

PO_EVB 板使用手册

产品编号:	
文档编号:	
作者:	
创建时间:	2026 年 1 月 8 日
审核时间:	2026 年 1 月 19 日
审核人:	
项目完成日期:	

修改记录

文件编号	版本号	拟制人/修改人	拟制/修改日期	更改理由	主要更改内容

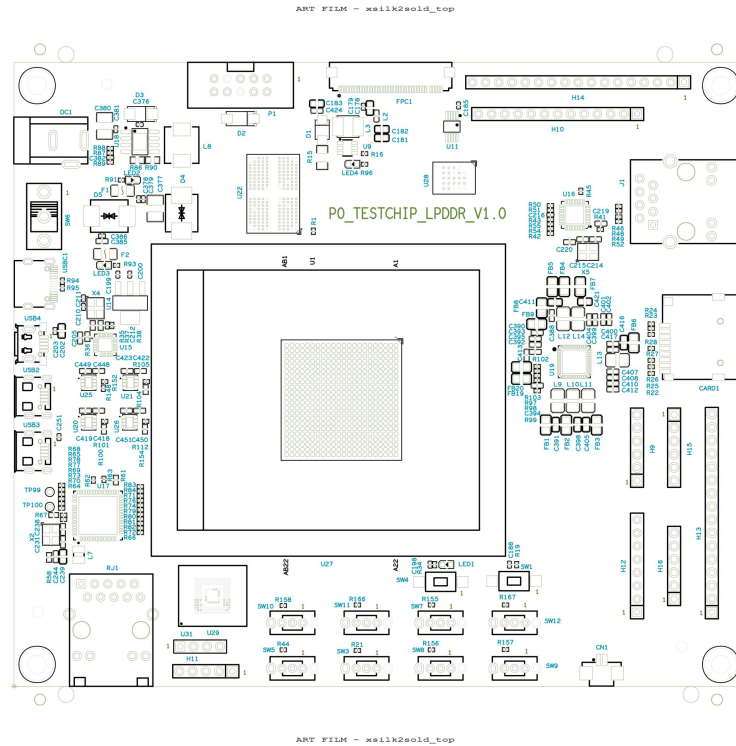
目录

1. 概述	4
2. P0 EVB 丝印及实物	5
3. P0 EVB 器件及功能说明	6
4. EVB 板使用说明	7
4.1 硬件配置	7
4.2 软件配置	8
4.3 应用示例	14
5. 常见问题解答	17
5.1 烧录串口输出乱码	17

1. 概述

本文档为明确 P0_EVB 板（以下简称“EVB 板”）的设计、开发、测试及维护过程，规范 EVB 板的各项功能指标、接口定义、操作流程及技术参数，为相关技术人员提供详细、准确的使用指导和参考依据。本文档旨在确保 EVB 板在项目开发中能够被正确、高效地应用，保障基于该 EVB 板的后续软件开发、算法验证及系统集成工作顺利进行。

2. P0 EVB 丝印及实物



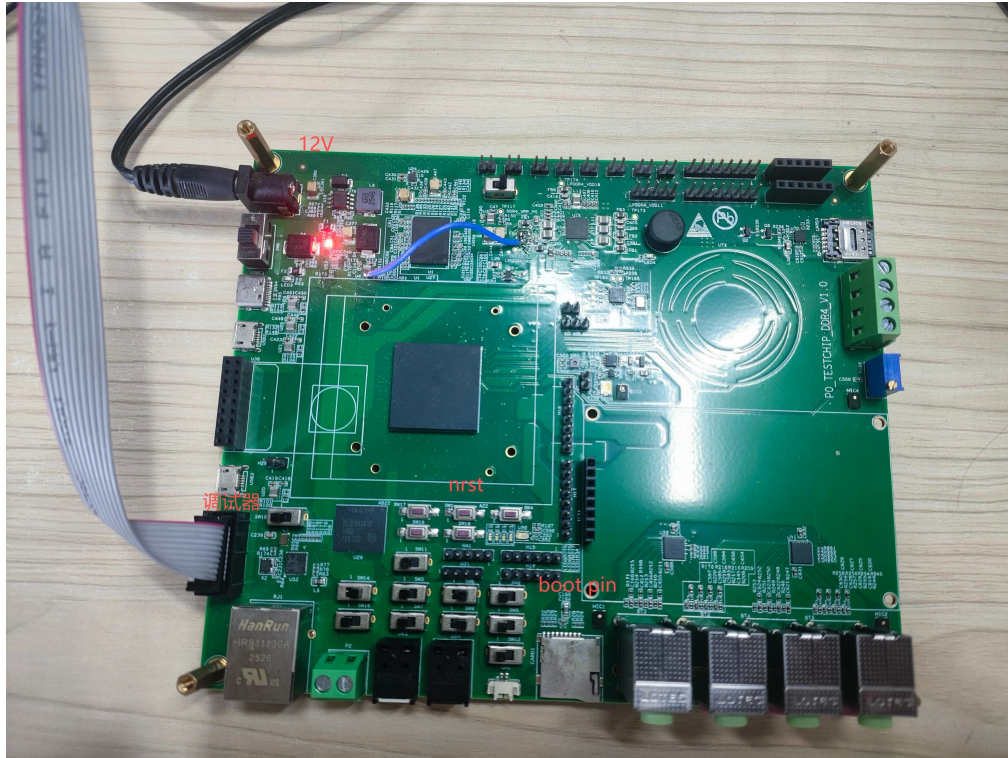
3. P0 EVB 器件及功能说明

针对 EVB 板上的各开关、排插、接口等器件，以 DC1 为左上角，自左到右，自上而下，依次说明各器件功能，如下表所列。

器件编号	功能说明	备注
DC1	直流电源	
P1	蜂鸟调试器插座	
FPC1	液晶显示排插座	
H14	通用复用引脚引出口，对应芯片 IOMUX49-IOMUX63 口	
H10	通用复用引脚引出口，对应芯片 IOMUX95-IOMUX105 口	
SW6	拨码拨到 1：使用直流 5V 供电并使用 JTAG 下载 拨码拨到 3：使用 USB 进行供电并下载	
J1	RJ45 10/100M 以太网口，网络芯片为 RTL8201F-VB-CG	
CARD1	Mini SD 卡槽	
USB2	供电及 USB 数据传输	
USB3	供电及 USB 数据传输	
H9	通用复用引脚引出口，对应芯片 IOMUX12-IOMUX15 口	
H15	通用复用引脚引出口，对应芯片 IOMUX_AOND_0-IOMUX_AOND_4 口	
SW4	配置 GPIO_INPUT 的高低电平（按下为配置高电平）	
SW1	复位开关（按下芯片复位）	
H12	通用复用引脚引出口，对应芯片 IOMUX127-IOMUX133 口	
H16	内部电源测试	
H13	通用复用引脚引出口，对应芯片 IOMUX151-IOMUX168 口	
RJ1	RJ45 10/100/1000M 以太网口, 网络芯片为 RTL8211EG-VE-CG	
U31	1：监测芯片内置的 BROM 是否完成对 eFuse 的读取 2：HSI 旁路模式信号传输 3：LSI 旁路模式信号传输 4：VSS	
SW10	对应 IOMUX_126，DDR 模块的 JTAG 调试通路开关（拨到 1 关闭）	
SW11	HSI 旁路模式开关（拨到 1 为关闭）	
SW7	BOOT3 启动开关, 对应 IOMUX_156	
SW12	LSI 旁路模式开关（拨到 1 关闭）	
H11	通用复用引脚引出口，对应 IOMUX_109，IOMUX_119，IOMUX_121	
SW5	未使用	
SW3	BOOT2 启动开关, 对应 IOMUX_157	
SW8	BOOT0 启动开关, 对应 IOMUX_159	
SW9	BOOT1 启动开关, 对应 IOMUX_158	

4. EVB 板使用说明

4.1 硬件配置



1、JTAG 设置

SW6 拨向 1；连接电源；连接 JTAG

2、运行模式设置

SW10/SW11/SW7/SW12/SW3 均拨向 1（正向 0）；
SW8/SW9 均拨向反向。

4.2 软件配置

一、环境配置

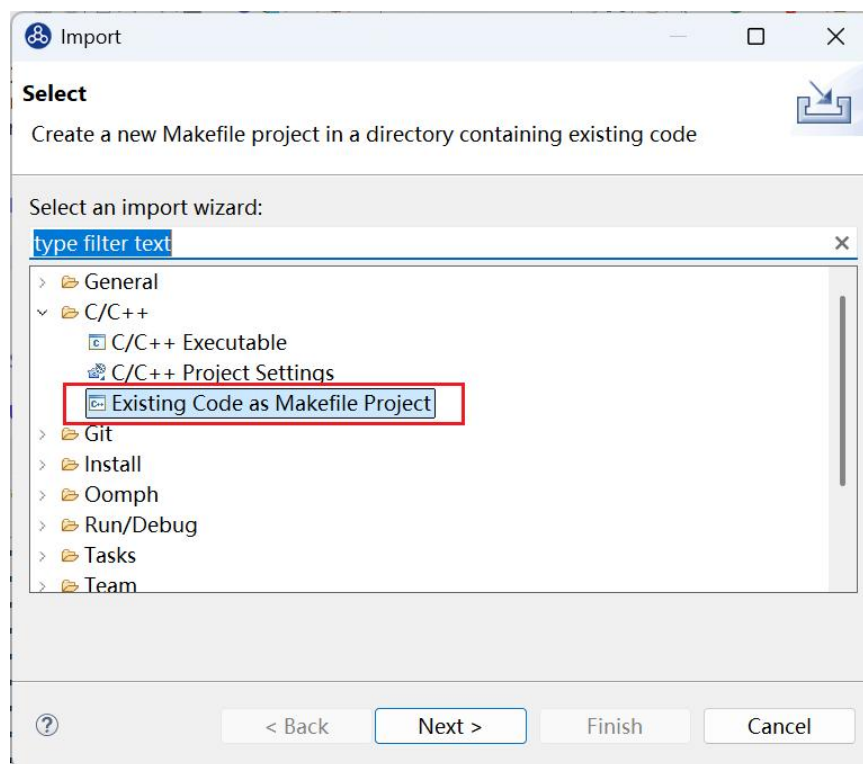
NucleiStudio: 2025.10

SDK: p0-main

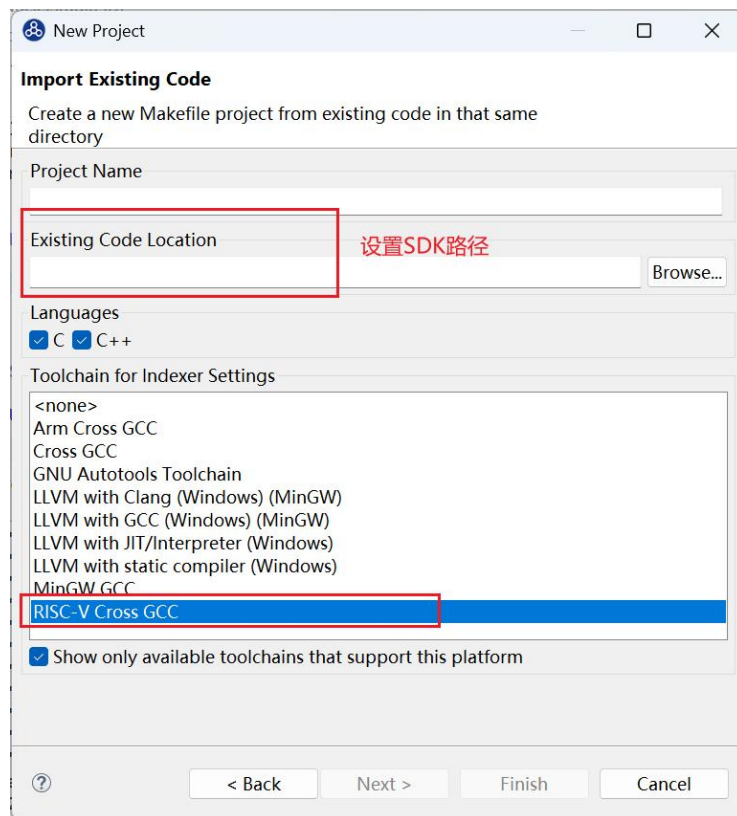
二、编译

1、SDK 包导入 Nuclei Studio

- ① 以“Existing Code as Makefile Project”方式导入 SDK 包。



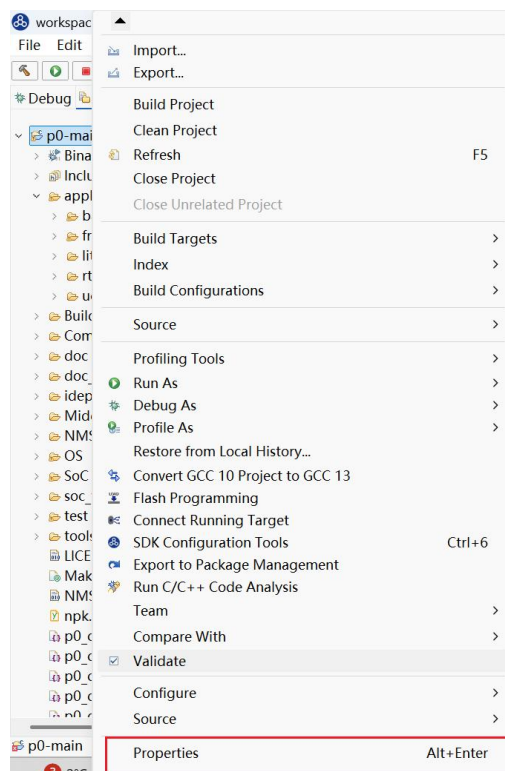
- ② 设置 SDK 路径，并以“RISC-V Cross GCC”配置 toolchain。



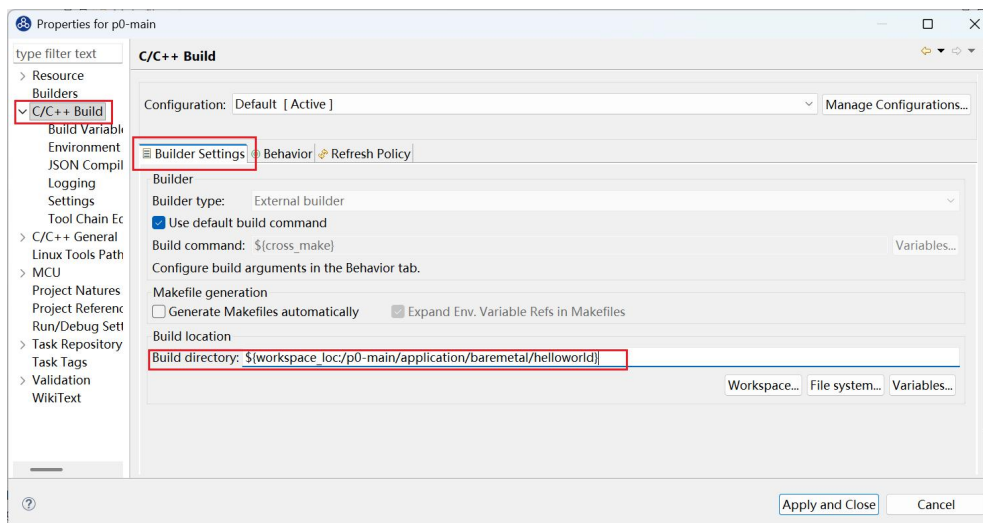
2、配置工程编译环境

(1) workspace 设置

- ① 在项目根目录右击，选择 properties 进行相关配置。

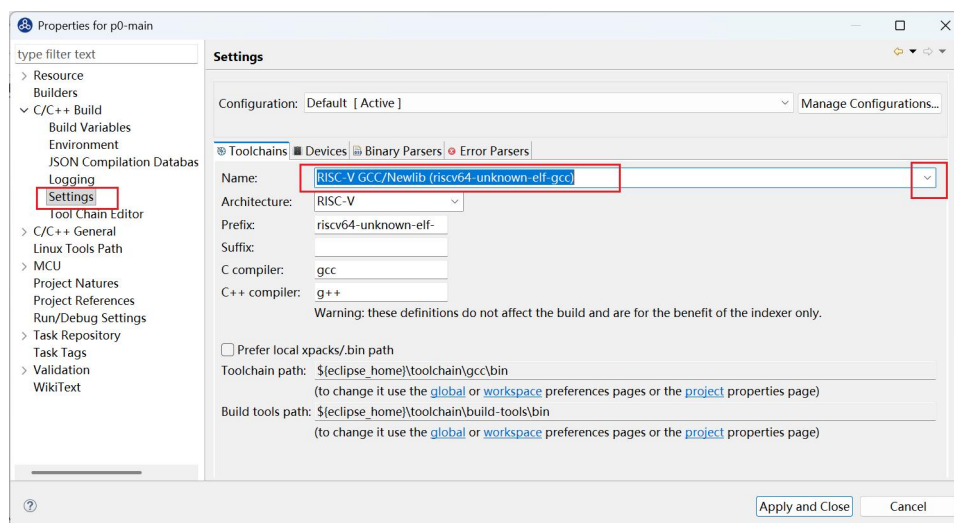


- ② 在 C/C++Build 中设置编译的路径，该路径为项目根目录下的 application 下的 demo 路径。以 helloworld 工程为例，配置如图所示。



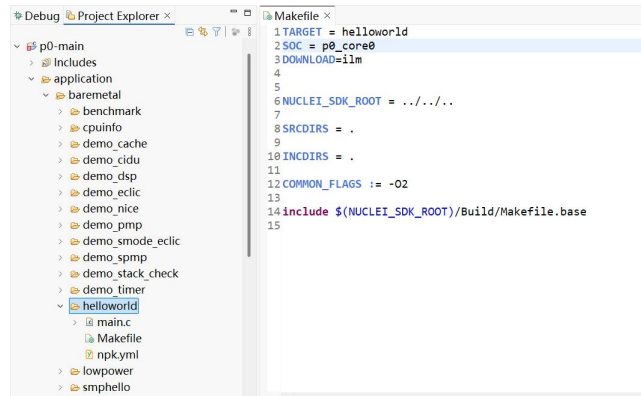
(2) Toolchain 设置

在 C/C++Build 下选择 Settings 进行 toolchain 的配置。点击 Name 后面的下拉箭头选择“RISC-V GCC/Newlib (riscv64-unknown-elf-gcc)”。选择完成后点击右下角“Apply and Close”应用配置并关闭。



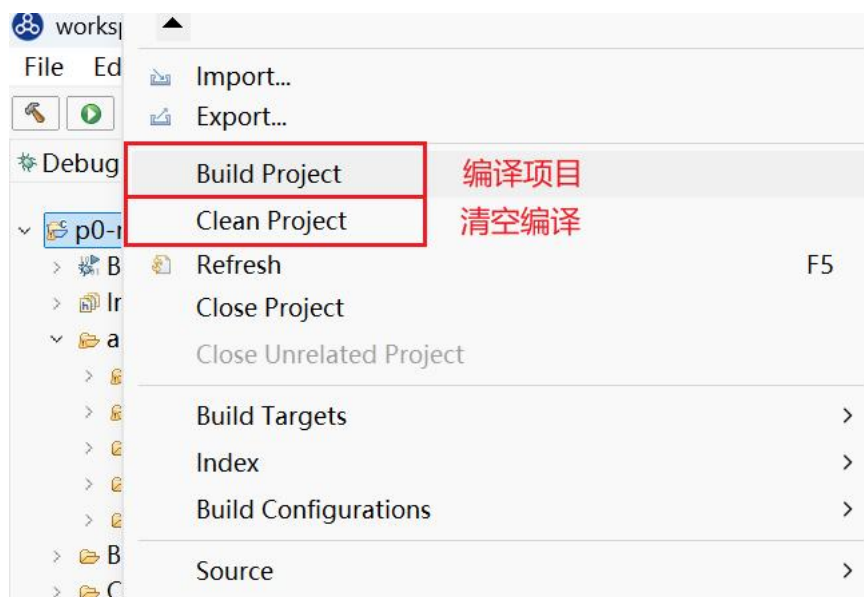
3、设置 Makefile

在要编译的 workspace 下找到 Makefile 文件进行相关配置，主要是指定核、下载模式、操作系统等。配置示例：将 helloworld 工程下的 SOC 设置为 p0_core0，DOWNLOAD 设置为 ilm 如下图所示。

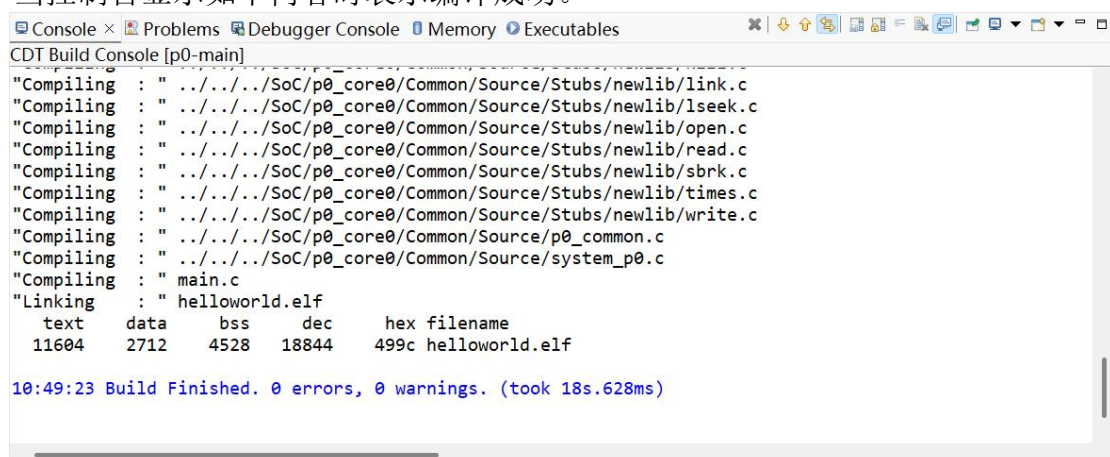


4、编译

右击项目根目录，可以看到“Build Project”和“Clean Project”分别是项目编译和项目清空。在对项目进行较大的改动后需要先清空编译在重新进行编译。



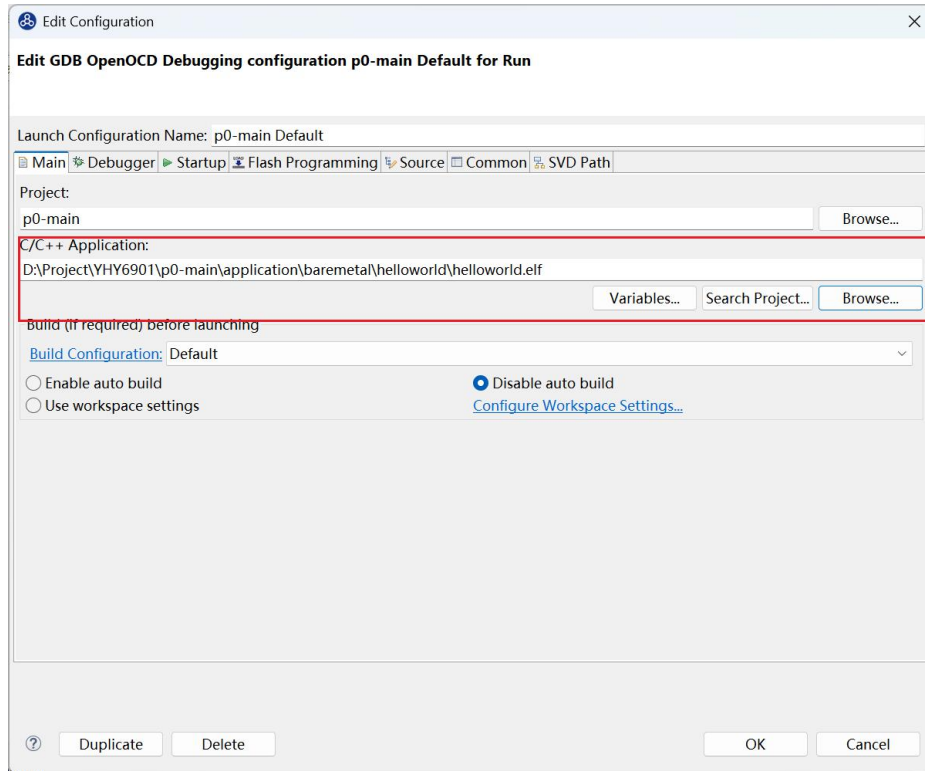
当控制台显示如下内容时表示编译成功。



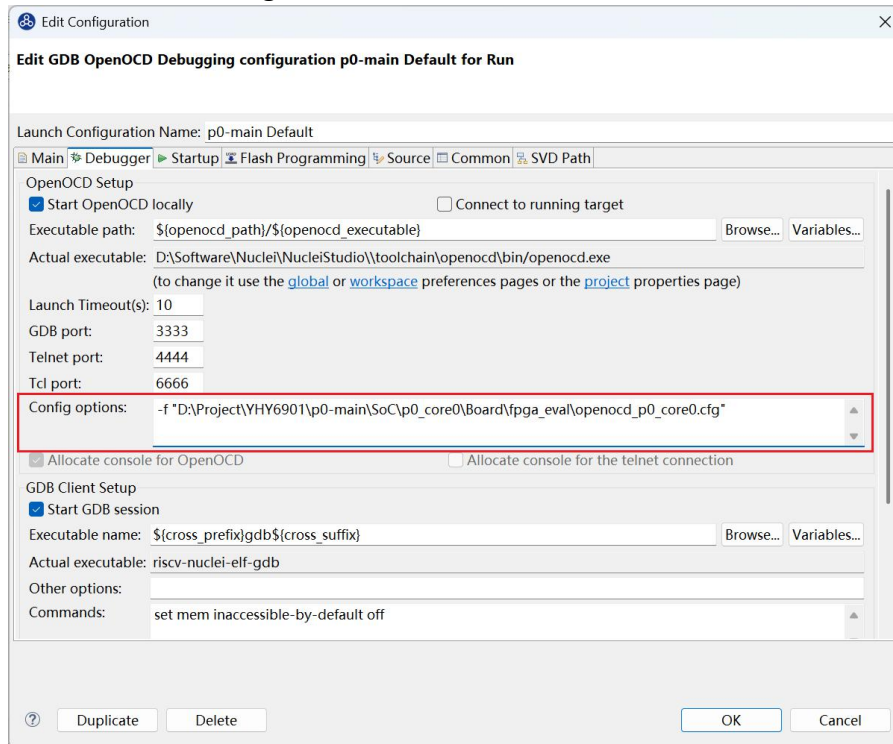
三、烧录

用 NucleiStudio 进行烧录，主要是配置好 launch 文件。在 launchbar 中进行相关配置。配置过程如下：

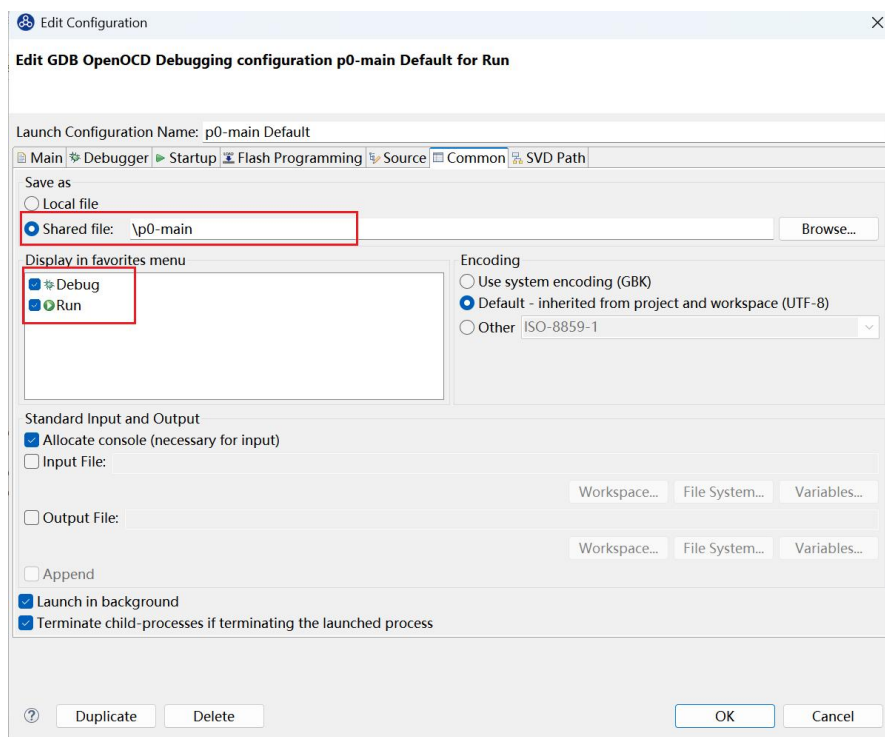
1、指定要烧录的文件



2、指定与核对应的 cfg 文件



3、若是新建的 launch 文件，可以将这个文件保存到根目录下以进行快速选择。



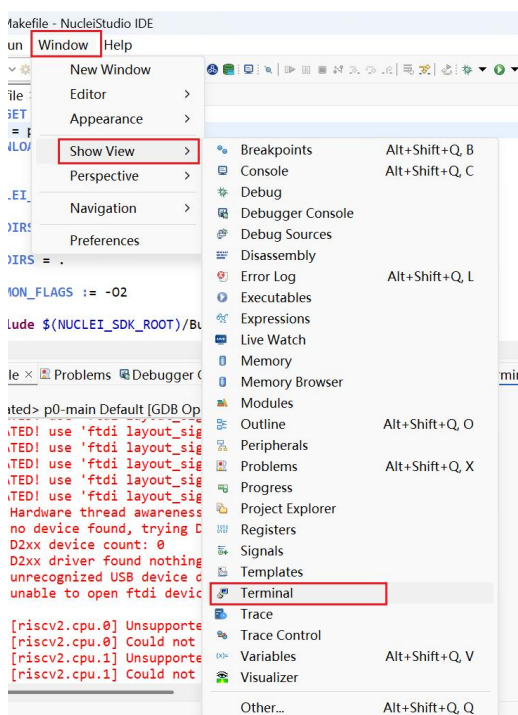
配置完成后点击下图所选按钮进行烧录



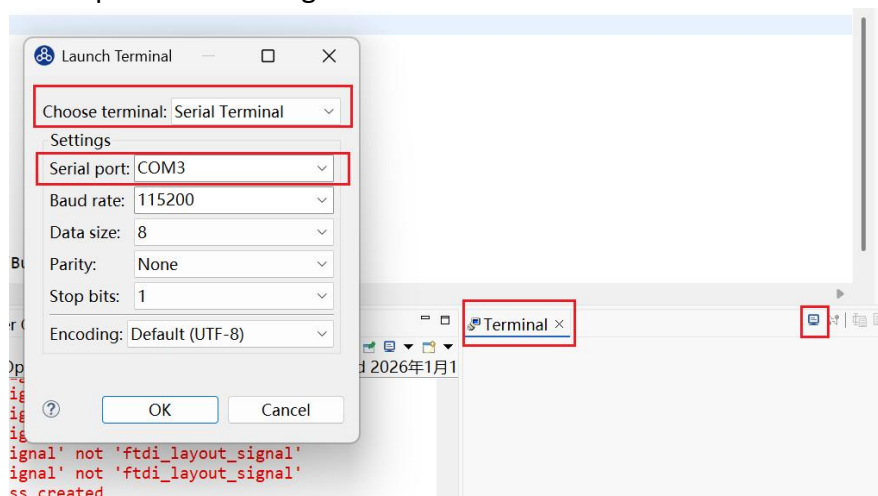
四、烧录结果查看

NucleiStudio 有自带的串口查看页面。

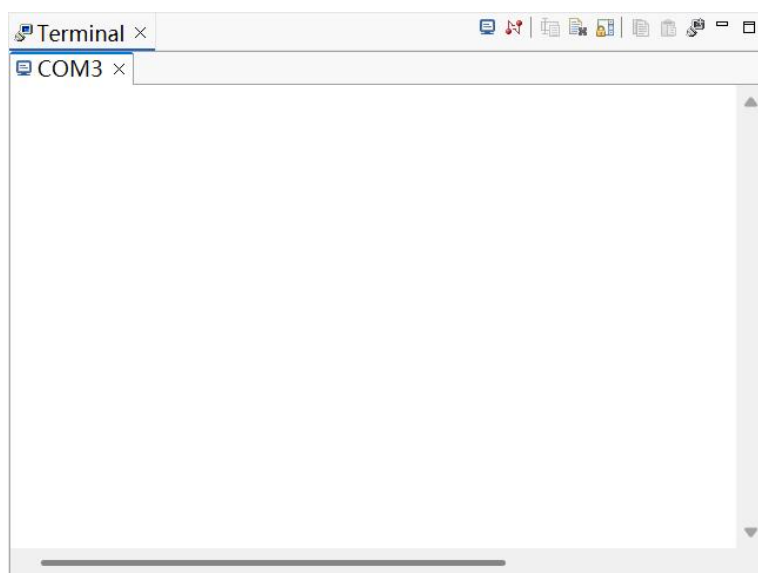
1、按如下方式打开“Window——Show View——Terminal”。



- 2、点击 Terminal 窗口旁边的配置按钮。
- (1) 在 Choose terminal 下拉选择 Serial Terminal。
 - (2) 在 Serial port 中选择 JTag 所在的 COM 口。



配置完成后 Terminal 界面如下：



4.3 应用示例

现在以 RTOS 操作系统和 linux 操作系统的编译烧录运行为例，详细说明在 P0 EVB 板上的操作流程。

一、RTOS 操作系统应用示例

首先，确保已完成 4.1 硬件配置中的 JTAG 设置和运行模式设置，即 SW6 拨向 1，SW10、SW11、SW7、SW12、SW3 均拨向 1，SW8、SW9 均拨向反向，并正确连接 JTAG 调试器与电源。

在软件配置方面，编译步骤如下：

(1) SDK 包导入 Nuclei Studio。

(2) 配置工程编译环境：

workspace 设置：在导入的项目根目录上右击，选择“Properties”。在“C/C++ Build”中，将编译路径设置为项目根目录下的“**applications\liteos_m\demo**”路径，如果没有 liteos 请选择其它 RTOS 或者 demo。

Toolchain 设置不变。

(3) 设置 Makefile：“applications\liteos_m\demo”目录下的 Makefile 文件。根据 EVB 板的硬件配置，将 SOC 设置为“**p0_core0**”，**DOWNLOAD** 设置为“**ilm**”，**RTOS 设置为“FreeRTOS”**。

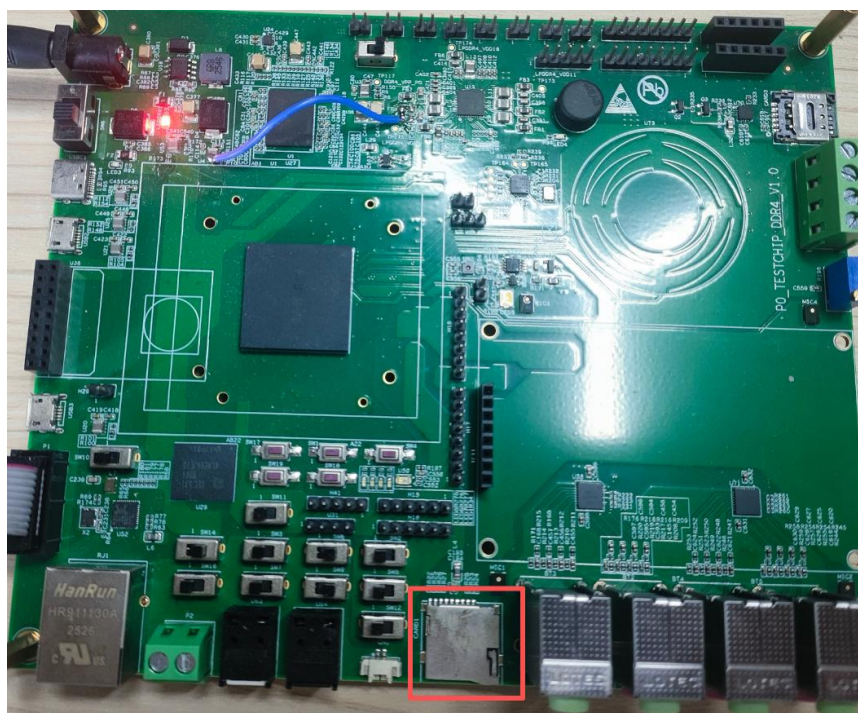
(4) 编译：右击项目根目录，选择“Build Project”。编译过程中，控制台会输出编译信息，若最终显示“Build Finished”且无错误提示，则表示编译成功，生成的可执行文件通常位于“applications\freertos\demo”目录下。

(5) 烧录前需要提前将 Terminal 窗口打开，烧录步骤与 4.2 中描述的通用烧录流程一致。

二、Linux 操作系统应用示例

Linux 操作系统所在的 SDK 包为：P0_LinuxSDK_IMG-1226

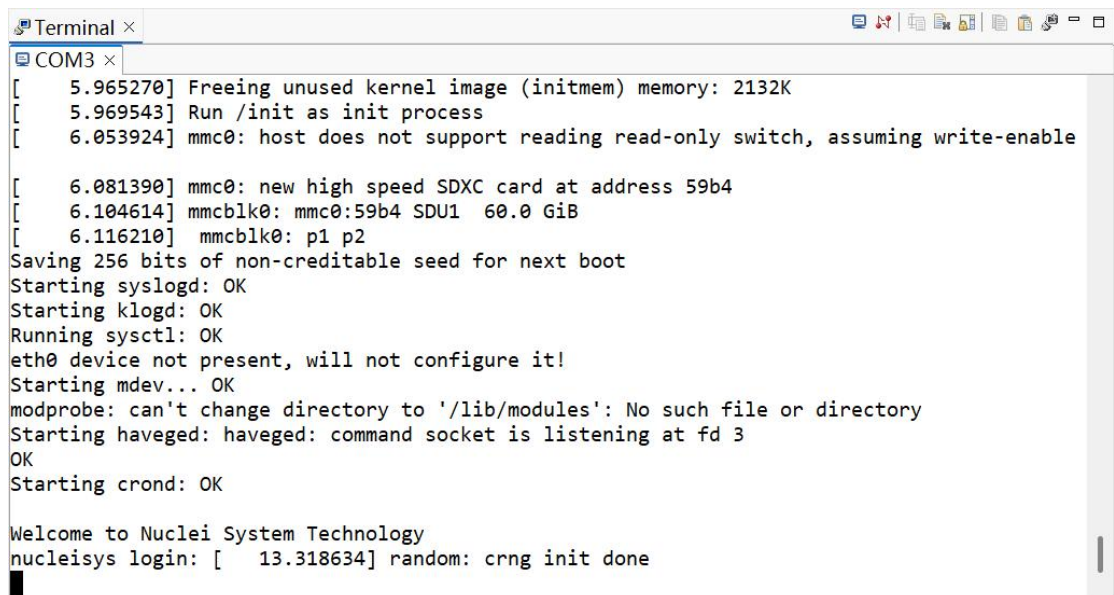
硬件配置与 RTOS 示例相同，确保 JTAG 连接和拨码开关设置正确。除此之外，运行 linux 还需要使用一张 SD 卡。使用读卡器，将 SDK 包中的 boot.scr,ulmage.lz4,ulnitrd.lz4 存放在 SD 卡上。并将 SD 卡插入 CARD1。CARD1 位置如下图所示。



软件方面，打开 NucleiStudio 后，需要进行两次烧录。

(1) 配置 launch 文件，要烧录的文件为 core0_boot.elf，其配套的 cfg 文件为 openocd_p0_core0.cfg，烧录完成后重新上电或者按 sw1 复位启动，启动 N300 程序，观察打印 "core0 boot init ok",再进行后续操作。

(2) 再重新配置 launch 文件，烧录的文件为 freeloader.elf，其配套的 cfg 文件为 openocd_p0_cluster2_core0.cfg，烧录完成后重新上电或者按 sw1 复位启动。最终串口输出结果如下。



```
Terminal x
COM3 x
[ 5.965270] Freeing unused kernel image (initmem) memory: 2132K
[ 5.969543] Run /init as init process
[ 6.053924] mmc0: host does not support reading read-only switch, assuming write-enable

[ 6.081390] mmc0: new high speed SDXC card at address 59b4
[ 6.104614] mmcblk0: mmc0:59b4 SDU1 60.0 GiB
[ 6.116210] mmcblk0: p1 p2
Saving 256 bits of non-creditable seed for next boot
Starting syslogd: OK
Starting klogd: OK
Running sysctl: OK
eth0 device not present, will not configure it!
Starting mdev... OK
modprobe: can't change directory to '/lib/modules': No such file or directory
Starting haveged: haveged: command socket is listening at fd 3
OK
Starting crond: OK

Welcome to Nuclei System Technology
nucleisys login: [ 13.318634] random: crng init done
```


5. 常见问题解答

5.1 烧录串口输出乱码

若在烧录时出现串口输出乱码的情况，需重点从以下几个方面进行检查：

- （1）USART0 重复初始化。若运行的是 CPU0 以外的核，需检查是否已将 `system_p0.c` 中的 USART0 初始化代码注释掉，每次使用 `p0_core0` 来初始化 USART0，不需要重复初始化。
- （2）芯片复位。按下 SW1 使芯片复位，重新查看输出情况。