



# HPM6P00

## HPM6P00EVK 用户使用手册

---

适用于上海先楫半导体 HPM6P00 系列高性能微控制器

# 目录

第一章 HPM6P00EVK 简介 .....	5
第二章 硬件电路 .....	7
2.1 电路模块介绍 .....	7
2.1.1 系统架构 .....	7
2.1.2 电源 .....	8
2.1.3 USER LED .....	8
2.1.4 按键 .....	8
2.1.5 USB 接口 .....	8
2.1.6 DEBUG 接口 .....	8
2.1.7 马达接口 .....	9
2.1.8 千兆网口 .....	9
2.1.9 音频接口 .....	9
2.1.10 CAN 接口 .....	9
2.1.11 ADC 接口 .....	9
2.1.12 FEMC/PPI 接口 .....	9
2.1.13 扩展 IO 接口 .....	10
2.1.14 BOOT 拨码开关设置 .....	11
第三章 软件开发套件 .....	12
3.1 简介 .....	12
3.2 环境以及依赖 .....	12
3.3 开发工具 .....	12
3.4 快速使用指南 .....	12
3.5 调试出错常见原因 .....	19
3.6 更新 sdk_env 中的 SDK/toolchain 指南 .....	23
3.6.1 更新 sdk_env 中的 SDK .....	24
3.6.2 更新 sdk_env 中的 toolchain .....	24
第四章 免责声明 .....	27
4.1 版本信息 .....	27

## 表格目录

表 1	主要器件位号对应功能名称 .....	6
表 2	扩展 IO 接口 .....	11
表 3	BOOT Pin 配置表 .....	11

# 图片目录

图 1	顶层器件位置图 .....	5
图 2	底层器件位置图 .....	5
图 3	HPM6P00EVK 硬件设计框图 .....	7
图 4	FEMC/PPI 子板 .....	10
图 5	安装 FTDI 驱动 .....	12
图 6	查看 Windows 设备管理器 .....	13
图 7	sdk_env 创建工程方式 .....	13
图 8	打开 sdk prompt .....	14
图 9	构建目标板工程 .....	14
图 10	构建目标板 flash_xip 工程 .....	14
图 11	generate_project 帮助 .....	15
图 12	Segger Embedded Studio 打开 hello_world 工程 .....	15
图 13	Segger Embedded Studio 编译 hello_world 工程 .....	16
图 14	Segger Embedded Studio 调试 hello_world 工程 .....	16
图 15	Segger Embedded Studio 中配置串口 .....	17
图 16	Segger Embedded Studio 中连接串口 .....	17
图 17	Segger Embedded Studio 打开串口 .....	17
图 18	Segger Embedded Studio 运行 hello_world .....	18
图 19	start_gui 工具 .....	18
图 20	GUI project generator 工具操作界面 .....	19
图 21	GUI project generator 生成 hello world 工程 .....	19
图 22	hello world 工程 .....	19
图 23	GDB Server 连接失败 .....	20
图 24	查看 openocd 配置 .....	21
图 25	GDB Server 默认配置 .....	21
图 26	使用 cmsis_dap 调试器 GDB Server 配置 .....	22
图 27	J-Link 驱动下载 .....	22
图 28	J-Link 驱动安装 .....	22
图 29	Target Connection 设置为 J-Link .....	23
图 30	Target Interface Type 设置为 JTAG .....	23
图 31	更新 SDK .....	24
图 32	覆盖 toolchain .....	24
图 33	更新 start cmd.cmd 中的 TOOLCHAIN_NAME .....	25
图 34	更新 start_gui.exe 中的 TOOLCHAIN_NAME .....	25
图 35	start_gui.exe 更新 TOOLCHAIN 完成 .....	26

# 第一章 HPM6P00EVK 简介

HPM6P00EVK 器件图如图 1，图 2 所示。表 1 给出了器件位置的对应器件名称。

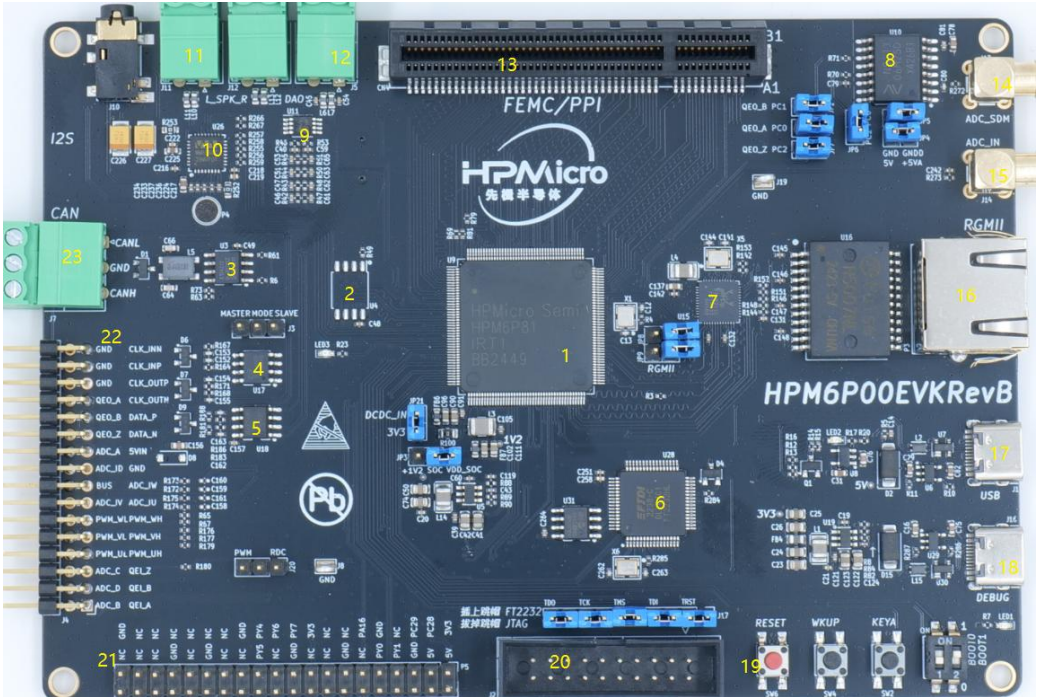


图 1 顶层器件位置图

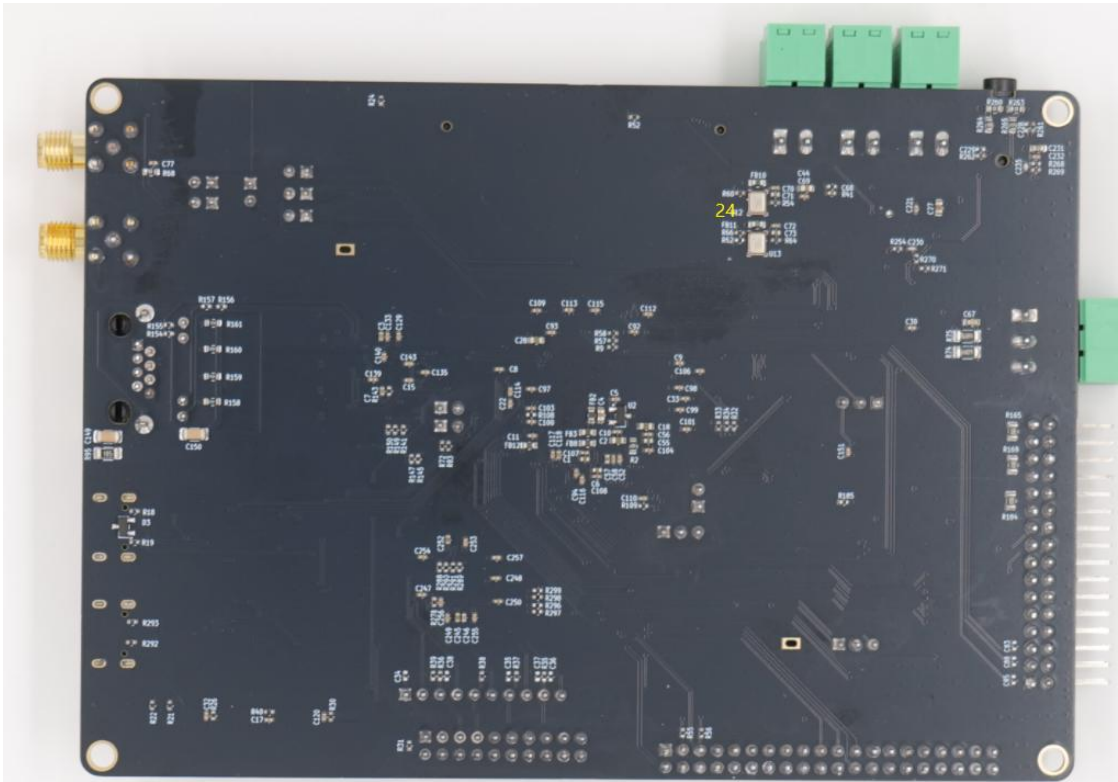


图 2 底层器件位置图

序号	名称	序号	名称
1	HPM6P81IRT1	2	Flash(DNP)
3	CAN transceiver(TJA1042T/3)	4	RS422 transceiver(SIT3490)
5	RS485 transceiver(MAX3485)	6	FT2232HL
7	RGMII PHY(RTL8211)	8	SDM decoder( NSI1306M25)
9	Amplifier(NS4150B)	10	Audio Codec (WM8960)
11	Speaker Connector	12	Dao Connector
13	FEMC/PPI Connector	14	ADC Connector
15	SDM Connector	16	Giga EtherNET Connector
17	Type-C USB-OTG	18	Type-C USB for Debug
19	Key and Boot Pin	20	20-pin JTAG Connector
21	Raspberry Pi Connector	22	Motor Connector
23	CAN Connector	24	Digital Mic(SPH0641)

表 1 主要器件位号对应功能名称

## 第二章 硬件电路

HPM6P00EVK 电源输入由 Debug Type-C 接口或者 USB Type-C 提供，供电不能超过 5.5V，防止过压导致板上器件损坏。I/O 接口是 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

HPM6P00EVK 系统机构如图 3

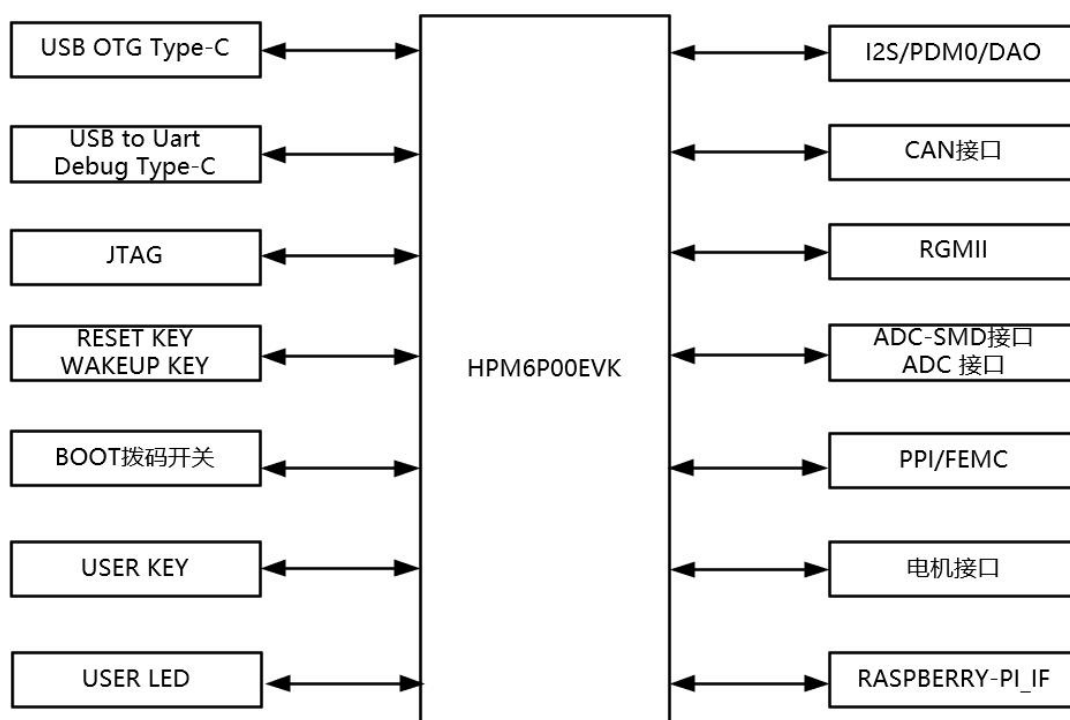


图 3 HPM6P00EVK 硬件设计框图

注意：

为了实现更多的功能，对部分引脚进行了重复利用，因此再使用时需要注意。具体的设置，用户可以参看 SDK 中相应功能的 readme 文档。

- USB 的 DP、DN 与网口的 PD0 和 PD1 重合，默认是 USB 功能，插上跳帽 JP8,JP9 则使能千兆网。
- I2S 的 TXD 引脚和 PPI 的 PB0 引脚重合，默认通过 R52 连接到 I2S，同时连接到 SDRAM\_A0,SRAM\_A16。
- TCK、TMS、TRST 三个引脚与 PPI、FEMC 功能重合。如果用户希望使用其 PPI、FEMC 需要查看子板电路。同时该操作会影响 JTAG 调试功能。
- QEO 和 SDM 共同使用 PC0 PC1 PC2 三个引脚，默认连接在 SDM 接口上，需要使用 QEO 时候，插上跳帽 JP1,JP2,JP7，摘掉 JP3,JP4,JP5

### 2.1.2 电源

HPM6P00EVK 供电方式如下

- USB Type-C USB 接口，可以做 host 或者 device
- Debug Type-C，板载 FT2232 芯片，通过该接口可以直接仿真，以及串口通讯功能
- HPM6P00 内核电源 1.2V，该 EVK 可以通过 JP3 选择使用内部和外部供电内核。

### 2.1.3 USER LED

HPM6P00EVK 板载 LED

- LED1 电源指示灯
- LED3 功能 LED
- LED2 USB 过流指示灯

### 2.1.4 按键

HPM6P00EVK 板载 3 个按键，2 个拨码开关

- SW1 BOOT PIN
- SW6 Reset 按键
- SW4 Wake Up 按键
- SW2 功能按键

### 2.1.5 USB 接口

HPM6P00EVK 板载 USB 接口，类型为 Type-C，支持 USB2.0 OTG。同时支持通过 USB 烧录，即通过 USB 接口下载 bin 文件。下载工具可以通过官网获取。

注意：使用 USB 时候需要保证 JP8 JP9 没有插上跳帽。此时只能使用 USB，无法使用千兆网。

### 2.1.6 DEBUG 接口

HPM6P00EVK 提供两种 DEBUG 接口，默认是通过 J16(USB Type-C)连接到 PC，通过 FT2232HL 实现 USB 到 JTAG 和 UART 的转换。通过这种方式，用户可以访问芯片 JTAG 接口和 UART0 接口。UART0 也可用于 UART 串行启动和 ISP。另外一路为 20Pin 牛角插座 JTAG 接口，用户可以通过调试器直接连接芯片的 DEBUG 口，此时无需使用板载 FT2232 调试接口。

需要注意的是，使用该接口时需要把牛角插座旁边的 TRST,TDI,TMS,TDO,TCK 的跳线帽取掉，以使得板载 DEBUG 功能由 U28 FT2232HL 切换到 JTAG 直连。



### 2.1.7 马达接口

HPM6P00EVK 提供一个马达接口，可以兼容适配先楫的马达驱动板。使用 PWM 时需要把 J20 跳帽的 PWM 端插上。

### 2.1.8 千兆网口

HPM6P00EVK 板载千兆网口，网口芯片为 RTL8211E-VB-CG。

注意：使用时需要插上 JP8 JP9 跳帽，此时可以使用千兆网，无法使用 USB。

### 2.1.9 音频接口

HPM6P00EVK 支持音频输入输出

- Audio Codec(WM8960)
  - 3.5mm 耳麦(默认美标，可以修改电阻为国标)
  - 左右声道(1W@8Ω 每声道)
  - Mic
- DAO
  - 1 路 3W@4Ω
- Digital Mic
  - 2 路 SPH0641LU4H

注意：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  
其次，音频 I2S\_TXD 引脚和 SDRAM 、SRAM 复用。

### 2.1.10 CAN 接口

HPM6P00EVK 板载 CAN 收发器，芯片为 TJA1024T/3。

### 2.1.11 ADC 接口

HPM6P00EVK 板载两种 ADC 接口

- J13，隔离型 Sigma-Delta 调制器，芯片为 NSI1306M25。
- J14，ADC 接口

使用 SDM 接口时，需要插上跳帽 JP1,JP2,JP7，摘掉 JP4,JP5,JP6

### 2.1.12 FEMC/PPI 接口

HPM6P00EVK 板载有 FEMC/PPI 接口。此接口包含 FEMC 功能以及 PPI 功能引线。客户可以根据自身的需要，自行设计子板适配。HPM6P00EVK 套装中包含一个子板，其功能包括

- 16bit SDRAM，M12L2561616A

## HPM6P00EVK

- 10bit 并口 ADC 转换, MS9280
- 16bit SRAM, IS62WV51216BLL

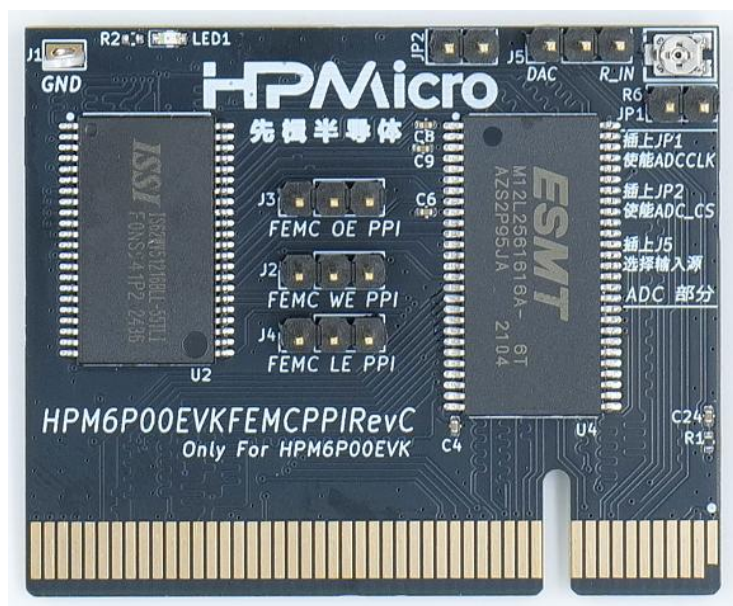


图 4 FEMC/PPI 子板

该子板兼容多个不同接口，调试时注意跳帽及使用的接口，下面注意事项：

- 因其引脚重合，FEMC 与 PPI 不可同时工作。使用 FEMC 时候将 J2 J3 J4 跳帽插到 FEMC 丝印侧即可。
- PPI\_ADC 与 FEMC\_SDRAM 有引脚复用，因此不能同时使能。
- PPI\_SRAM 与 FEMC\_SDRAM 有引脚复用，因此不能同时使能。
- FEMC\_SDRAM 和 FEMC\_SRAM 可分时工作，通过 CS 选择使能。
- PPI\_SRAM 和 PPI\_ADC 可同时工作，通过 CS 选择使能。

HPM6P00EVK 上的 FEMC/PPI 接口的料号是 C404103，UMaxconn. 1mm 间距 98pin，用户客户在先楫网站上获取上述子板的设计文件。

因为采购供应链问题，先楫保留更换器件型号的可能，但是先楫会确保不同的器件保持同样的性能。上述的设计，是为了再最少的板子的情况下实现更多的例程。先楫并不推荐客户将这些功能复合使用。

### 2.1.13 扩展 IO 接口

HPM6P00EVK 提供板载扩展 IO，即排针 P5。接口机械尺寸与树莓派接口兼容。P5 的信号列表如下。

引脚名	功能名	连接器编号		功能名	引脚名
	3.3V	1	2	5.0V	
PC28	SDA1	3	4	5.0V	
PC29	SCL1	5	6	GND	
NC	GPIO7	7	8	RXD	PY1
	GND	9	10	TXD	PY0
PA16	GPIO0	11	12	GPIO1	NC
NC	GPIO2	13	14	GND	
NC	GPIO3	15	16	GPIO4	NC

	3.3V	17	18	GPIO5	NC
PY7	MOSI	19	20	GND	
PY6	MISO	21	22	GPIO6	NC
PY4	SCLK	23	24	CE0	PY5
	GND	25	26	CE1	NC
NC	SDA0	27	28	SCL0	NC
NC	GPIO21	29	30	GND	
NC	GPIO22	31	32	GPIO26	NC
NC	GPIO23	33	34	GND	
NC	GPIO24	35	36	GPIO27	NC
NC	GPIO25	37	38	GPIO28	NC
	GND	39	40	GPIO29	NC

表 2 扩展 IO 接口

## 2.1.14 BOOT 拨码开关设置

HPM6P00 是通过拨码开关设置对应 BOOT\_MODE[1:0]=[PA03:PA02] 引脚选择启动模式，配置所示。

拨码开关 [1:0]		启动模式	说明
OFF	OFF	XPI NOR FLASH 启动	从连接在 XPI0 上的串行 NOR FLASH 启动(芯片内部自带 FLASH)
OFF	ON	系统编程(ISP)/串行启动	从 UART0/USB0 上烧写固件，OTP， 或从 UART0/USB0 启动
ON	OFF	系统编程(ISP)/串行启动	从 UART0/USB0 上烧写固件，OTP， 或从 UART0/USB0 启动
ON	ON	保留模式	保留模式

表 3 BOOT Pin 配置表

## 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。

### 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 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，在 Segger 官网申请 license。

### 3.4 快速使用指南

1. 下载安装 Segger Embedded Studio，本文所使用的版本号为 8.10
2. 下载最新版本 sdk\_env\_vx.x.x.zip 压缩包后解压(本文 sdk\_env\_v1.9.0 为例)  
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。
3. 运行 sdk\_env\_v1.9.0\tools\FTDI\_InstallDriver.exe 以安装可用于调试的 FT232 驱动。

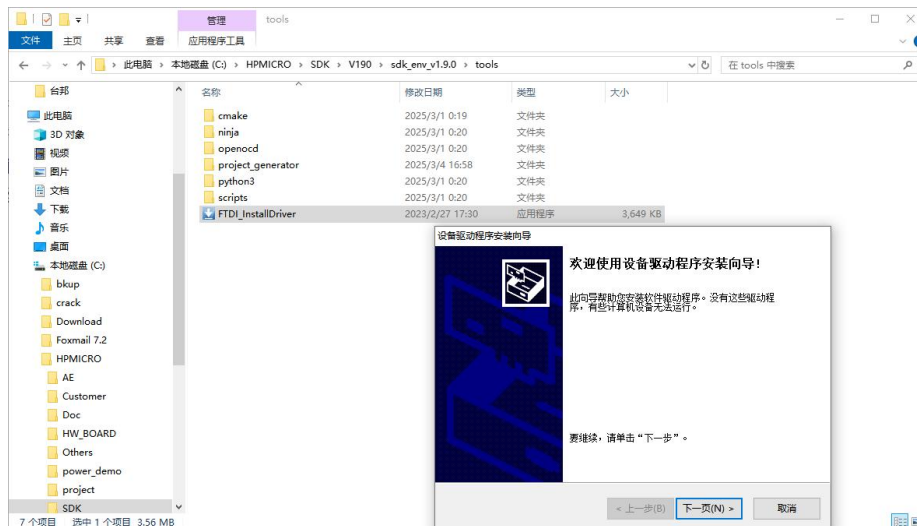


图 5 安装 FTDI 驱动

正确安装驱动后，使用 USB Type-C 线缆将 HPM6P00EVK 上的 J16 连接到 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)，

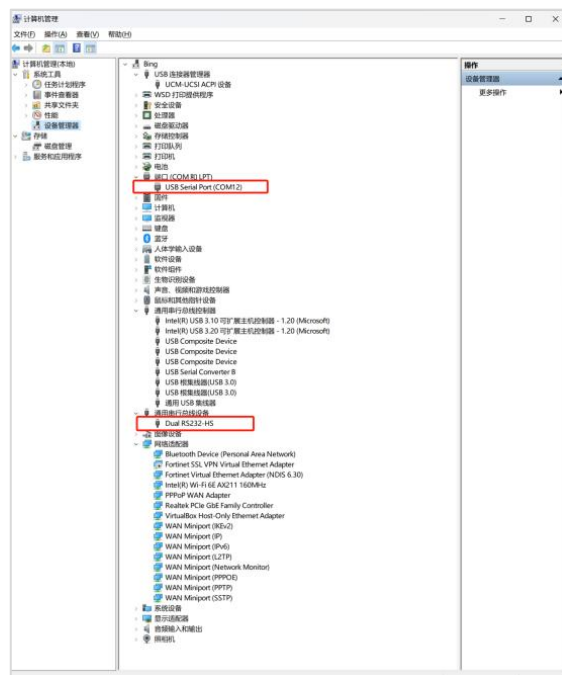


图 6 查看 Windows 设备管理器

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。

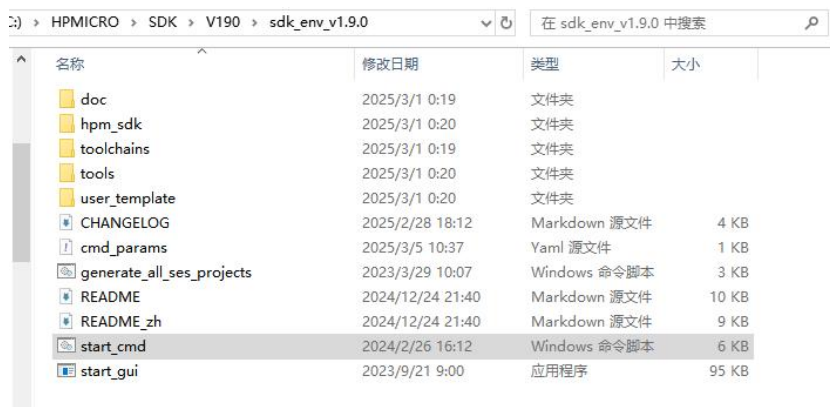


图 7 sdk\_env 创建工程方式

以命令行工具为例，双击打开 sdk\_env\_v1.9.0 下 start\_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

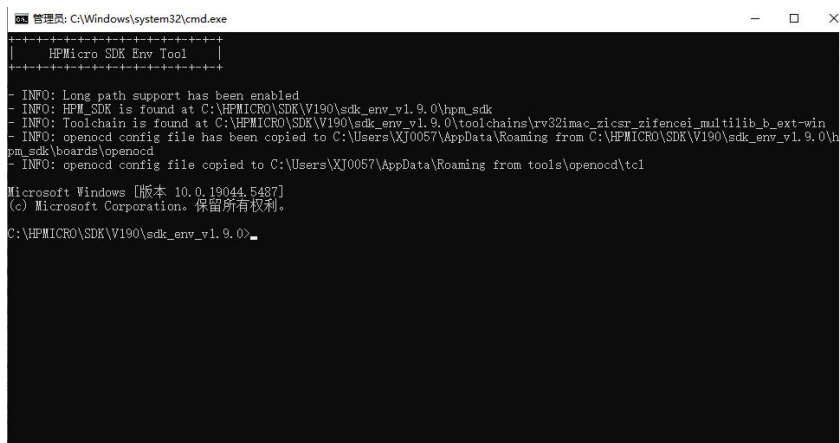
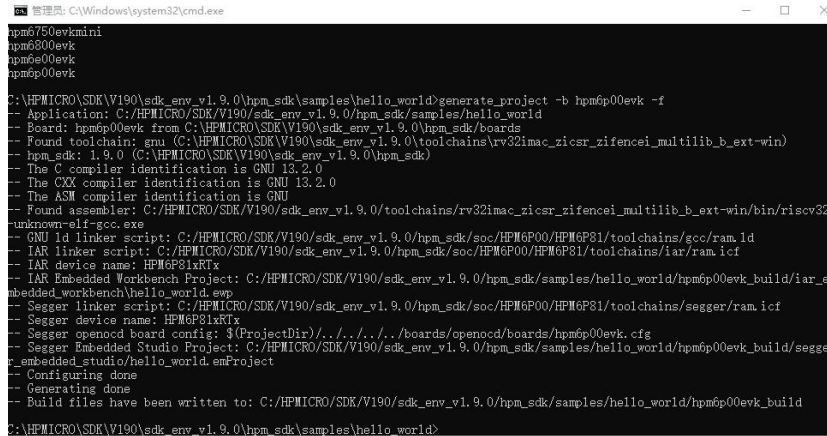


图 8 打开 sdk prompt

5. 再 sdk\_prompt 中切换路径到 SDK 具体的一个示例例程，以 hello\_world 为例  
`>cd hpm_sdk\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 HPM6P00EVK 为例)可以通过运行以下命令进行工程构建，若构建成功，将看到如下提示  
`>generate_project -b hpm6p00evk -f`

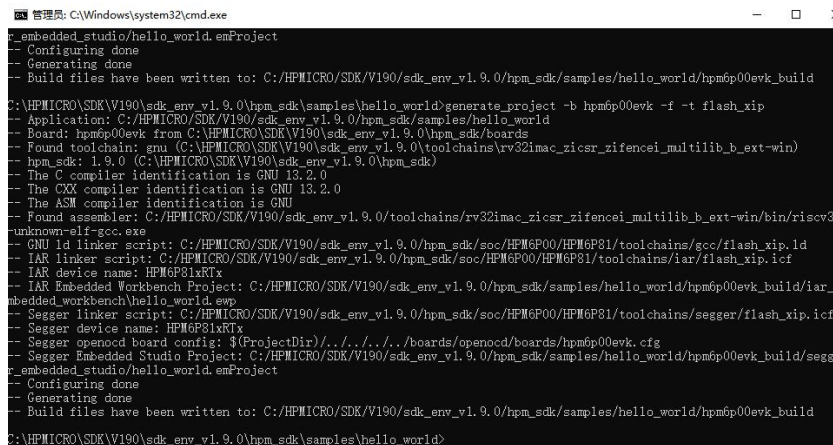


```
C:\HPMICRO\SDK\V190\sdk_env_v1.9.0\hpm_sdk\samples\hello_world>generate_project -b hpm6p00evk -f
-- Application: C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/hpm_sdk/samples/hello_world
-- Board: hpm6p00evk from C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/hpm_sdk/boards
-- Found toolchain: gnu (C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/toolchains/rv32imac_zicrs_zifencei_multilib_b_ext-win)
-- hpm_sdk: 1.9.0 (C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/hpm_sdk)
-- The C compiler identification is GNU 13.2.0
-- The CXX compiler identification is GNU 13.2.0
-- The ASM compiler identification is GNU
-- Found assembler: C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/toolchains/rv32imac_zicrs_zifencei_multilib_b_ext-win/bin/riscv32
unknown-elf-gcc.exe
-- GNU ld linker script: C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/hpm_sdk/soc/HPM6P00/HPM6P81/toolchains/gcc/ram.ld
-- IAR linker script: C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/hpm_sdk/soc/HPM6P00/HPM6P81/toolchains/iar/ram.icf
-- IAR device name: HPM6P81xRTx
-- IAR Embedded Workbench Project: C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/hpm_sdk/samples/hello_world/hpm6p00evk_build/iar_e
mbedded_workbench/hello_world.ewp
-- Segger linker script: C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/hpm_sdk/soc/HPM6P00/HPM6P81/toolchains/segger/ram.icf
-- Segger device name: HPM6P81xRTx
-- Segger openocd board config: $(ProjectDir)/../../boards/openocd/boards/hpm6p00evk.cfg
-- Segger Embedded Studio Project: C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/hpm_sdk/samples/hello_world/hpm6p00evk_build/segge
r_embedded_studio/hello_world.emProject
-- Configuring done
-- Generating done
-- Build files have been written to: C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/hpm_sdk/samples/hello_world/hpm6p00evk_build
C:\HPMICRO\SDK\V190\sdk_env_v1.9.0\hpm_sdk\samples\hello_world>
```

图 9 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 flash 地址空间原地执行），ram（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。

`>generate_project -b hpm6p00evk -f -t flash_xip`



```
r_embedded_studio/hello_world.emProject
-- Configuring done
-- Generating done
-- Build files have been written to: C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/hpm_sdk/samples/hello_world/hpm6p00evk_build
C:\HPMICRO\SDK\V190\sdk_env_v1.9.0\hpm_sdk\samples\hello_world>generate_project -b hpm6p00evk -f -t flash_xip
-- Application: C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/hpm_sdk/samples/hello_world
-- Board: hpm6p00evk from C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/hpm_sdk/boards
-- Found toolchain: gnu (C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/toolchains/rv32imac_zicrs_zifencei_multilib_b_ext-win)
-- hpm_sdk: 1.9.0 (C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/hpm_sdk)
-- The C compiler identification is GNU 13.2.0
-- The CXX compiler identification is GNU 13.2.0
-- The ASM compiler identification is GNU
-- Found assembler: C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/toolchains/rv32imac_zicrs_zifencei_multilib_b_ext-win/bin/riscv32
unknown-elf-gcc.exe
-- GNU ld linker script: C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/hpm_sdk/soc/HPM6P00/HPM6P81/toolchains/gcc/flash_xip.ld
-- IAR linker script: C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/hpm_sdk/soc/HPM6P00/HPM6P81/toolchains/iar/flash_xip.icf
-- IAR device name: HPM6P81xRTx
-- IAR Embedded Workbench Project: C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/hpm_sdk/samples/hello_world/hpm6p00evk_build/iar_e
mbedded_workbench/hello_world.ewp
-- Segger linker script: C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/hpm_sdk/soc/HPM6P00/HPM6P81/toolchains/segger/flash_xip.icf
-- Segger device name: HPM6P81xRTx
-- Segger openocd board config: $(ProjectDir)/../../boards/openocd/boards/hpm6p00evk.cfg
-- Segger Embedded Studio Project: C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/hpm_sdk/samples/hello_world/hpm6p00evk_build/segge
r_embedded_studio/hello_world.emProject
-- Configuring done
-- Generating done
-- Build files have been written to: C:/HPMICRO/SDK/V190/sdk_env_v1.9.0/hpm_sdk/samples/hello_world/hpm6p00evk_build
C:\HPMICRO\SDK\V190\sdk_env_v1.9.0\hpm_sdk\samples\hello_world>
```

图 10 构建目标板 flash\_xip 工程

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

`>generate_project -h`



```

C:\Windows\system32\cmd.exe
r_embedded_studio\hello_world.emProject
-- Configuring done
-- Generating done
-- Build files have been written to: C:/HFMICRO/SDK/V190/sdk_env_v1.9.0/hpm_sdk/samples/hello_world/hpm6p00evk_build

C:\HFMICRO\SDK\V190\sd... generate_project -h
generate_project [-f] [-b board] [-x board_search_path] [-a] [-list] [-h]
-f: force clean already existed build directory
-b board: specify board for project generation
-x board_search_path: specify board search path for project generation
-a: generate projects for all supported boards
-list: list all supported boards
-t type: specify build type
-h: show this text

Here're supported build types:
- release
- debug
- flash_xip
- flash_xip_release
- flash_sram_xip
- flash_sram_xip_release
- flash_uf2
- flash_uf2_release
- flash_sram_uf2
- flash_sram_uf2_release
- sec_core_img
- sec_core_img_release
C:\HFMICRO\SDK\V190\sd...

```

图 11 generate\_project 帮助

- 当前目录下将生成名为 hpm6p00evk\_build 的文件夹，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，双击可打开该工程。

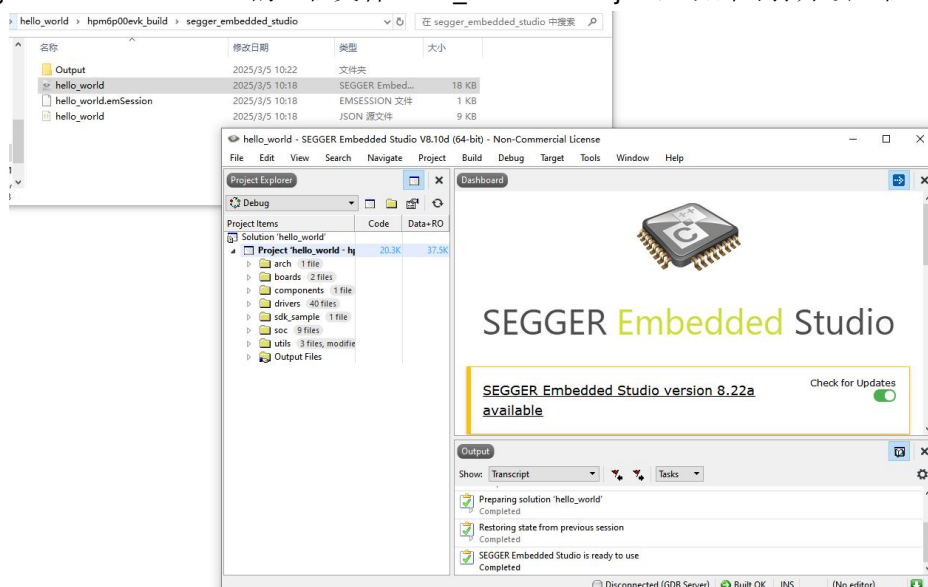


图 12 Segger Embedded Studio 打开 hello\_world 工程

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

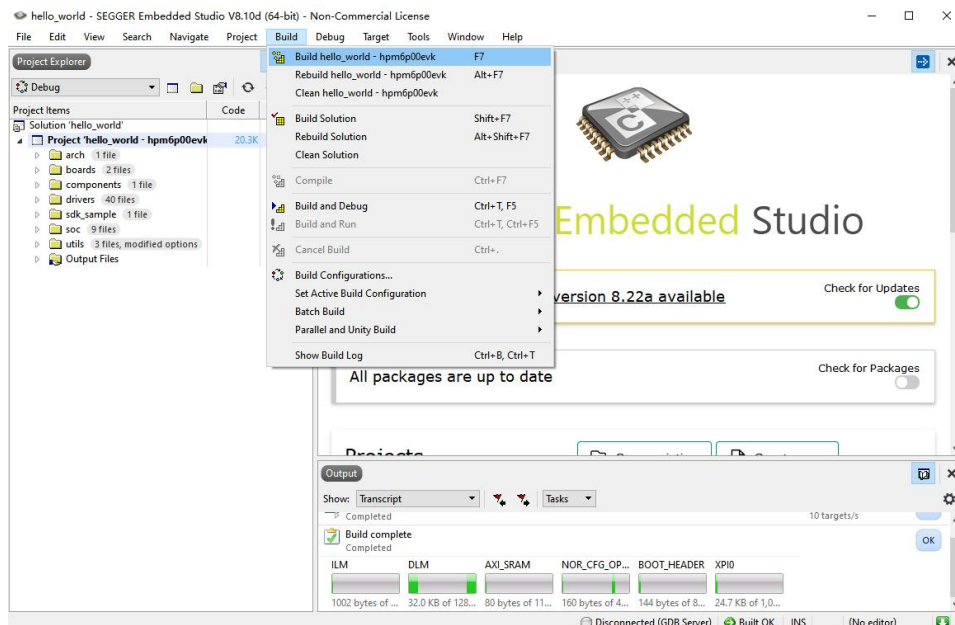


图 13 Segger Embedded Studio 编译 hello\_world 工程

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

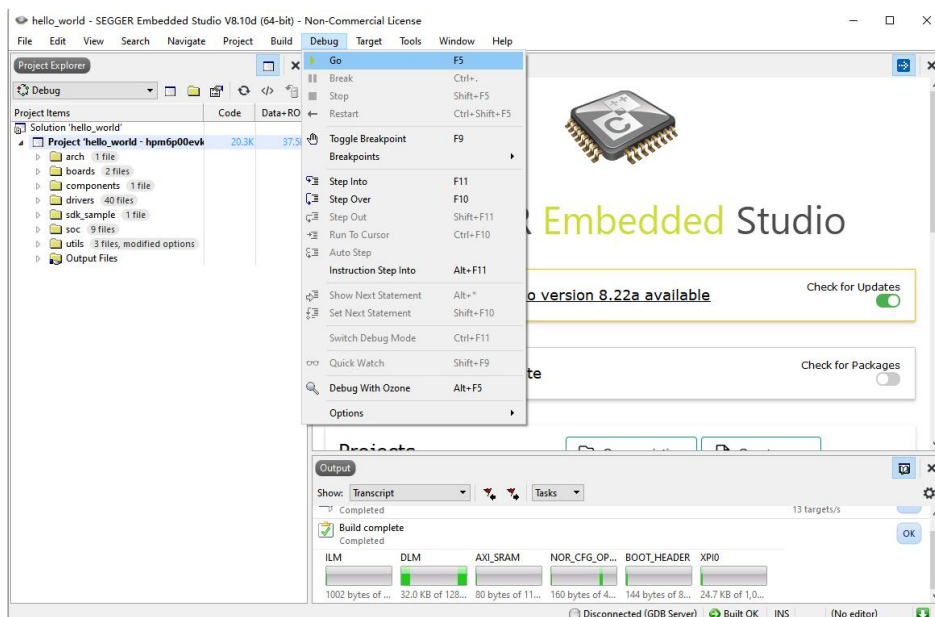


图 14 Segger Embedded Studio 调试 hello\_world 工程

## 11. 在 Segger Embedded Studio 中配置串口



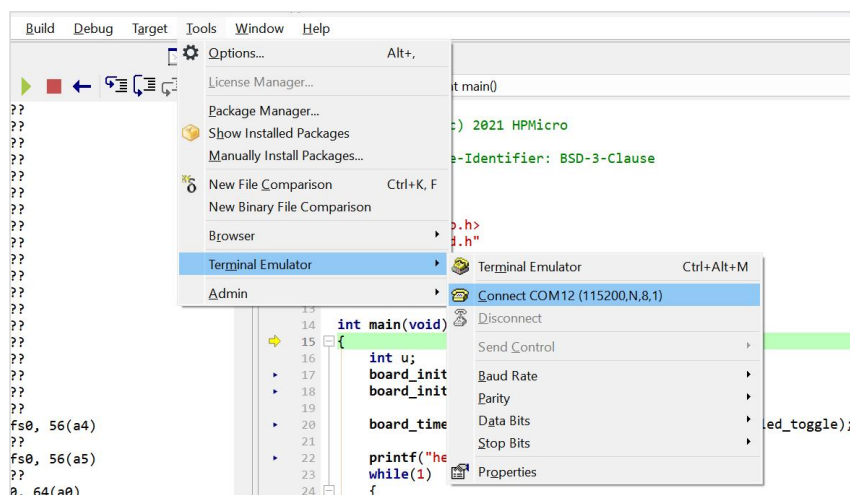


图 15 Segger Embedded Studio 中配置串口

## 12. 在Segger Embedded Studio 中连接串口

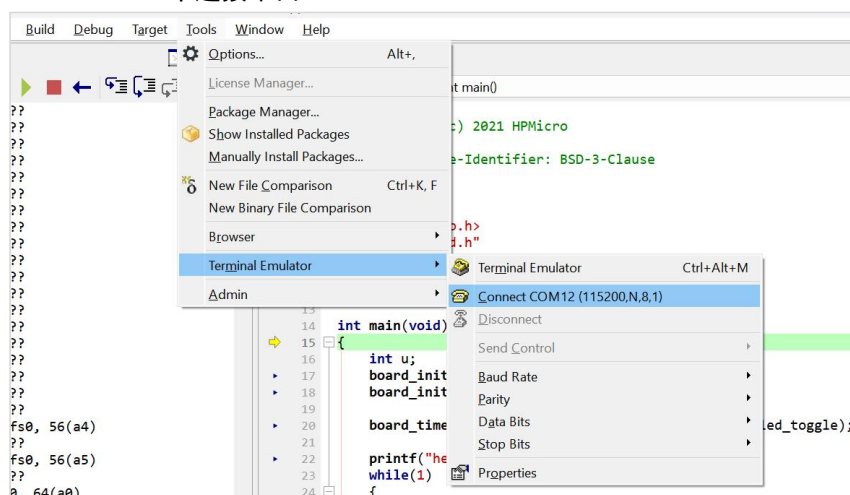


图 16 Segger Embedded Studio 中连接串口

## 13. 在Segger Embedded Studio 中打开串口

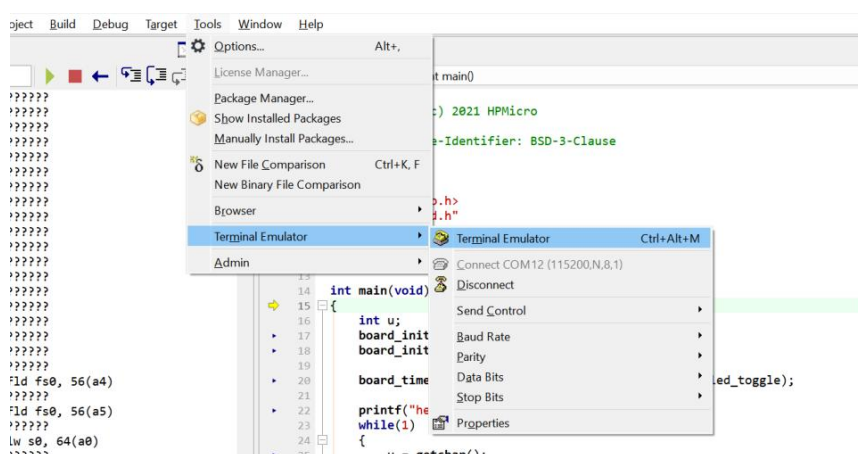


图 17 Segger Embedded Studio 打开串口

## 14. 运行 hello\_world

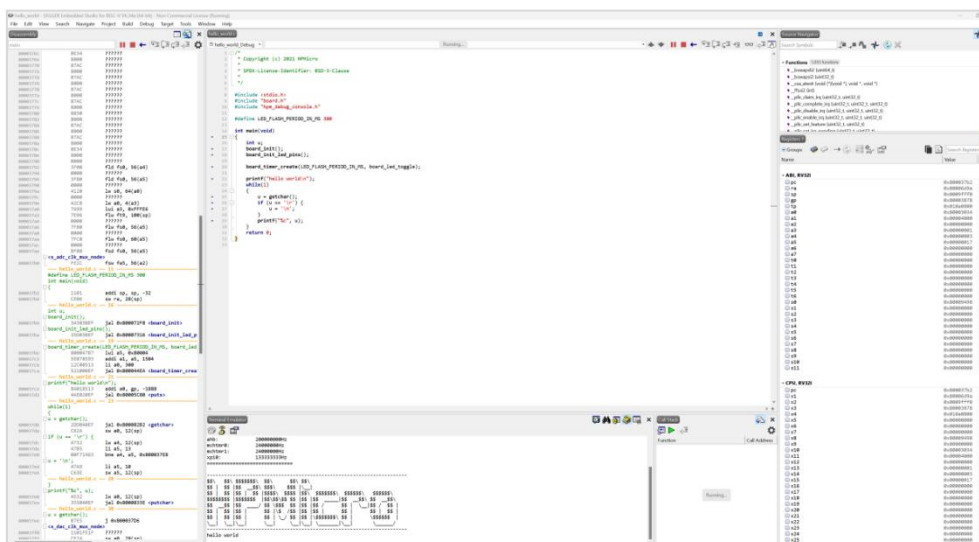


图 18 Segger Embedded Studio 运行 hello\_world

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

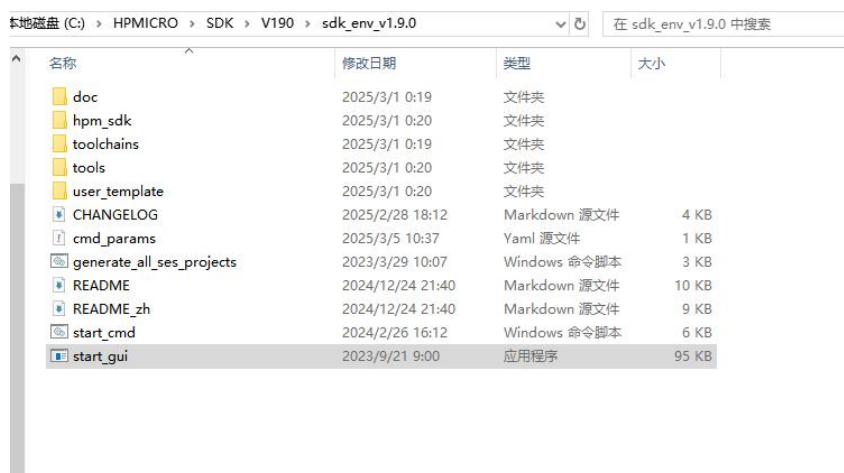


图 19 start\_gui 工具

16. 双击打开 start\_gui.exe

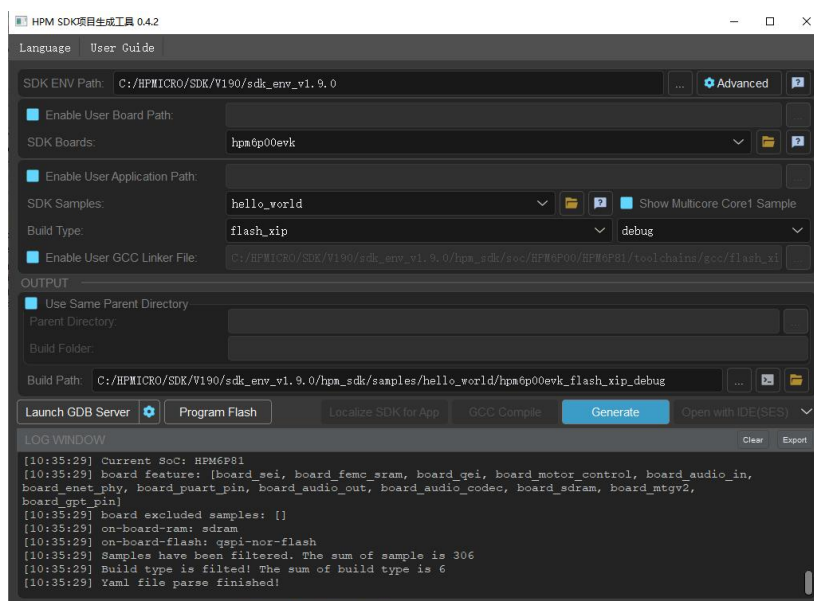


图 20 GUI project generator 工具操作界面

- 在 GUI project generator 界面中的“SDK Samples”下拉列表中选择“hello\_world”，在“SDK Boards”下拉列表中选择“hpm6p00evk”，在“Build Type”下拉列表中选择“flash\_xip”。点击“Generate”按钮，即可生成 flash\_xip 类型的 hello\_world 工程。

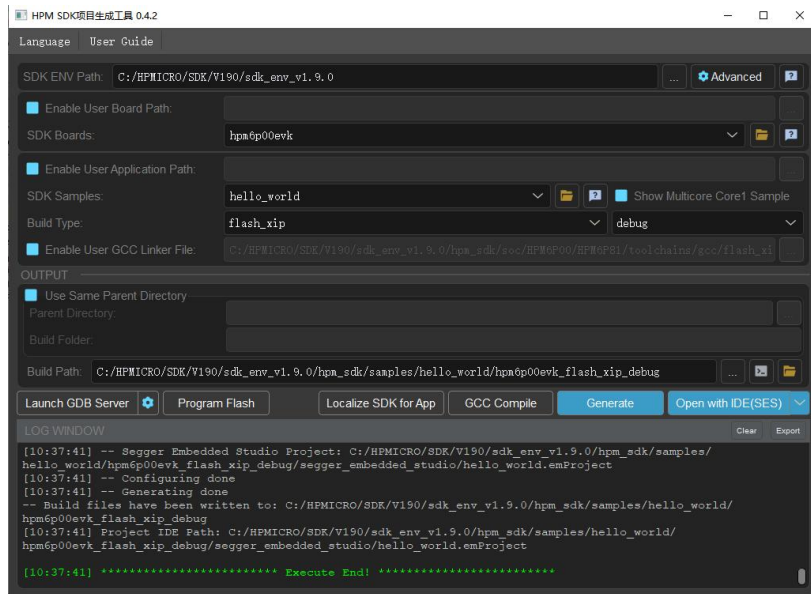


图 21 GUI project generator 生成 hello world 工程

- 点击“Open with IDE(SES)”即可打开 hello\_world 工程

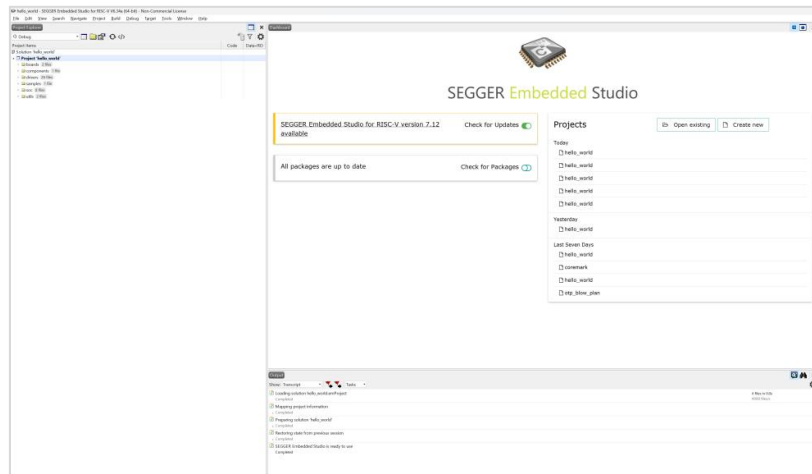


图 22 hello world 工程

## 3.5 调试出错常见原因

### ● FT2232 驱动没有正确安装

HPM6P00EVK 配备有板载的 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 23 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。

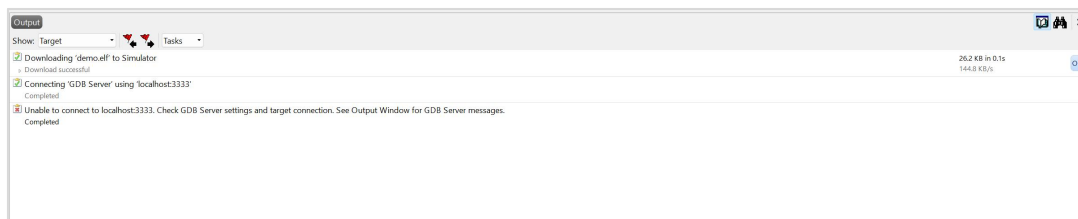


图 23 GDB Server 连接失败

- BOOT Pin 配置异常

HPM6P00EVK 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，之后再把 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。

- 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，也有可能導致类似现象。

- Debug 跳线帽没有正确配置

为了方便用户调试，HPM6P00EVK 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，如果要用 FT2232-to-JTAG 模式，需要把 TRST,TDI,TMS,TDO,TCK 跳线帽安装上；如果需要使用 JTAG 直连方式，则需要把这五个跳线帽拔掉。

- 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。

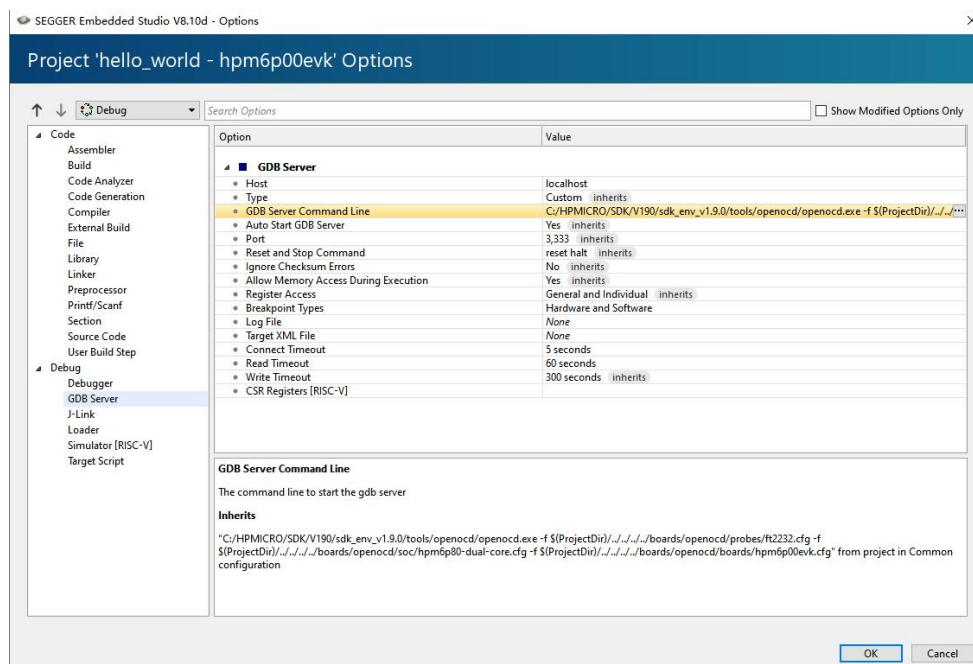


图 24 查看 openocd 配置

SDK 默认配置如图，使用 ft2232 进行调试

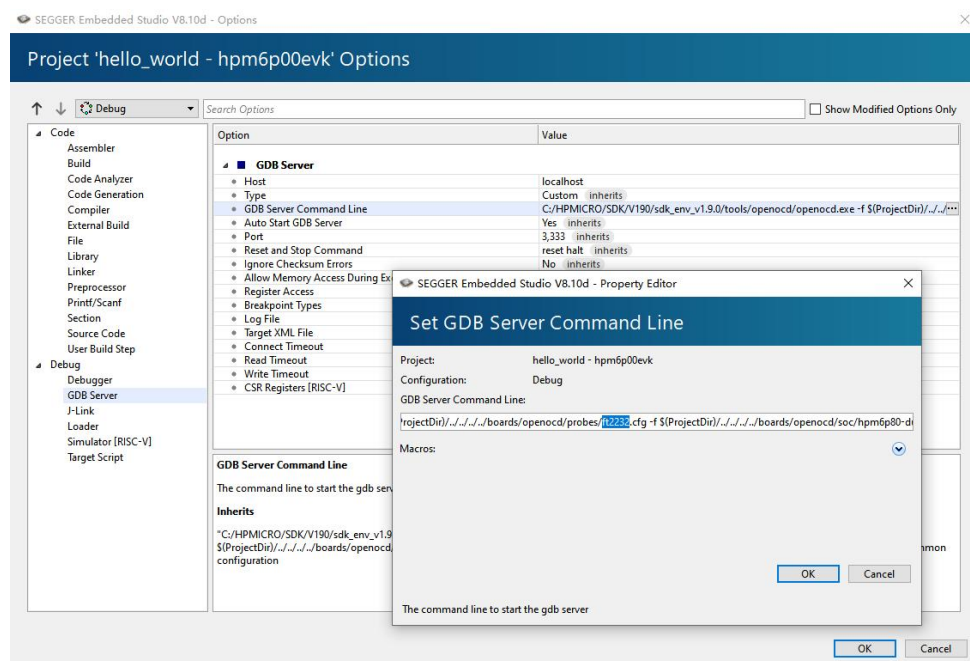


图 25 GDB Server 默认配置

如果用户使用其他调试器，则需要更改此配置文件。以 cmsis\_dap 调试器为例，要更改此配置文件如图

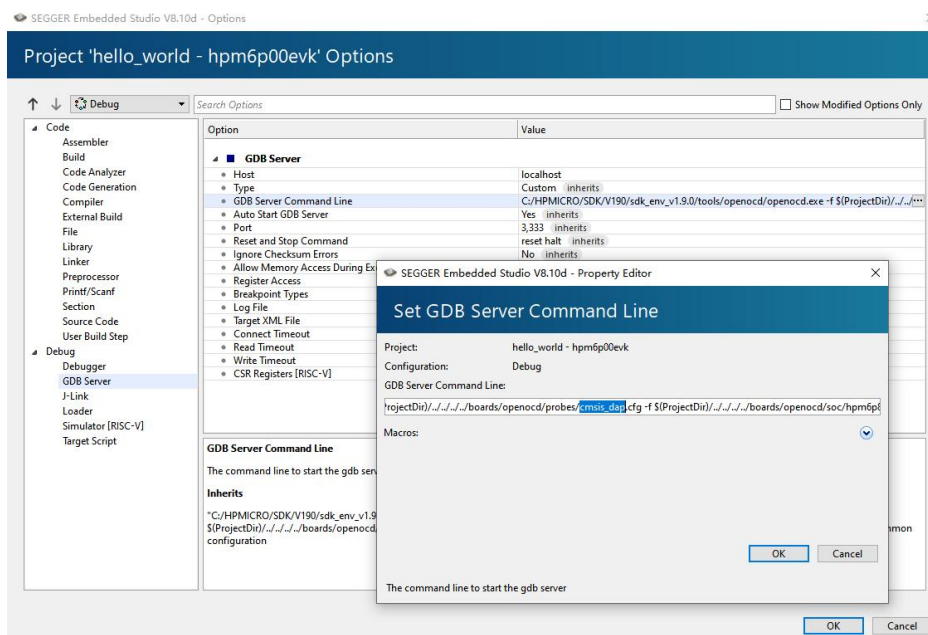


图 26 使用 cmsis\_dap 调试器 GDB Server 配置

## ● 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。

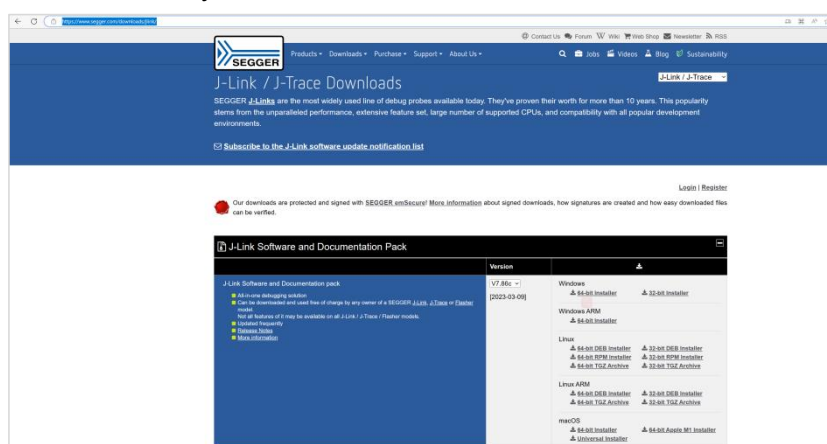


图 27 J-Link 驱动下载

## 安装 J-Link 驱动

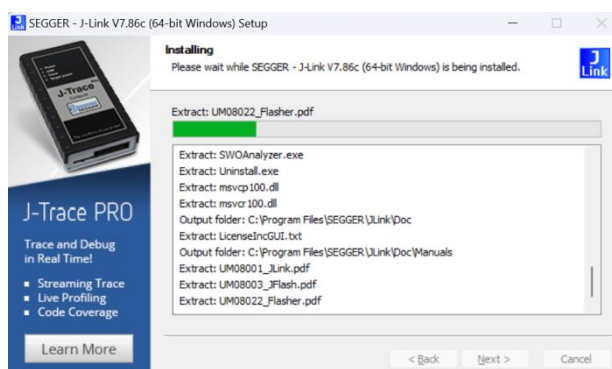


图 28 J-Link 驱动安装

安装驱动完成后，正确连接 J-Link JTAG 接口到 HPM6P00EVK J2 20pin 牛角插座，同时拔掉



## HPM6P00EVK

TRST,TDI,TMS,TDO,TCK 跳线帽。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。

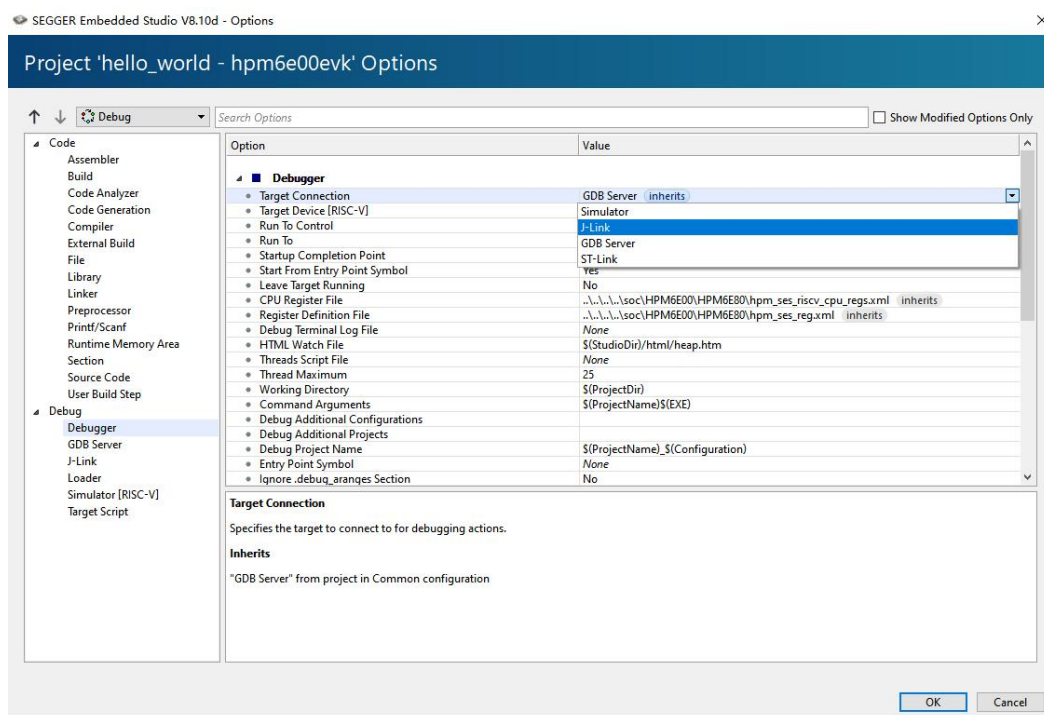


图 29 Target Connection 设置为 J-Link

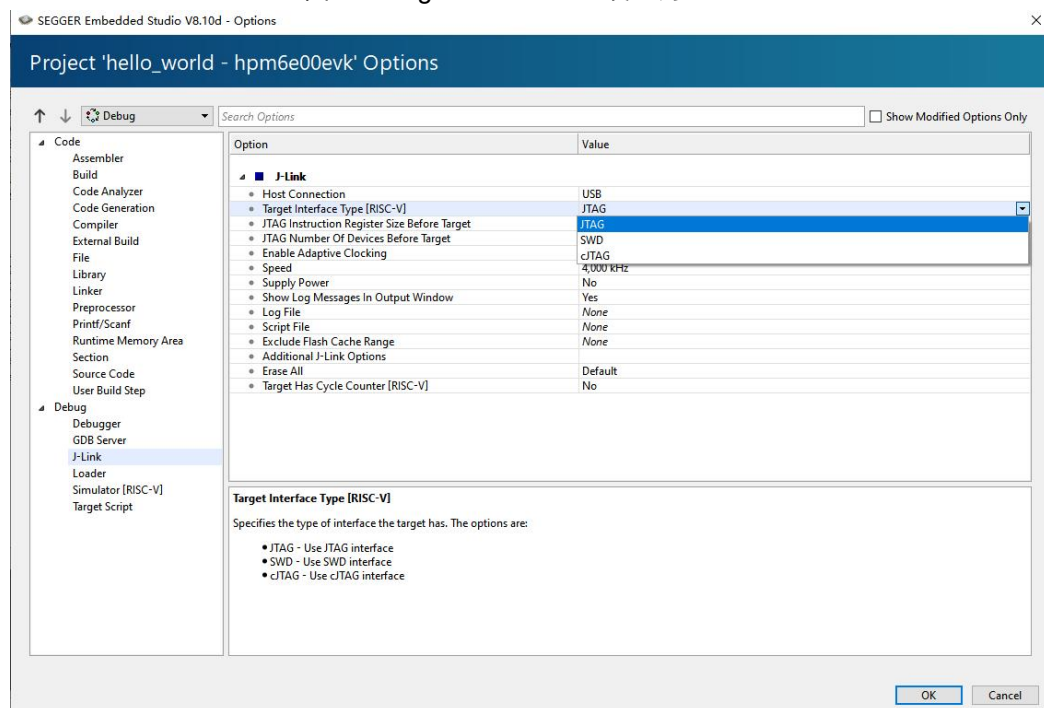


图 30 Target Interface Type 设置为 JTAG

## 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 或者 toolchain。

### 3.6.1 更新 sdk\_env 中的 SDK

- 下载 hpm\_sdk 并解压缩
- 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

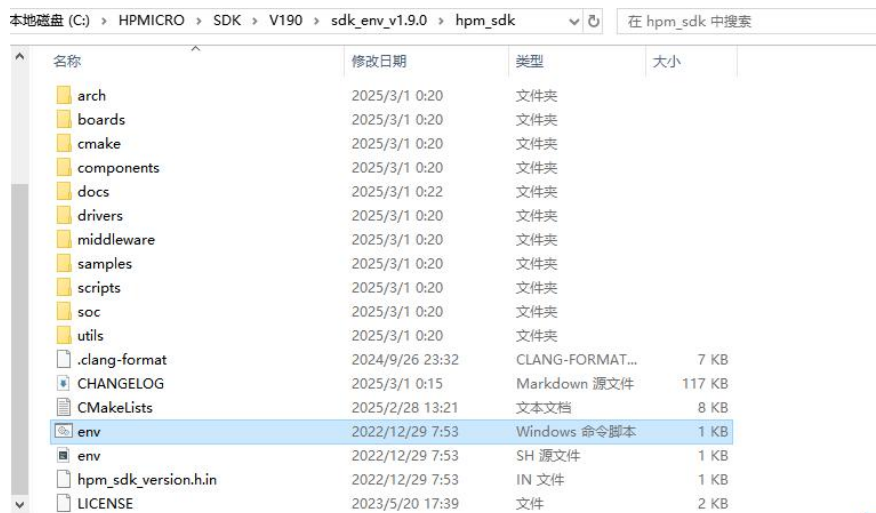


图 31 更新 SDK

### 3.6.2 更新 sdk\_env 中的 toolchain

- 下载 toolchain (以 rv32imac\_zicsr\_zifencei\_multilib\_b\_ext-win.zip 为例)
- 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。

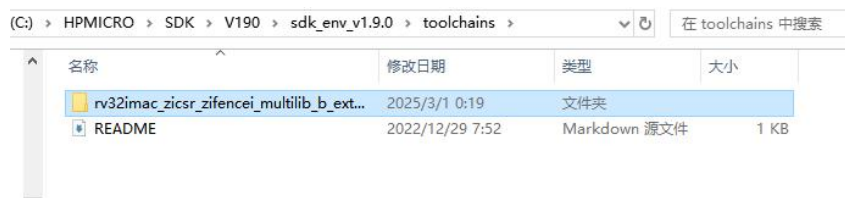
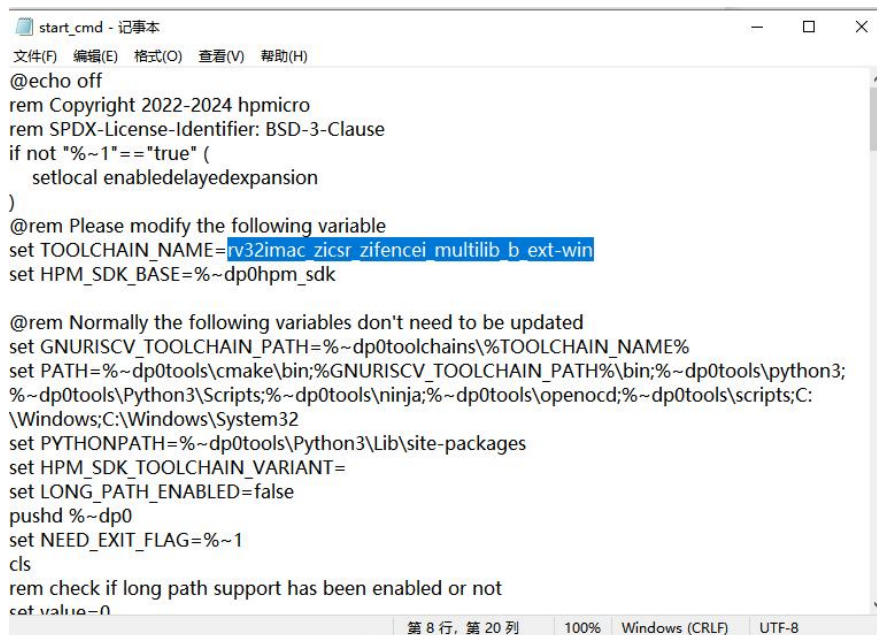


图 32 覆盖 toolchain

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





```

start_cmd - 记事本
文件(F)  编辑(E)  格式(O)  查看(V)  帮助(H)
@echo off
rem Copyright 2022-2024 hpmicro
rem SPDX-License-Identifier: BSD-3-Clause
if not "%~1"=="true" (
    setlocal enabledelayedexpansion
)
@rem Please modify the following variable
set TOOLCHAIN_NAME=rv32imac_zicsr_zifencei_multilib_b_ext-win
set HPM_SDK_BASE=%~dp0hpm_sdk

@rem Normally the following variables don't need to be updated
set GNURISCV_TOOLCHAIN_PATH=%~dp0toolchains\%TOOLCHAIN_NAME%
set PATH=%~dp0tools\cmake\bin;%GNURISCV_TOOLCHAIN_PATH%\bin;%~dp0tools\python3;%~dp0tools\Python3\Scripts;%~dp0tools\ninja;%~dp0tools\openocd;%~dp0tools\scripts;C:\Windows;C:\Windows\System32
set PYTHONPATH=%~dp0tools\Python3\Lib\site-packages
set HPM_SDK_TOOLCHAIN_VARIANT=
set LONG_PATH_ENABLED=false
pushd %~dp0
set NEED_EXIT_FLAG=%~1
cls
rem check if long path support has been enabled or not
cat value=0
    
```

图 33 更新 start\_cmd.cmd 中的 TOOLCHAIN\_NAME

- 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“C:/HPMICRO/SDK/V190/sdk\_env\_v1.9.0/toolchains/rv32imac\_zicsr\_zifencei\_multilib\_b\_ext-win”目录，点击“SAVE”。即可看到 LOG WINDOW 中更新完成的提示。

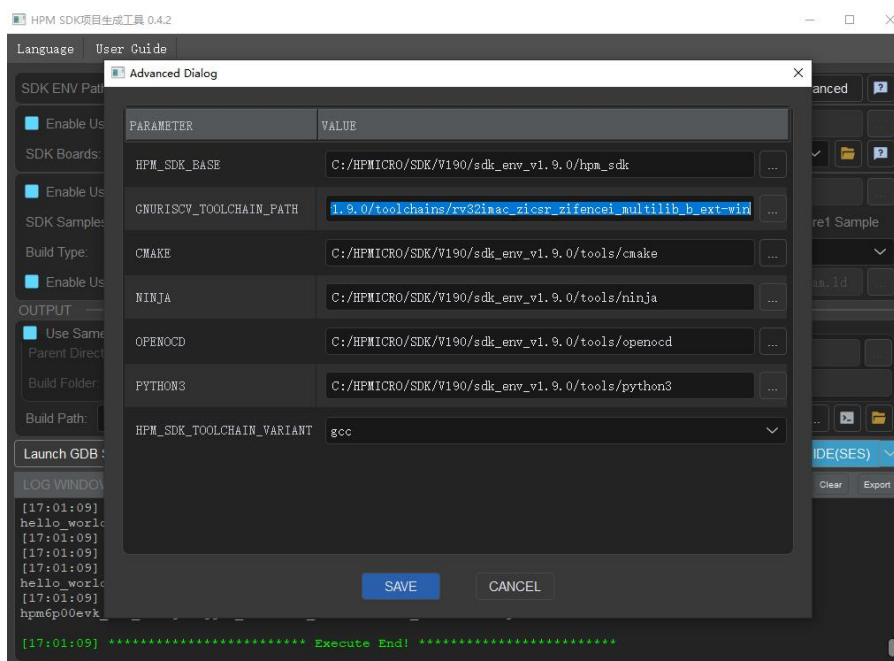


图 34 更新 start\_gui.exe 中的 TOOLCHAIN\_NAME

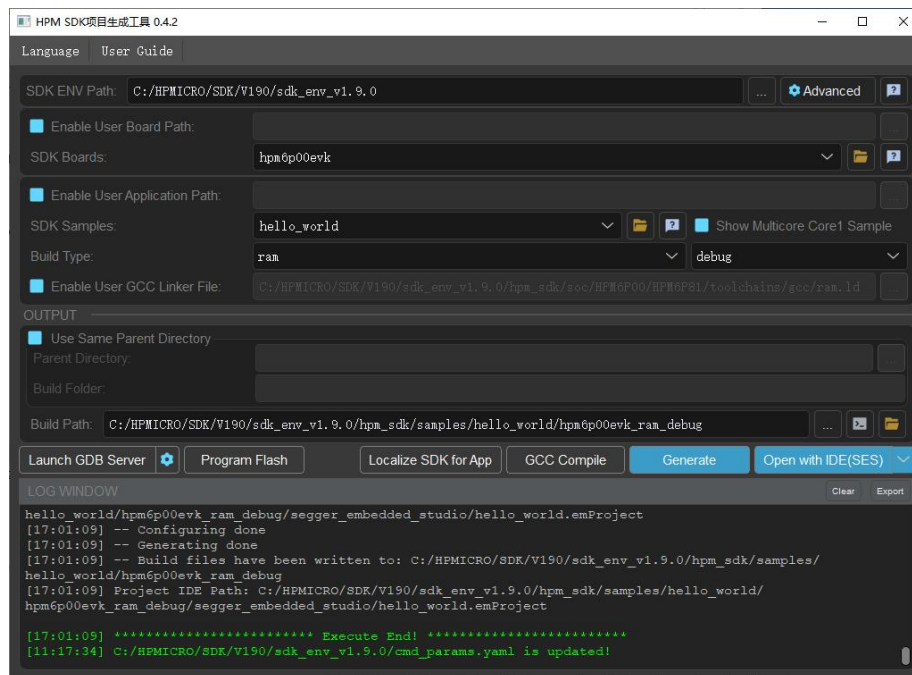


图 35 start\_gui.exe 更新 TOOLCHAIN 完成

## 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。

### 4.1 版本信息

日期	版本	描述
Rev1.0	2025/03/15	初版发布。