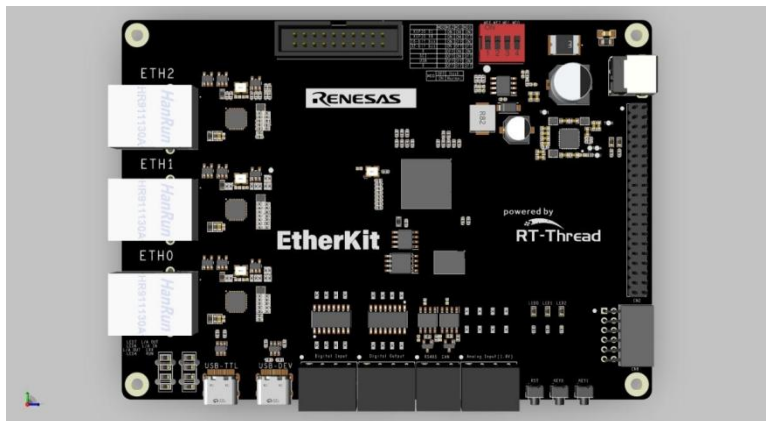


RZN2L Quick Start-----基于 Etherkit 开发板

简介

本应用笔记介绍了如何使用 RZ/N2 Etherkit 开发板快速入门。描述硬件配置和 IDE IAR 和 E2studio 软件环境搭建。



开发工具

- IDE: IAR EW for Arm 9.50.2
或 E2studio 2024-01.1
- FSP: RZ/N2 FSP V2.0
- 仿真器: Jlink V12

实验材料

- Etherkit 开发板
- Jlink 仿真器, 需支持瑞萨 R52 内核

实验部分

1	硬件设置	2
2	IAR 环境开发入门	3
3	E2studio 环境开发入门	5

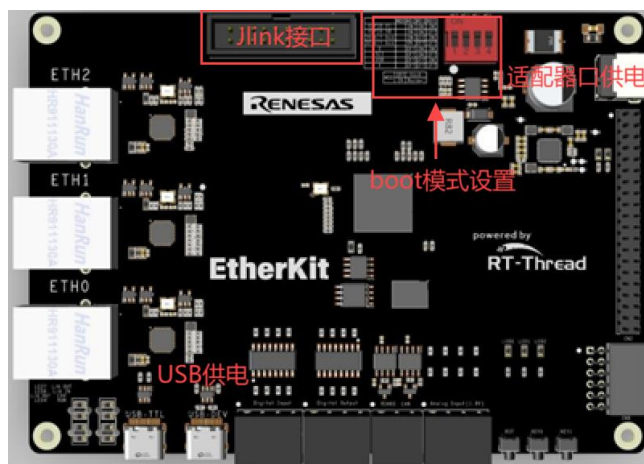
1 硬件设置

本节介绍使用 IAR 环境 和 E2studio 软件安装和环境配置。

1.1

开发板设置：

- 供电：可选 USB 供电或适配器供电
- Boot 模式设置：推荐 xSPI0 x1 boot mode
- Jlink v12



本节完

2 IAR 环境开发入门

本节介绍使用 IAR 环境软件安装和环境配置。

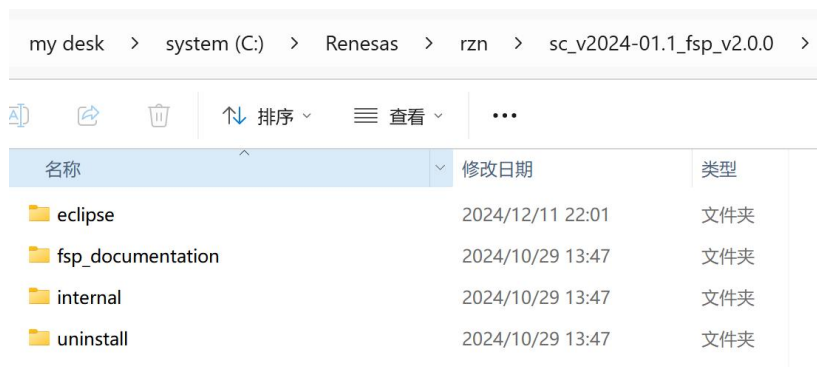
2.1

IAR 软件安装：

- 安装 IAR EW for Arm 9.50.2 IAR 安装不做详细说明
- 安装 FSP 2.0: RZN2L: [setup_rznfsp_v2_0_0_rzsc_v2024-01.1.exe](#)

打开 setup_rznfsp_v2_0_0_rzsc_v2024-01.1.exe 安装软件，安装后的目录如下图：

FSP 软件使用说明参考 fsp_documentation 文件夹下面的说明文件，里面每个外设参数都有说明，还有一些例程参考。

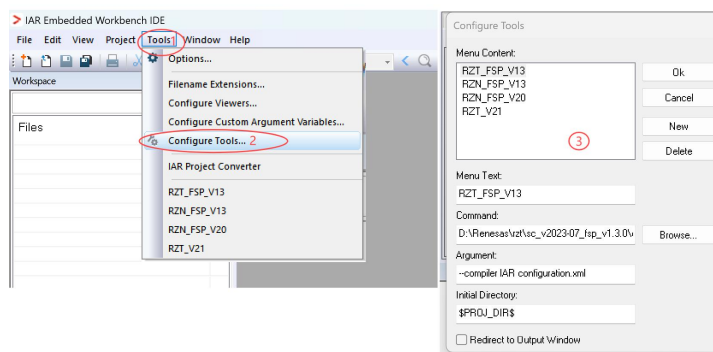


2.2

FSP 导入 IAR

FSP 可以作为 Tools 导入到 IAR 环境，方便后续随时打开 FSP，进行配置修改更新。

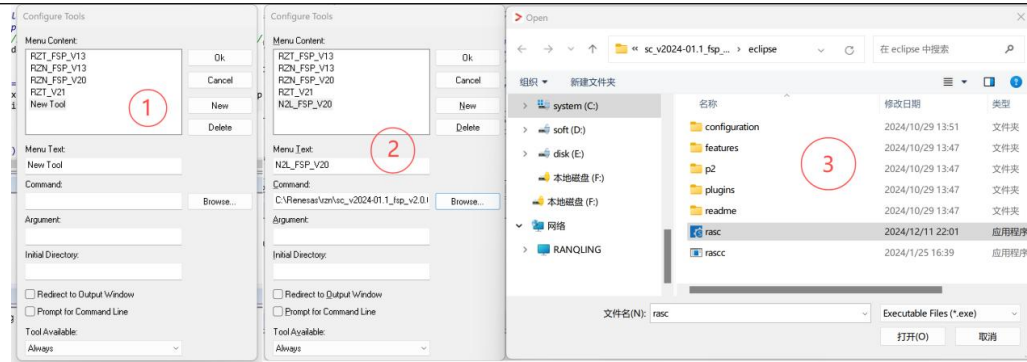
打开 IAR-Tools-Configure Tools...如下图：



从上图可以看到，已经导入多个版本 FSP，新增点击 New，

出现如下图界面 1,出现 New Tools，修改名称为：N2L_FSP_V20。

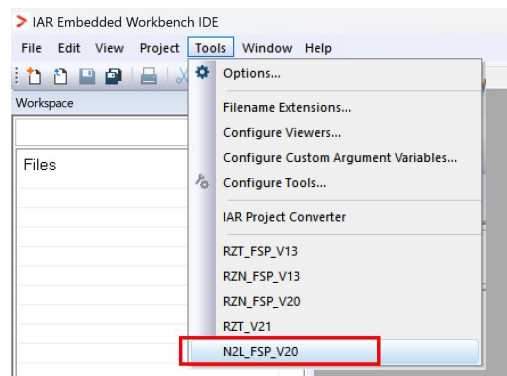
然后点击 Browse，选择 FSP 安装目录，选定 rase 文件，点击打开。



Argument 输入: --compiler IAR configuration.xml

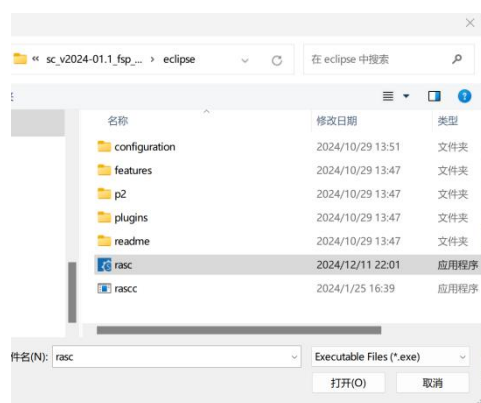
Initial Directory 输入: \$PROJ_DIR\$

- 点击 OK, 即添加成功。
- IAR 点击 Tools 即可看到刚添加的 FSP 工具 N2L_FSP_V20, 打开一个例程, 点击该工具, 即可调出该工程的 FSP 配置。空工程无法打开该工具。

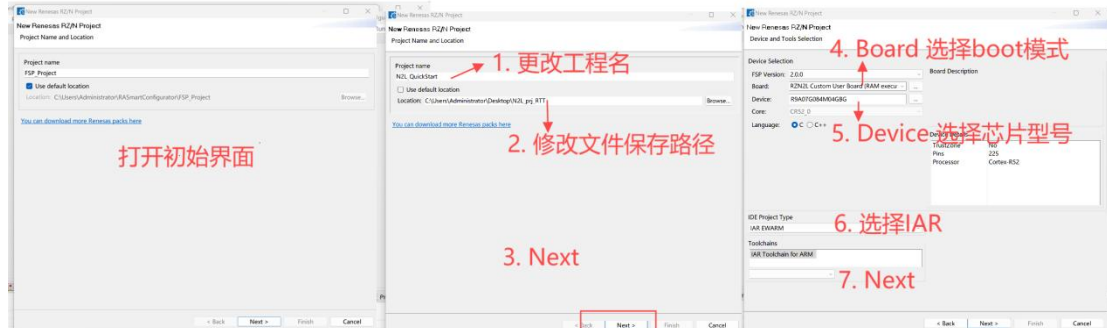


2.3

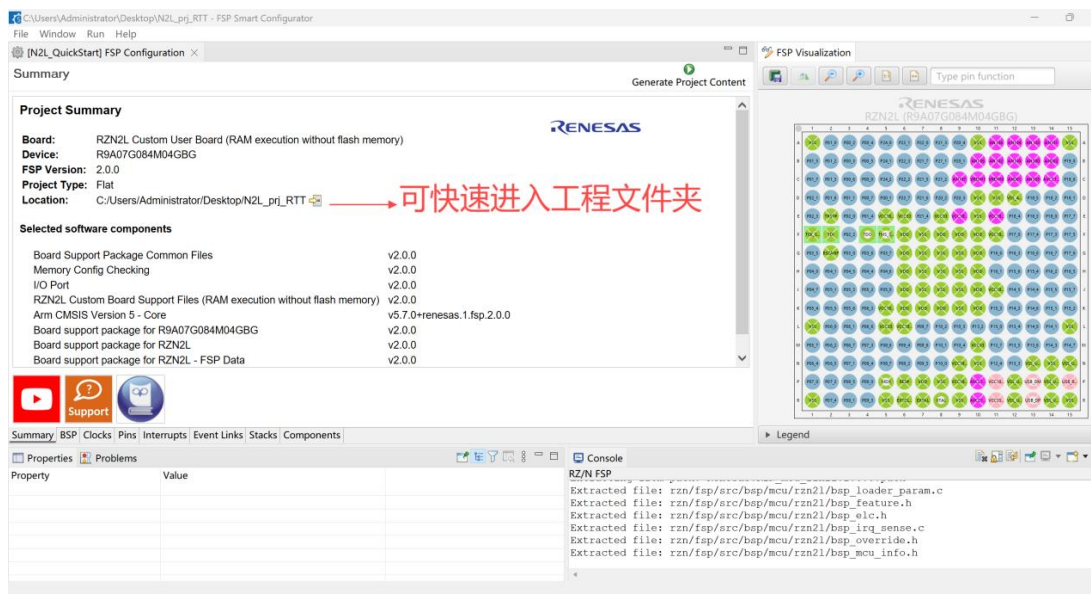
- 利用 FSP 新建 IAR 工程



- 打开 rasc.exe-- File > New > FSP Project...更改工程名、文件路径--下一步--boot 模式、芯片型号、IDE 选择--下一步



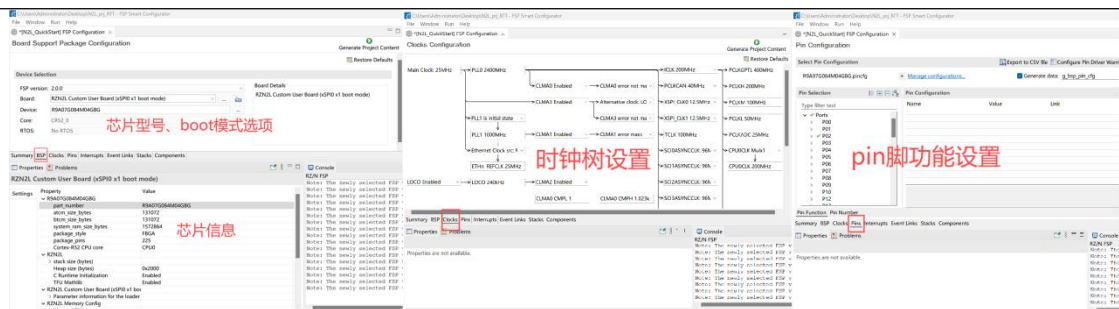
- Boot 模式选择：RZN2L Custom User Board (xSPI0 x1 boot mode)
- Next 进入到如下界面：



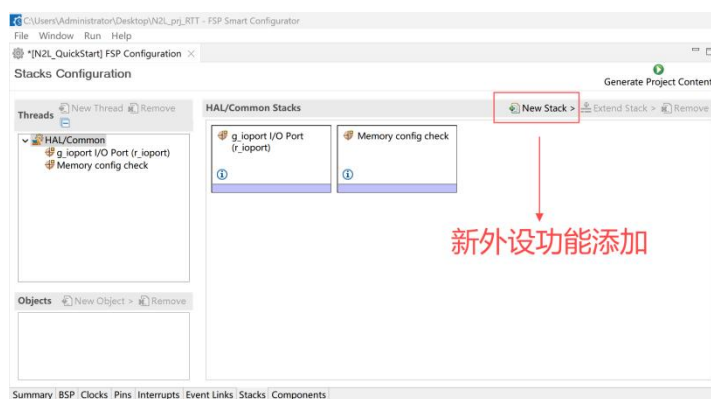
2.4

FSP 功能窗口简介

- BSP 窗口： 芯片型号 boot 模式选择 芯片信息
- Clocks 窗口： 时钟树设置
- Pins 窗口： pin 脚功能设置



- Stacks 窗口：新外设功能添加

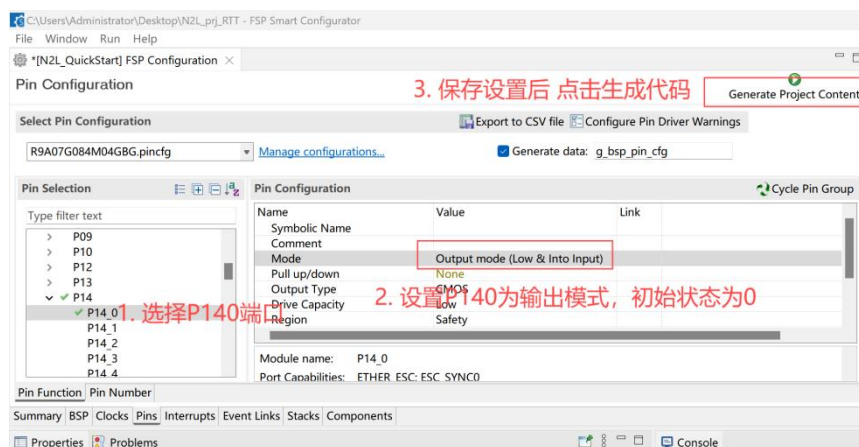


2.5

生成 IAR 代码

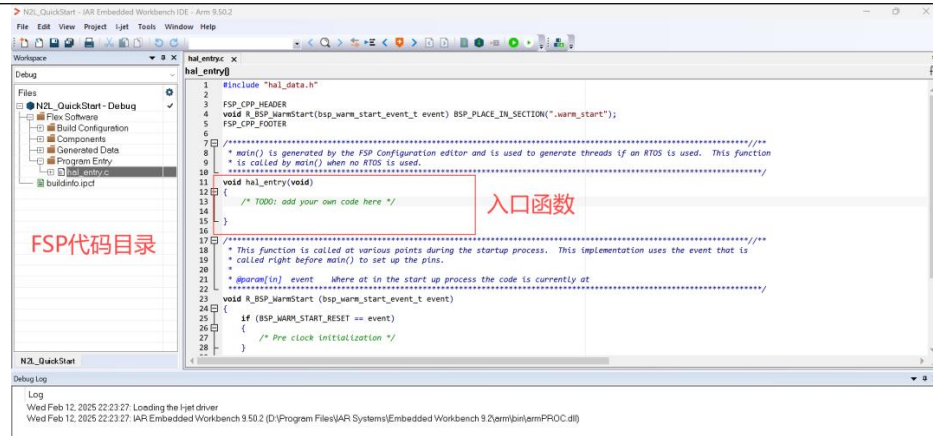
- 生成一个简单的代码，设置 P140 为输出，控制 LED 灯。

设置 P140 为输出模式，初始状态为低电平，然后点击 **Generate Project Content** 生成代码。



进入工程目录，打开 IAR 工程：

下图为 FSP 代码目录结构，void hal_entry(void) 为入口函数，用户代码在此添加。

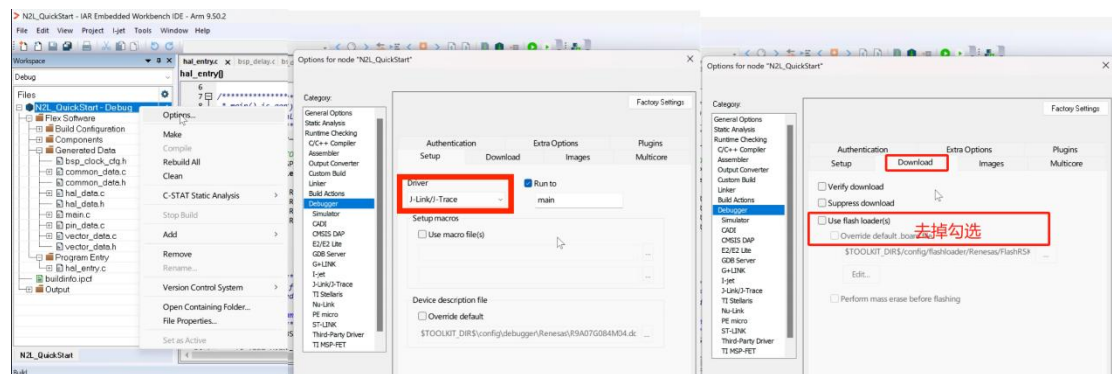


2.6

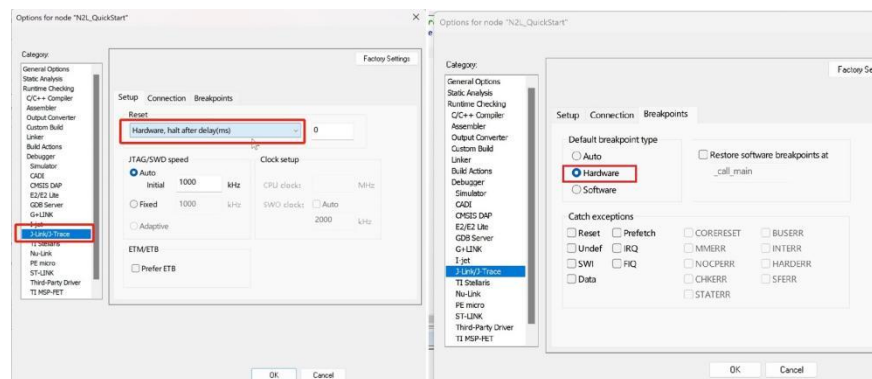
Jlink 设置

代码生成后，IAR 工程默认使用 IJET 仿真器，需要更改为 Jlink，方法如下：

工程名右键--Options--Debugger 设置修改



J-Link/J-Trace 设置修改

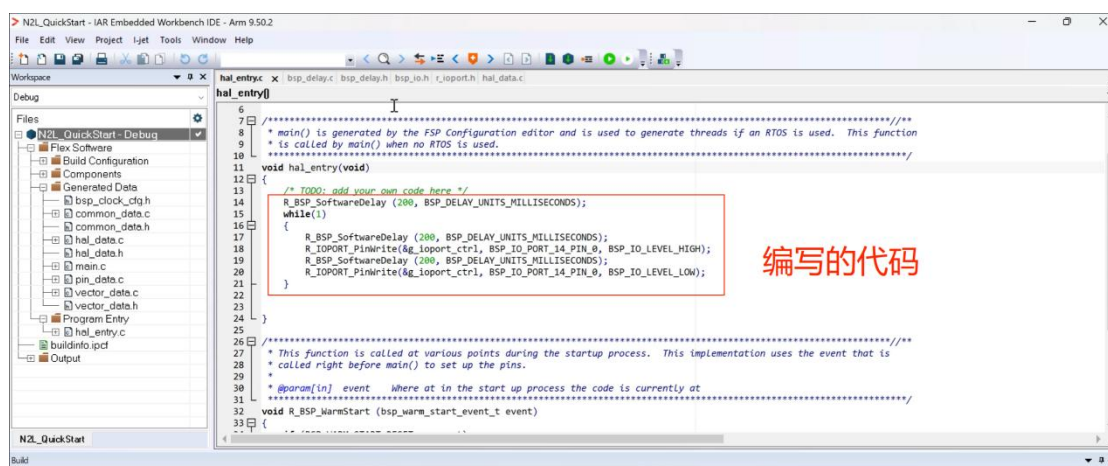


2.7

编写代码

- 利用系统生成延时函数：

void R_BSP_SoftwareDelay (uint32_t delay, bsp_delay_units_t units), 控制 P140 电平输出，使 LED1 亮灭 200ms 反转一次。



2.8

Xspi0 boot 模式 debug 注意事项

在 xSPI0 boot 模式下 debug 的时候，程序不会停止用户入口函数处，需要在工程编译前插入如下代码到 startup_core.c:

BSP_TARGET_ARM void system_init (void)

{

#if 1 // Software loops are only needed when debugging.

__asm volatile (

" mov r0, #0 \n"

" movw r1, #0xf07f \n"

" movt r1, #0x2fa \n"

"software_loop: \n"

" adds r0, #1 \n"

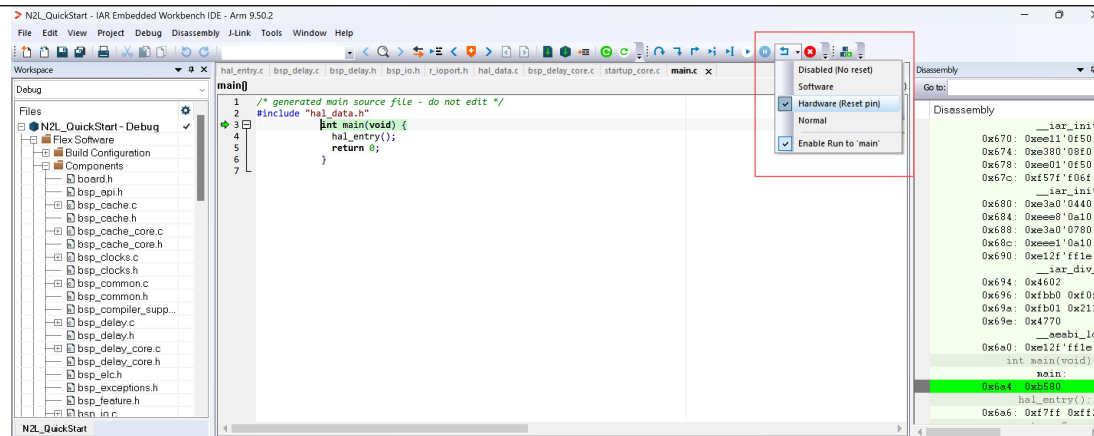
" cmp r0, r1 \n"

" bne software_loop \n"

::: "memory");

#endif

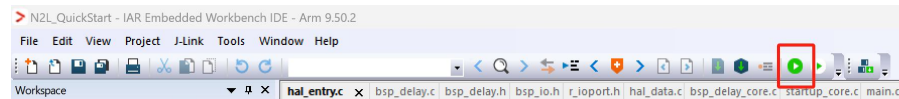
如果第一次进入仿真界面，程序没有停止在用户程序入口 int main () 位置，需暂停程序，IAR 快捷 RESET 设置为：Hardware (Reset pin)，点击 RESET 后，程序就会停在函数入口 int main () 位置。如下图：



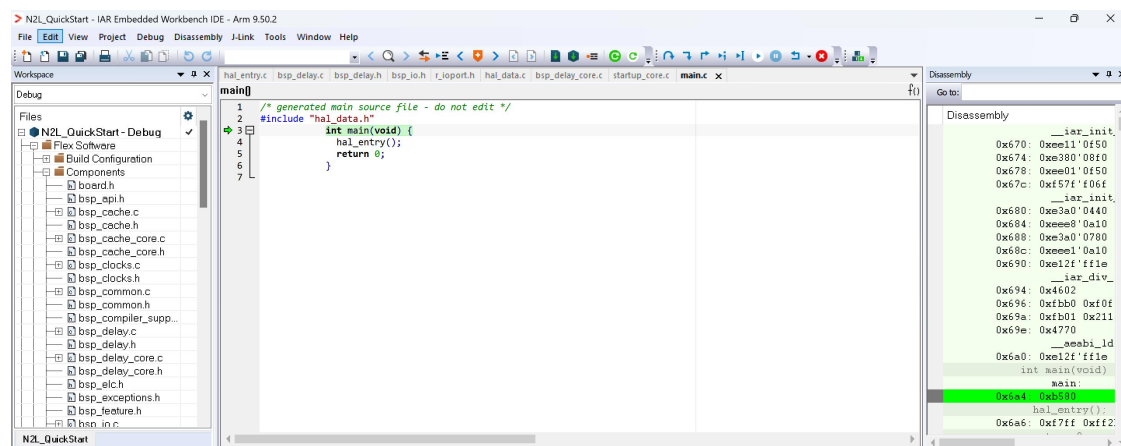
2.9

编译&运行

开发板连上 JLINK, 上电。编译程序, 没有错误后, 点击 **download and debug** 图标进入仿真。



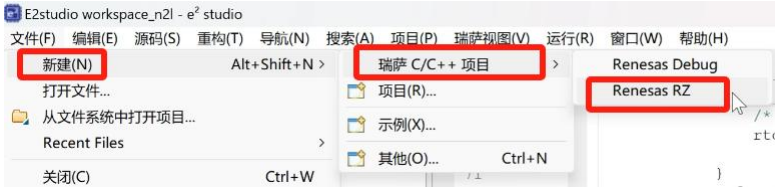

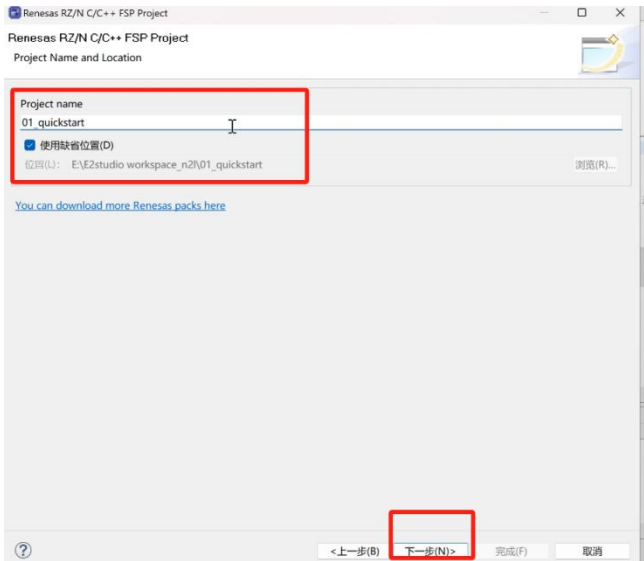
程序进入仿真界面, 如下图, 点击全速运行, 观察开发板 LED1 的亮灭变化。



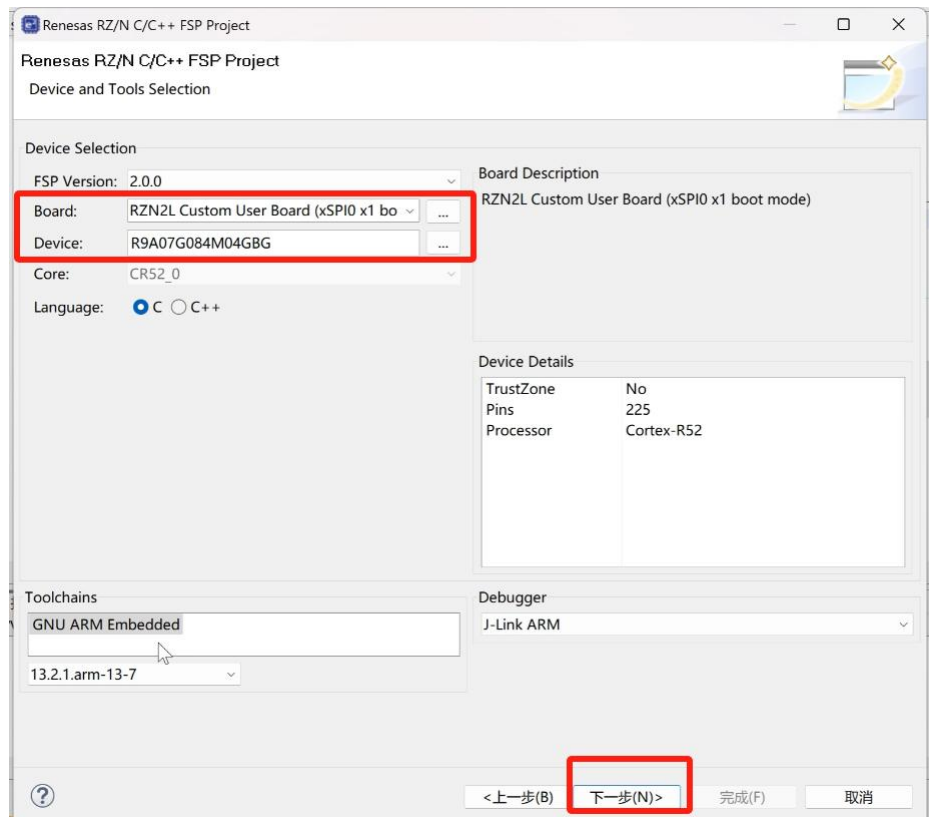
本节完

3 E2studio 环境开发入门

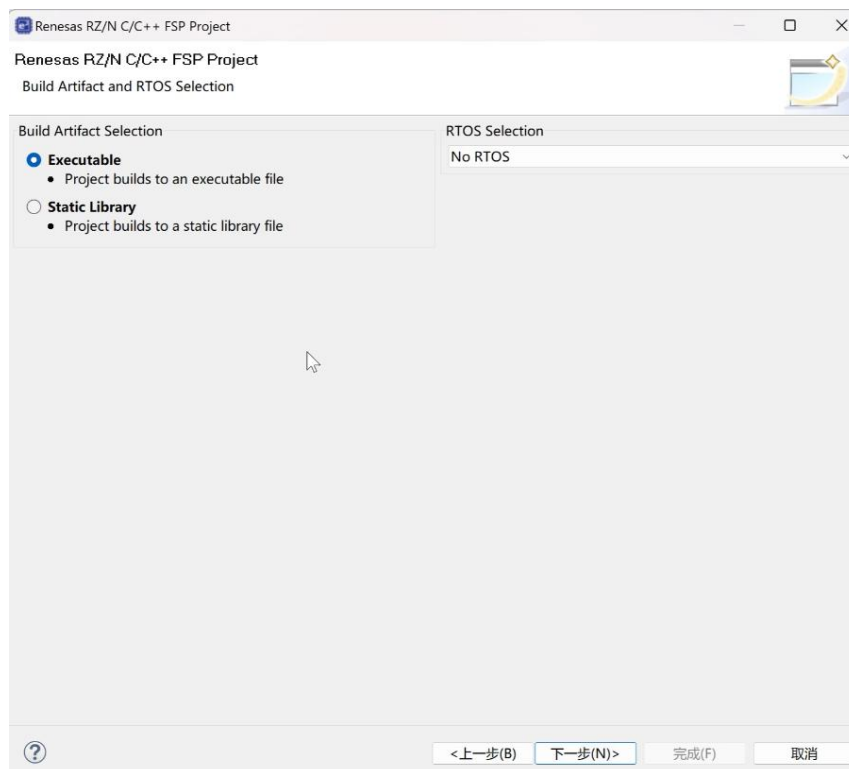
本节介绍使用 E2studio 环境软件安装和环境配置。

3.1	<p>E2studio 软件安装：</p> <ul style="list-style-type: none"> 安装 E2studio : RZN2L: setup_rznfsp_v2_0_0_e2s_v2024-01.1.exe
3.2	<p>E2studio 新建工程</p> <ul style="list-style-type: none"> 点击：文件 > 新建 > 瑞萨 C/C++ 项目 > Renesas RZ. 
3.3	<ul style="list-style-type: none"> 选择 Renesas RZ/N C/C++ FSP Project，点击下一步  <ul style="list-style-type: none"> 新建工程名 01_quickstart，设置文件保存路径，点击下一步 

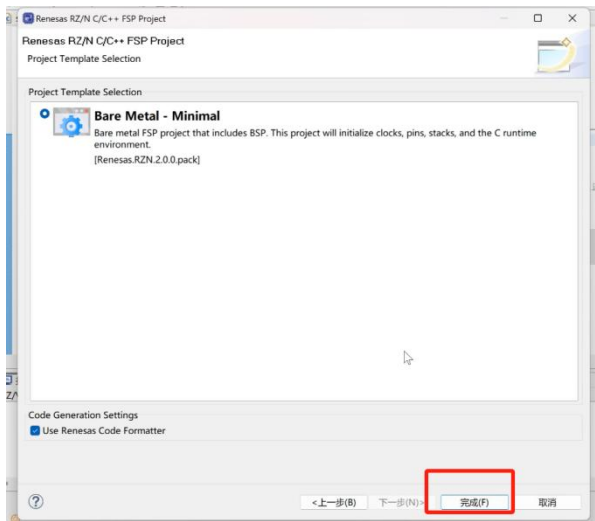
- 选择 RZN2L Custom User Board (xSPI0 x1 boot mode)
- 芯片型号 R9A07G084M04GBG 点击下一步：



- 进入如下界面，点击下一步：



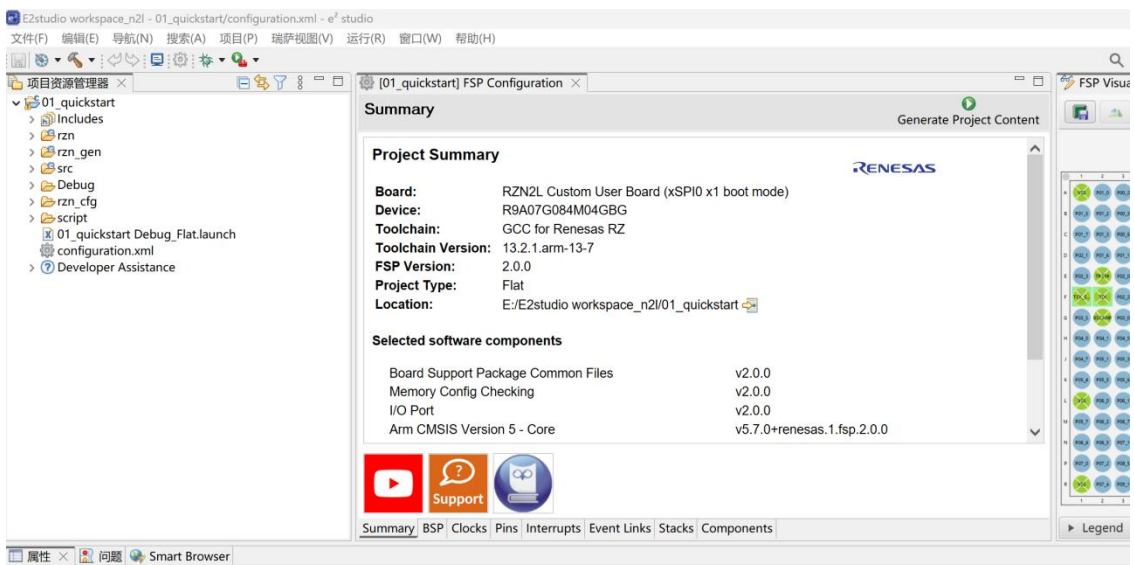
● 点击 完成



● 跳出对话框：选择 打开透视图

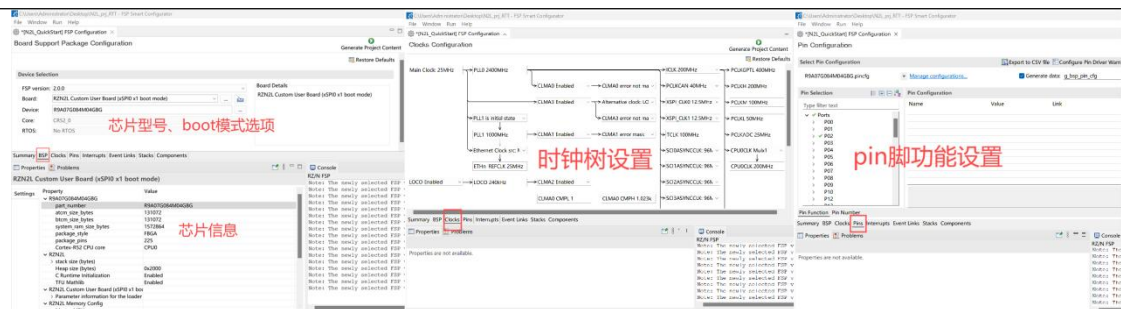


● 进入如下工程界面

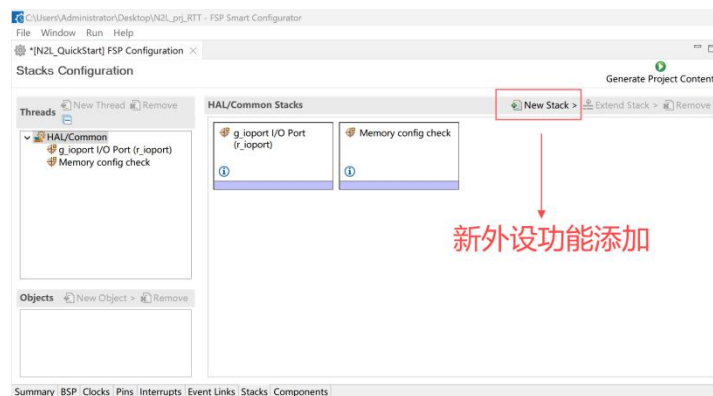


3.4 FSP 功能窗口简介 和 IAR 中 FPS 一致

- BSP 窗口： 芯片型号 boot 模式选择 芯片信息
- Clocks 窗口： 时钟树设置
- Pins 窗口： pin 脚功能设置



● Stacks 窗口：新外设功能添加

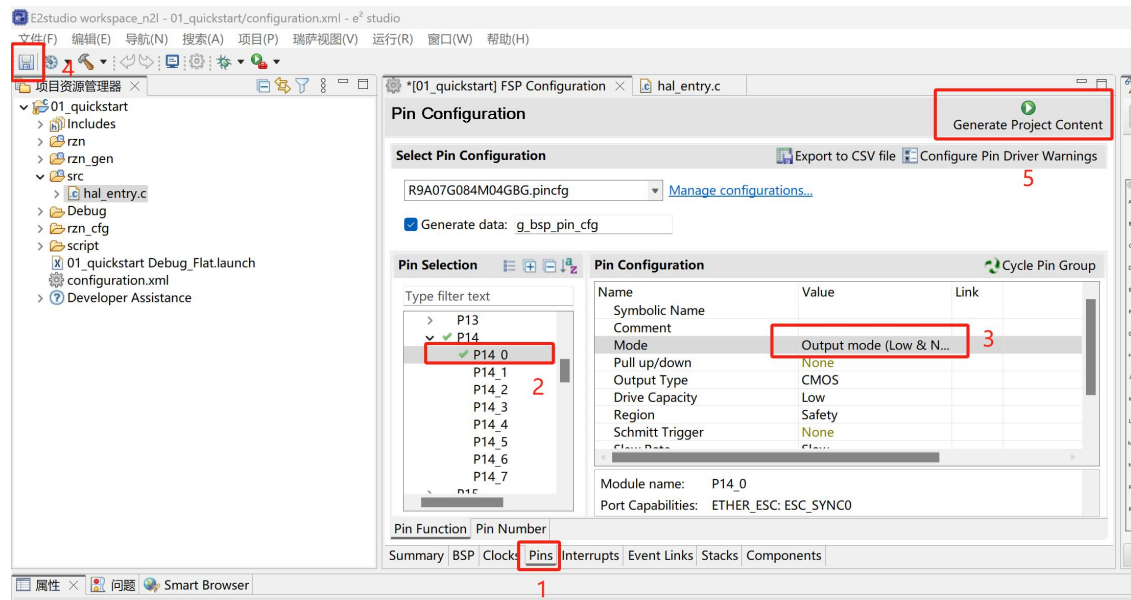


3.5

利用 FSP 配置引脚，生成驱动代码

- 生成一个简单的代码，设置 P140 为输出，控制 LED 灯。

设置 P140 为输出模式，初始状态为低电平，然后点击 **Generate Project Content** 生成代码。



编写用户代码

- 利用系统生成延时函数：

void R_BSP_SoftwareDelay (uint32_t delay, bsp_delay_units_t units), 控制 P140 电平输出，使 LED1 亮灭 200ms 反转一次。

- 打开 hal_entry.c 进入 void hal_entry(void) 为入口函数，添加用户代码。

```
R_BSP_SoftwareDelay (200, BSP_DELAY_UNITS_MILLISECONDS);

while(1)

{

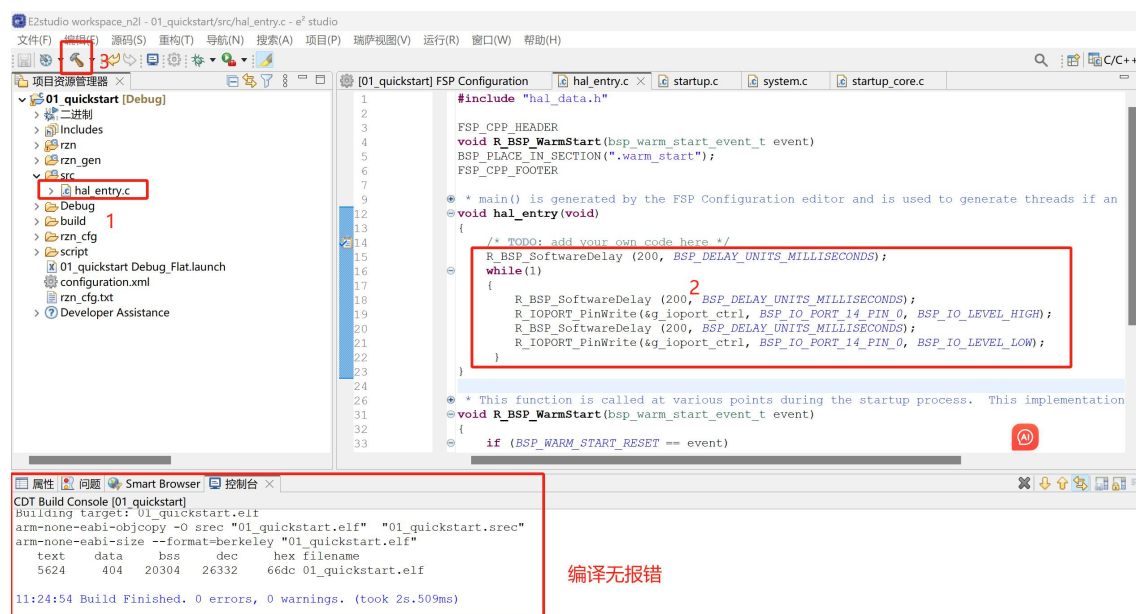
    R_BSP_SoftwareDelay (200, BSP_DELAY_UNITS_MILLISECONDS);

    R_IOPORT_PinWrite(&g_ioport_ctrl, BSP_IO_PORT_14_PIN_0,
        BSP_IO_LEVEL_HIGH);

    R_BSP_SoftwareDelay (200, BSP_DELAY_UNITS_MILLISECONDS);

    R_IOPORT_PinWrite(&g_ioport_ctrl, BSP_IO_PORT_14_PIN_0,
        BSP_IO_LEVEL_LOW);

}
```

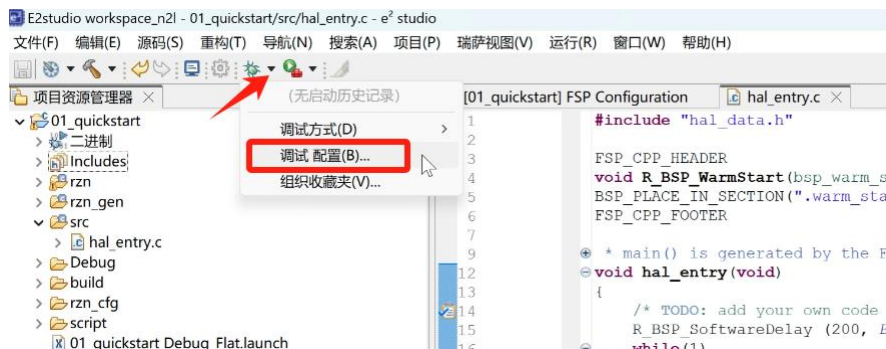


编译无报错

3.6

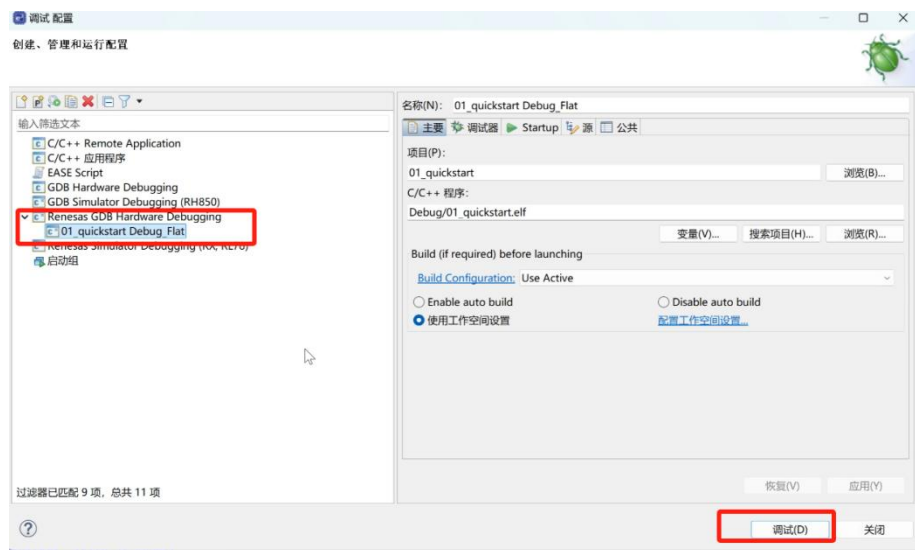
调试：开发板 Jlink 正确连接，上电。

- 新工程第一次调试，需进行调试配置，配置方法如下：

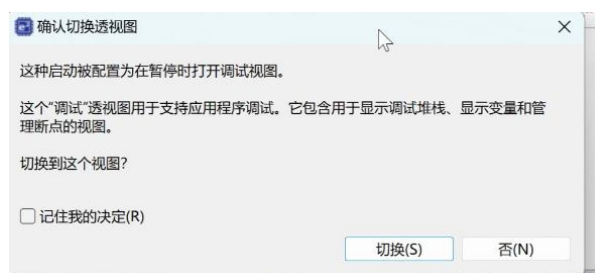


- 打开调试配置

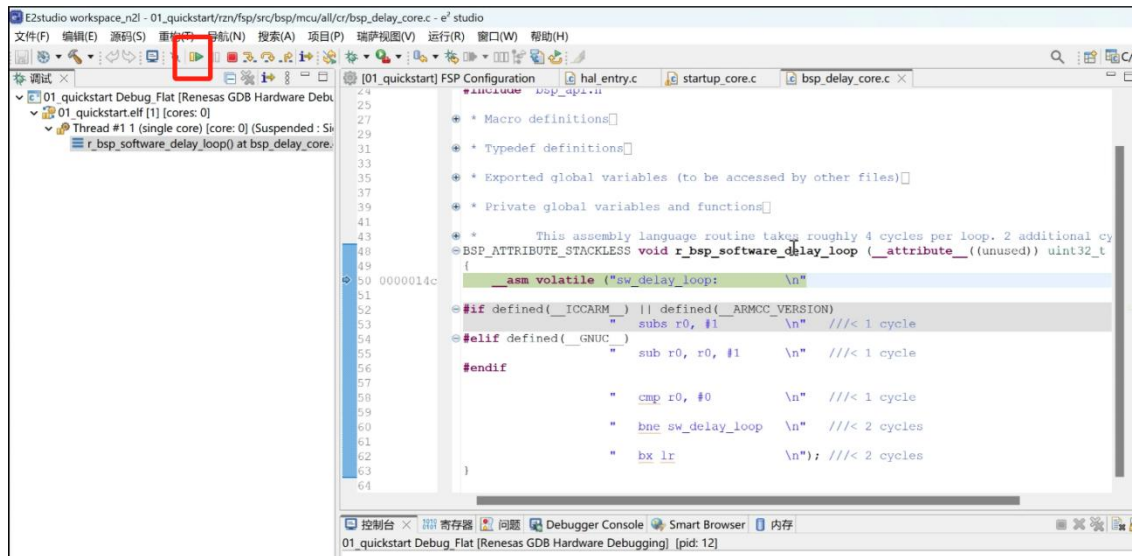
选择我们创建的工程，然后点击 调试 按钮



- 跳出如下窗口，点击 切换 按钮



- 进入调试界面后，点击全速运行按钮，运行代码



观察 开发板 LED 灯 亮灭变化

本节完