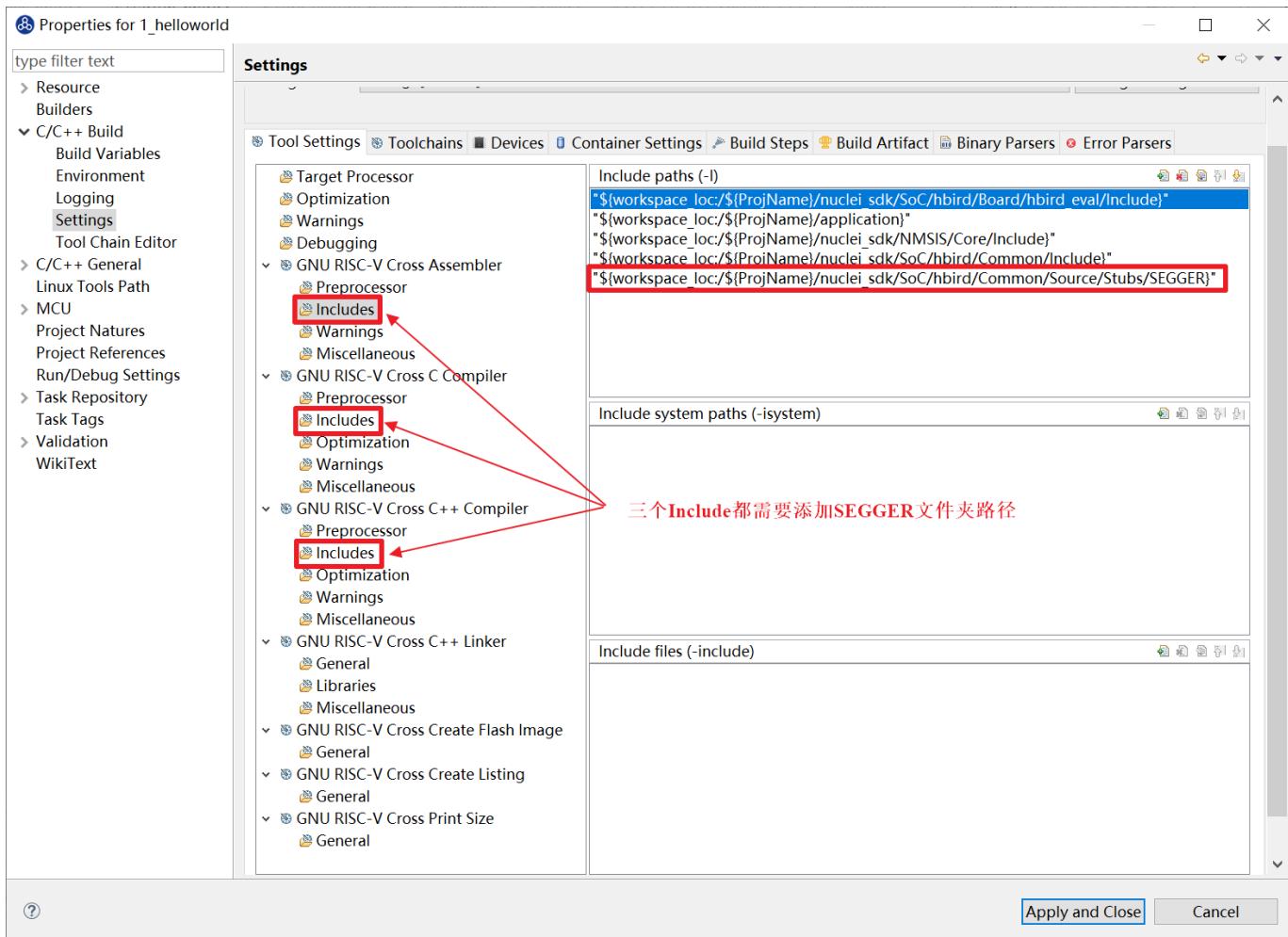


图 6-23 添加 SEGGER 文件夹路径至 Include 中

接下来移除原有的 write 函数。如图 6-24，在“nuclei_sdk/SoC/hbird/Common/Source/Stubs”下的“write.c”文件处右击，选择“Resource Configurations → Exclude from Build”。如图 6-25，选择“Select All”，点击“OK”。以后如果想切换回使用串口打印，可以使用相同的方式移除 SEGGER 文件夹并把“write.c”文件添加回工程。

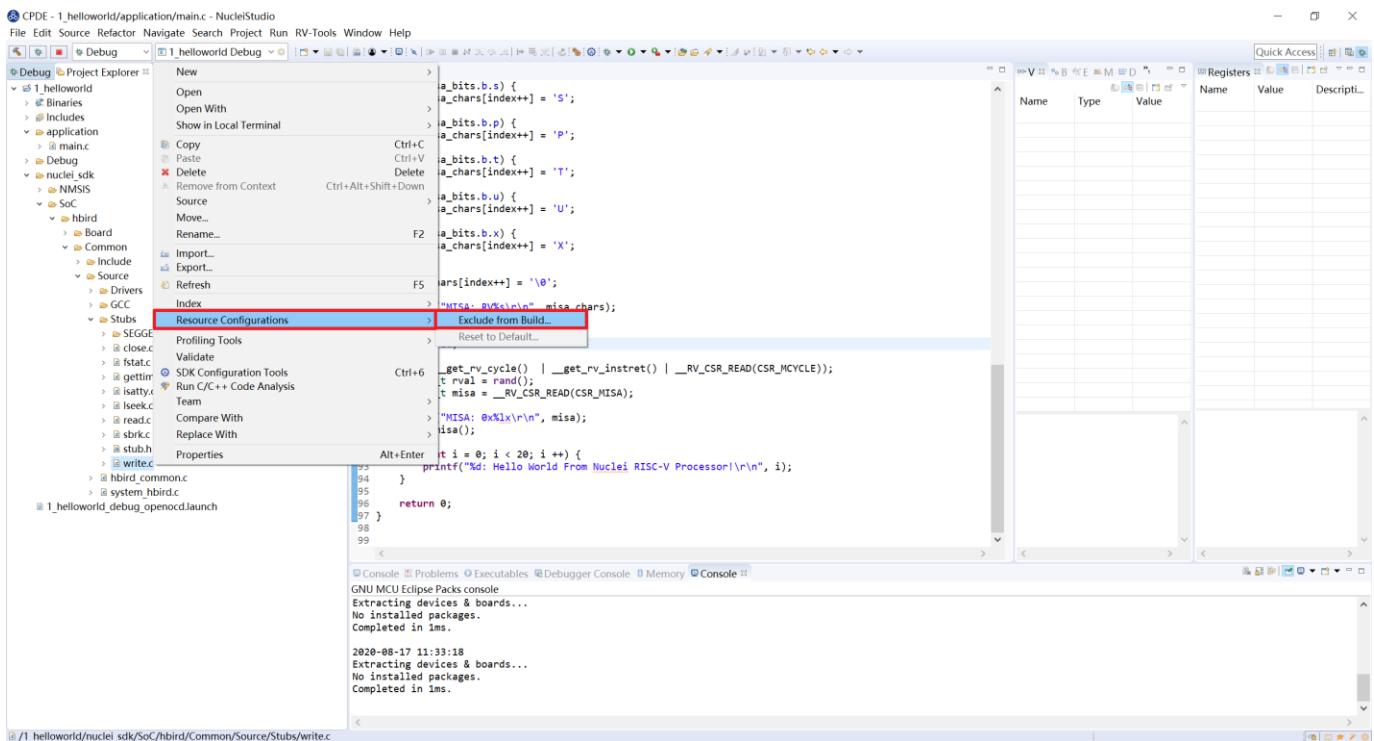


图 6-24 打开 Exclude from Build

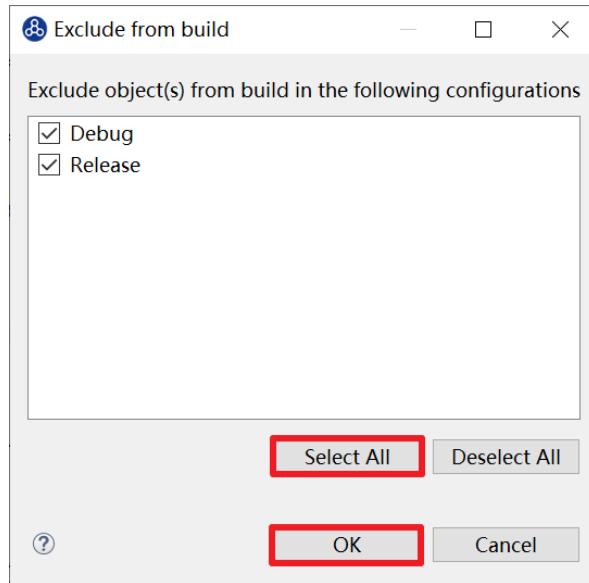


图 6-25 移除 write.c 文件

6.2.2. 新建并配置 Debug Configuration

新建并配置 J-Link 调试下载的 Debug Configuration 步骤如下：

- 在菜单栏中选择“Run—>Debug Configurations”。
- 如图 6-26，在弹出的窗口中，如果没有当前工程的调试设置内容，右键单击“GDB SEGGER J-Link Debugging”，选择“New Configuration”，将会为本项目新建出一个调试项目“1_helloworld Debug”，如图 6-27。确保“Project”是当前需要调试的工程，“C/C++ Application”中选择了正确的需要调试的 ELF 文件。

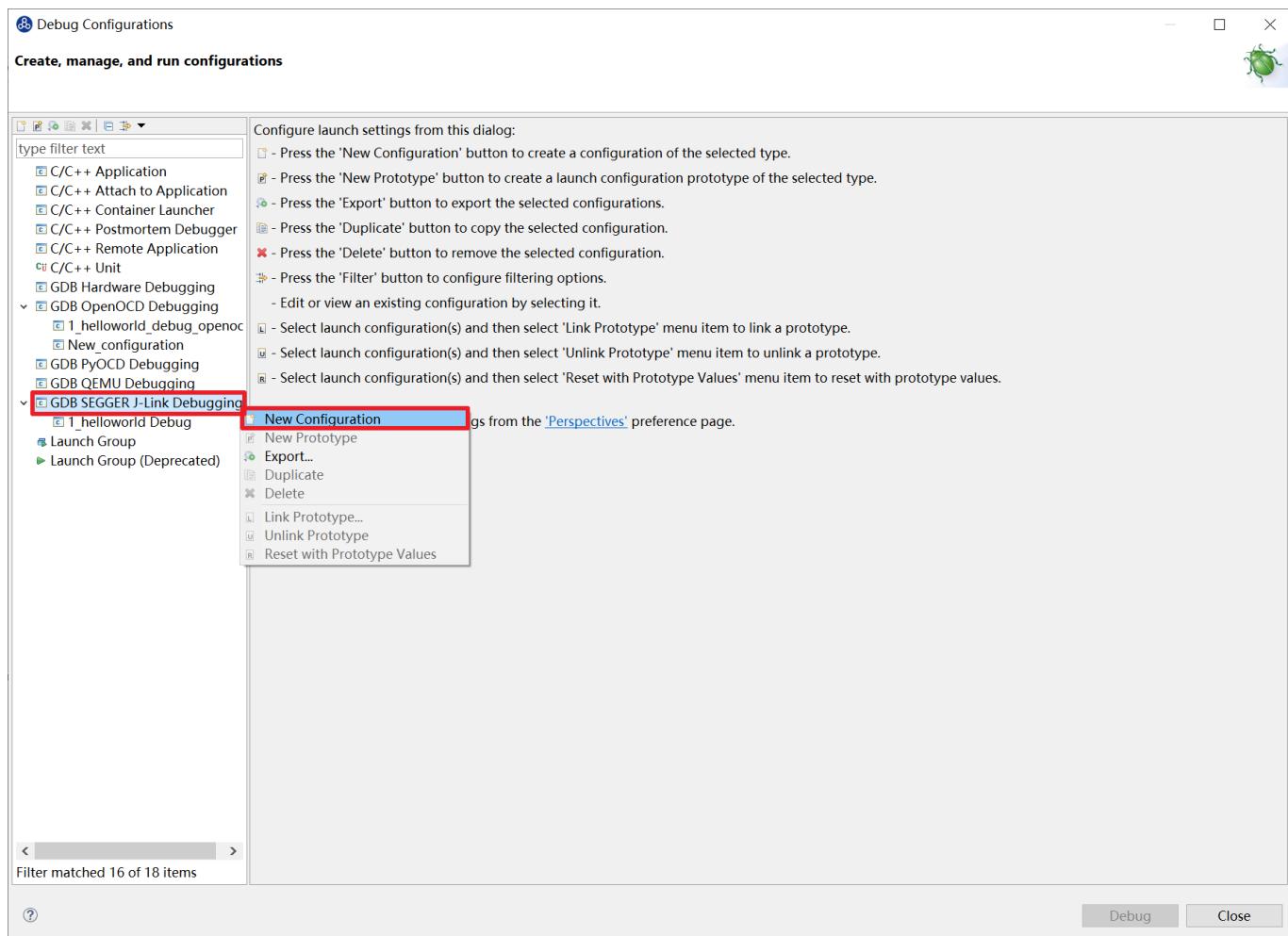


图 6-26 新建 J-Link Debug Configuration

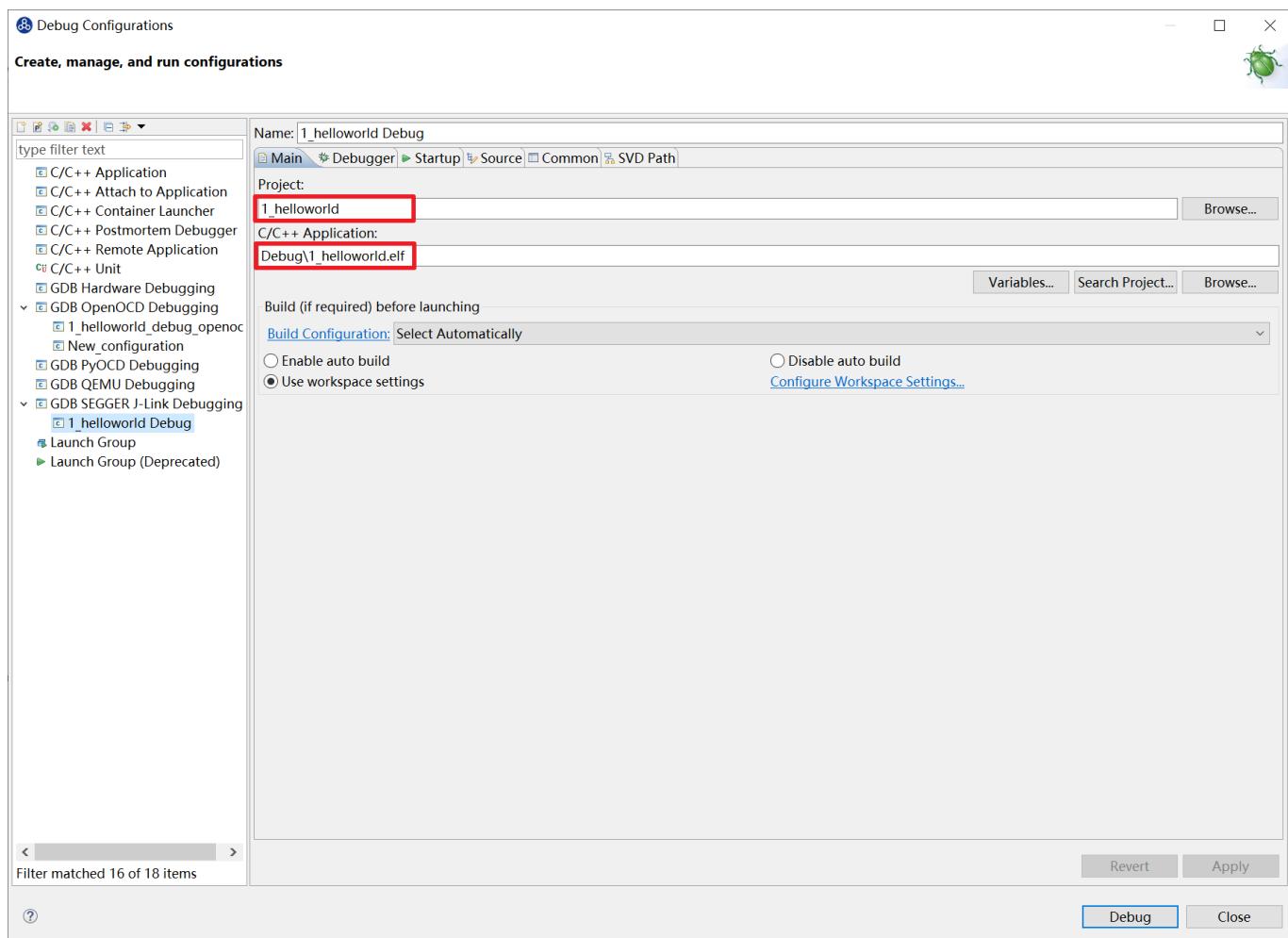


图 6-27 确保 Project 和 C/C++ Application 与工程一致

打开 Debugger 栏目，如图 6-28。

确保 1 号位置 “Start the J-Link GDB server locally” 被选中。

2 号位置正确指向 JLinkGDBServerCL.exe 的路径。

3 号内是当前使用的内核，这里以 N307 为例，输入 N307 即可。如果使用 RV-STAR 开发板，这里输入 GD32VF103VBT6。如果使用其他开发板请参考 J-Link Support Device 网页，链接如下：<https://www.segger.com/downloads/supported-devices.php>

4 号选择 “Interface” 为 JTAG，“initial speed” 为 Auto。

5号确认与使用的GDB设置一致。

如果有修改的内容，点击6号位置“Apply”保存。

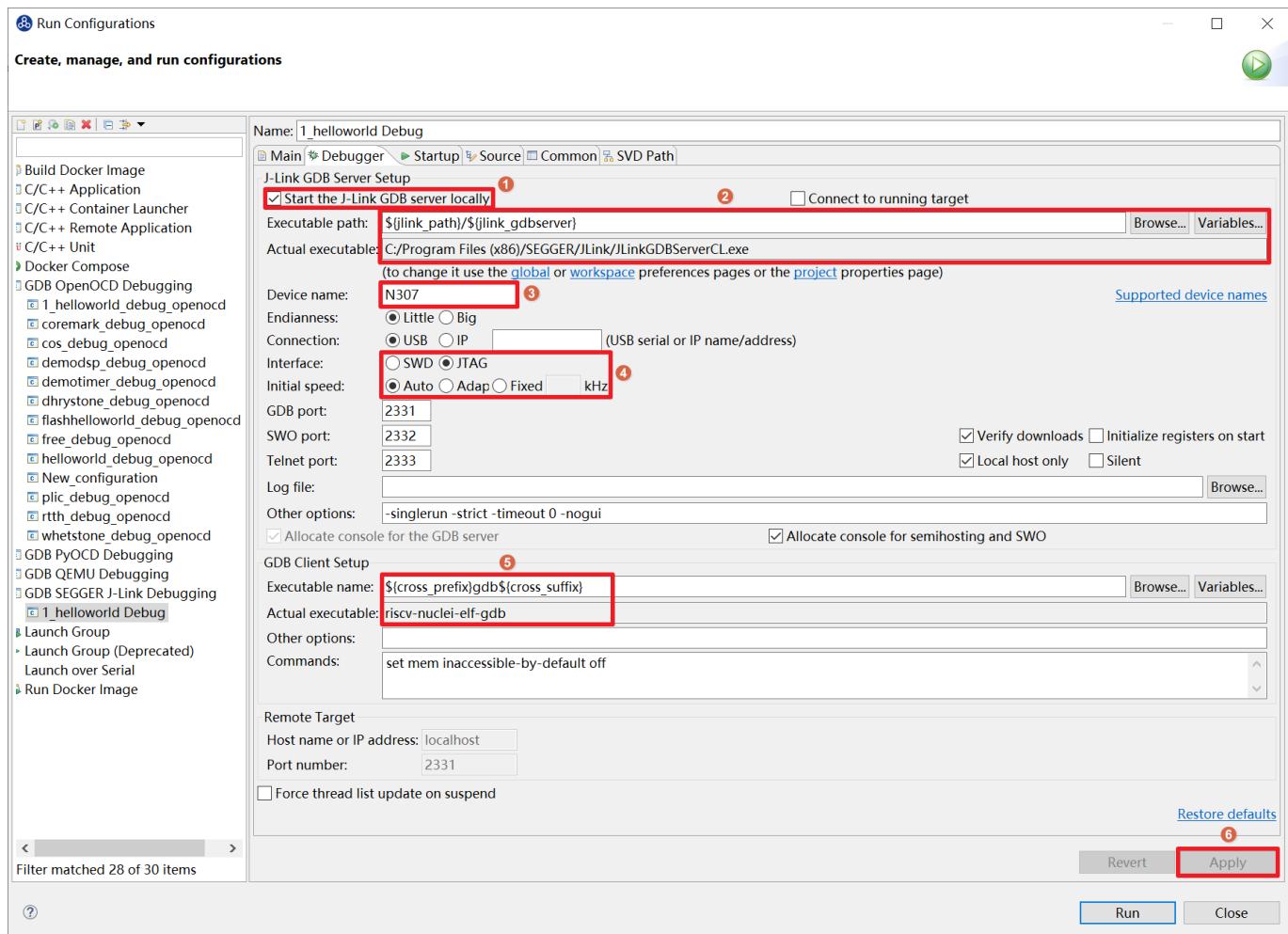


图 6-28 设置 Debugger 栏目内容

打开“Startup”栏目，如图 6-29，确保 JTAG/SWD Speed 为 Auto，“set Breakpoint at main”，“Continue”，“Pre-run/Restart reset”和“RAM application”选项被勾选，并且取消勾选“Initial Reset and Halt”选项。

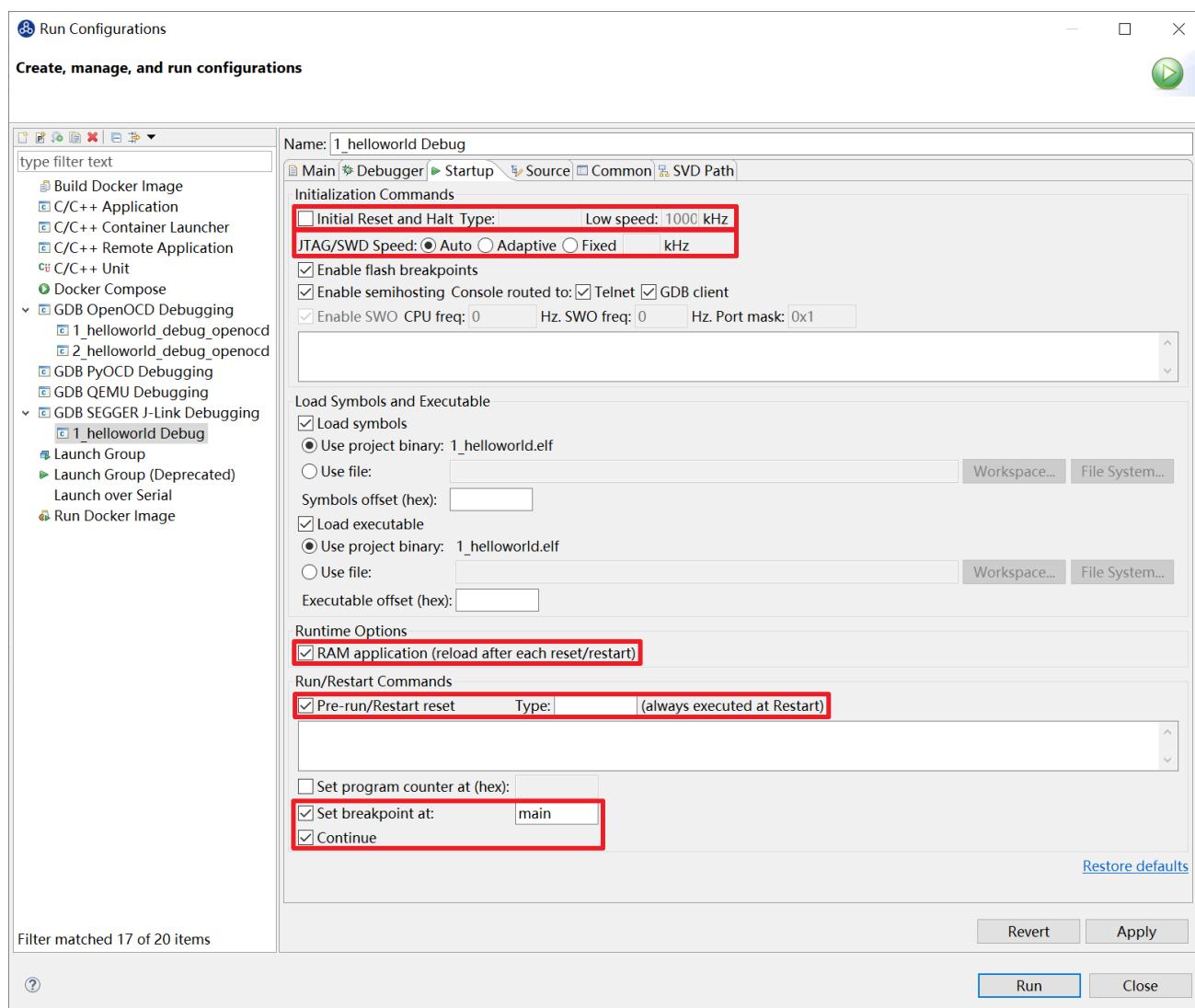


图 6-29 设置 Startup 栏目内容

以上设置内容完成后，如果有变动需要点击右下角 Apply 保存设置，如果没有变动点击 close 即可。

6.2.3. 在原型开发板上调试程序

使用 J-Link 在 HummingBird Evaluation Board 调试需要连接跳线。如图 6-30，红框标注的部分是 J-Link 需要连接到板子的部分。其中 VTref 连接到板子上 V3.3 的接口，其他部分连接到 JTAG 接口，各引脚的丝印就在旁边，一一对应连接即可，最后实物连接如图 6-31。

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

图 6-3 J-Link 引脚图

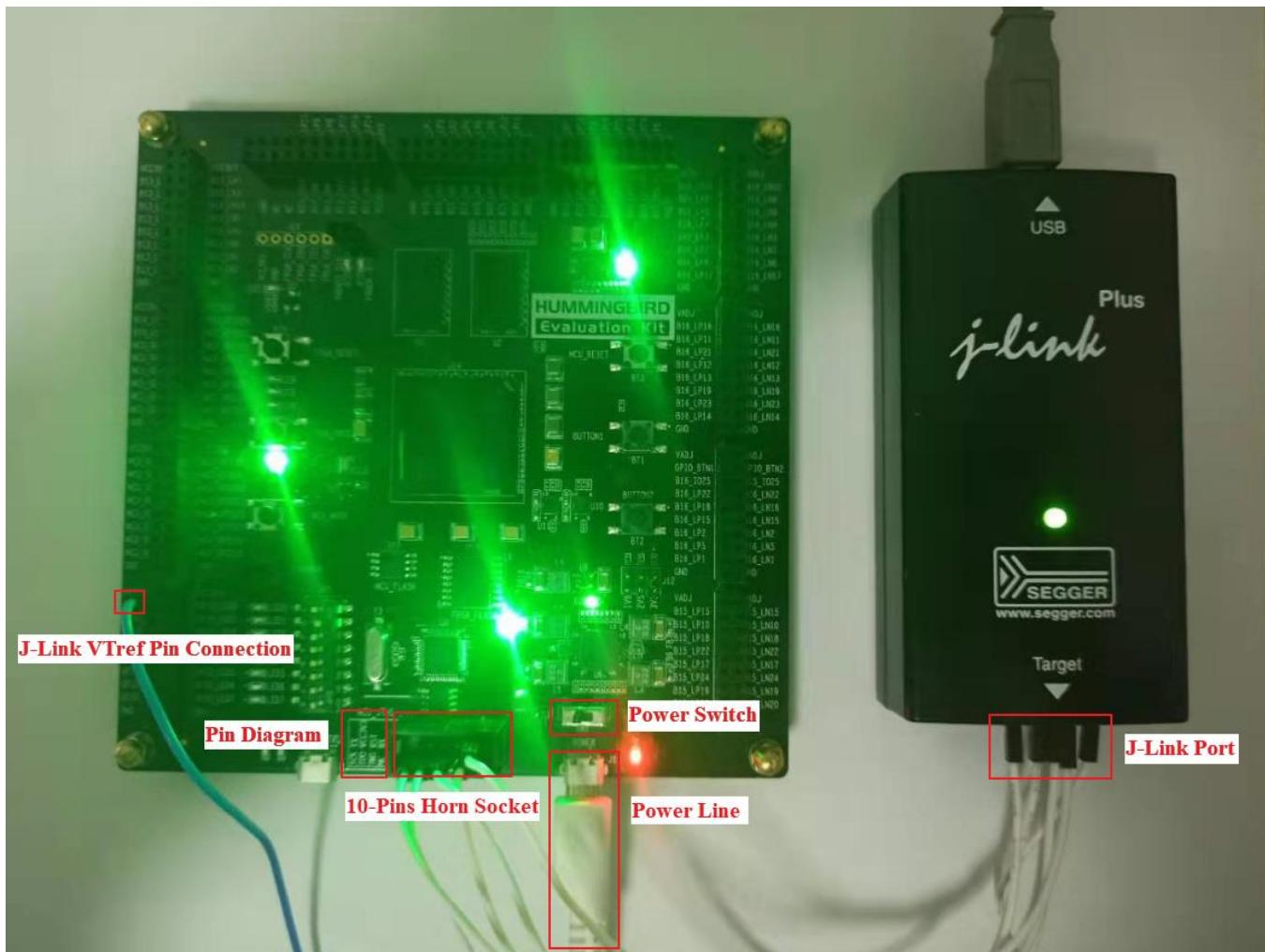


图 6-31 J-Link 实物连接图

在开发板上调试之前，如果使用串口打印，需要连接 JTAG 上的串口引脚到自己的主机上，再打开串口以便观察 `Printf` 函数打印信息。如果使用 RTT 打印，需要打开 J-Link RTT Viewer 查看 `printf` 打印信息。如图 6-32，按照图中内容设置，选择 USB 方式连接。Specify Target Device 根据使用的内核来修改，这里以 N307 为例。Target Interface & Speed 设置为 1000kHz，可根据实际使用情况来修改。RTT Control Block 选择 Auto Detection。

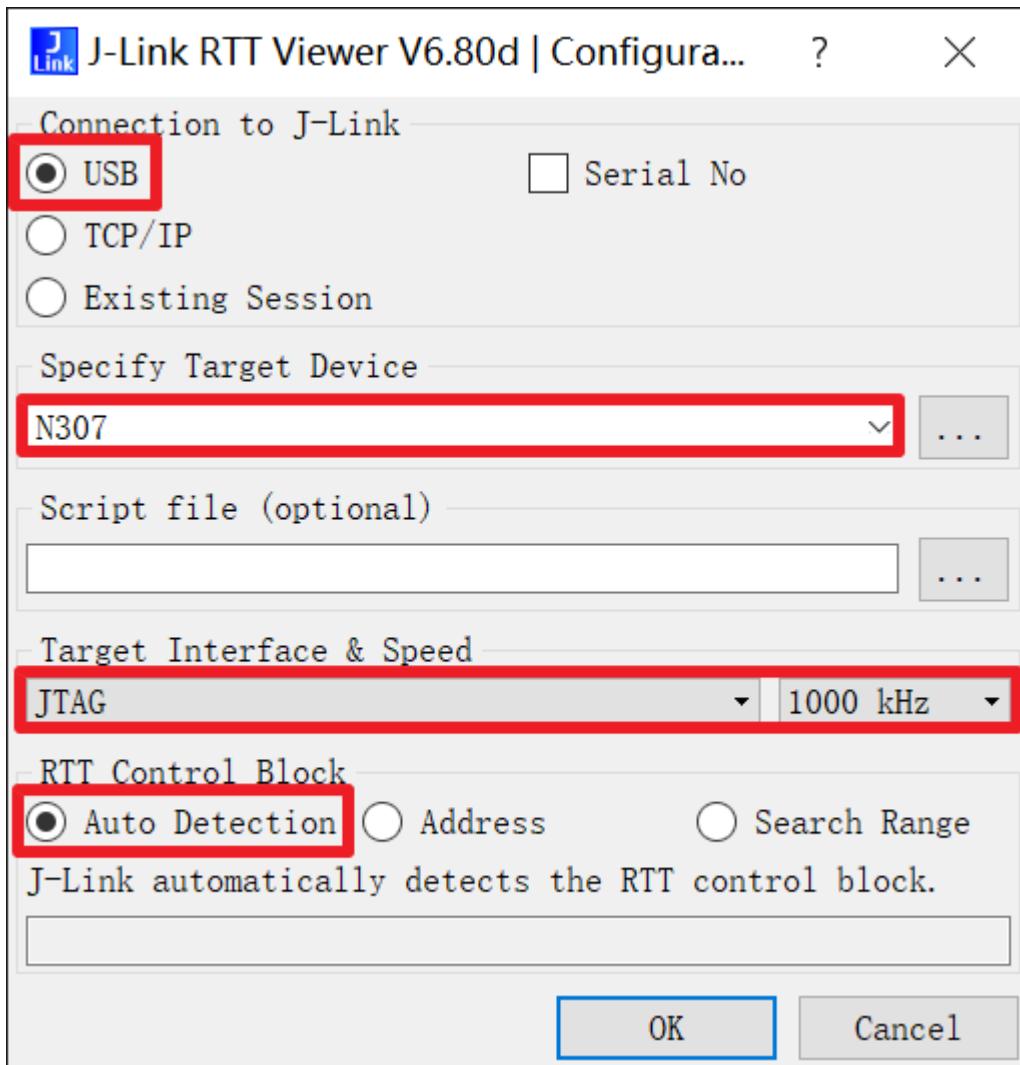


图 6-32 设置 RTT Viewer

图 6-33 中下拉框选中 Debug，之后左侧图标会变为甲虫图标，单击即可进入调试模式并下载程序进入开发板中。

如果下载成功，则如图 6-34 所示，并且会启动调试界面。

- 如图 6-34 的 1 号标注位置，这里功能包括单步，运行，汇编级调试等。
- 如图 6-34 的 2 号标注位置，这个箭头表示当前程序运行位置。
- 如图 6-34 的 3 号标注位置，在代码的左侧双击即可在该行设置断点，再次双击可以取消断点。
- 如图 6-34 的 4 号标注位置，这里可以切换编辑模式和调试模式。
- 如图 6-34 的 5 号标注位置，这里是函数内变量显示的位置。
- 如图 6-34 的 6 号标注位置，这里是查看寄存器数值的位置。
- 如图 6-34 的 7 号标注位置，点击这里红色按钮可以退出调试模式。
- 如图 6-34 的 8 号标注位置下方，这里可以使用 GDB 控制台指令进行调试。

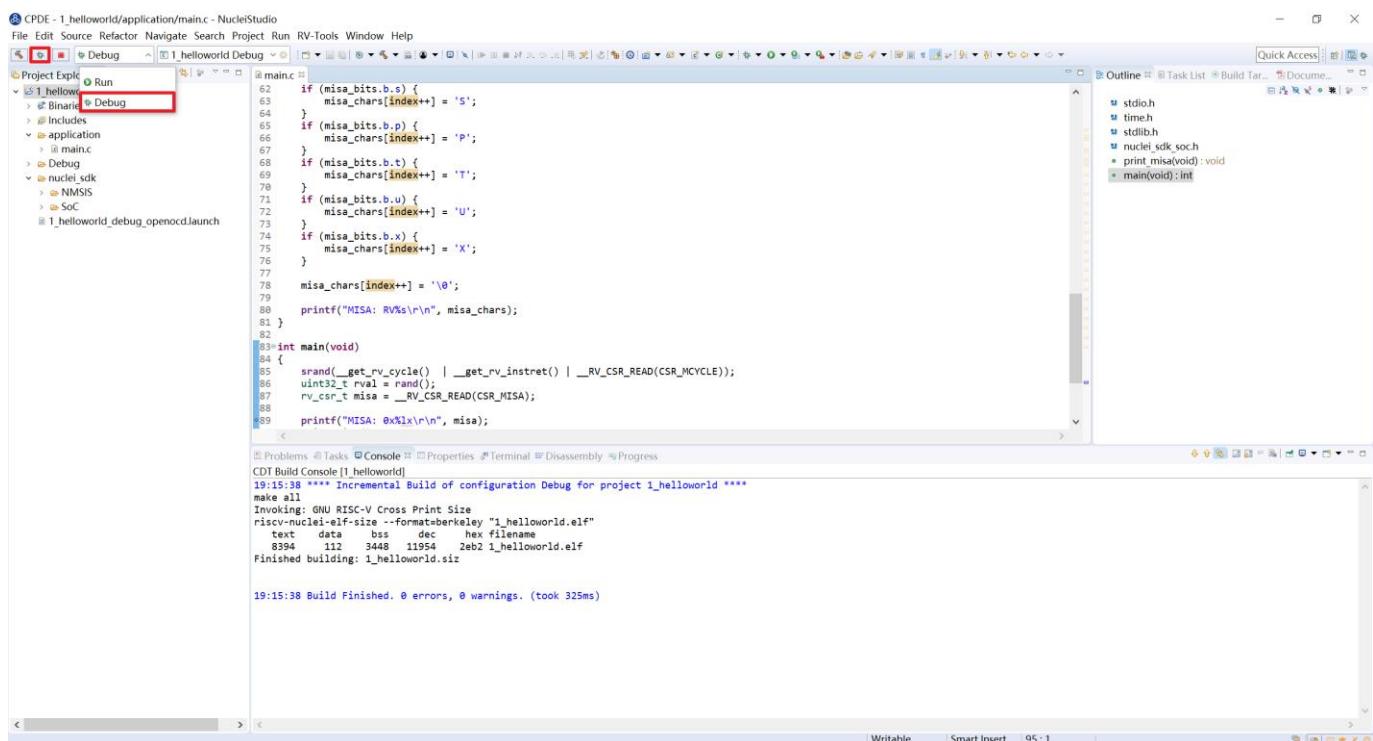


图 6-33 切换至下载模式

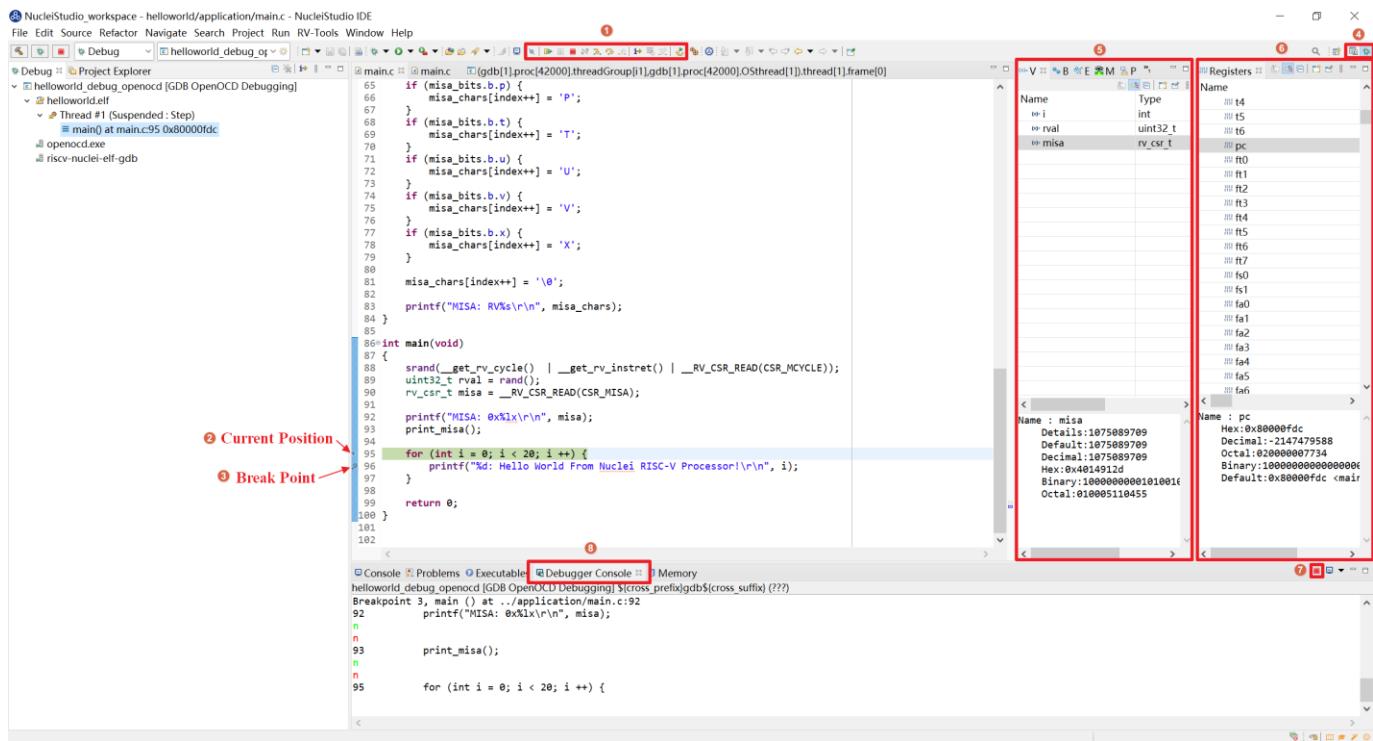


图 6-34 下载完成后进入调试页面

6.2.4. 下载运行程序

调试程序没有出现问题后，可以将程序下载进开发板。如图 6-35，点击下拉框切换至运行模式，此时左侧图标会切换为绿色运行按键，单击即可将程序下载至开发板并运行。由于调试和下载使用相同的设置文件，所以不需要再次设置。如图 6-36，可以看到 RTT Viewer 正确打印出 helloworld 等信息。如果需要断开连接，如图 6-37，在 console 栏目下点击红色按钮即可断开连接。

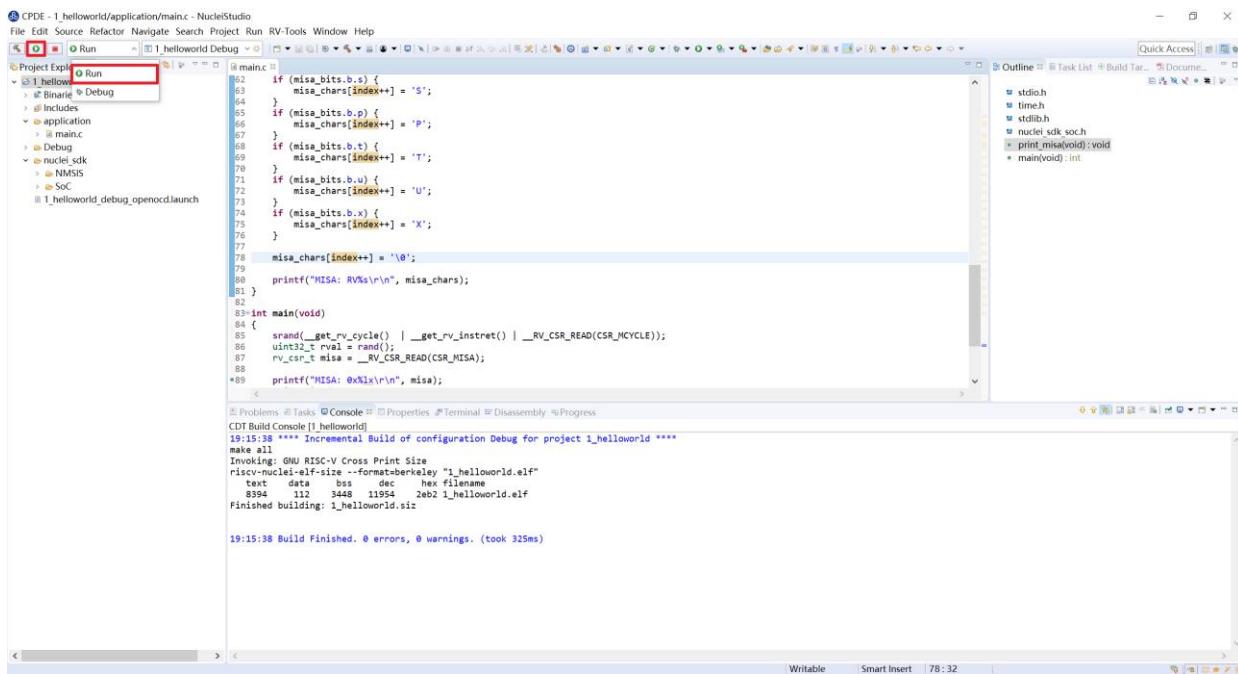


图 6-35 下载程序到开发板中运行

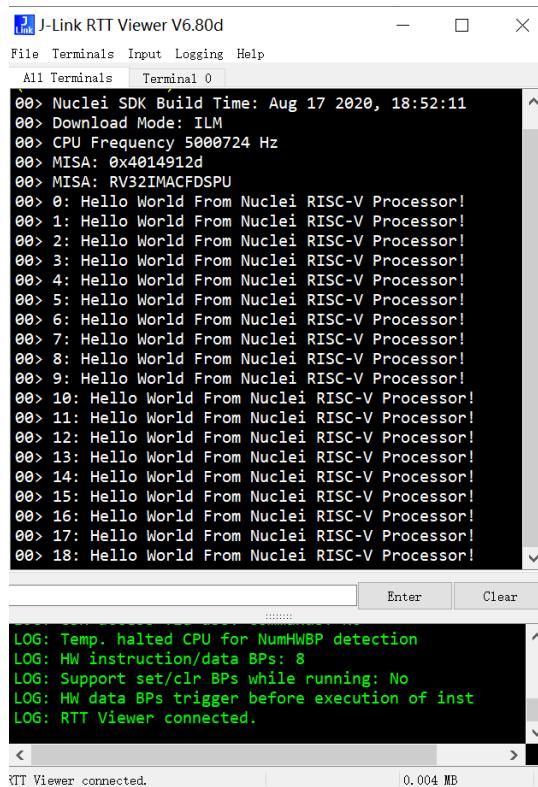


图 6-36 打印输出 Hello World

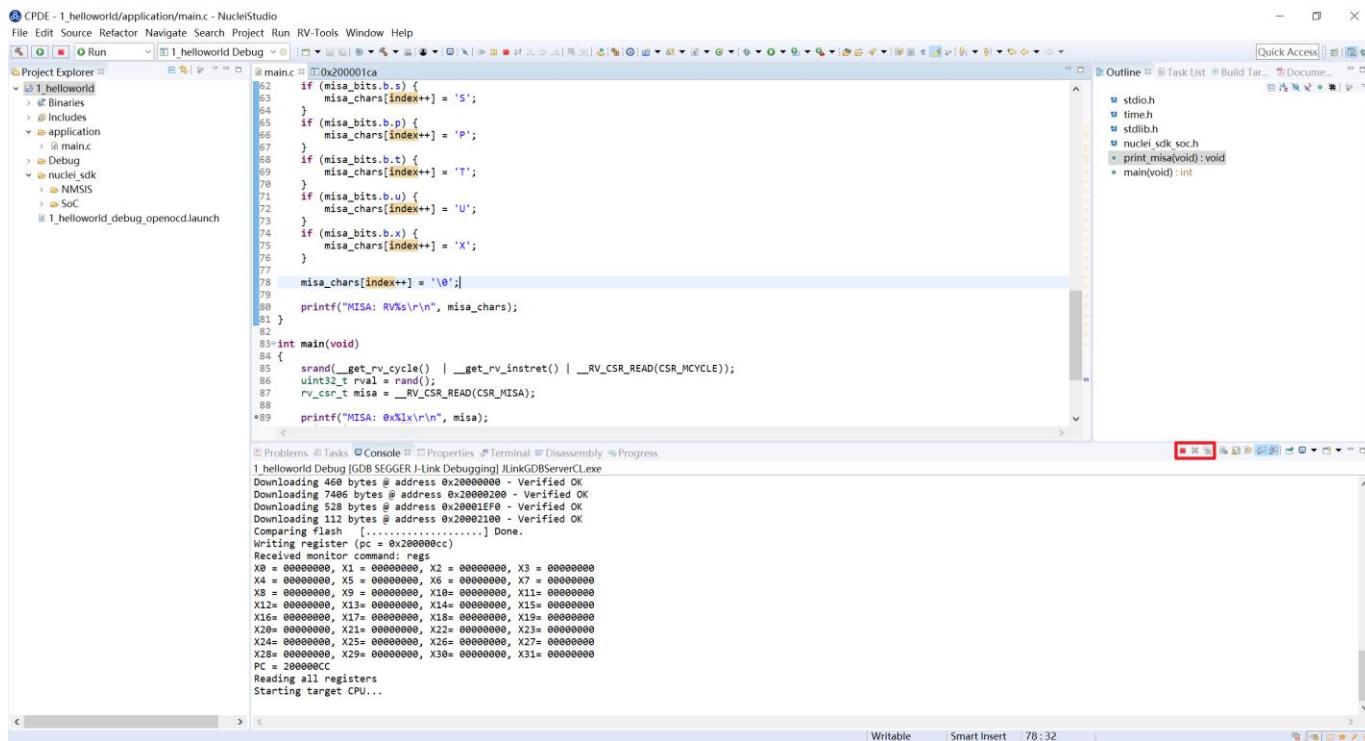


图 6-37 下载完成单击红色按钮断开连接

7. 导入旧版本 Nuclei Studio 创建的工程

Nuclei Studio 从 2020.08 版本开始，官方工具链从 GNU MCU RISC-V GCC (riscv-none-embed-gcc) 升级到 RISC-V Nuclei GCC (riscv-nuclei-elf-gcc)，因为编译前缀发生变化，所以使用 201909 及其之前版本的 IDE 生成的工程，经过调整设置后才可以在新版本 IDE 中使用。这里以 201909 版本的 Nuclei Studio 生成的 helloworld 工程为例，其导入及修改设置的详细步骤如下：

- 导入 201909 版本生成的 helloworld 工程，详细的导入方式请参考 4.2 节，这里不做赘述。
 - 导入工程后右击选择 “Properties” 打开设置页面，选择 “C/C++ Build → Settings” ，打开 Toolchains 栏目然后修改 Name 下拉选项为 “RISC-V Nuclei GCC (riscv-nuclei-elf-gcc)” ，如所示。修改后点击 Apply 保存修改。

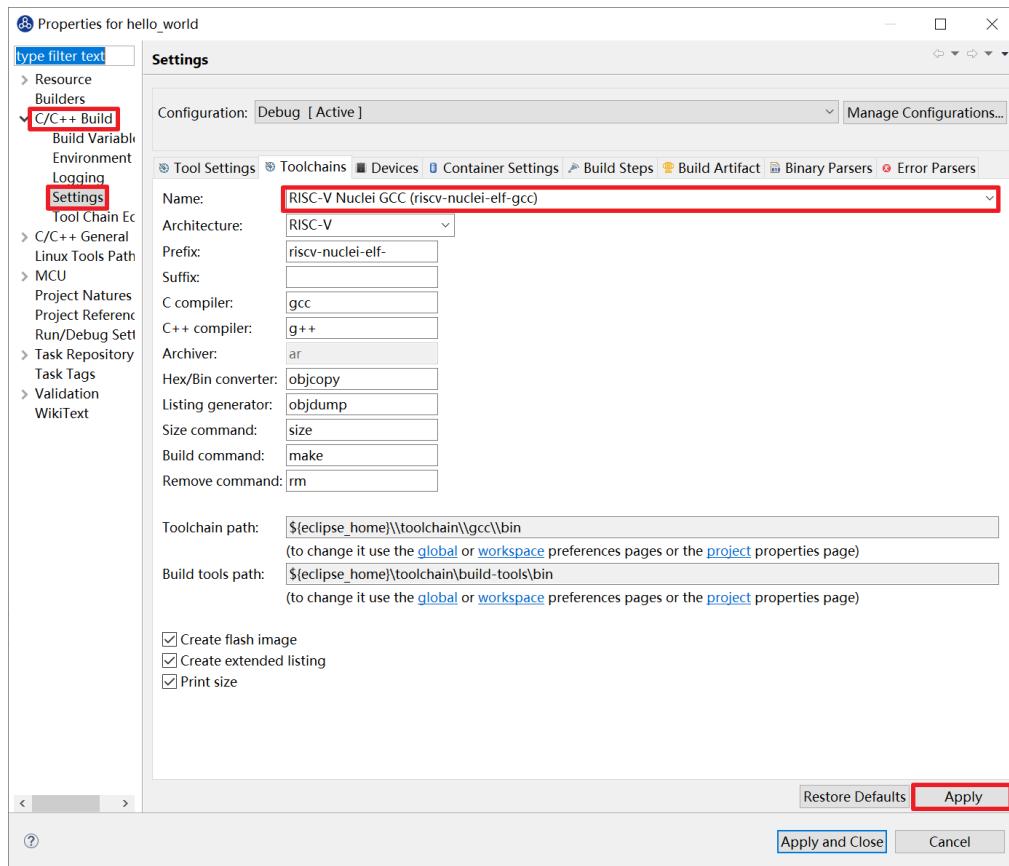


图 7-1 修改 Nuclei Studio 工具链

- 修改后右击工程选择“Clean Project”再选择“Build Project”即可。

8. 常见问题

8.1. 打开/关闭 Launch Bar

打开菜单栏“Window -> Preferences”，搜索“bar”，如图 8-1，取消勾选第一个选项即可关闭 Launch Bar。同理，想要使用 Launch Bar 功能，请勾选此选项。

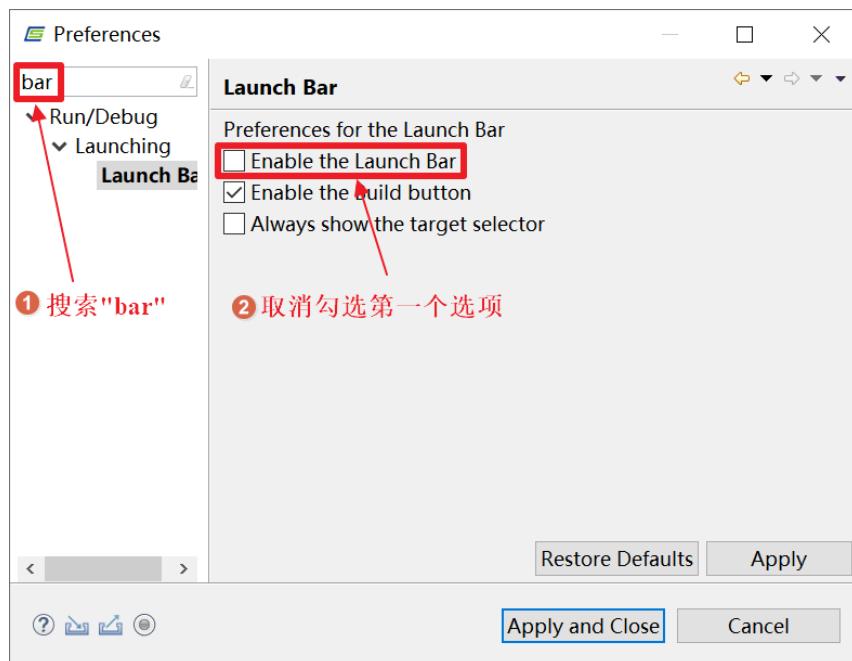


图 8-1 关闭 Launch Bar

8.2. 使用 Launch Bar

Smartchip Studio 的 Launch Bar（下图中蓝色框内标注部分）是对后边下拉框中选中的调试配置对应的工程进行编译、调试和下载等操作，如需使用此部分请注意，图标是与下拉框中调试配置内容相绑定。下面详细介绍 Launch Bar 各部分功能：

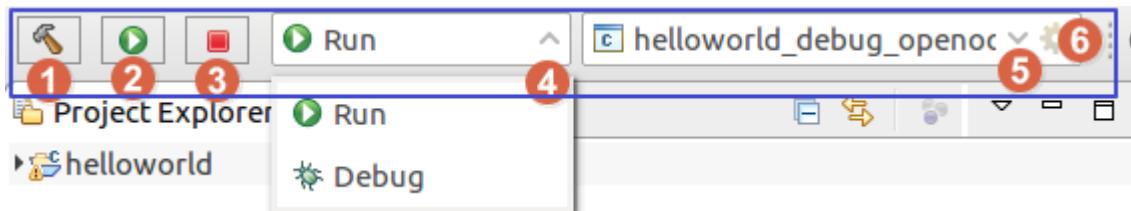


图 8-2 Launch Bar 功能介绍

图标 1：编译选中的调试配置对应的工程。此处选中的工程为 5 号下拉框选中的调试配置对应工程，并非 Project Explorer 中选中的工程。

图标 2：运行/调试选中的调试配置对应的工程。此处图标会根据下拉框 4 的内容变化。

图标 3：退出下载/调试模式。

下拉框 4：切换选择下载/调试模式。切换后会改变图标 2 的显示内容。

下拉框 5：选中调试配置文件。此处的内容会影响图标 1、2、3 对应的工程。

齿轮图标 6：打开当前选中的调试配置页面，可直接进行修改保存。

8.3. 不使用 Launch Bar 进行运行/调试

对于初次新建的工程，如果不使用 Launch Bar 功能，可以右键工程，选择“Debug As -> Debug Configurations”。在弹窗中选择工程对应的设置选项，这里以 helloworld 工程为例，如图 8-3，选择“helloworld_debug_openocd”，之后选择“Debug”开始调试。

同理，初次运行需要右击工程，选择“Run As -> Run Configurations”。在弹窗中选择工程对应的设置选项，之后选择“Run”开始运行。

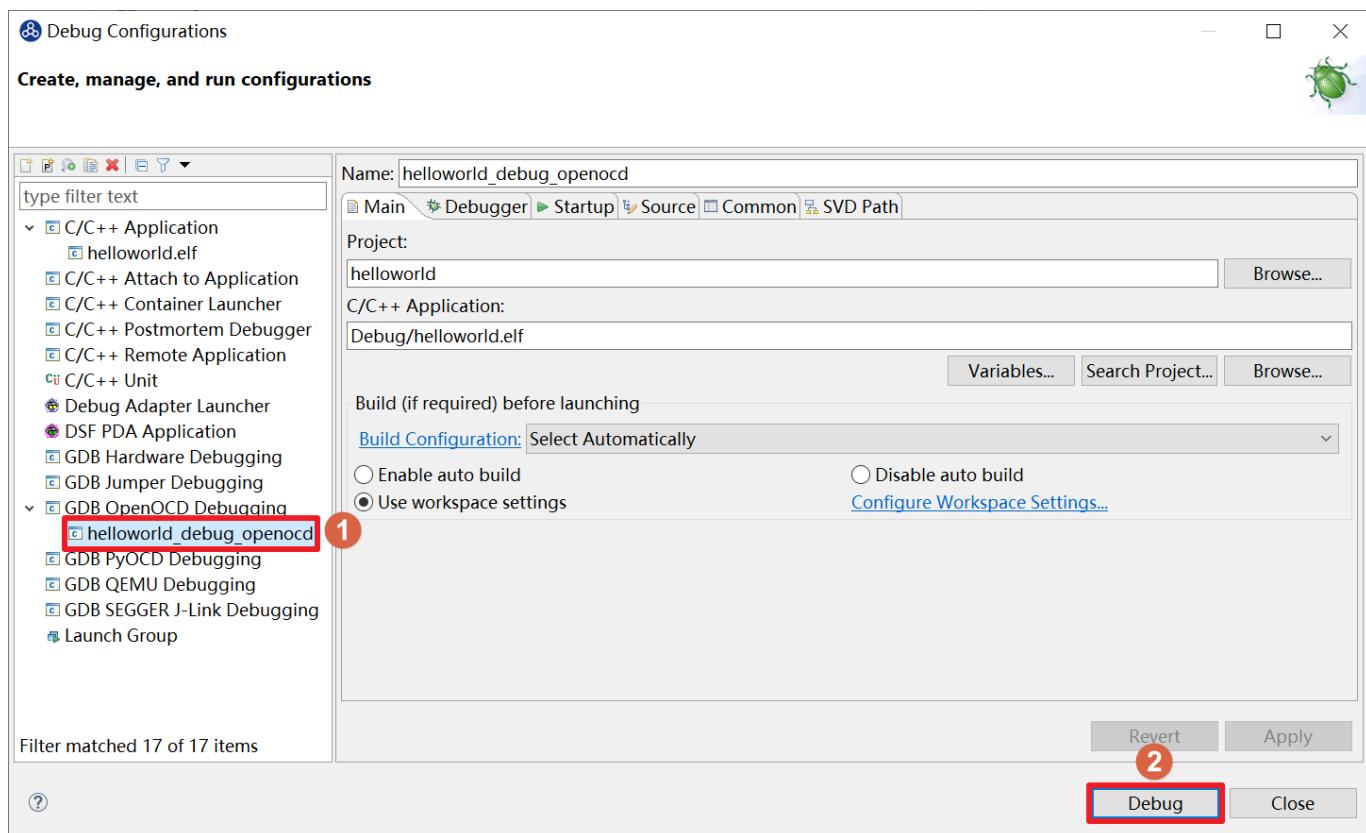


图 8-3 不使用 Launch Bar 首次进行调试

若当前工程已按以上方式进行过调试或运行操作，可在上方按键中选择  的下拉选项，快速选择对应的工程设置进行调试。对于运行，可以在  的下拉选项，快速选择对应的设置。

8.4. Debug 页面查看寄存器

进入 Debug 模式后，可能有些页面并不能第一时间找到，这里以查看寄存器栏目为例。在菜单栏选择“Window -> Show View -> Registers”打开寄存器栏目。同样，如果想查看内存，可以选择“Window -> Show View -> Memory”，想打开调试控制台，可以选择“Window -> Show View -> Debugger Console”，其他页面此处不做过多介绍，请用户自行体验。

8.5. Debug 页面结束进程

在 Debug 页面要退出调试，可点击  退出调试。为防止打开多个进程导致报错，在 Debug 页面，右击 Debug 栏目中的进程，选择“Terminate / Disconnect All”关闭所有 Debug 进程。

8.6. Nuclei Studio 工具栏中各按键功能

在 Nuclei Studio 中，常见的工具栏图标如下，本节将依次介绍各按键的功能。



新建按键 ，包括新建各种文件和工程。

保存当前文件  和保存所有未保存文件 。

切换编译设置 ，默认只有 Debug 和 Release，用户可自行添加，这里不做介绍。

编译工程 ，可以在下拉选项中编译 Project Explorer 中选中的工程。

编译所有工程 ，一次性编译所有 Project Explorer 中的工程。

新建功能     ，包含工程创建，目录创建，文件创建和类创建。

调试 ，可根据下拉选项选择不同调试设置来调试不同的工程。

运行 ，可根据下拉选项选择不同的设置来选择不同的工程。

Profile ，使用 Profile 功能，暂不支持，此处不做介绍。

用户自定义工具 ，此处不做介绍，用户可自己深入研究。

搜索工具 ，可搜索任务，文本等。

文本阅读工具 ，可以展开或折叠文本等功能。

控制台工具 ，可选择打开各种控制台或串口等。

Debug 用  | ，前者跳过所有断点，后者实现 Restart 功能。

查看 CMSIS ，eclipse 自带功能，暂不支持。

Nuclei SDK 工程设置工具 ，参见 5.1。

文本阅读工具 ，可设置快速跳转，具体功能此处不作过多介绍。

8.7. 显示其他窗口

在 IDE 的右上角  按键可以切换不同的显示模式，常用的是 C/C++ 页面和 Debug 页面。其中，在进入 Debug 模式时，若不在 Debug 页面，会有弹窗提示，此时点击“Switch”选项可以切换至 Debug 页面。在菜单栏中选择“Window -> Perspective -> Open Perspective”可以快速切换显示窗口。

在不同的窗口下，“Window -> Show View”显示的内容也不尽相同。如图 8-4，在 Debug 窗口中可以选择显示如寄存器（Registers），内存（Memory）和调试控制台（Debugger Console）。

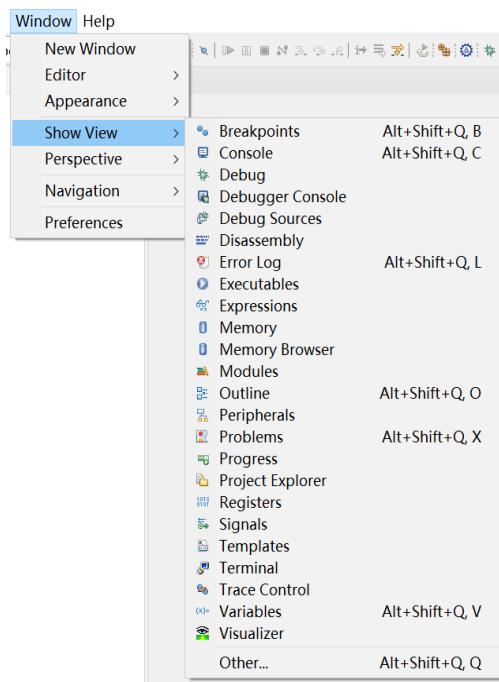


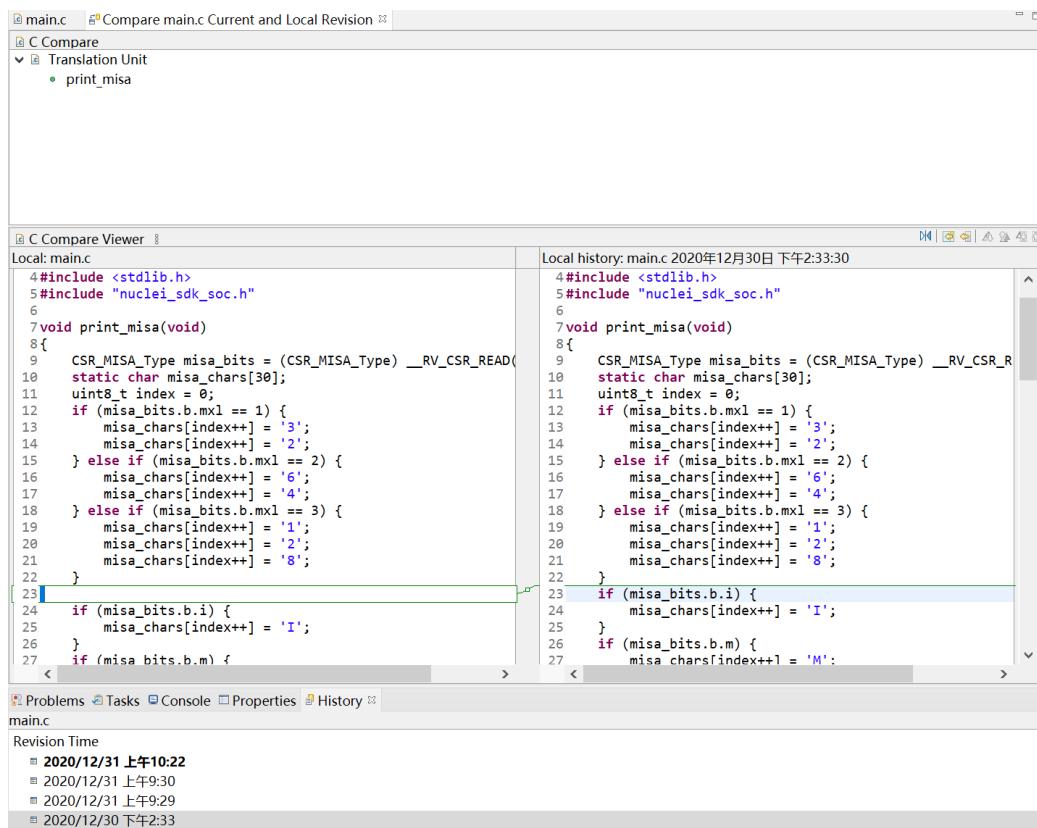
图 8-4Debug 模式查看功能窗口

8.8. 恢复默认窗口布局

在菜单栏中选择“Window -> Perspective -> Reset Perspective”，在弹窗中选择“Reset Perspective”可以恢复窗口默认布局。

8.9. 对比历史文件

右击要查看历史的文件，选择“Compare With -> Local History”打开历史记录。如图 8-5，在打开的 History 栏目双击选择要对比的文件历史版本，可以查看文件变动历史。



```

main.c Compare main.c Current and Local Revision
C Compare
Translation Unit
print_misa

C Compare Viewer
Local: main.c
4 #include <stdlib.h>
5 #include "nuclei_sdk_soc.h"
6
7 void print_misa(void)
8{
9    CSR_MISA_Type misa_bits = (CSR_MISA_Type) __RV_CSR_READ(0x40000000);
10   static char misa_chars[30];
11   uint8_t index = 0;
12   if (misa_bits.b.mx1 == 1) {
13       misa_chars[index++] = '3';
14       misa_chars[index++] = '2';
15   } else if (misa_bits.b.mx1 == 2) {
16       misa_chars[index++] = '6';
17       misa_chars[index++] = '4';
18   } else if (misa_bits.b.mx1 == 3) {
19       misa_chars[index++] = '1';
20       misa_chars[index++] = '2';
21       misa_chars[index++] = '8';
22   }
23   if (misa_bits.b.i) {
24       misa_chars[index++] = 'I';
25   }
26   if (misa_bits.b.m) {
27       misa_chars[index++] = 'M';
28   }
}
Local history: main.c 2020年12月30日下午2:33:30
4 #include <stdlib.h>
5 #include "nuclei_sdk_soc.h"
6
7 void print_misa(void)
8{
9    CSR_MISA_Type misa_bits = (CSR_MISA_Type) __RV_CSR_READ(0x40000000);
10   static char misa_chars[30];
11   uint8_t index = 0;
12   if (misa_bits.b.mx1 == 1) {
13       misa_chars[index++] = '3';
14       misa_chars[index++] = '2';
15   } else if (misa_bits.b.mx1 == 2) {
16       misa_chars[index++] = '6';
17       misa_chars[index++] = '4';
18   } else if (misa_bits.b.mx1 == 3) {
19       misa_chars[index++] = '1';
20       misa_chars[index++] = '2';
21       misa_chars[index++] = '8';
22   }
23   if (misa_bits.b.i) {
24       misa_chars[index++] = 'I';
25   }
26   if (misa_bits.b.m) {
27       misa_chars[index++] = 'M';
28   }
}

```

图 8-5 对比历史文件

8.10. 新建工程时可能出现报错

新建工程时 IDE 需要索引整个工程，根据主机性能其所用时间不同。如果主机性能较差，可能会看到索引时产生 error，索引结束后 error 会消失。如出现以上现象，请以编译时是否报错为准。

8.11. 新增 Include 路径出现缓存

在 include 页面新增路径时，可能会出现缓存的路径内容，此时点击 Workspace 或 File System 选中后可覆盖其内容。

8.12. 设置页面栏目找不到

有时可能因为弹窗大小，部分设置栏目被隐藏起来，如图 8-6，可点击红框中左右方向图标显示隐藏栏目。

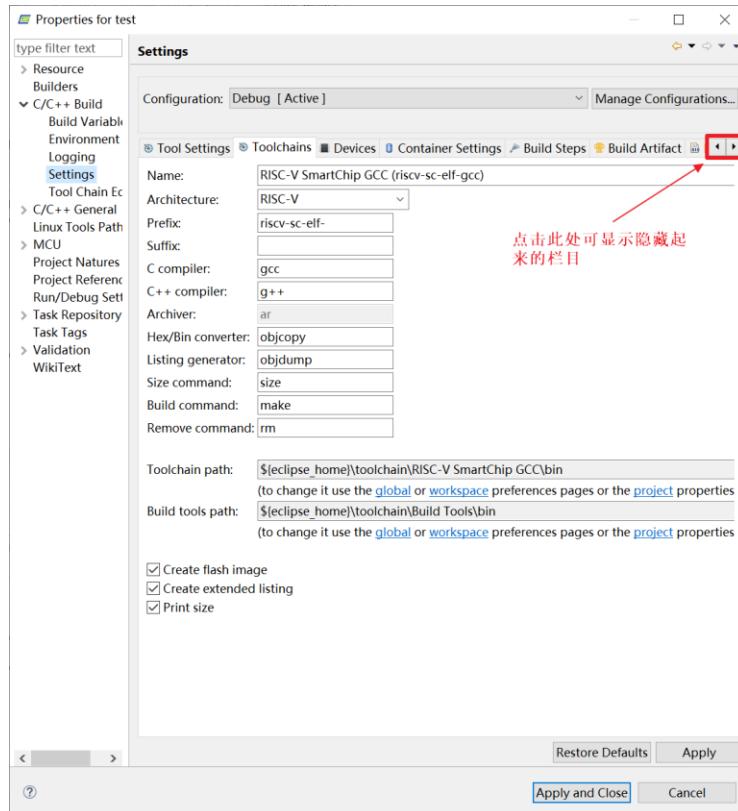


图 8-6 显示隐藏栏目

8.13. 开发板下载速度很慢

如果遇到开发板下载速度很慢，甚至出现超时的报错，请切换至使用 USB3.0 接口，如果使用虚拟机开发，也请同时将 USB 接口设置为 3.0。

8.14. Linux 环境下多用户使用 Nuclei Studio

如果需要多用户同时使用 Nuclei Studio（不推荐），用户在运行 Nuclei Studio 时首先要打开菜单栏的“Window->Preferences”。在弹窗中需要修改三个地方：

- 打开“MCU->Global OpenOCD Path”，“Executable”输入“openocd”，“Folder”输入“\${eclipse_home}/toolchain/openocd/bin”。

- 打开“MCU->Global QEMU Path”，“Executable”输入“qemu-system-riscv32”，“Folder”输入“\${eclipse_home}/tools/qemu”。
- 打开“MCU->Global RISC-V Toolchains Paths”，“Default toolchain”选中“RISC-V Nuclei GCC”，“Toolchain folder”输入“\${eclipse_home}/toolchain/gcc/bin”。

修改后点击“Apply and Close”保存并关闭设置弹窗。

9. Nuclei Studio 升级

一般情况下，如果 Nuclei Studio 没有大的版本变动，Nuclei Studio 升级工作可以由用户下载最新的 GNU 工具链以及其他相关工具和升级 IDE Plugins 的方式来完成。

9.1. GCC/OpenOCD 等工具链的安装

Nuclei Studio 已经把工具链集成在 IDE 内部，工具链存放在 NucleiStudio_IDE_XXX 文件夹内，路径为：NucleiStudio_IDE_XXX\NucleiStudio\toolchain。IDE 默认使用此路径的工具链，所以请不要移动“toolchain”此文件夹，使用 IDE 创建工程后也不需要进行工具链的相关配置。

由于工具链升级的速度会比 IDE 快，所以后期需要用户手动升级工具链，工具链升级方法如下：

- 首先删除需要更新的 GCC 或者 OpenOCD 文件内的内容，然后在芯来科技官网下载最新版本的 GCC 或者 OpenOCD。
- 将 GCC 或者 OpenOCD 的内容复制到对应的文件夹下，即可完成 IDE 工具链的更新，IDE 自带的工具链的版本号记录在 ReadMe.txt 中，如图 9-1 所示。注意：要保证 gcc 所在的这一级目录文件夹名字不变，替换后保证 bin 文件夹所在层级是图中文件夹的子目录，中间不要有其他文件夹。

build-tools	2020/7/30 10:35	文件夹
gcc	2020/7/30 10:35	文件夹
openocd	2020/7/30 10:35	文件夹
ReadMe.txt	2019/9/25 13:48	文本文档

图 9-1 IDE 工具链路径

9.2. IDE Plugins 升级

Nuclei Studio 支持 IDE 内在线升级，步骤如下。

- 如图 9-2，打开安装软件弹窗，在菜单栏选择“Help → Install New Software”。
- 如图 9-3，点击“already installed”打开已安装界面。
- 如图 9-4，选择“Nuclei Studio”，点击“update”即可自动完成在线更新。

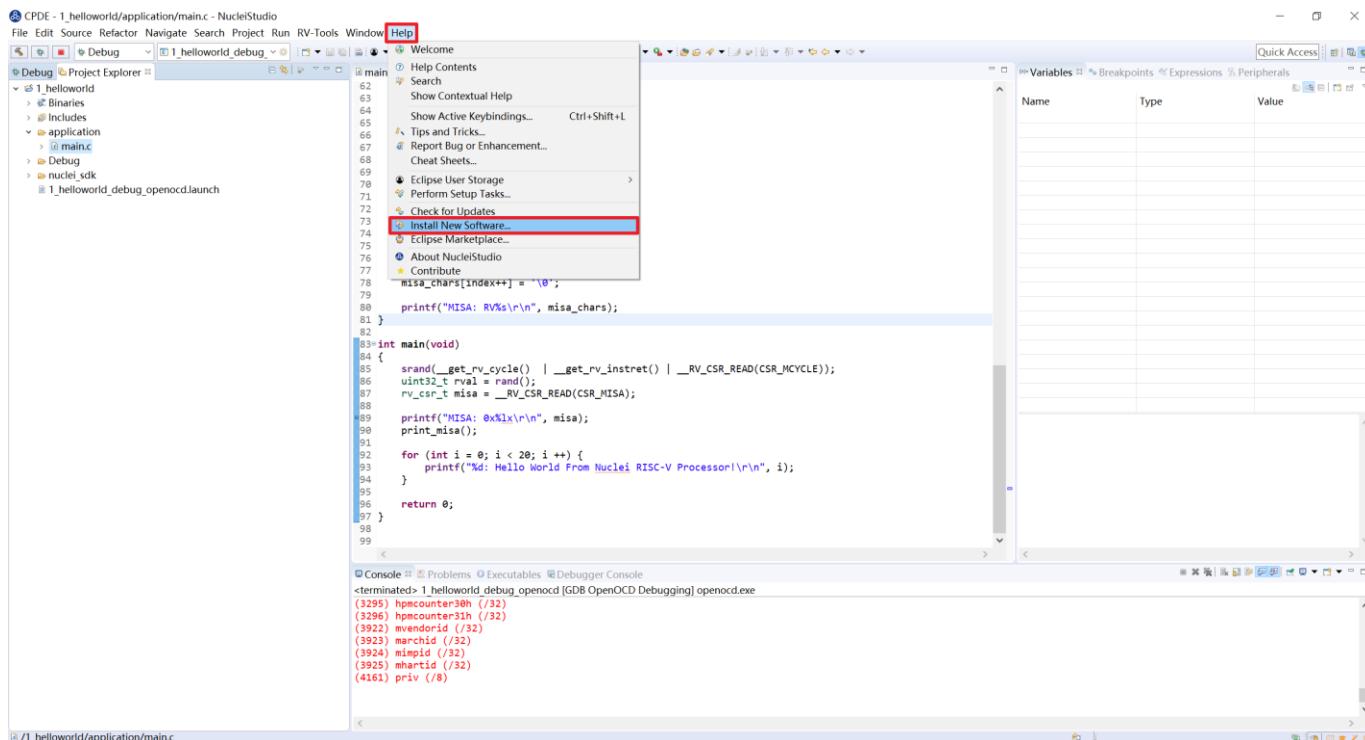


图 9-2 打开安装软件弹窗

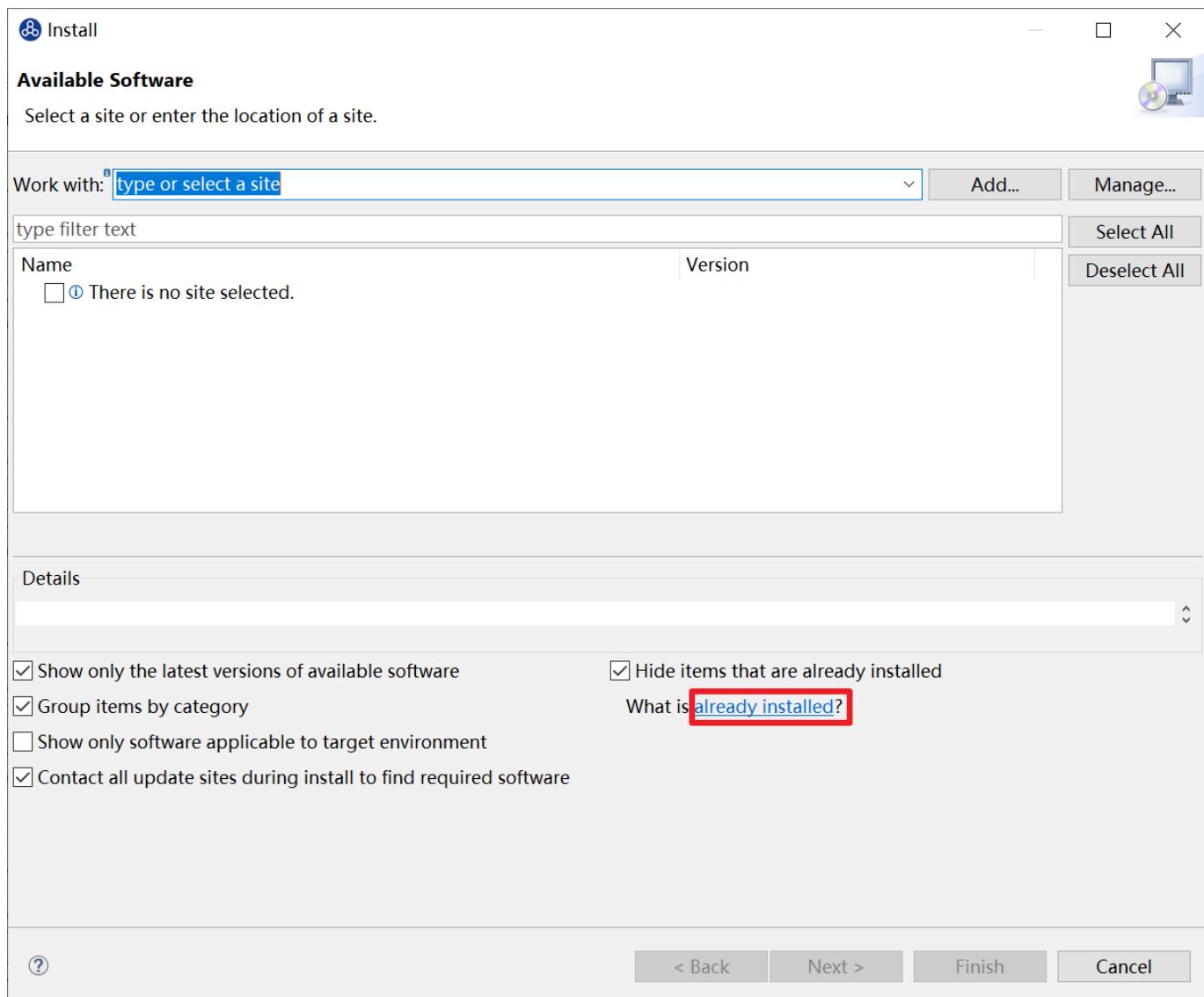


图 9-3 打开已安装界面

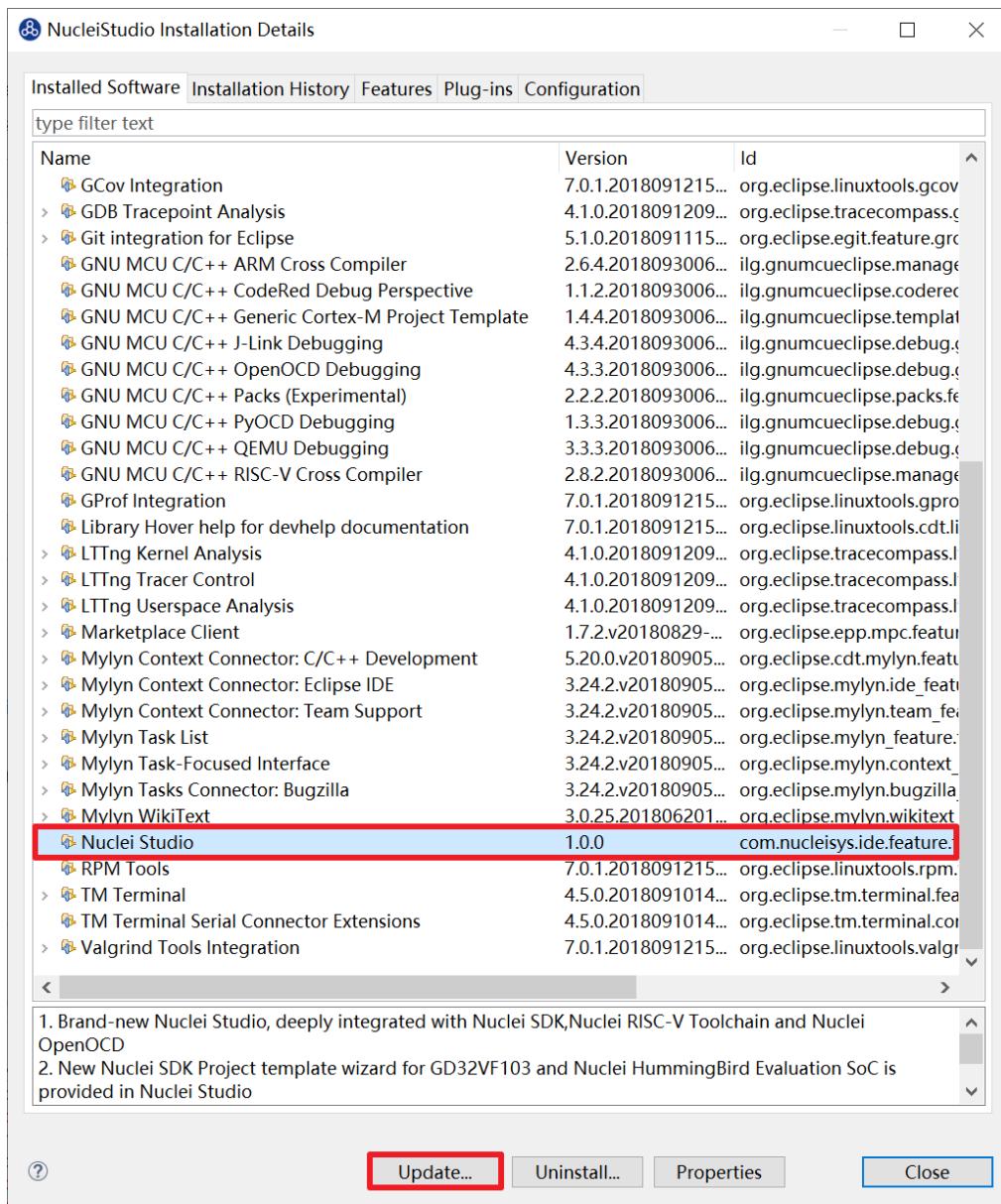


图 9-4 选择“Nuclei Studio”更新

10. 使用 QEMU 仿真调试

最新版本的 Nuclei Studio 整合了 QEMU，可对 RVSTAR 工程实现仿真调试功能。

按照前文的内容新建一个 RVSTAR 工程，这里以 helloworld 工程为例，我们新建一个名为 helloworld 的工程。

右击新建的工程，选择“Debug As -> Debug Configurations”打开调试设置弹窗。如图 10-1，在弹窗中选择“GDB QEMU Debugging”下带有“qemu”后缀的设置选项。我们工程为“helloworld”，所以此处选择“helloworld_debug_qemu”，选中后点击右下方的“Debug”开始仿真调试。

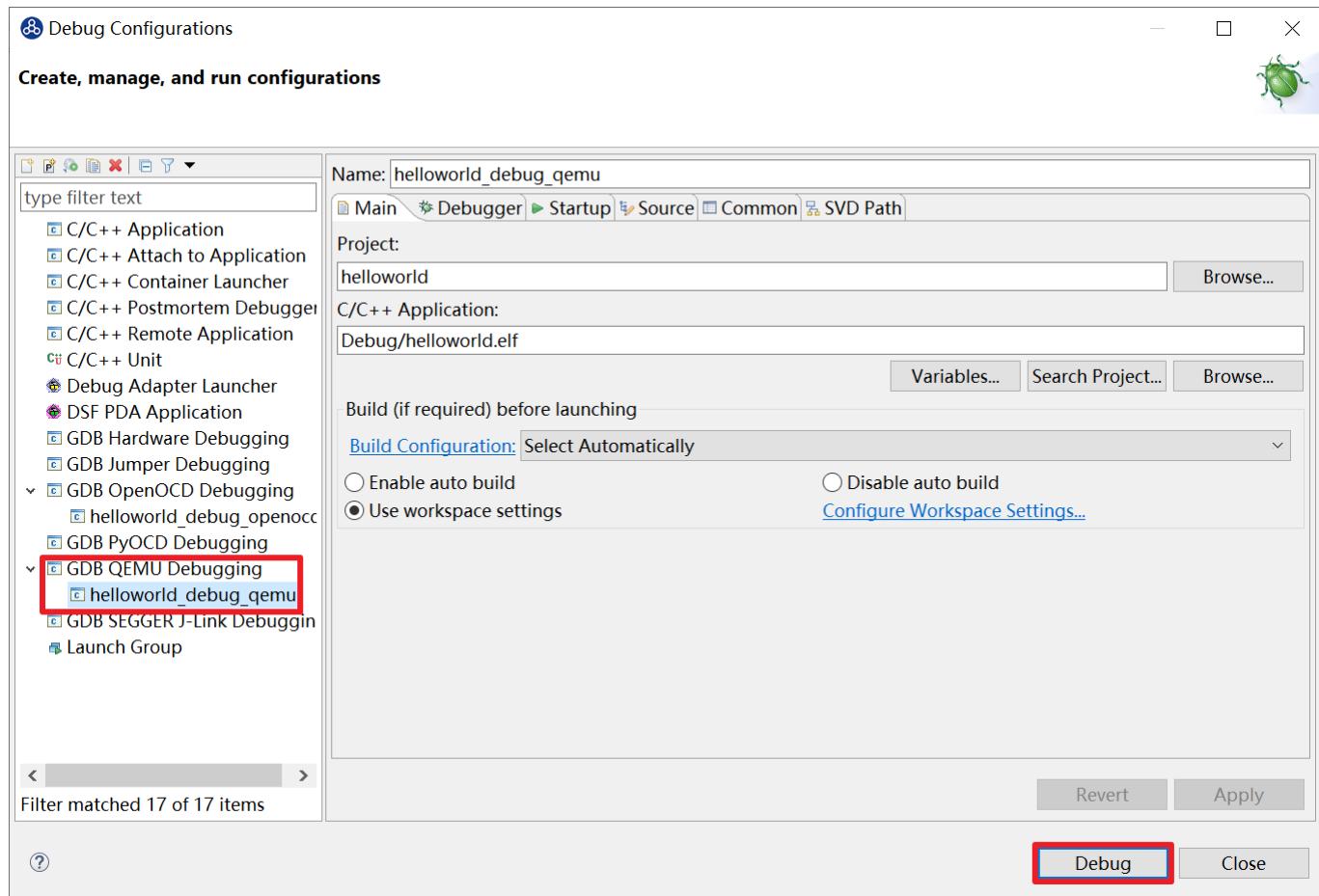


图 10-1 选择 QEMU 进行仿真调试

如图 10-2 所示，仿真调试页面与一般的调试页面完全相同，其中，串口打印信息可直接在 Console 中查看。寄存器可能有部分未实现的会报错，属于正常现象。

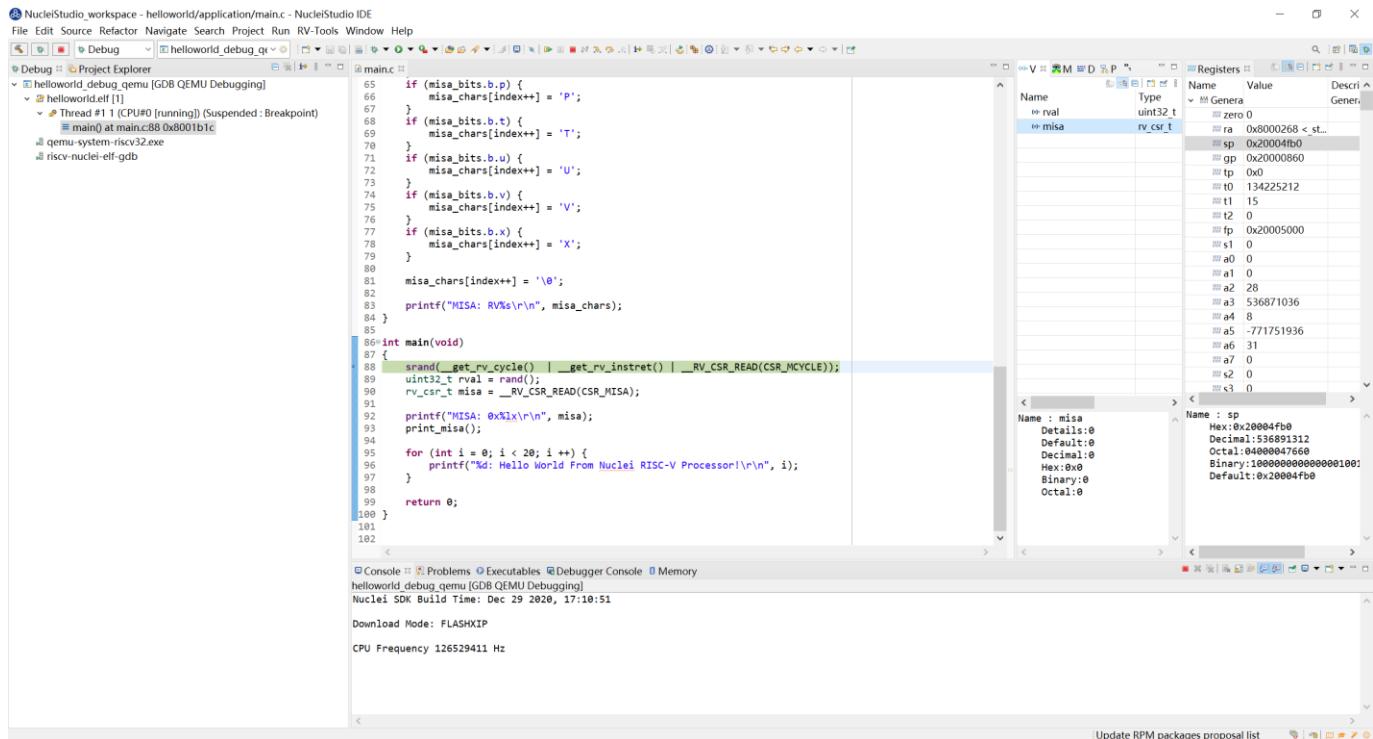


图 10-2QEMU 仿真调试界面

仿真运行流程与仿真调试相类似，右击 RVSTAR 工程选择“Run As -> Run Configurations”，弹窗中选择“GDB QEMU Debugging”下带有“qemu”后缀的设置选项。选中后点击“Run”，可以在控制台（Console 栏）看到运行的打印输出。