

UG

# HPM6750

HPM6750EVK2 用户使用手册

---

适用于先楫半导体 HPM6750 系列高性能微控制器

# 目录

目录.....	2
表格目录.....	3
第一章 HPM6750EVK2 简介.....	6
第二章 硬件电路.....	8
2.1 电路模块介绍 .....	8
第三章 软件开发套件.....	14
3.1 简介.....	14
3.2 环境以及依赖 .....	14
3.3 开发工具 .....	14
3.4 sdk_env/Segger Embedded Studio For RISC-V 使用快速指南.....	14
3.5 调试出错常见原因.....	24
1. FT2232驱动没有正确安装.....	24
2. Boot Pin配置异常.....	24
3. 调试没有正常退出 .....	25
4. Debug电阻没有正确配置.....	25
5. openocd没有正确配置 .....	25
6. J-Link调试器没有正确配置 .....	26
3.6 更新 sdk_env 中的 SDK/toolchain 指南.....	28
3.7 版本信息 .....	32
第四章 免责声明.....	33

## 表格目录

表 1: 主要器件位号对应器件功能名称 .....	7
表 2: 启动配置表.....	12
表 3: 电机接口管脚列表.....	12
表 4: 扩展IO表 .....	13
表 5: 版本信息 .....	32

## 图片目录

图 1: 顶层器件位置图.....	6
图 2: 底层器件位置图.....	6
图 3: HPM6360EVK 硬件设计框图.....	8
图 4: 安装FTDI 驱动.....	15
图 5: 查看Windows 设备管理器.....	15
图 6: sdk_env创建工程方式.....	16
图 7: 打开sdk prompt.....	16
图 8: 构建目标板工程.....	17
图 9: 构建目标板flash_xip 工程.....	17
图 10: generate_project 帮助.....	18
图 11: Segger Embedded Studiohello_world 工程.....	18
图 12: Segger Embedded Studio 打开hello_world 工程.....	19
图 13: Segger Embedded Studio 编译hello_world 工程.....	19
图 14: Segger Embedded Studio 调试hello_world 工程.....	20
图 15: Segger Embedded Studio 配置串口.....	20
图 16: Segger Embedded Studio 连接串口.....	21
图 17: Segger Embedded Studio 打开串口.....	21
图 18: Segger Embedded Studio 运行hello_world.....	22
图 19: start_gui 工具.....	22
图 20: GUI project generator工具操作界面.....	23
图 21: GUI project generator生成hello_world工程.....	23
图 22: hello_world工程.....	24
图 23: GDB Server连接失败.....	24
图 24: 查看openocd配置.....	25
图 25: GDB Server默认配置.....	26
图 26: 使用cmsis-dap 调试器GDB Server配置.....	26
图 27: J-Link驱动下载.....	26
图 28: J-Link驱动安装.....	27
图 29: Target Connection 设置为J-Link.....	27
图 30: Target Interface Type设置为JTAG.....	27
图 31: 更新SDK.....	28

---

图 32: 拷贝toolchain .....	29
图 33: 更新start_cmd.cmd中TOOLCHAIN_NAME .....	29
图 34: 更新start_gui.exe中TOOLCHAIN_NAME .....	30
图 35: 选择TOOLCHAIN目录 .....	30
图 36: start_gui.exe更新TOOLCHAIN完成 .....	31

# 第一章 HPM6750EVK2 简介

HPM6750EVK2 板的器件位置如 图1, 图2 所示。表1 给出了器件位置对应器件的名称。

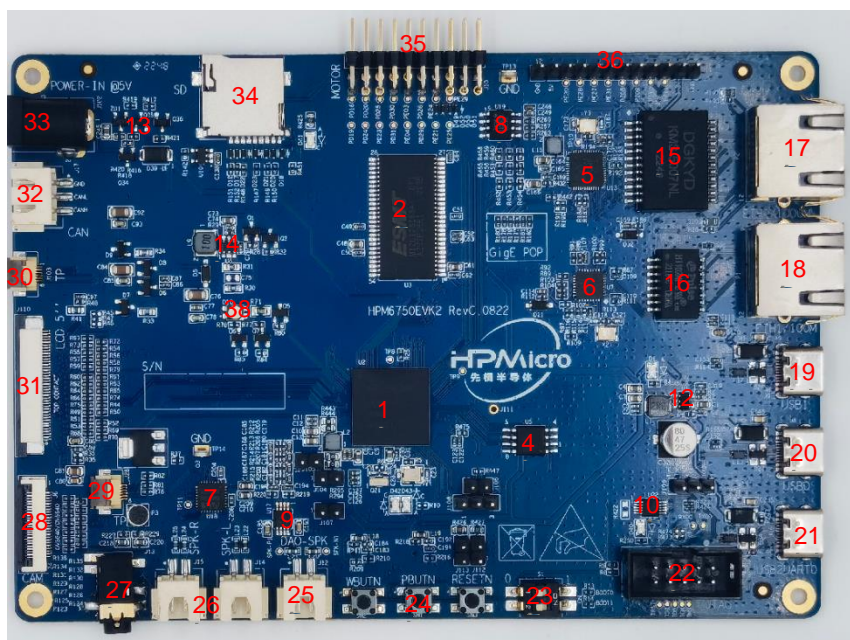


图 1: 顶层器件位置图

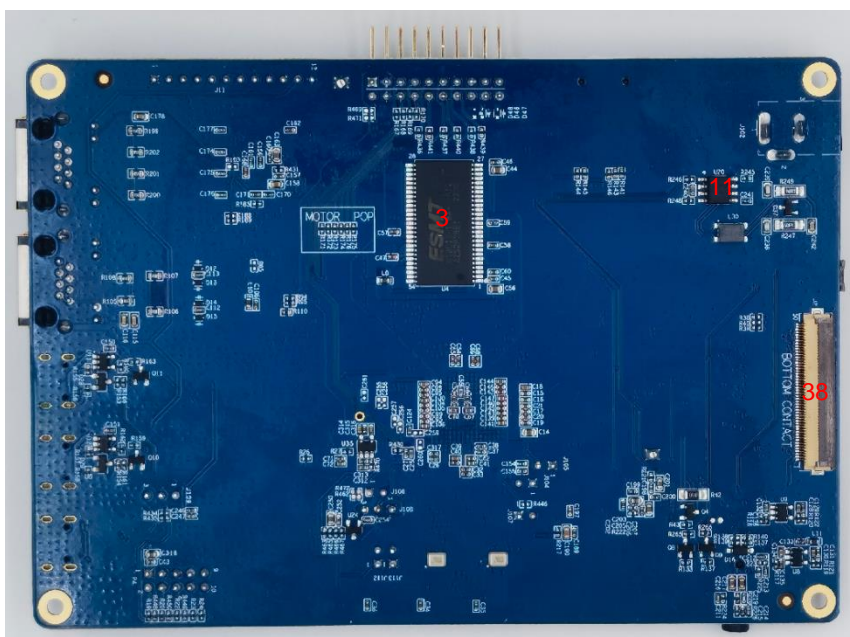


图 2: 底层器件位置图

序号	名称	序号	名称
1	HPM6750	2/3	SDRAM, 256Mb, 32bit
4	FLASH, 128Mb	5	RGMII PHY

6	RMII PHY	7	AUDIO CODEC
8	EEPROM,	9	POWER AMPLIFIER
10	USB-UART	11	CAN收发器
12	DCDC, 3V3输出	13	过压保护
14	DCDC, 10V4输出	15	千兆网变压器
16	百兆网变压器	17	千兆网口
18	百兆网口	19	USB1 Type-C
20	USB0 Type-C	21	DEBUG Type-C
22	JTAG接口	23	BOOT 配置拨码开关
24	按键	25	DAO接口
26	左右声道	27	耳机接口
28	摄像头接口	29/30	触摸屏接口
31	LCD接口(无背光电压输出)	32	CAN接口
33	DC输入	34	T-Flash卡槽
35	马达接口	36	扩展IO接口
37	三色LED	38	LCD接口(带背光电压输出)

表 1: 主要器件位号对应器件功能名称

## 第二章 硬件电路

HPM6750EVK2 可以通过外接5V直流供电，也可以通过USB供电。板载的3个USB Type-C均可以作为电源供电。HPM6750的I/O 接口均为 3.3V 电平，如外接其他设备，需确保电平匹配。如不匹配可能导致不能正常工作或损坏芯片。

### 2.1 电路模块介绍

#### 2.1.1 系统架构

HPM6750EVK2 系统架构如图3所示。

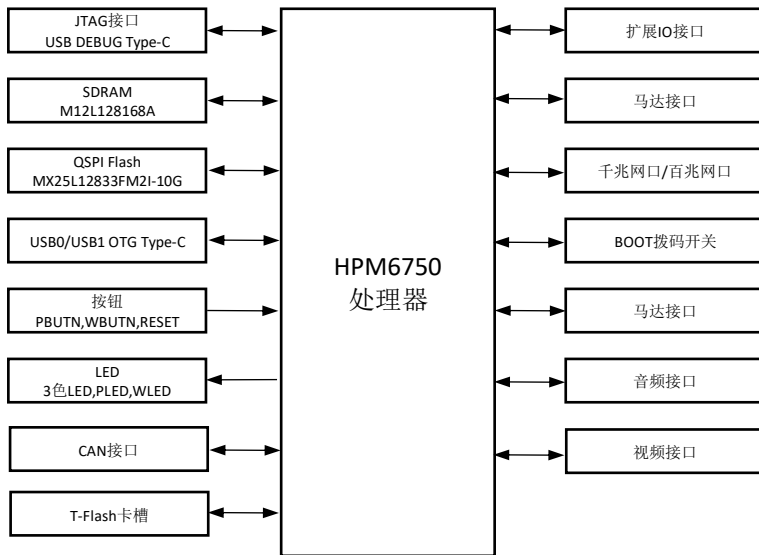


图 3: HPM6750EVK2 硬件设计框图

#### 2.1.2 电源

HPM6750EVK2供电方式如下

- 外接DC5V输入
- USB1 Type-C
- USB0 Type-C
- DEBUG Type-C



### 2.1.3 SDRAM

HPM6750EVK2 板载2颗16bit SDRAM，容量 256Mb，型号 M12L128168A。

注：由于市场供应链问题，先楫半导体会根据实际情况变更SDRAM料号，但是确保用户使用没有影响。

### 2.1.4 数据存储

HPM6750EVK2 板载单颗QSPI接口 NOR FLASH，容量 128Mb，型号MX25L12833FM

HPM6750EVK2 板载单颗EEPROM，容量 2Kb，型号24C02。

### 2.1.5 USB 接口

HPM6750EVK2 板载2个USB 接口，连接器类型均是Type-C。支持USB 2.0 OTG。同时USB0支持USB串行下载和ISP，即通过USB0下载固件，下载工具通过官网获取。

### 2.1.6 DEBUG

HPM6750EVK2 板载 DEBUG 接口，通过 Tpye-C链接USB转UART(U22)。该UART也可用于UART串行下载和ISP。即通过UART0下载固件，下载工具通过官网获取。

HPM6750EVK2支持10pin的JTAG接口(P4)。用户可以通过该接口连接调试器。

HPM6750EVK2 REVC

1	5V	TMS	2
3	GND	TCK	4
5	GND	TDO	6
7	URT0.TXD	TDI	8
9	URT0.RXD	TRST	10

注：如果用户需要使用P4的UART功能，需要焊接R473,R474方可使用

HPM6750EVK2 REVB

1	3V3	TMS	2
---	-----	-----	---

3	GND	TCK	4
5	GND	TDO	6
7	NC	TDI	8
9	NC	TRST	10

### 2.1.7 网口

HPM6750EVK2 板载百兆网口，网口芯片为RTL8201F-VB-CG。

HPM6750EVK2 板载千兆网口，网口芯片为RTL8211E-VB-CG。

HPM6750可以支持2个网口同时工作。

### 2.1.8 音频接口

HPM6750EVK2支持音频输入输出

- Audio Codec(WM8960)
  - 3.5mm 耳麦
  - 左,右声道(1W@8Ω每声道)
  - Mic
- DAO
  - 3W@4Ω
- Digital Mic
  - 2 \* SPH0641LU4H

注：外置喇叭均由5V供电，因此用户在使用时要确保5V的带载能力

### 2.1.9 视频接口

HPM6750EVK2支持视频输入输出

- 摄像头(J6), DVP接口
- 触摸屏(J26,J103)

- LCD接口, 支持24bit
  - 接口1(J110), 接口不支持背光
  - 接口2(J2), 接口支持背光

注: LCD接口不能同时使用。

### 2.1.10 CAN总线接口

HPM6750EVK2 板载CAN总线接口, 通过J17端子引出, CAN总线芯片为TJA1042T-3。

### 2.1.11 TF接口

HPM6750EVK2 板载T-Flash接口。

### 2.1.12 LED

HPM6750EVK2 板载功能型LED如下

- 三色LED(D10)
- PLED(D42)
- WLED(D43)

### 2.1.13 按键

HPM6750EVK2 板载3个按键, SW1为PBUTN按键, SW2为WBUTN按键, SW3为RESET按键。

用户通过 PBUTN 按键实现对 MCU 内部的电源进行管理。当系统处于运行状态时, PBUTN 上检测到一次有效的超长按键 (输入保持低电平约 16 秒), 就会指示电源管理系统关闭电源管理域的各个电源, 使系统休眠状态。

用户通过 WBUTN 按键实现对 MCU 的唤醒。当系统处于掉电状态时, 唤醒按键 WBUTN 上检测到一次有效的按键 (输入保持低电平约 0.5 秒), 可以重新打开电源域 VPMC 里的各个电源, 使系统重新工作。

上述功能需要经过软件配置方可使用, 另外, 用户也可以把它们当作普通的按键使用。

用户通过 RESET 按键对 MCU 进行外部复位。

### 2.1.14 BOOT 拨码开关设置

芯片默认是通过S1 拨码开关设置对应BOOT\_MODE[1:0]=[PZ07:PZ06] 引脚选择启动模式，配置如表2。

S1 拨码开关[1:0]		启动模式	说明
OFF	OFF	XPI NOR 启动	从连接在XPI0/1 上的串行NOR FLASH 启动
OFF	ON	串行启动UART0/USB-HID	从UART0/USB0 上启动
ON	OFF	在系统编程 (ISP)	从UART0/USB0 上烧写固件，OTP
ON	ON	保留模式	保留模式

表 2: 启动配置表

### 2.1.15 电机控制接口

HPM6750EVK2 板载电机20PIN控制接口，6路PWM输出，4路ADC，6路互联管理。

引脚名	功能名	连接器编号		功能名	引脚名
-	-	1	2	-	-
GNDA	GNDA	3	4	GNDA	GNDA
PE29	ADC3.INA2	5	6	ADC0.VINP11	PE25
PE24	ADC0.VINP10	7	8	ADC0.VINP7	PE21
PD28	PWM2.P5	9	10	PWM2.P4	PD29
PE03	PWM2.P3	11	12	PWM2.P2	PE04
PD30	PWM2.P1	13	14	PWM2.P0	PD31
PD25/ PE28	TRGM2.P08/ SPI2.MISO	15	16	TRGM2.P11	PD23
PD20/ PE30	TRGM2.P07/ SPI2.MOSI	17	18	TRGM2.P10	PD24
PD16/ PE31	TRGM2.P06/ SPI2.CSN	19	20	TRGM2.P09	PD19

表 3:电机接口管脚列表

注：由于引脚资源受限，用户在使用马达的时候，需要去掉连接在RGMII PHY的电阻，分别为R177,R178,R179,R180,R181,R182

### 2.1.16 扩展 IO 接口

HPM6750EVK2 板载12PIN的扩展 IO 接口。具体信息如表 4所示。

编号	功能名	引脚名
1	GND	
2	3.3V	
3	I2C0.SCL	PZ11
4	I2C0.SDA	PZ10
5	URT13.TXD	PZ09
6	URT13.RXD	PZ08
7	SPI2.CSN	PE31
8	SPI2.SCLK	PE27
9	SPI2.MISO	PE28
10	SPI2.MOSI	PE30
11	5V	
12	GND	

表 4: 扩展IO接口

## 第三章 软件开发套件

### 3.1 简介

HPM SDK (HPM 软件开发套件, 以下简称SDK) 是基于BSD 3-Clause 许可证, 针对HPM 出品的系列SoC 底层驱动软件包, 提供了 SoC 上所集成 IP 模块底层驱动代码, 集成多种中间件与 RTOS。用户可以从官网下载最新的SDK, 本文中所使用的版本是SDK1.0.0。

### 3.2 环境以及依赖

- 使用sdk\_env 工具
- 手工搭建SDK 开发环境, 具体参考请参考SDK 目录下README.md 文件。

### 3.3 开发工具

SDK 支持第三方IDE 开发, 如Segger Embedded Studio For RISC-V, 该IDE 可以在[Segger 官网下载](#)下载最新版本。先楫半导体为开发者购买了商业的license, 用户可以通过邮件的方式, 向Segger 申请license。

由于HPM6750EVK2没有板载调试器, 因此用户需要自备调试器。目前可以支持的调试器有DAP-link, Jlink。

默认的工程使用的调试器是DAP-link, 用户可以参考3.5章节中的提示, 使用Jlink调试。

### 3.4 sdk\_env/Segger Embedded Studio For RISC-V 使用快速指南

1. 下载安装 Segger Embedded Studio For RISC-V
2. 下载的sdk\_env.zip 解压

注: 解压目标路径中只可包含英文字母以及下划线, 不可包含空格、中文等字符。

注: 由于HPM6750EVK2本身不包括板载调试器, 故下面描述的第3步可以跳过。对于调试器的驱动的安装, 不在本文的讨论范围内。

3. 运行sdk\_env\tools\FTDI\_InstallDriver.exe 以安装可用于调试的FT2232 驱动, 如图4所示。

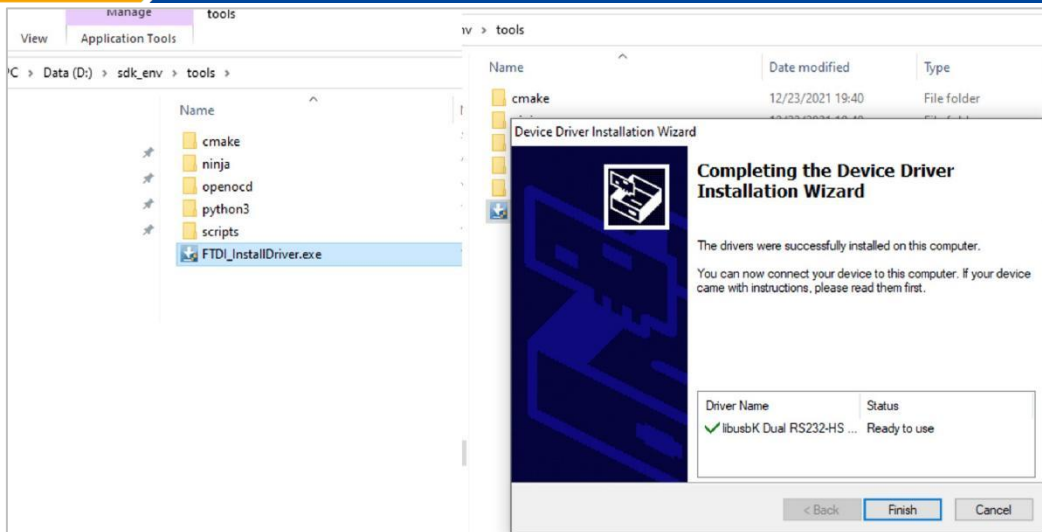


图 4: 安装FTDI 驱动

正确安装驱动后，使用USB Type-C 线缆连接板载的DEBUG USB Type-C接口到 PC 后，在 Windows 设备管理器中应能看到一个USB Serial Port 以及一个 Dual RS232-HS (Interface 0)，如图 5所示:

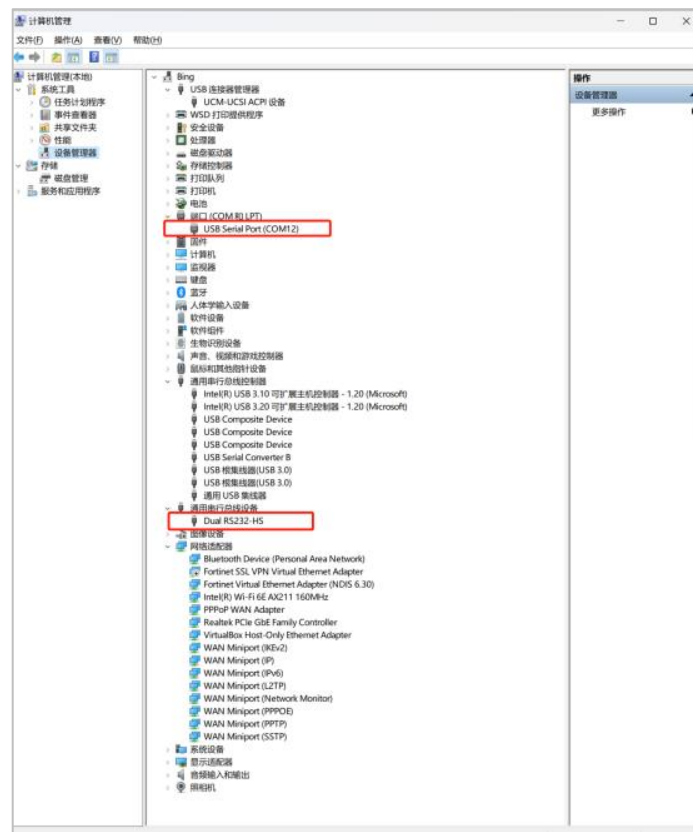


图 5: 查看Windows 设备管理器

4. sdk\_env目录下有两种创建工程的方式，即命令行工具和GUI Project Generator工具,用户可根据自己的喜好选择适合自己的方式。

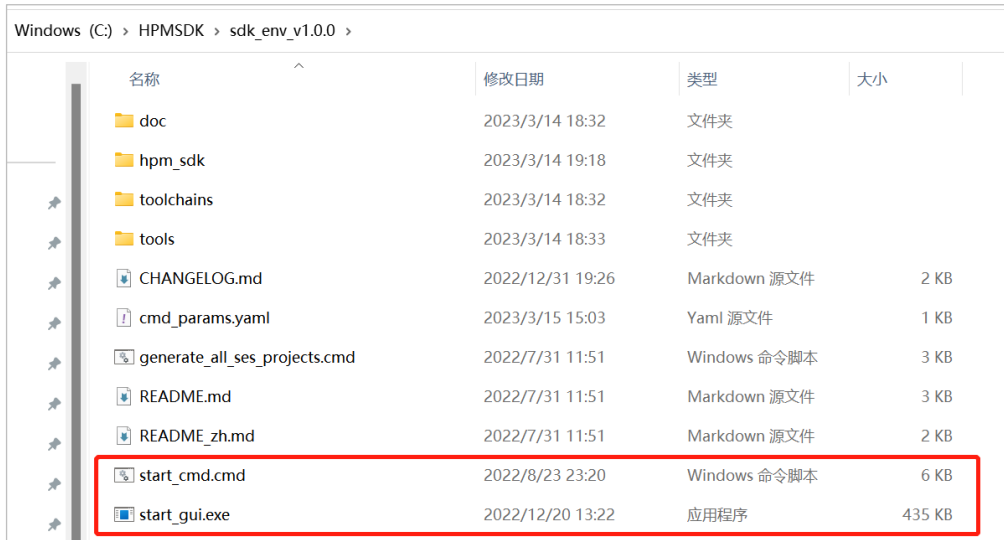


图 6: sdk\_env创建工程方式

以命令行工具为例，双击打开sdk\_env\_v1.0.0下start\_cmd.cmd，该脚本将打开一个Windows command prompt（以下将此Windows cmd prompt 简称为sdk prompt），如果之前步骤配置正确，将会看到图 7 所示。

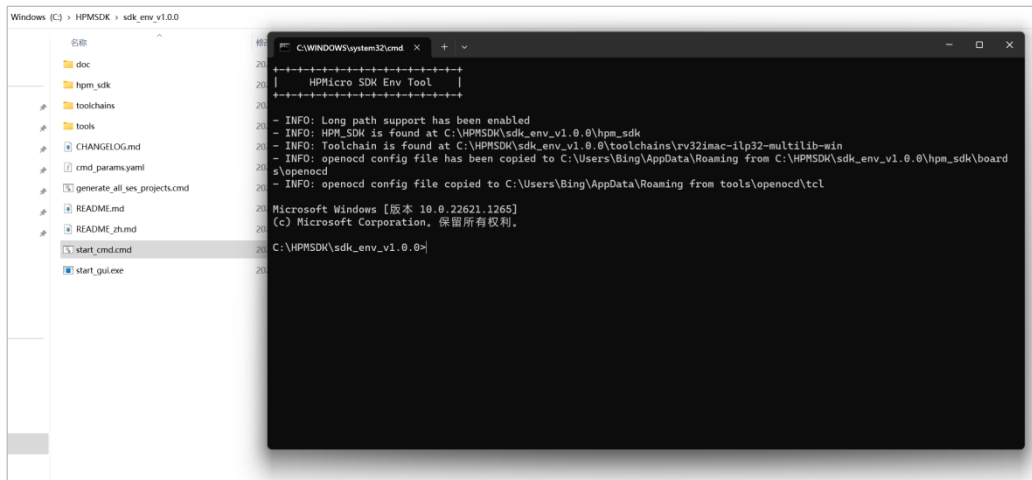


图 7: 打开sdk prompt

5. 在sdk prompt 中切换路径至SDK 具体的一个示例程序，以hello\_world 为例：  
> cd %HPM\_SDK\_BASE%\samples\hello\_world
6. 运行以下命令进行支持目标板查询  
> generate\_project -list
7. 确认目标板名称后（以HPM6750EVK2 为例）可以通过运行以下命令进行工程构建，若构建成功，将看



到如下类似提示

> generate\_project -b hpm6750evk2 -f

```
C:\Windows\system32\cmd.e... X + - □ X
- flash_uf2_release
- flash_sdram_uf2
- flash_sdram_uf2_release
- sec_core_img
- sec_core_img_release

C:\HPMICRO\SDK\V100\sdk_env_v1.0.0\hpm_sdk\samples\hello_world>
C:\HPMICRO\SDK\V100\sdk_env_v1.0.0\hpm_sdk\samples\hello_world>generate_project -b hpm6750evk2
hpm6750evk2_build directory exists, please remove it manually or run this script with -f option
C:\HPMICRO\SDK\V100\sdk_env_v1.0.0\hpm_sdk\samples\hello_world>generate_project -b hpm6750evk2 -f
-- Application: C:/HPMICRO/SDK/V100/sdk_env_v1.0.0/hpm_sdk/samples/hello_world
-- Board: hpm6750evk2
-- Found toolchain: gnu (C:\HPMICRO\SDK\V100\sdk_env_v1.0.0\toolchains\rv32imac-ilp32-multilib-win)
-- The C compiler identification is GNU 11.1.0
-- The CXX compiler identification is GNU 11.1.0
-- The ASM compiler identification is GNU
-- Found assembler: C:/HPMICRO/SDK/V100/sdk_env_v1.0.0/toolchains/rv32imac-ilp32-multilib-win/bin/riscv32-unknown-elf-gc
c.exe
-- hpm_sdk: 1.0.0 (C:\HPMICRO\SDK\V100\sdk_env_v1.0.0\hpm_sdk)
-- Segger linker script: C:/HPMICRO/SDK/V100/sdk_env_v1.0.0/hpm_sdk/soc/HPM6750/toolchains/segger/ram.icf
-- Segger device name: HPM6750xVMX
-- Segger Embedded Studio Project: C:/HPMICRO/SDK/V100/sdk_env_v1.0.0/hpm_sdk/samples/hello_world/hpm6750evk2_build/segger_embedded_studio/hello_world.emProject
-- Configuring done
-- Generating done
-- Build files have been written to: C:/HPMICRO/SDK/V100/sdk_env_v1.0.0/hpm_sdk/samples/hello_world/hpm6750evk2_build

C:\HPMICRO\SDK\V100\sdk_env_v1.0.0\hpm_sdk\samples\hello_world>
```

图 8: 构建目标板工程

注: generate\_project 可以生成多种工程类型, 如: flash\_xip (链接完成后的应用程序将会在norflash 地址空间原地执行), debug (链接完成后的应用程序将会在片上sram中执行, 掉电后程序不能保存) 等。

```
C:\Windows\system32\cmd.e... X + - □ X
-- Found assembler: C:/HPMICRO/SDK/V100/sdk_env_v1.0.0/toolchains/rv32imac-ilp32-multilib-win/bin/riscv32-unknown-elf-gc
c.exe
-- hpm_sdk: 1.0.0 (C:\HPMICRO\SDK\V100\sdk_env_v1.0.0\hpm_sdk)
-- Segger linker script: C:/HPMICRO/SDK/V100/sdk_env_v1.0.0/hpm_sdk/soc/HPM6750/toolchains/segger/ram.icf
-- Segger device name: HPM6750xVMX
-- Segger Embedded Studio Project: C:/HPMICRO/SDK/V100/sdk_env_v1.0.0/hpm_sdk/samples/hello_world/hpm6750evk2_build/segger_embedded_studio/hello_world.emProject
-- Configuring done
-- Generating done
-- Build files have been written to: C:/HPMICRO/SDK/V100/sdk_env_v1.0.0/hpm_sdk/samples/hello_world/hpm6750evk2_build

C:\HPMICRO\SDK\V100\sdk_env_v1.0.0\hpm_sdk\samples\hello_world>generate_project -b hpm6750evk2 -t flash_xip -f
-- Application: C:/HPMICRO/SDK/V100/sdk_env_v1.0.0/hpm_sdk/samples/hello_world
-- Board: hpm6750evk2
-- Found toolchain: gnu (C:\HPMICRO\SDK\V100\sdk_env_v1.0.0\toolchains\rv32imac-ilp32-multilib-win)
-- The C compiler identification is GNU 11.1.0
-- The CXX compiler identification is GNU 11.1.0
-- The ASM compiler identification is GNU
-- Found assembler: C:/HPMICRO/SDK/V100/sdk_env_v1.0.0/toolchains/rv32imac-ilp32-multilib-win/bin/riscv32-unknown-elf-gc
c.exe
-- hpm_sdk: 1.0.0 (C:\HPMICRO\SDK\V100\sdk_env_v1.0.0\hpm_sdk)
-- Segger linker script: C:/HPMICRO/SDK/V100/sdk_env_v1.0.0/hpm_sdk/soc/HPM6750/toolchains/segger/flash_xip.icf
-- Segger device name: HPM6750xVMX
-- Segger Embedded Studio Project: C:/HPMICRO/SDK/V100/sdk_env_v1.0.0/hpm_sdk/samples/hello_world/hpm6750evk2_build/segger_embedded_studio/hello_world.emProject
-- Configuring done
-- Generating done
-- Build files have been written to: C:/HPMICRO/SDK/V100/sdk_env_v1.0.0/hpm_sdk/samples/hello_world/hpm6750evk2_build

C:\HPMICRO\SDK\V100\sdk_env_v1.0.0\hpm_sdk\samples\hello_world>
```

图 9: 构建目标板flash\_xip 工程

注: 当调试flash 目标时, 建议把启动配置 (具体请参考表2) 拨为在系统编程 (ISP) 模式, 以免flash 内已烧录的程序对当前调试过程产影响。

注: 更多generate\_project使用方法可以通过执行以下命令查看

> generate\_project -h

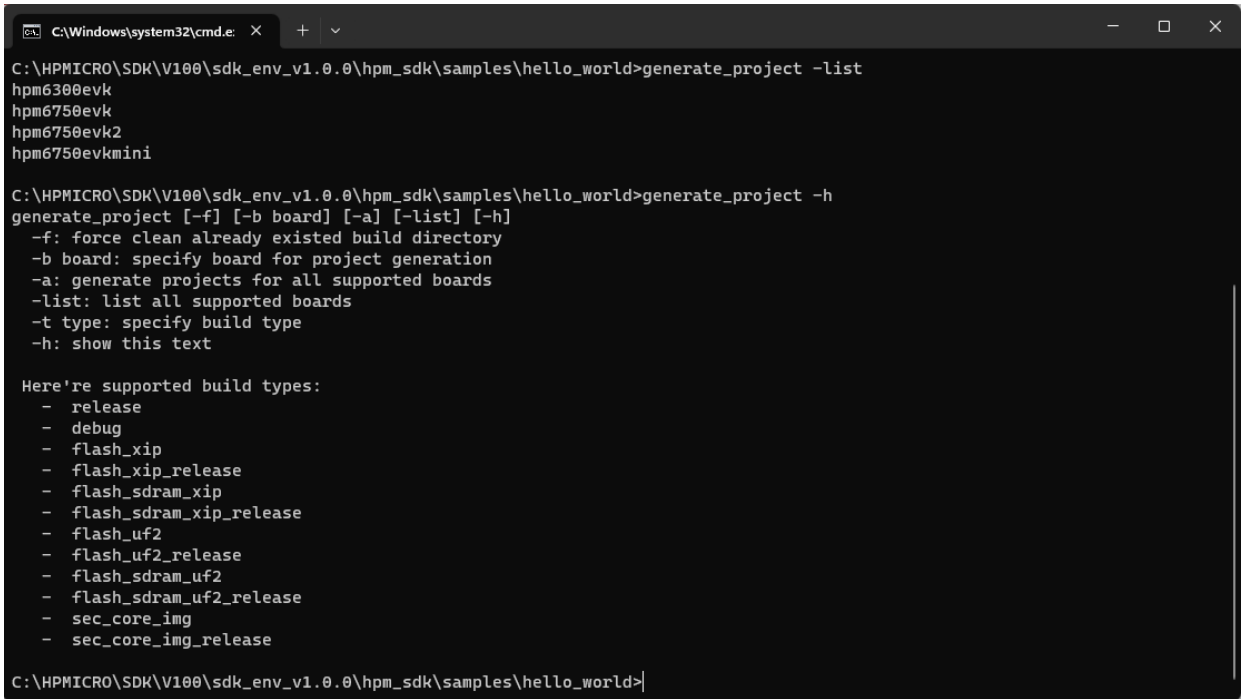


图 10: generate\_project帮助

- 8. 当前目录下将生成名为hpm6750evk2\_build的目录。该目录下segger\_embedded\_studio的目录中可找到Segger Embedded Studio的工程文件, hello\_world.emProject。

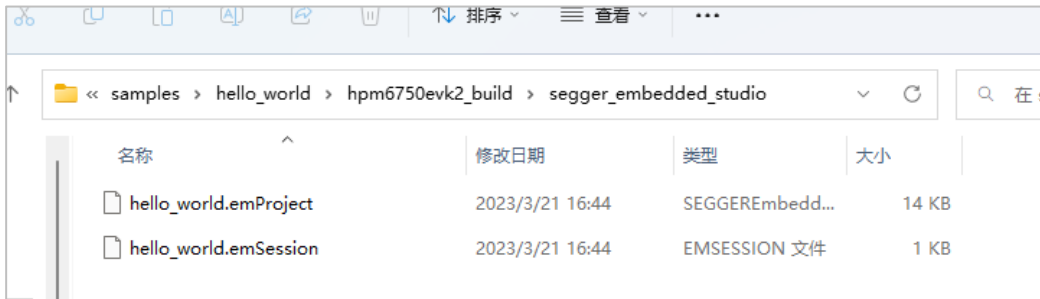


图 11: Segger Embedded Studiohello\_world工程

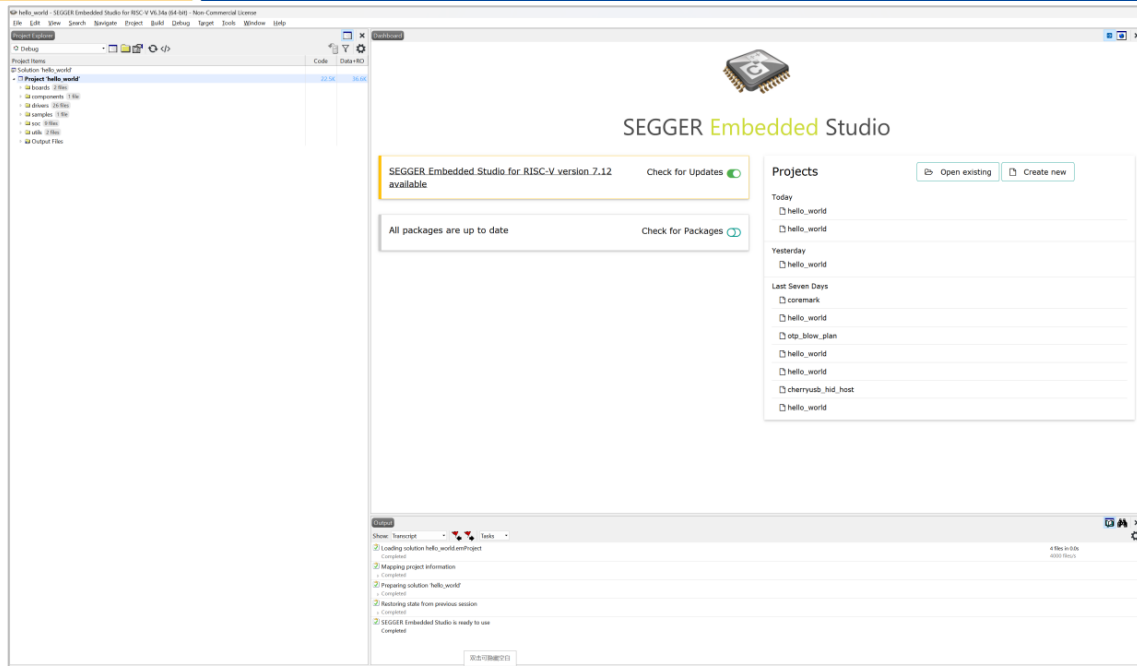


图 12: Segger Embedded Studio 打开hello\_world 工程

9. 使用Segger Embedded Studio 打开hello\_world 工程即可进行编译。

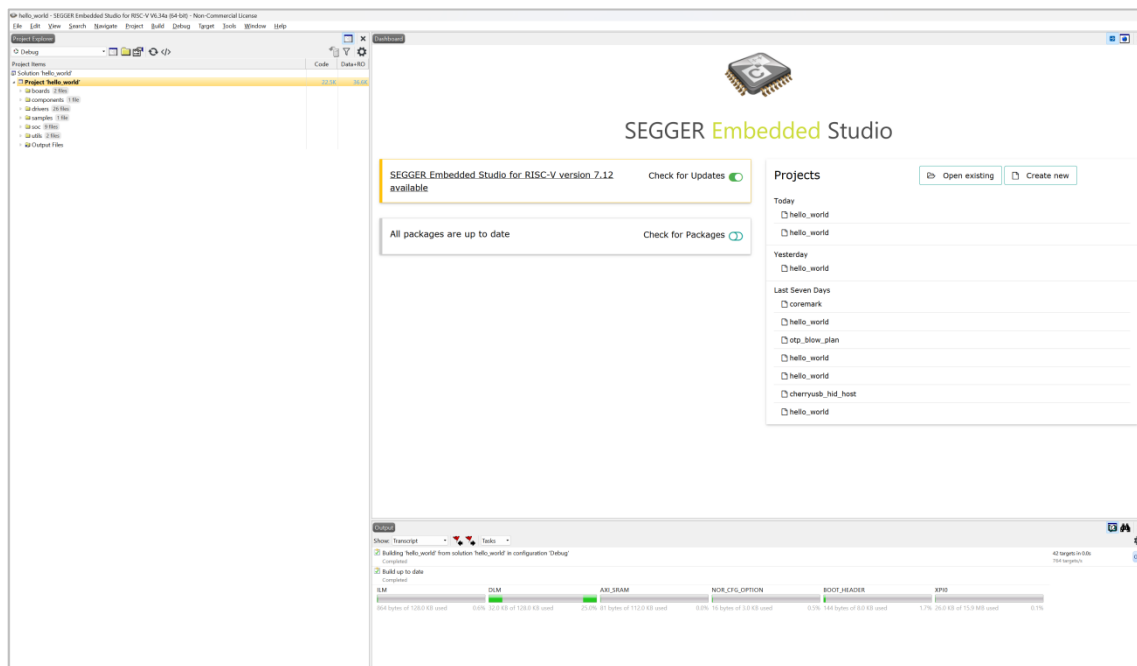


图 13: Segger Embedded Studio 编译hello\_world 工程

10. 使用Segger Embedded Studio 进行hello\_world 调试。

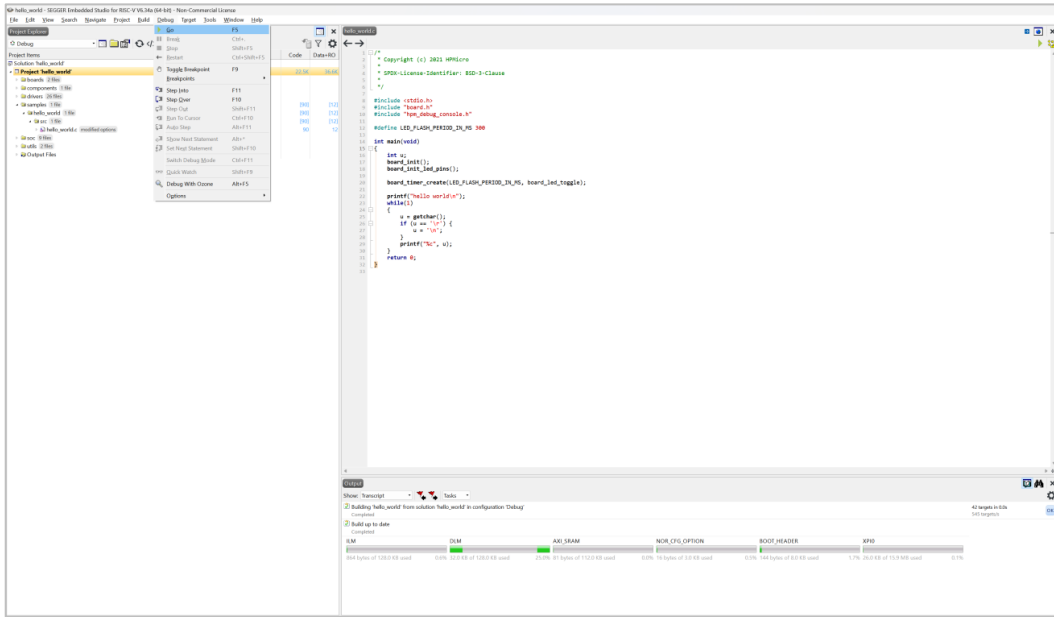


图 14: Segger Embedded Studio 调试hello\_world 工程

### 11. 在Segger Embedded Studio 中配置串口

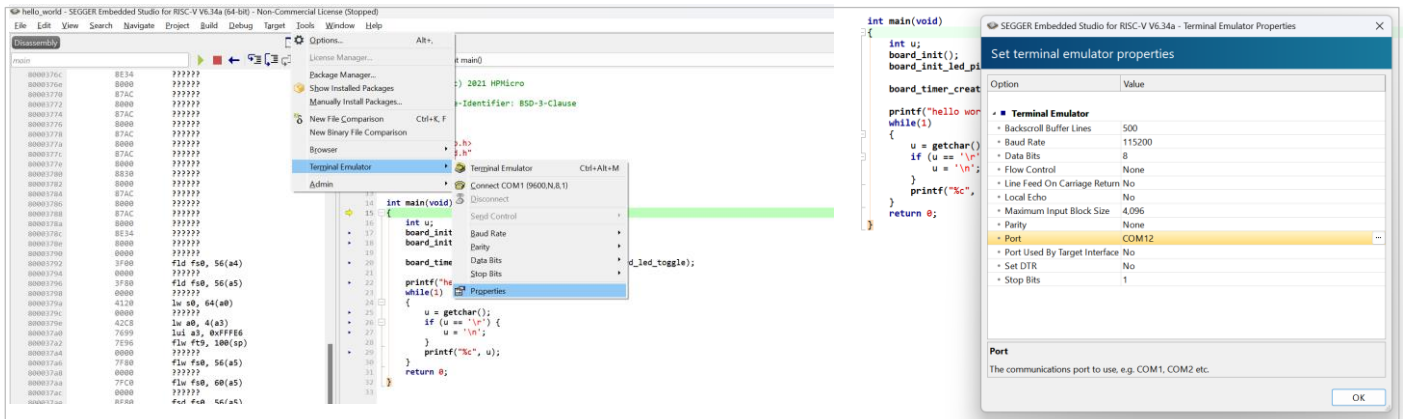


图 15: Segger Embedded Studio 配置串口

### 12. 在Segger Embedded Studio 中连接串口

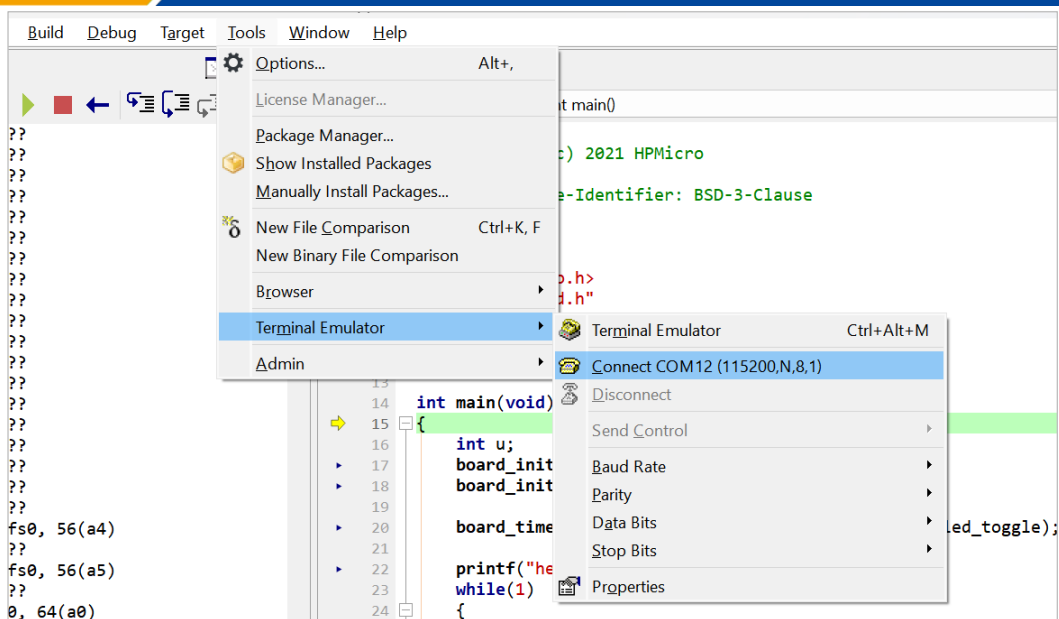


图 16: Segger Embedded Studio 连接串口

### 13. 在Segger Embedded Studio 中打开串口

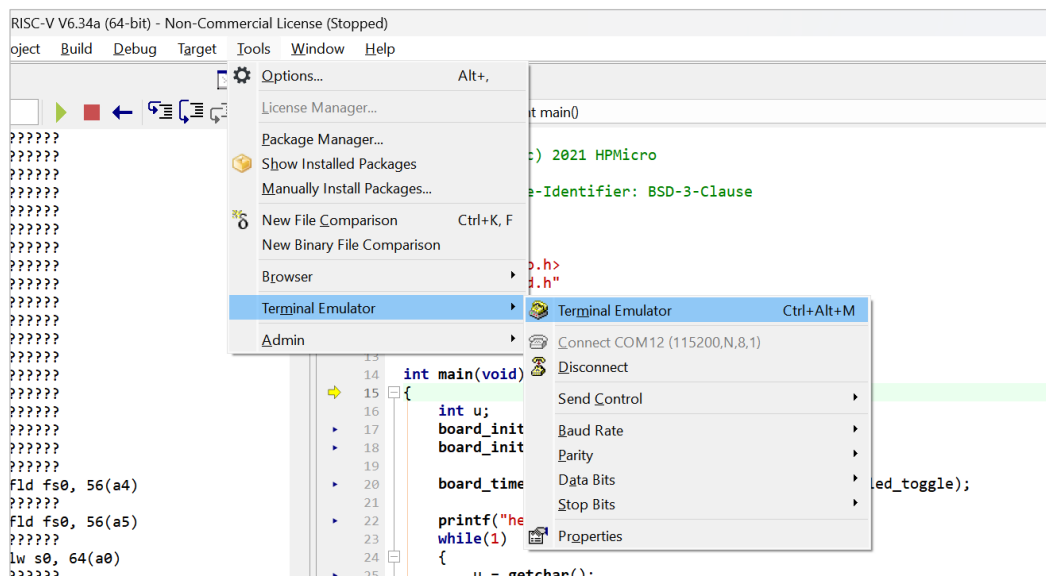


图 17: Segger Embedded Studio 打开串口

### 14. 运行hello\_world

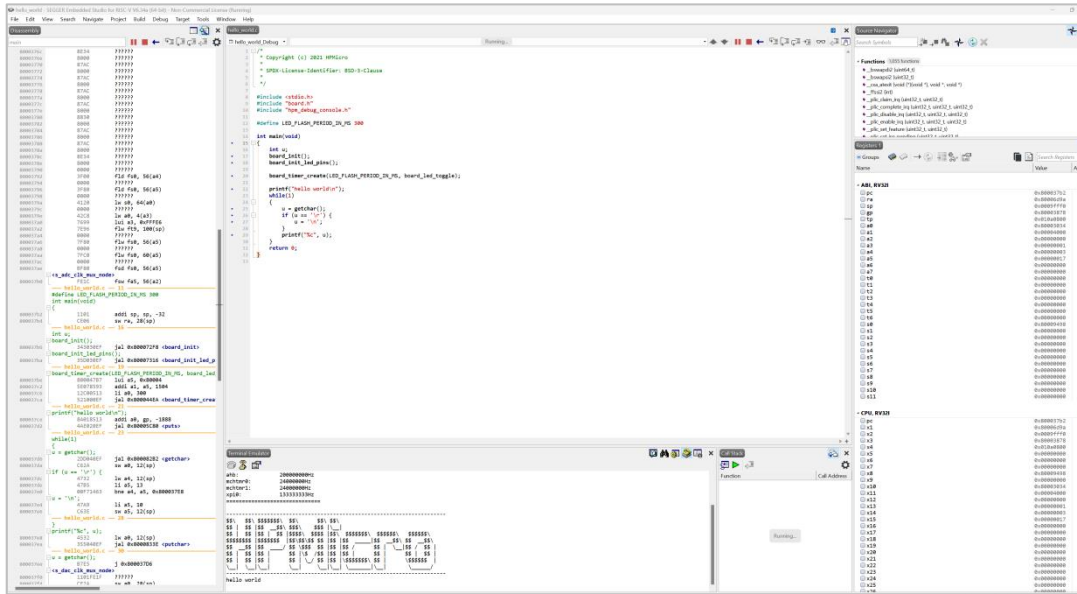


图 18: Segger Embedded Studio 运行hello\_world

15. sdk\_env提供了GUI project generator工具，用户亦可使用该工具生成工程。

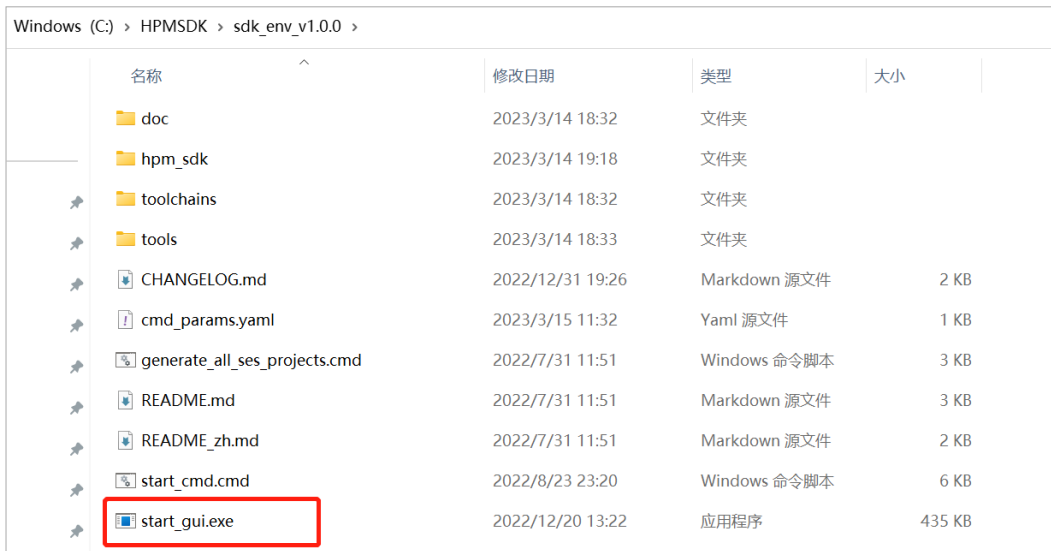


图 19: start\_gui 工具

16. 双击打开start\_gui.exe

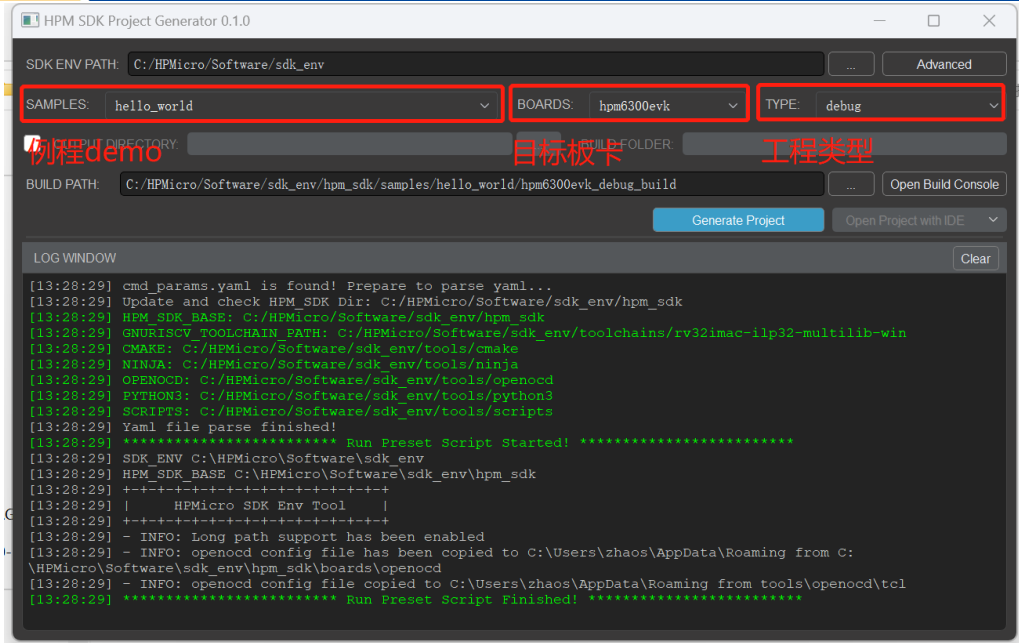


图 20: GUI project generator工具操作界面

17. 在GUI project generator界面中的” SAMPLES” 下拉列表中选择” hello\_world” ，在” BOARDS” 下拉列表中选择” hpm6750evk2” ,在” TYPE” 下拉列表中选择” debug” 。点击” Generate Project” 按钮，即可生成debug类型的hello\_world工程。如图所示。

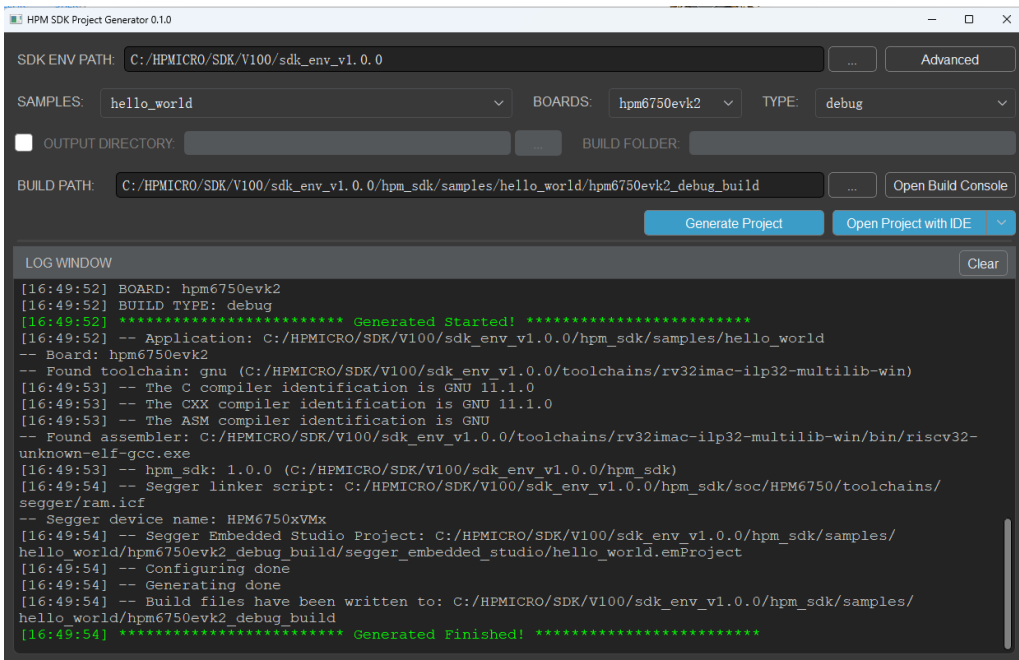


图 21: GUI project generator生成hello\_world工程

18. 点击” Open Project with IDE” 即可打开hello\_world工程。

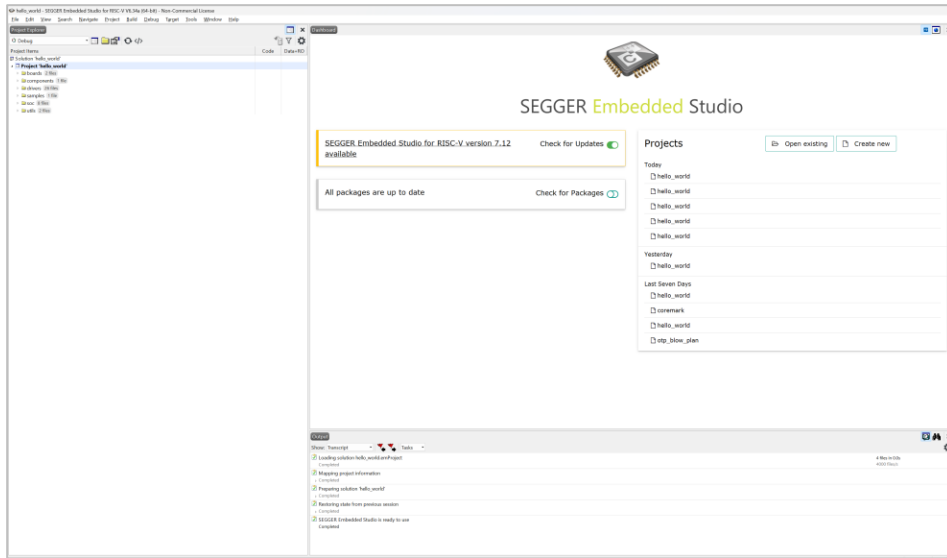


图 22: hello\_world工程

### 3.5 调试出错常见原因

本章节所描述的为先楫半导体的开发板的常见问题，并非只是发生在HPM6750EVK2上

#### 1. FT2232驱动没有正确安装

若开发板配有板载的FT2232调试器，方便用户直接调试程序。当使用FT2232调试器时遇到GDB server 连接失败的时候（如图所示），首先确认FT2232的驱动是否正确安装。可以在设备管理器中检查总线和串口驱动是否正确：一个USB Serial Port，一个Dual RS232-HS。

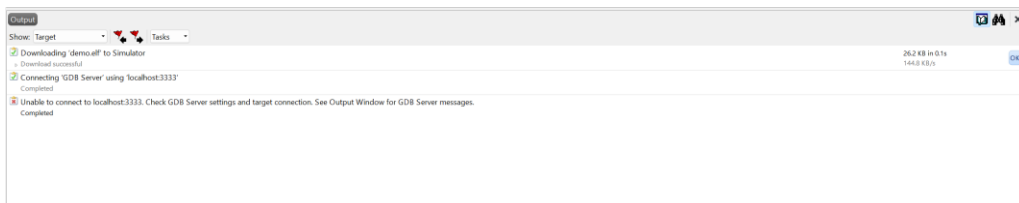


图 23: GDB Server连接失败

（请注意，当开发板连接到同一PC的不同USB端口时，也可能需要重新安装驱动。）

#### 2. Boot Pin配置异常

HPM6750EVK2的Boot Pin配置也有可能影响到芯片调试。如果发生调试失败，可以尝试调整boot pin配置如下：BOOT0=0，BOOT1=1，并且复位。

原因在于，有时flash内部执行的代码，特别是中断发生较频繁时，有可能影响到芯片进入debug模式。通过Boot Pin配置，将微控制器置于bootloader模式下，可以避免未知的中断状态。



如果是生成的Flash调试工程，为了避免Flash内已有代码执行的影响，从而导致debug无法连接。可以先将Boot Pin调整为：BOOT0=0，BOOT1=1，将芯片复位或者重新上电，之后再吧Boot Pin调整到：BOOT0=0，BOOT1=0，即调整到从NOR FLASH启动。最后，在点击debug按钮，开始程序调试。

### 3. 调试没有正常退出

如果调试环境依赖openocd，有时调试没有正常退出，可能导致openocd进程驻留，影响下一次调试。在调试出错时，可以考虑进入Windows的任务管理器，寻找openocd.exe进程，如果有的话，关闭此进程。同样的，打开多个Segger Embedded Studio窗口，当其中一个在debug中未退出，再开始另一个环境的debug时，也有可能导致类似现象。

### 4. Debug电阻没有正确配置

为了方便用户调试，有些开发板配置了两种调试接口（FT2232-to-JTAG, JTAG直连），两种调试模式不能同时进行，如果需要使用JTAG直连方式，此时需要去掉板载的电阻，具体器件需要根据电路图的标识。

### 5. openocd没有正确配置

点击工程，右击选择“options”，在弹出的对话框中查看GDB Server，如图所示，在GDB Server Command Line中查看openocd配置文件。

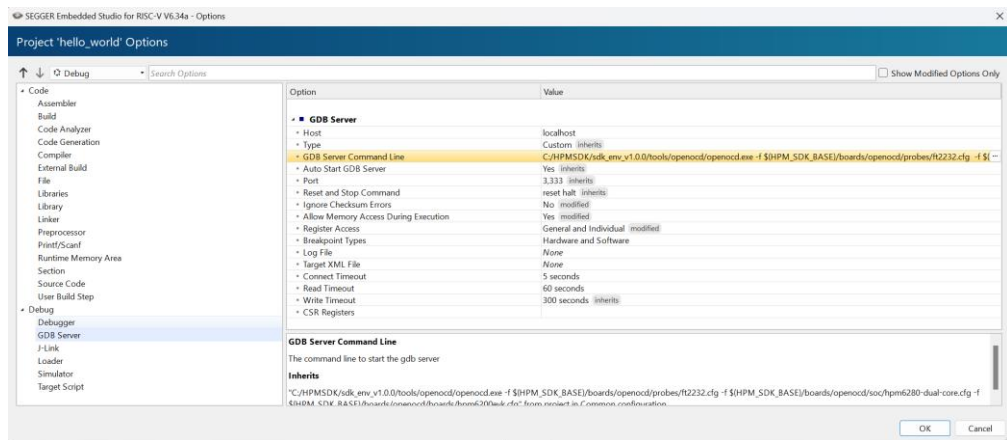


图 24: 查看openocd配置

SDK默认配置如图所示，默认使用ft2232调试器。在HPM6750EVK2上，默认使用的调试器是DAP-link，即使用cmsis-dap作为配置文件。

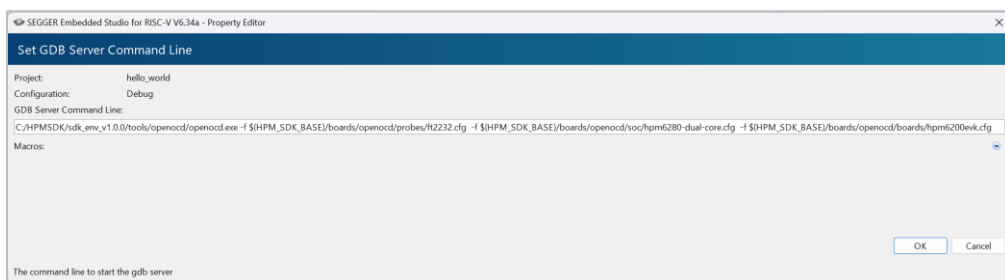


图 25: GDB Server默认配置

如果用户使用其他调试器，则需要更改此配置文件。以DAP-link调试器为例，要更改此配置文件为如图所示。

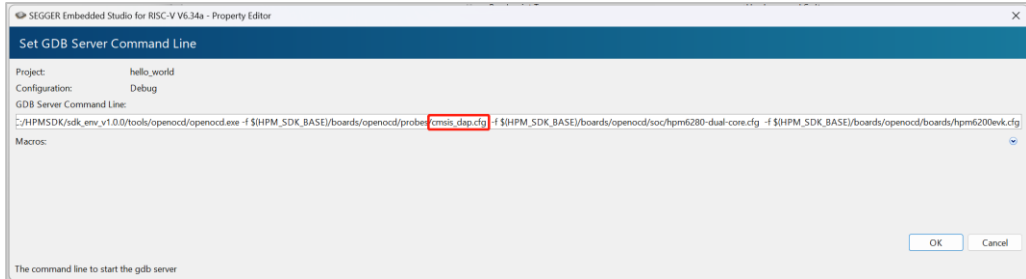


图 26: 使用DAP-link调试器GDB Server配置

## 6. J-Link调试器没有正确配置

如果用户使用Segger授权的J-Link调试器，则需要安装J-Link驱动，用户可以在 <https://www.segger.com/downloads/jlink/> 网站下载J-Link驱动程序。

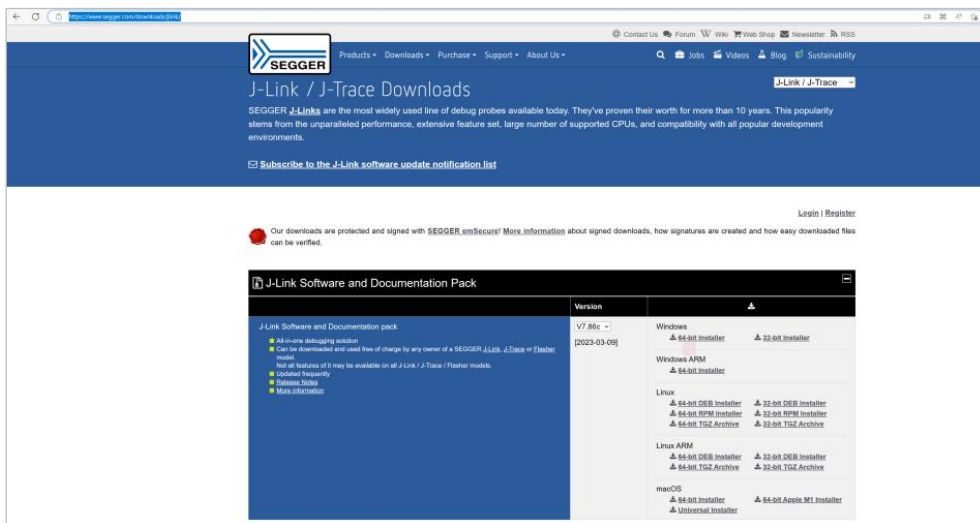


图 27: J-Link驱动下载

下载完成后安装J-Link驱动

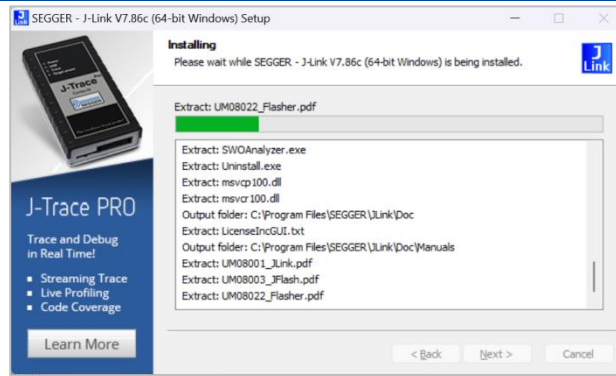


图 28: J-Link驱动安装

安装驱动完成后，正确连接J-Link JTAG接口到HPM6750EVK2上的P4接口 通过 Project->Options打开现有工程配置界面，点击Debugger配置项，确保“Target Connection”配置值为J-Link，选中J-Link配置项确认Target Interface Type选择的为JTAG选项。

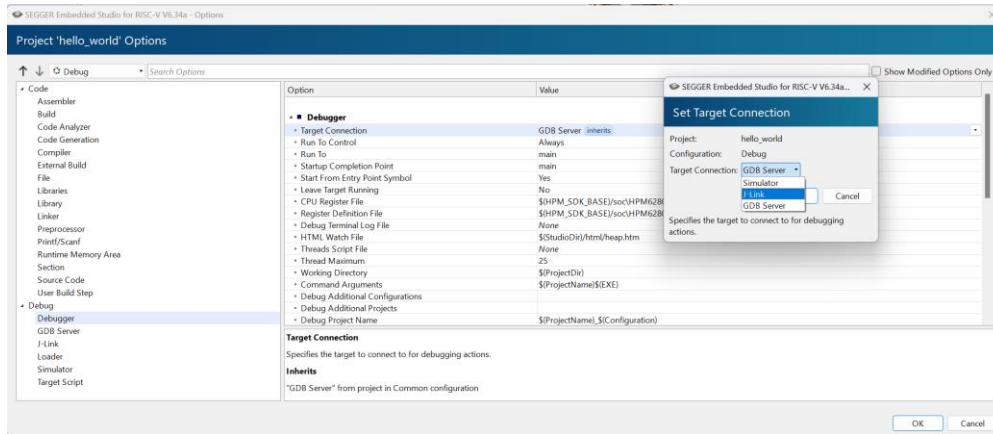


图 29: Target Connection 设置为J-Link

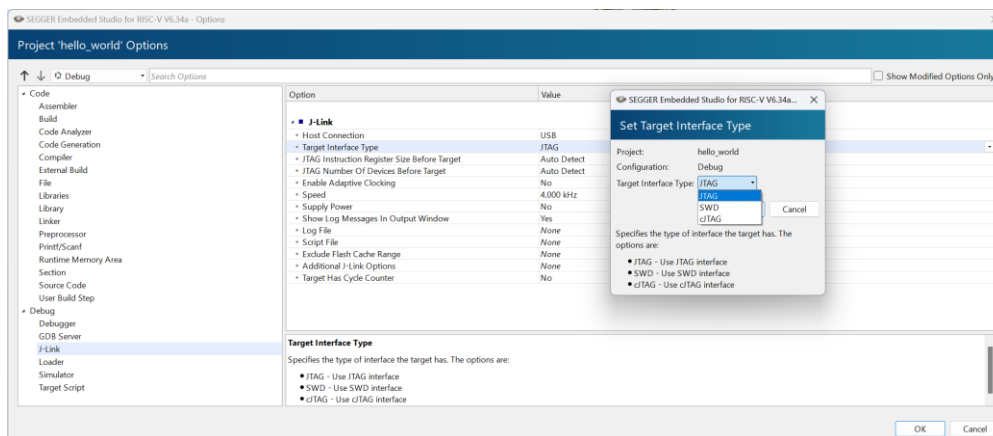


图 30: Target Interface Type设置为JTAG

### 3.6 更新 sdk\_env 中的 SDK/toolchain 指南

在这一部分将说明如何更新 sdk\_env 中的 SDK 以及 toolchain。用户可根据自身需求，按照如下描述更新 SDK 或 toolchain。

#### 3.6.1 更新 sdk\_env 中的 SDK

1. 下载的 hpm\_sdk.zip
2. 将解压后的 hpm\_sdk 放至 sdk\_env 目录下，确保可以在 sdk\_env\hpm\_sdk\ 目录中可以找到 env.cmd

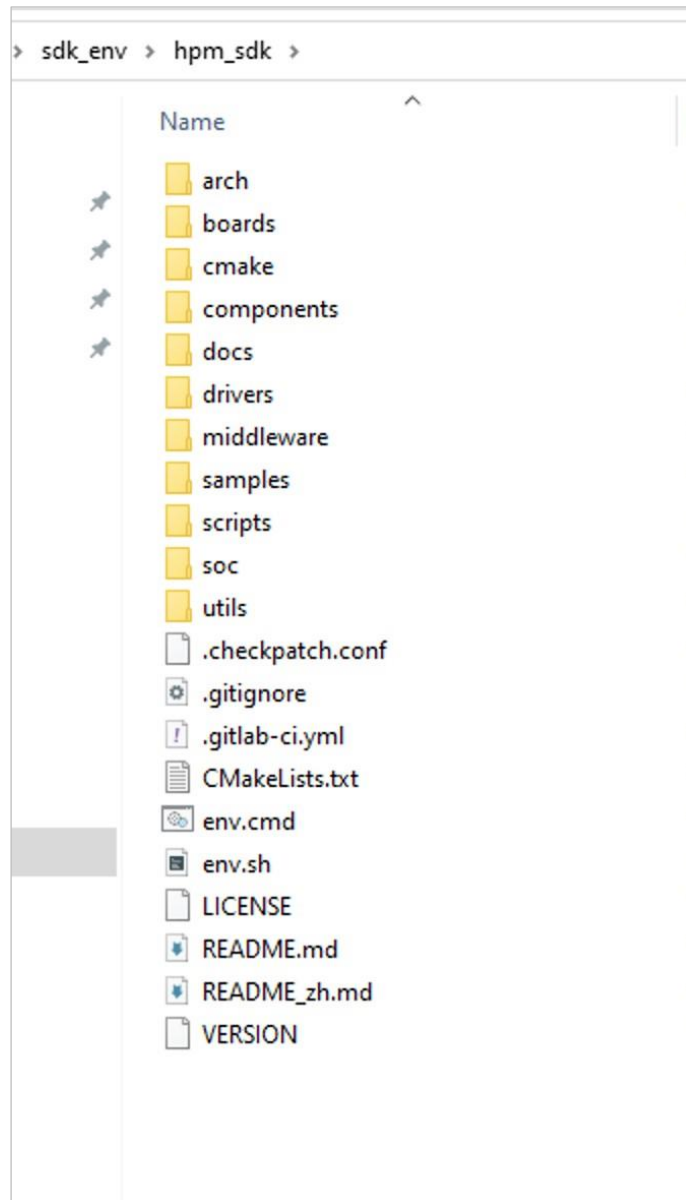


图 31: 更新 SDK

### 3.6.2 更新 sdk\_env 中的 toolchain

3. 下载toolchain (以rv32imafdc-ilp32d-x86\_64-w64-mingw32.zip 为例)
4. 将解压后的toolchain 放至sdk\_env\toolchains\ 目录下，确保可以在 sdk\_env\toolchains\rv32imafdc-ilp32d-x86\_64-w64-mingw32 目录中可以找到bin 文件夹。

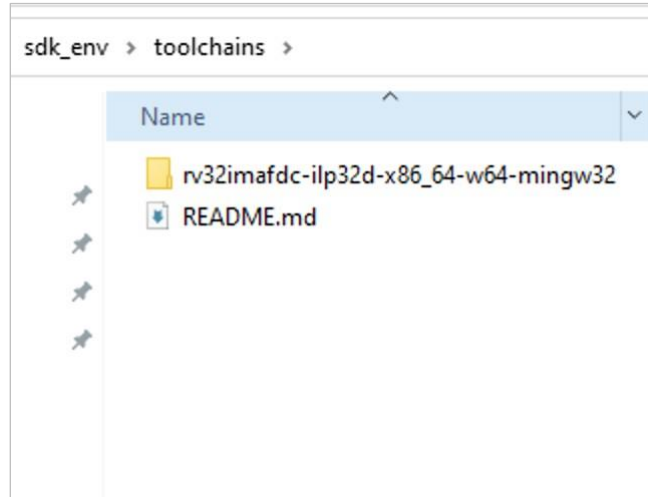


图 32: 拷贝toolchain

5. 编辑start\_cmd.cmd，更新环境变量TOOLCHAIN\_NAME

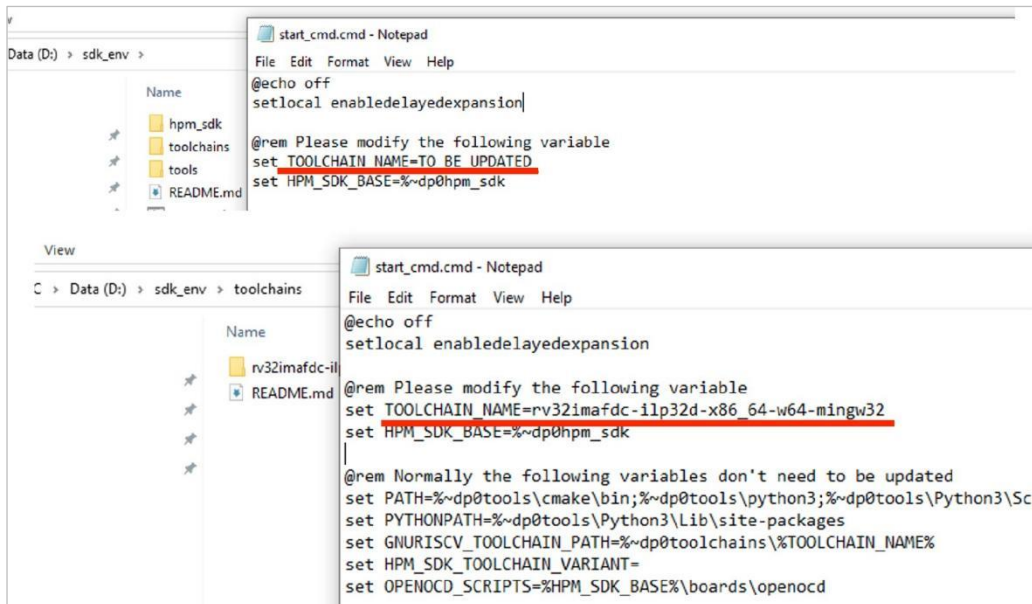


图 33: 更新start\_cmd.cmd 中TOOLCHAIN\_NAME

6. 双击打开start\_gui.exe，在界面中点击右上角” Advanced” 按钮,在设置列表找到 GNURISCV\_TOOLCHAIN\_PATH行，点击右侧浏览按钮，选择” sdk\_env\_v1.0.0\toolchains\rv32imafdc-

ilp32d-x86\_64-w64-mingw32” 目录，点击” Save Advanced Configuration”。即可看到LOG WINDOW中更新完成的提示。

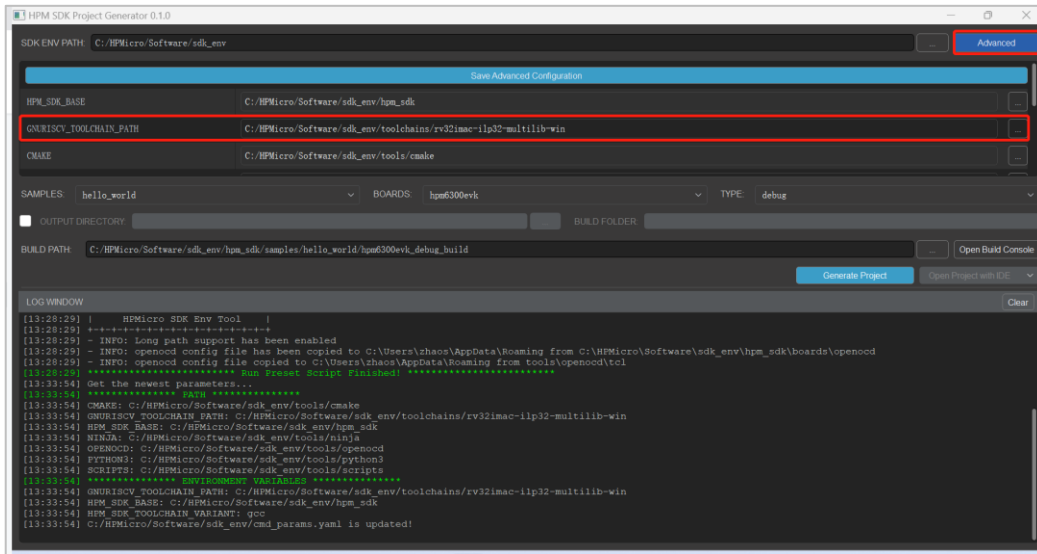


图 34: 更新start\_gui.exe 中TOOLCHAIN\_NAME

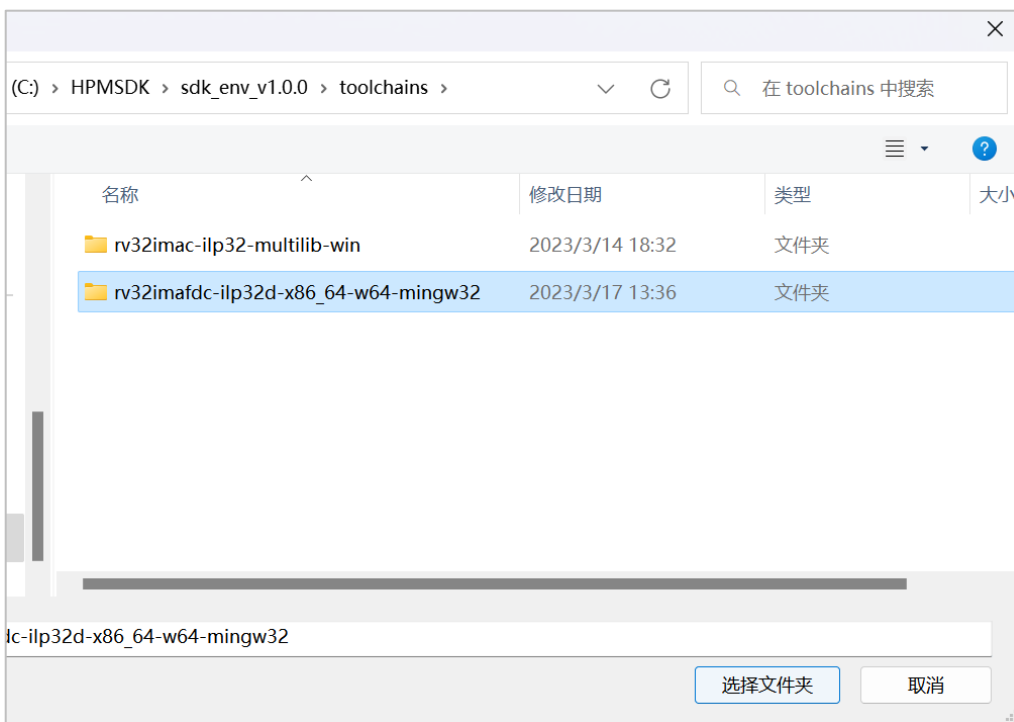


图 35: 选择TOOLCHAIN目录

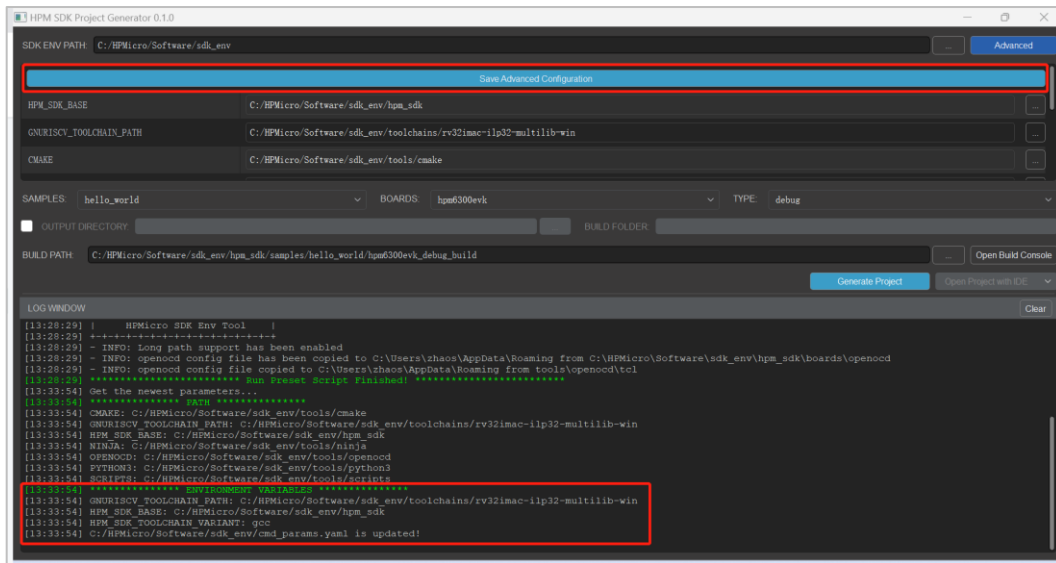


图 36: start\_gui.exe更新TOOLCHAIN完成

## 3.7 版本信息

日期	版本	描述
Rev1.0	2023/03/27	初版发布。

表 5: 版本信息



## 第四章 免责声明

上海先楫半导体科技有限公司（以下简称：“先楫”）保留随时更改、更正、增强、修改先楫半导体产品和/或本文档的权利，恕不另行通知。用户可在先楫官方网站<https://www.hpmicro.com> 获取最新相关信息。

本声明中的信息取代并替换先前版本中声明的信息。