

HPM6200

HPM6200EVK 用户使用手册

适用于上海先楫半导体 HPM6200 系列高性能微控制器

目录

| | |
|---|----|
| 表格目录..... | 3 |
| 第一章 HPM6200EVK 简介..... | 5 |
| 第二章 硬件电路..... | 6 |
| 2.1 电路模块介绍..... | 6 |
| 第三章 软件开发套件..... | 11 |
| 3.1 简介..... | 11 |
| 3.2 环境以及依赖..... | 11 |
| 3.3 开发工具..... | 11 |
| 3.4 sdk_env/Segger Embedded Studio For RISC-V 使用快速指南..... | 11 |
| 3.5 调试出错常见原因..... | 21 |
| 3.6 更新 sdk_env 中的 SDK/toolchain 指南..... | 25 |
| 3.7 版本信息..... | 28 |
| 第四章 免责声明..... | 29 |

表格目录

| | |
|-------------------------|----|
| 表 1：主要器件位号对应器件功能名称..... | 5 |
| 表 2：电机接口管脚列表..... | 7 |
| 表 3：启动配置表..... | 8 |
| 表 4：P1 连接器列表..... | 9 |
| 表 5：P2 连接器列表..... | 10 |
| 表 6：版本信息..... | 28 |

图片目录

| | |
|--|----|
| 图 1：顶层器件位置图..... | 5 |
| 图 2：底层器件位置..... | 5 |
| 图 3：HPM6200EVK 硬件设计框图..... | 6 |
| 图 4：安装 FTDI 驱动..... | 11 |
| 图 5：查看 Windows 设备管理器..... | 12 |
| 图 6：sdk_env创建工程方式..... | 12 |
| 图 7：打开 sdk prompt..... | 13 |
| 图 8：构建目标板工程..... | 14 |
| 图 9：构建目标板 flash_xip 工程..... | 14 |
| 图 10：generate_project 帮助..... | 15 |
| 图 11：Segger Embedded Studio hello_world 工程..... | 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 配置串口..... | 17 |
| 图 16：Segger Embedded Studio 连接串口..... | 17 |
| 图 17：Segger Embedded Studio 打开串口..... | 18 |
| 图 18：Segger Embedded Studio 运行 hello_world..... | 18 |
| 图 19：start_gui 工具..... | 19 |
| 图 20：GUI project generator工具操作界面..... | 19 |
| 图 21：GUI project generator生成hello_world工程..... | 20 |
| 图 22：hello_world工程..... | 20 |
| 图 23：GDB Server连接失败..... | 21 |
| 图 24：查看openocd配置..... | 22 |
| 图 25：GDB Server默认配置..... | 22 |
| 图 26：使用cmsis-dap 调试器GDB Server配置..... | 22 |
| 图 27：J-Link驱动下载..... | 23 |
| 图 28：J-Link驱动安装..... | 23 |
| 图 29：Target Connection 设置为J-Link..... | 23 |
| 图 30：Target Interface Type设置为JTAG..... | 24 |
| 图 31：更新 SDK..... | 25 |
| 图 32：拷贝 toolchain..... | 26 |
| 图 33：更新 start_cmd.cmd 中 TOOLCHAIN_NAME..... | 26 |
| 图 34：更新 start_gui.exe 中 TOOLCHAIN_NAME..... | 27 |
| 图 35：start_gui.exe更新TOOLCHAIN完成..... | 27 |

第一章 HPM6200EVK 简介

HPM6200EVK 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。

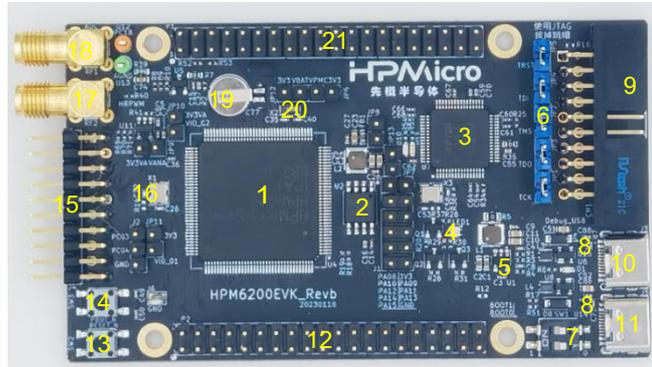


图 1：顶层器件位置图

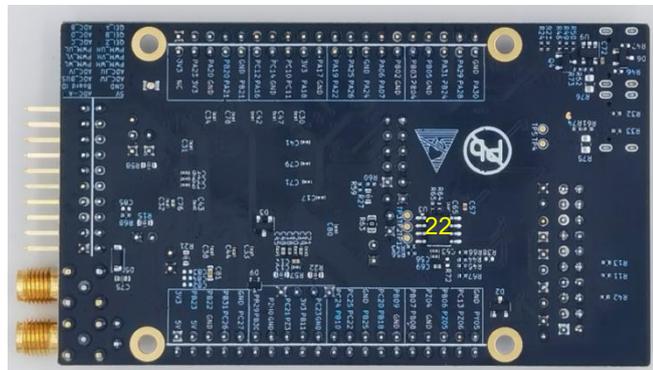


图 2：底层器件位置

| 序号 | 名称 | 序号 | 名称 |
|----|----------------|----|------------------|
| 1 | HPM6280 | 2 | FLASH |
| 3 | DEBUG 芯片 | 4 | 三色状态指示灯 |
| 5 | DCDC 芯片 | 6 | DEBUG选择跳针 |
| 7 | BOOT 配置拨码开关 | 8 | 过压保护芯片 |
| 9 | JTAG DEBUG接口 | 10 | Type-C DEBUG 接口 |
| 11 | USB0 Type-C 接口 | 12 | 扩展接口 P2 |
| 13 | RESET按键 | 14 | PBUTN 按键 |
| 15 | 电机驱动板接口 | 16 | 24MHZ 晶体 |
| 17 | HRPWM接口 | 18 | ADC接口 |
| 19 | 法拉电容 | 20 | 32.768K 晶体 |
| 21 | 扩展接口 P1 | 22 | FT2232HL配置EEPROM |

表 1：主要器件位号对应器件功能名称

第二章 硬件电路

HPM6200EVK 电源输入由Debug Type-C接口或者USB0 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

HPM6200EVK 系统架构如图 3。

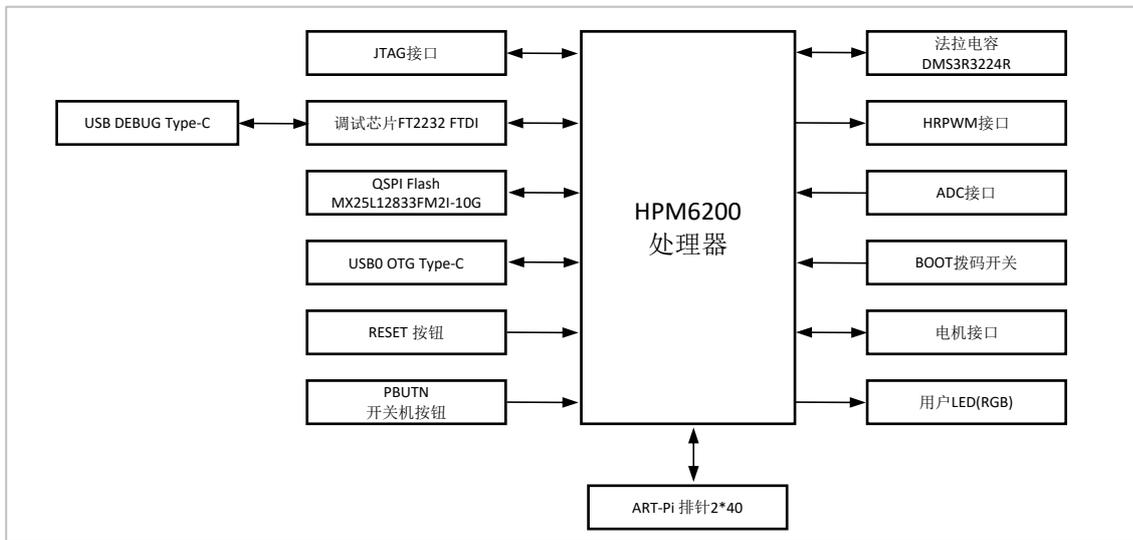


图 3：HPM6200EVK 硬件设计框图

2.1.2 电源

HPM6200EVK具有两种供电方式，可以选择Debug USB Type-C 或 USB0 OTG Type-C接口来为整板供电。板上提供板载0.22F/3.3V法拉电容，为 MCU 电池备份域供电。

2.1.3 DEBUG接口

HPM6200EVK提供两种DEBUG接口，默认是连接J3 DEBUG 接口到PC，通过 U7:FT2232HL 实现 USB到JTAG 和 UART 的转换。通过这种方式，用户可以通过USB同时访问芯片 JTAG 接口和 UART0 接口。UART0也可用于UART 串行下载和ISP。另外一路为20Pin牛角插座JTAG接口，用户可以通过调试直接连接芯片的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, TD0, TCK的跳线帽取掉，以使得板载DEBUG功能由U7:FT2232HL切换到JTAG直连。

2.1.4 FLASH

U2 是 HPM6200EVK 板上的 NOR FLASH 器件，4 位数据线，容量 128Mb，封装 SOP8 208mil。型号为 MX25L12833FM2I-10G。

2.1.5 USB 接口

J5 是 HPM6200EVK 板上的 USB 接口，连接器类型是 Type-C。支持 USB 2.0 OTG。同时支持USB串行下载和ISP，即通过USB给芯片下载bin文件，下载工具通过官网获取。

2.1.6 按键

SW2、SW3 分别是 EVK 板上对应的 RESET、PBUTN 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。

当系统处于电源管理域复位或 VPMC 掉电时，PBUTN 上检测到一次有效的按键（输入保持低电平约 0.5 秒），可以唤醒电源管理域电路，使系统重新工作。

更多PBUTN的使用请参考HPM6200用户手册。

用户通过 RESET 按键对 MCU 进行外部复位。

2.1.7 USER LED

LED1 是 HPM6200EVK 板上的一颗RGB LED ， R、G、B分别受PA27、PB01、PB19控制。

2.1.8 电机接口

J4是 HPM6200EVK 板上的20Pin电机驱动板接口，具有3对共6路PWM输出，9路ADC输入，3路QEI/HALL接口。J4 的信号列表如表 2所示。

| 引脚名 | 功能名 | 连接器编号 | | 功能名 | 引脚名 |
|------|------------|-------|----|------------|------|
| PC09 | ADC1.INA01 | 1 | 2 | 5.0V IN | |
| PC19 | ADC0.INA15 | 3 | 4 | GND | |
| PC05 | ADC0.INA01 | 5 | 6 | ADC2.INA04 | PC16 |
| PC17 | ADC1.INA09 | 7 | 8 | ADC0.INA11 | PC15 |
| PB17 | PWM0.P05 | 9 | 10 | PWM0.P04 | PB16 |
| PB15 | PWM0.P03 | 11 | 12 | PWM0.P02 | PB14 |
| PB13 | PWM0.P01 | 13 | 14 | PWM0.P00 | PB12 |
| PC06 | ADC0.INA02 | 15 | 16 | TRGM0.P08 | PB28 |
| PC08 | ADC1.INA00 | 17 | 18 | TRGM0.P07 | PB27 |
| PC07 | ADC0.INA03 | 19 | 20 | TRGM0.P06 | PB26 |

表 2: 电机接口管脚列表

2.1.9 BOOT 拨码开关设置

芯片默认是通过 SW1 拨码开关设置对应 BOOT_MODE[1:0]=[PA21:PA20] 引脚选择启动模式，配置如表 3所示。

| SW1 拨码开关 [1:0] | | 启动模式 | 说明 |
|----------------|-----|--------------------|-------------------------------|
| OFF | OFF | XPI NOR FLASH 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 3: 启动配置表

2.1.10 ADC 接口

RF1 是 HPM6200EVK 板上的 ADC接口，连接MCU PC18引脚。

2.1.11 HRPWM 接口

RF2 是 HPM6200EVK 板上的 HRPWM接口，连接MCU PB00引脚。

2.1.12 法拉电容

C77是 HPM6200EVK 板上的一颗法拉电容，对应型号DMS3R3224R，容量为0.22F，电压3.3V。为 MCU 电池备份域供电。

2.1.13 扩展 I0 接口

扩展 I0 接口包括 P1、P2 两个连接器接口。P1、P2接口机械尺寸与ART-Pi兼容。P1、P2 的信号列表如表 4 和表 5。

| 引脚名 | 功能名 | 连接器编号 | | 功能名 | 引脚名 |
|------|-----------|-------|----|-----------|------|
| | 3.3V | 1 | 2 | 5.0V | |
| PB23 | I2C0.SDA | 3 | 4 | 5.0V | |
| PB22 | I2C0.SCL | 5 | 6 | GND | |
| PB31 | GPIOB.31 | 7 | 8 | UART2.TXD | PC26 |
| | GND | 9 | 10 | UART2.RXD | PC27 |
| PB29 | GPIOB.29 | 11 | 12 | GPIOB.30 | PB30 |
| PZ00 | GPIOZ.00 | 13 | 14 | GND | |
| PC21 | GPIOC.21 | 15 | 16 | GPIOZ.03 | PZ03 |
| | 3.3V | 17 | 18 | UART6.RXD | PB11 |
| PC23 | SPI2.MOSI | 19 | 20 | GND | |
| PC24 | SPI2.MISO | 21 | 22 | UART6.TXD | PB10 |
| PC25 | SPI2.SCLK | 23 | 24 | SPI2.CSN | PC22 |
| | GND | 25 | 26 | GPIOB.25 | PB25 |
| PC20 | GPIOC.20 | 27 | 28 | GPIOB.18 | PB18 |
| PB09 | GPIOB.09 | 29 | 30 | GND | |
| PB07 | GPIOB.07 | 31 | 32 | GPIOB.08 | PB08 |
| PZ04 | CAN3.TXD | 33 | 34 | GND | |
| PB06 | GPIOB.06 | 35 | 36 | CAN3.RXD | PZ05 |
| PC13 | GPIOC.13 | 37 | 38 | GPIOZ.06 | PZ06 |
| | GND | 39 | 40 | GPIOY.05 | PY05 |

表 4: P1 连接器列表

| 引脚名 | 功能名 | 连接器编号 | | 功能名 | 引脚名 |
|------|------------|-------|----|-----------|------|
| | 3.3V | 1 | 2 | NC | |
| PA23 | GPIOA.23 | 3 | 4 | 3.3V | |
| PA20 | GPIOA.20 | 5 | 6 | GND | |
| PB20 | CAN0.TXD | 7 | 8 | GPIOA.21 | PA21 |
| | GND | 9 | 10 | CAN0.RXD | PB21 |
| PC12 | ADC0_INA8 | 11 | 12 | GPIOA.16 | PA16 |
| PC14 | ADC0_INA10 | 13 | 14 | GND | |
| PC10 | ADC0_INA6 | 15 | 16 | ADC1_INA3 | PC11 |
| | GND | 17 | 18 | GPIOA.18 | PA18 |
| PA17 | GPIOA.17 | 19 | 20 | GND | |
| PA19 | GPIOA.19 | 21 | 22 | GPIOA.22 | PA22 |
| PA25 | GPIOA.25 | 23 | 24 | GPIOA.26 | PA26 |
| | GND | 25 | 26 | GPIOA.24 | PA24 |
| PA6 | GPIOA.6 | 27 | 28 | GPIOA.7 | PA07 |
| PB02 | SPI1.CSN | 29 | 30 | GND | |
| PB03 | SPI1.MISO | 31 | 32 | SPI1.SCLK | PB04 |
| PB05 | SPI1.MOSI | 33 | 34 | GND | |
| PA31 | GPIOA.31 | 35 | 36 | GPIOB.24 | PB24 |
| PA29 | GPIOA.29 | 37 | 38 | GPIOA.28 | PA28 |
| | GND | 39 | 40 | GPIOA.30 | PA30 |

表 5: P2 连接器列表

注：P1和P2机械尺寸和部分管脚功能与ART-Pi兼容，用户如果想驱动ART-Pi扩展板时请先查看扩展板与P1，P2管脚功能是否能够匹配，确认功能无误后就可以开启自己的DIY之旅了。

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](#) 下载最新版本。先楦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 `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. 下载最新版本 `sdk_env_vx.x.x.zip` 压缩包后解压 (本文 `sdk_env_v1.1.0-rc2` 为例, 推荐获取最新版本 `sdk_env`)。

Note: 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_v1.1.0-rc2\tools\FTDI_InstallDriver.exe` 以安装可用于调试的 FT2232 驱动。

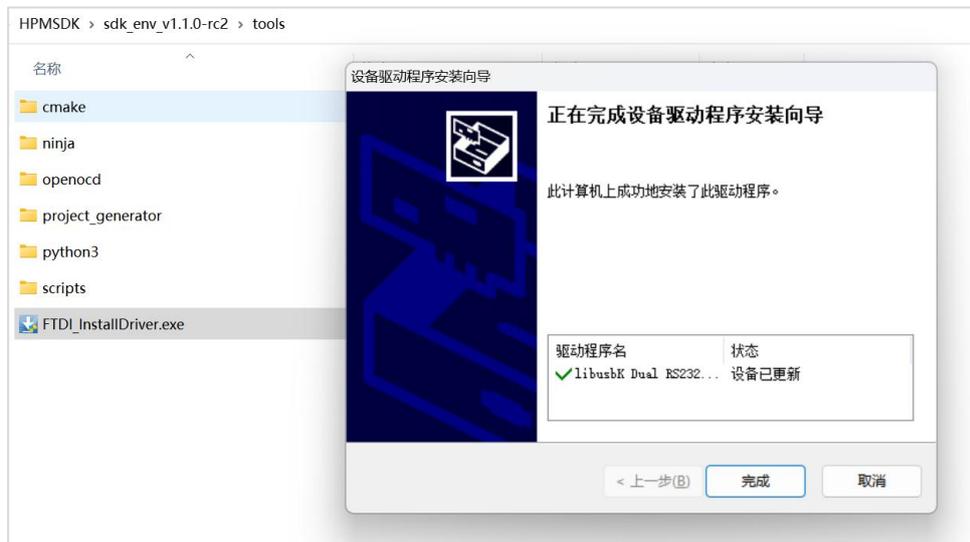


图 4: 安装 FTDI 驱动

正确安装驱动后, 使用 USB Type-C 线缆将 HPM6200EVK 上的 J3 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), 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。

```
HPMSDK > sdk_env_v1.1.0-rc2
```

| 名称 | 修改日期 | 类型 | 大小 |
|-------------------------------|-----------------|--------------|--------|
| doc | 2023/3/29 10:19 | 文件夹 | |
| hpm_sdk | 2023/3/29 10:19 | 文件夹 | |
| toolchains | 2023/3/17 7:18 | 文件夹 | |
| tools | 2023/3/29 10:20 | 文件夹 | |
| CHANGELOG.md | 2023/3/17 7:16 | Markdown 源文件 | 2 KB |
| cmd_params.yaml | 2023/2/13 9:06 | Yaml 源文件 | 1 KB |
| generate_all_ses_projects.cmd | 2022/12/29 7:52 | Windows 命令脚本 | 3 KB |
| README.md | 2022/12/29 7:52 | Markdown 源文件 | 3 KB |
| README_zh.md | 2022/12/29 7:52 | Markdown 源文件 | 2 KB |
| start_cmd.cmd | 2022/12/29 7:52 | Windows 命令脚本 | 6 KB |
| start_gui.exe | 2023/2/13 9:06 | 应用程序 | 314 KB |

图 6: sdk_env创建工程方式

以命令行工具为例，双击打开 sdk_env_v1.1.0-rc2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。

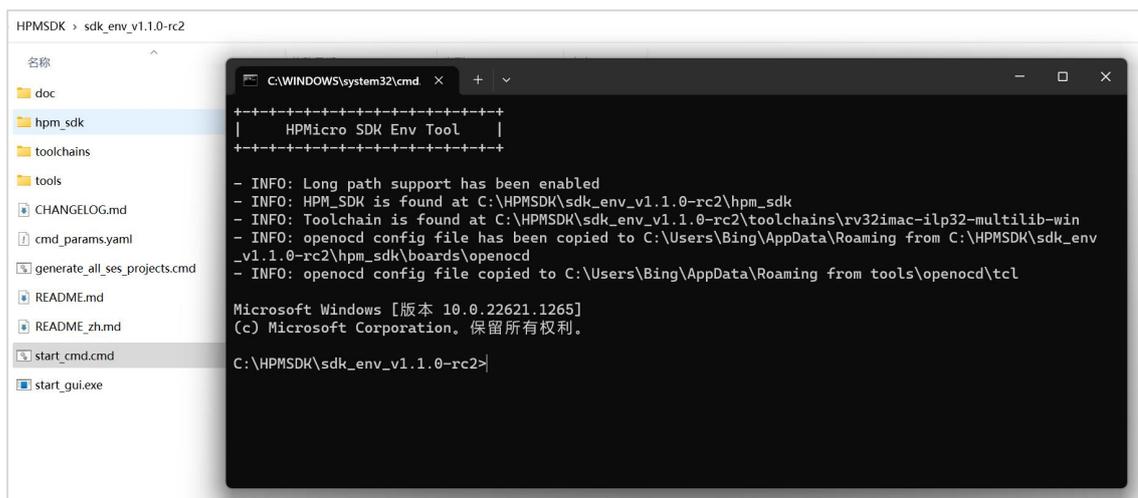


图 7: 打开 sdk prompt

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 HPM6200EVK 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 hpm6200evk

```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [版本 10.0.22621.1265]
(c) Microsoft Corporation. 保留所有权利。

C:\HPMSDK\sdk_env_v1.1.0-rc2>cd %HPM_SDK_BASE%\samples\hello_world

C:\HPMSDK\sdk_env_v1.1.0-rc2\hpm_sdk\samples\hello_world>generate_project -list
hpm6200evk
hpm6300evk
hpm6750evk
hpm6750evk2
hpm6750evkmini

C:\HPMSDK\sdk_env_v1.1.0-rc2\hpm_sdk\samples\hello_world>generate_project -b hpm6200evk
-- Application: C:/HPMSDK/sdk_env_v1.1.0-rc2/hpm_sdk/samples/hello_world
-- Board: hpm6200evk
-- Found toolchain: gnu (C:\HPMSDK\sdk_env_v1.1.0-rc2\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:/HPMSDK/sdk_env_v1.1.0-rc2/toolchains/rv32imac-ilp32-multilib-win/bin/riscv32-unknown-elf-gcc.exe
-- hpm_sdk: 1.1.0 (C:\HPMSDK\sdk_env_v1.1.0-rc2\hpm_sdk)
-- Segger linker script: C:/HPMSDK/sdk_env_v1.1.0-rc2/hpm_sdk/soc/HPM6280/toolchains/segger/ram.icf
-- Segger device name: HPM6280xPax
-- Segger Embedded Studio Project: C:/HPMSDK/sdk_env_v1.1.0-rc2/hpm_sdk/samples/hello_world/hpm6200evk_build/segger_embedded_studio/hello_world.emProject
-- Configuring done
-- Generating done
-- Build files have been written to: C:/HPMSDK/sdk_env_v1.1.0-rc2/hpm_sdk/samples/hello_world/hpm6200evk_build

C:\HPMSDK\sdk_env_v1.1.0-rc2\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 nor flash 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. x + v
C:\HPMSDK\sdk_env_v1.1.0-rc2\hpm_sdk\samples\hello_world>generate_project -b hpm6200evk -t flash_xip
hpm6200evk_build directory exists, please remove it manually or run this script with -f option

C:\HPMSDK\sdk_env_v1.1.0-rc2\hpm_sdk\samples\hello_world>generate_project -b hpm6200evk -t flash_xip
-- Application: C:/HPMSDK/sdk_env_v1.1.0-rc2/hpm_sdk/samples/hello_world
-- Board: hpm6200evk
-- Found toolchain: gnu (C:\HPMSDK\sdk_env_v1.1.0-rc2\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:/HPMSDK/sdk_env_v1.1.0-rc2/toolchains/rv32imac-ilp32-multilib-win/bin/riscv32-unknown-elf-gcc.exe
-- hpm_sdk: 1.1.0 (C:\HPMSDK\sdk_env_v1.1.0-rc2\hpm_sdk)
-- Segger linker script: C:/HPMSDK/sdk_env_v1.1.0-rc2/hpm_sdk/soc/HPM6280/toolchains/segger/flash_xip.icf
-- Segger device name: HPM6280xPax
-- Segger Embedded Studio Project: C:/HPMSDK/sdk_env_v1.1.0-rc2/hpm_sdk/samples/hello_world/hpm6200evk_build/segger_embedded_studio/hello_world.emProject
-- Configuring done
-- Generating done
-- Build files have been written to: C:/HPMSDK/sdk_env_v1.1.0-rc2/hpm_sdk/samples/hello_world/hpm6200evk_build

C:\HPMSDK\sdk_env_v1.1.0-rc2\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 3) 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生影响。

注：更多 generate_project 使用方法可以通过执行以下命令查看。

> generate_project -h

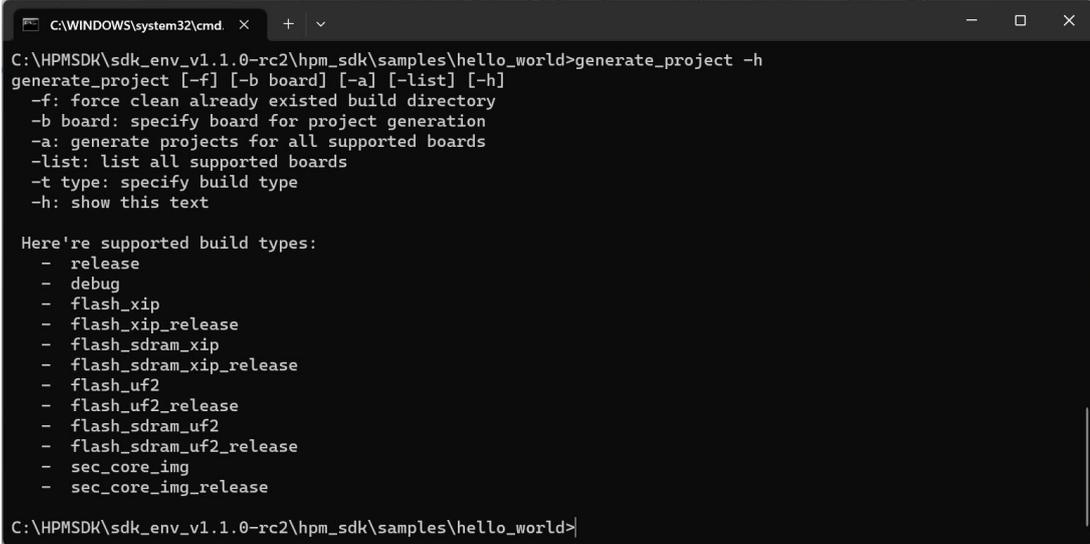


图 10: generate_project 帮助

- 8. 当前目录下将生成名为 hpm6200evk_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，双击可打开该工程。

HPMSDK > sdk_env_v1.1.0-rc2 > hpm_sdk > samples > hello_world > hpm6200evk_build > segger_embedded_studio

| 名称 | 修改日期 | 类型 | 大小 |
|-----------------------|-----------------|--------------------|-------|
| hello_world.emProject | 2023/3/29 10:37 | SEGGER Embedded... | 13 KB |
| hello_world.emSession | 2023/3/29 10:37 | EMSESSION 文件 | 1 KB |

图 11: Segger Embedded Studio hello_world 工程

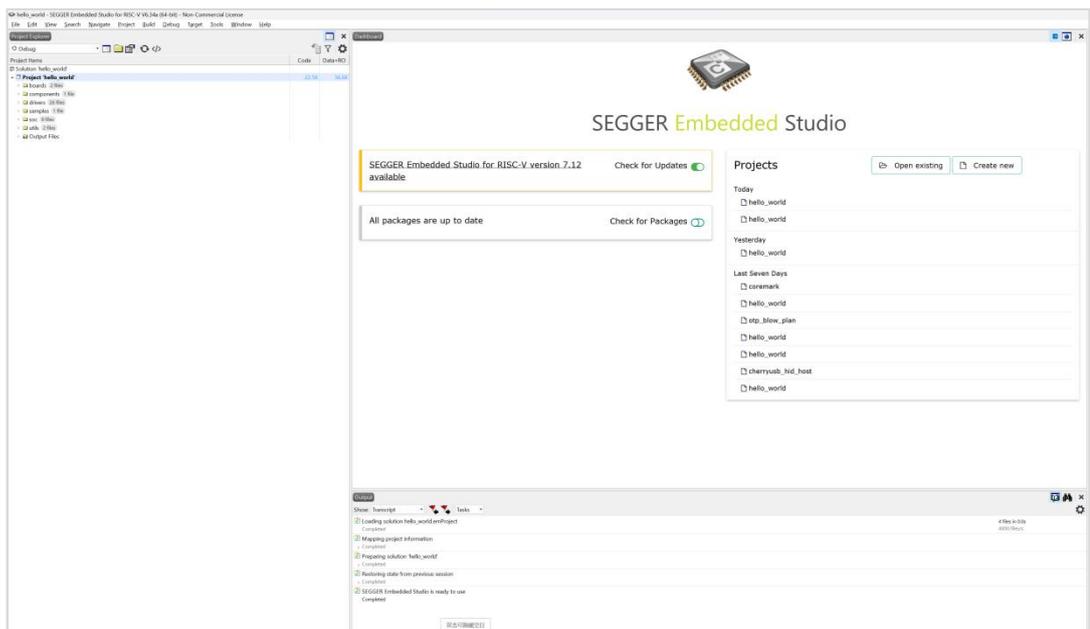


图 12: Segger Embedded Studio 打开 hello_world 工程

9. 使用 Segger Embedded Studio 打开 hello_world 工程即可进行编译。

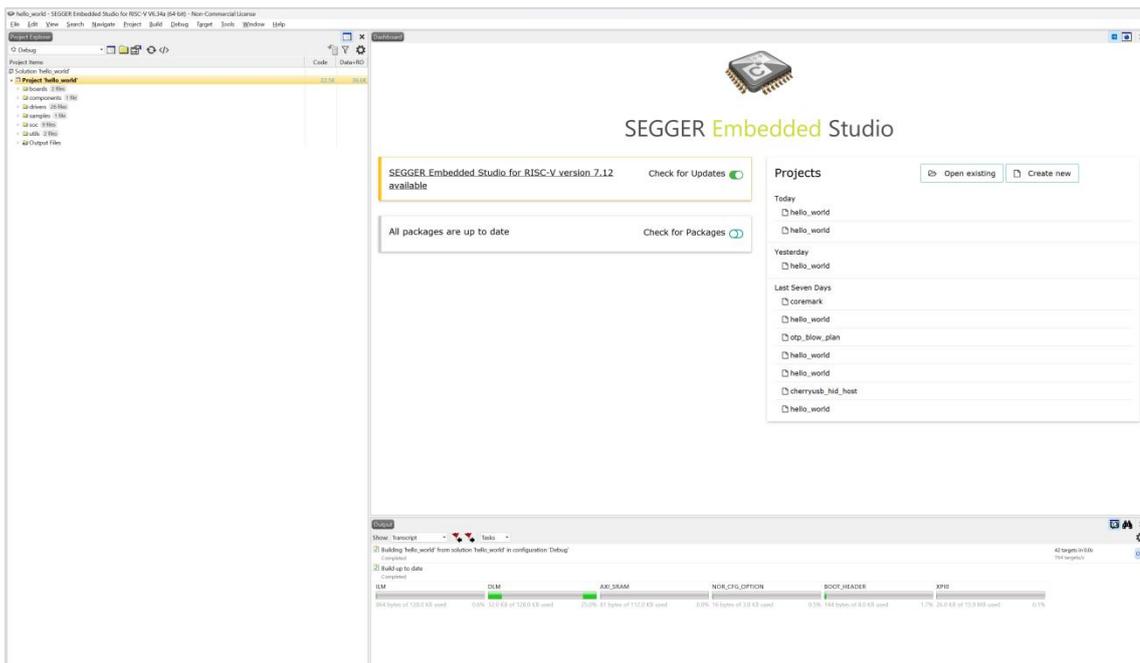


图 13: Segger Embedded Studio 编译 hello_world 工程

10. 使用 Segger Embedded Studio 进行 hello_world 调试。

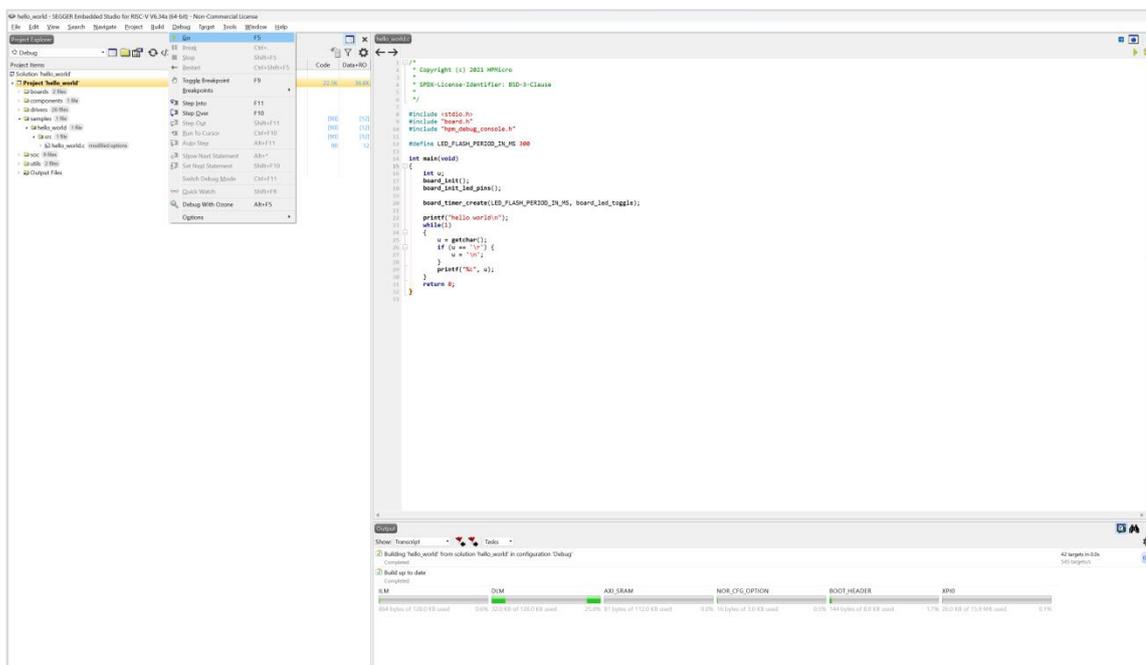


图 14: Segger Embedded Studio 调试 hello_world 工程

11. 在 Segger Embedded Studio 中配置串口。

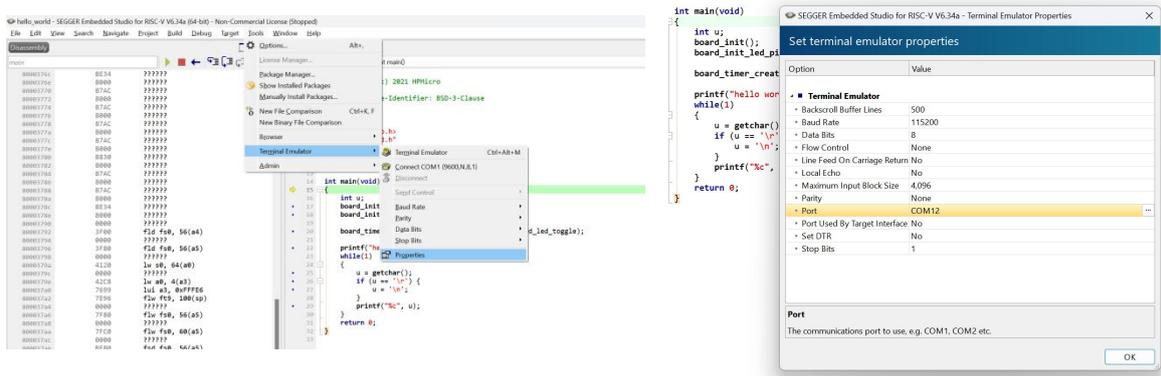


图 15: Segger Embedded Studio 配置串口

12. 在 Segger Embedded Studio 中连接串口。

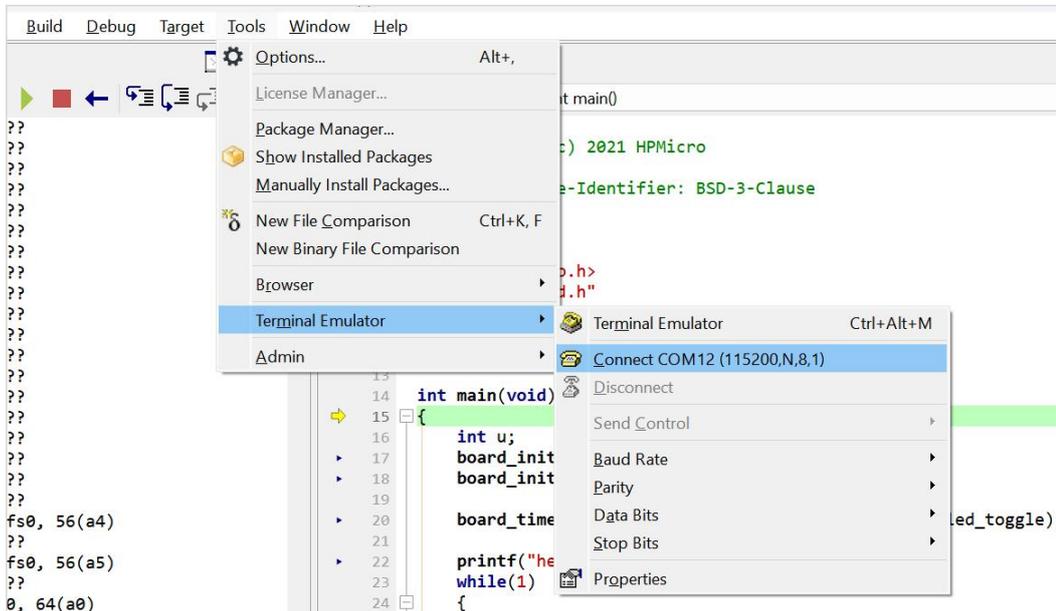


图 16: Segger Embedded Studio 连接串口

13. 在 Segger Embedded Studio 中打开串口。

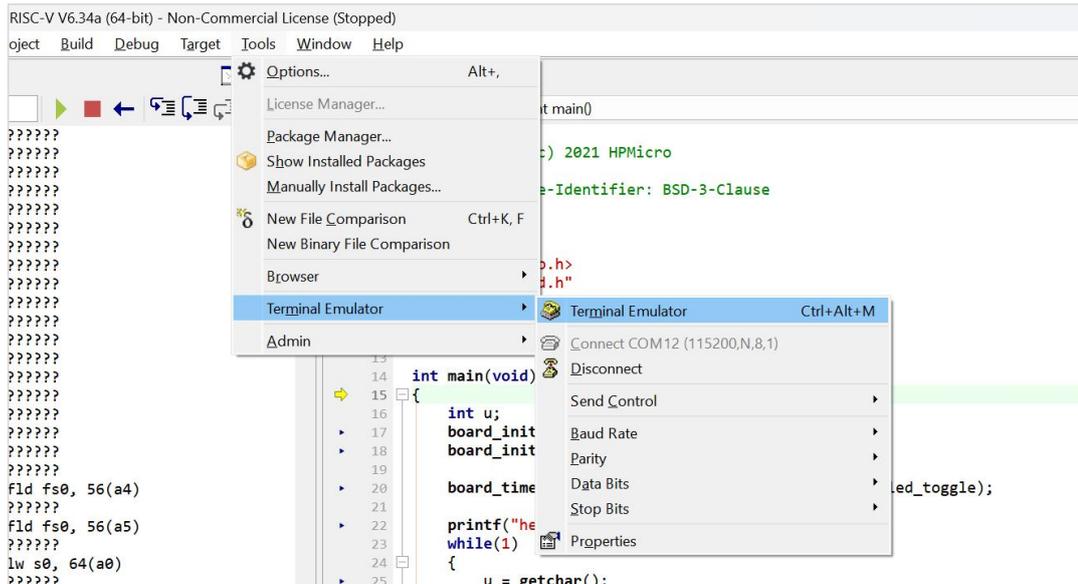


图 17: Segger Embedded Studio 打开串口

14. 运行 hello_world。

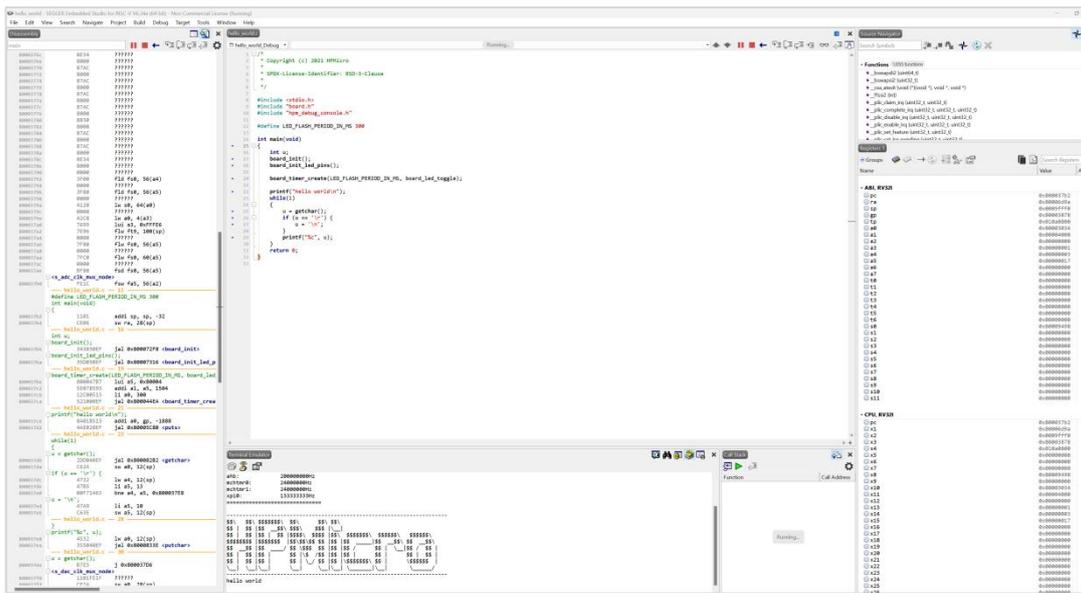


图 18: Segger Embedded Studio 运行 hello_world

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

| 名称 | 修改日期 | 类型 | 大小 |
|-------------------------------|-----------------|--------------|--------|
| doc | 2023/3/29 10:19 | 文件夹 | |
| hpm_sdk | 2023/3/29 10:19 | 文件夹 | |
| toolchains | 2023/3/29 10:19 | 文件夹 | |
| tools | 2023/3/29 10:20 | 文件夹 | |
| CHANGELOG.md | 2023/3/17 7:16 | Markdown 源文件 | 2 KB |
| cmd_params.yaml | 2023/2/13 9:06 | Yaml 源文件 | 1 KB |
| generate_all_ses_projects.cmd | 2022/12/29 7:52 | Windows 命令脚本 | 3 KB |
| README.md | 2022/12/29 7:52 | Markdown 源文件 | 3 KB |
| README_zh.md | 2022/12/29 7:52 | Markdown 源文件 | 2 KB |
| start_cmd.cmd | 2022/12/29 7:52 | Windows 命令脚本 | 6 KB |
| start_gui.exe | 2023/2/13 9:06 | 应用程序 | 314 KB |

图 19: start_gui 工具

16. 双击打开start_gui.exe。

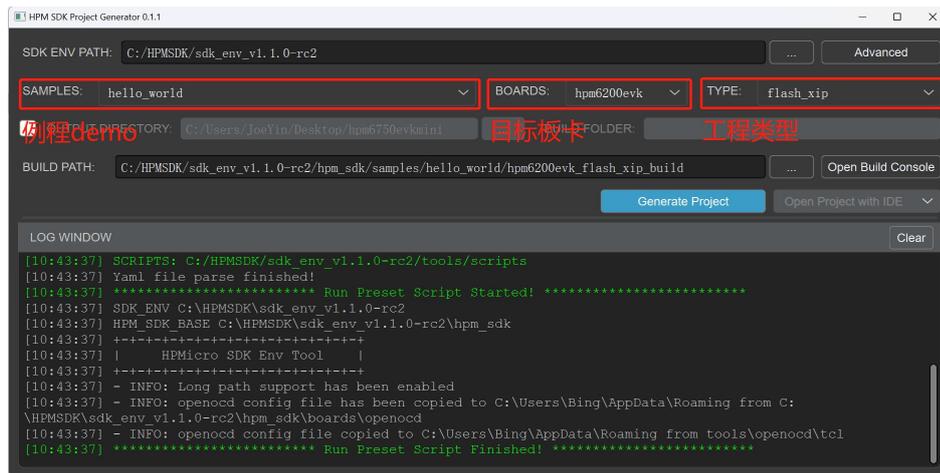


图 20: GUI project generator工具操作界面

- 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” hpm6200evk”，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 21所示。

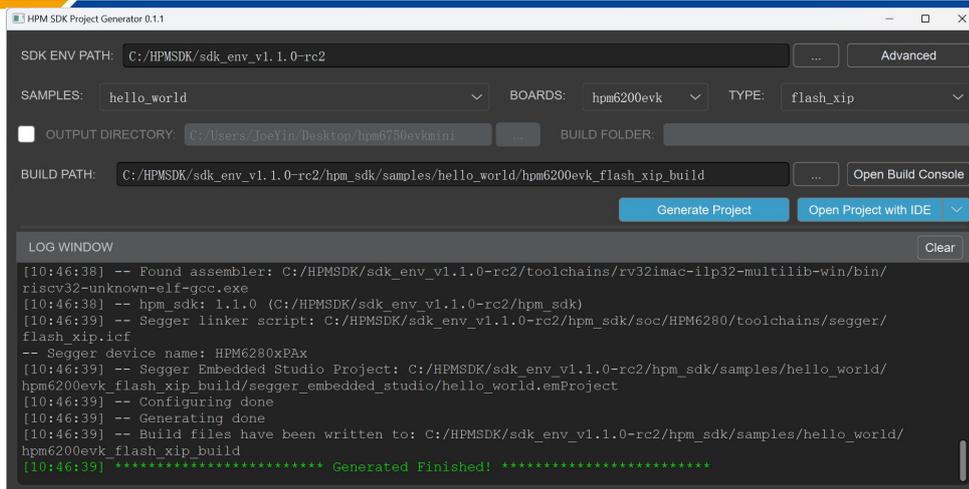


图 21: GUI project generator生成hello_world工程

18. 点击” Open Project with IDE” 即可打开hello_world工程。

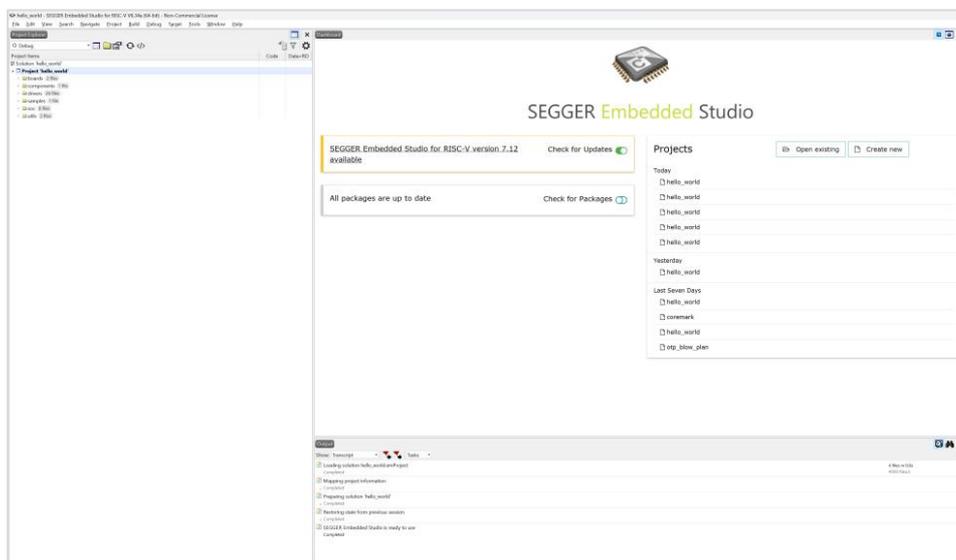


图 22: hello_world工程

3.5 调试出错常见原因

1. FT2232驱动没有正确安装

HPM6200EVK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。

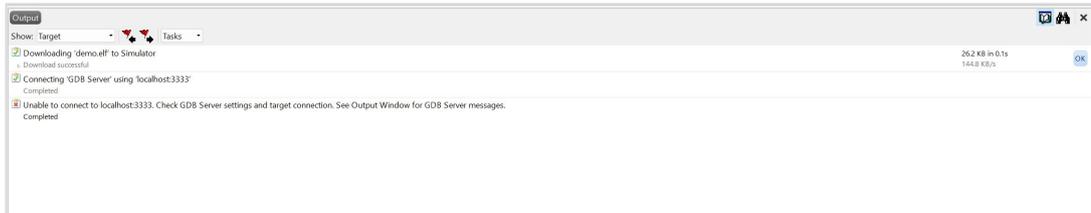


图 23: GDB Server连接失败

（请注意，当HPM6200EVK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

HPM6200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boot loader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跳线帽没有正确配置

为了方便用户调试，HPM6200EVK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。

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 24所示，在GDB Server Command Line中查看openocd配置文件。

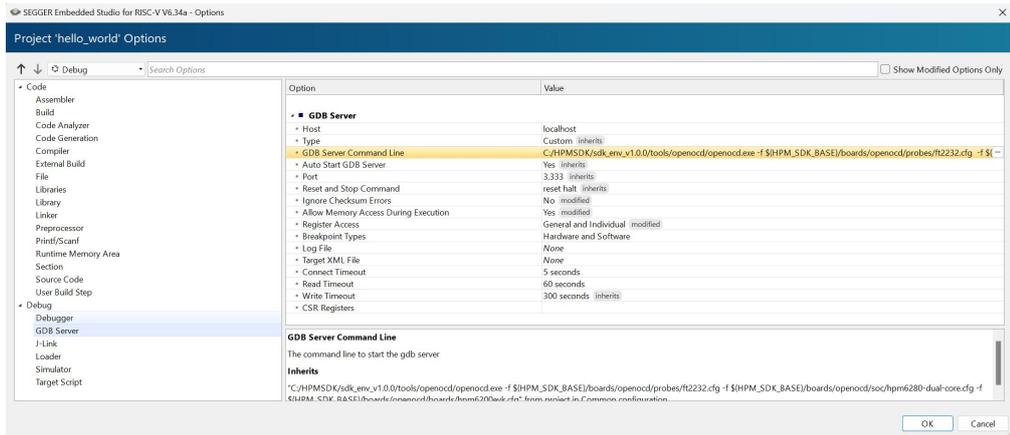


图 24: 查看openocd配置

SDK默认配置如图 25所示，默认使用ft232l调试器。

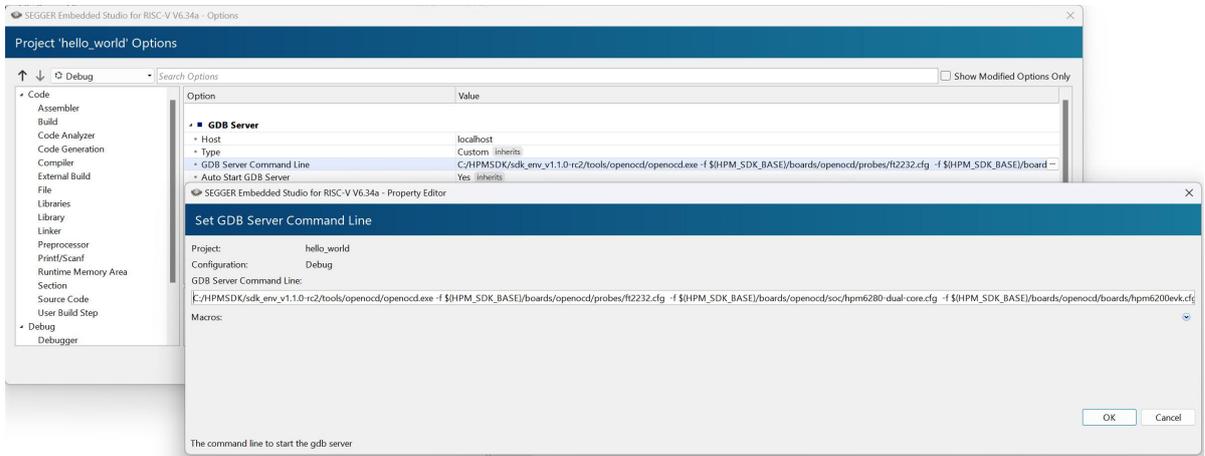


图 25: GDB Server默认配置

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，要更改此配置文件为如图 26所示。

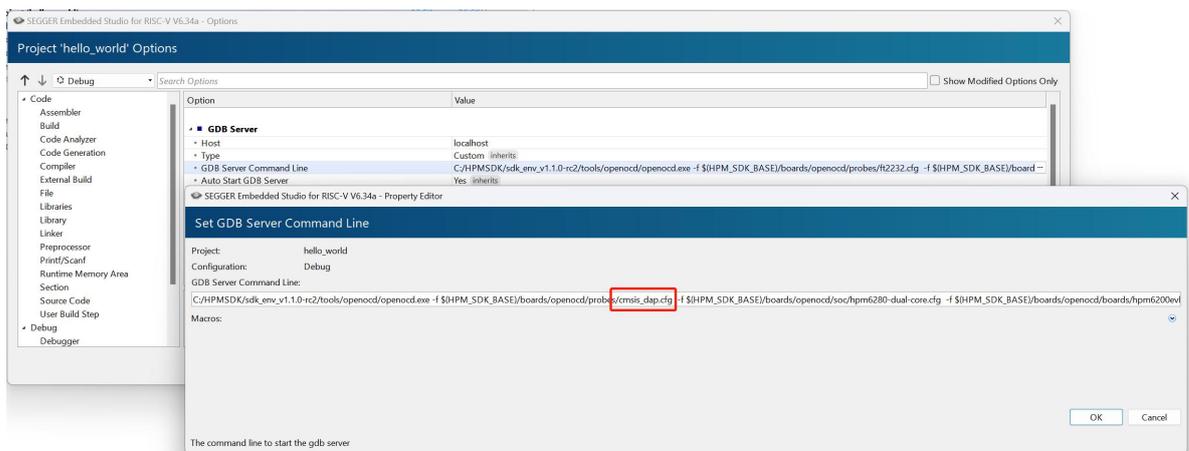


图 26: 使用cmsis-dap 调试器GDB Server配置

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。

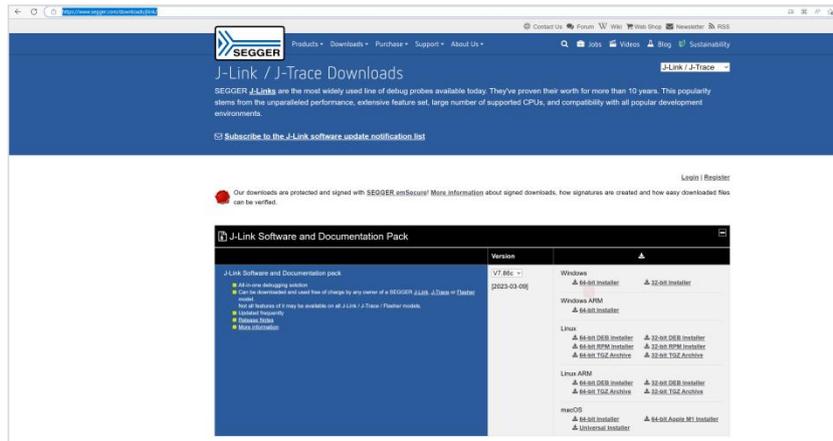


图 27: J-Link驱动下载

下载完成后安装J-Link驱动。

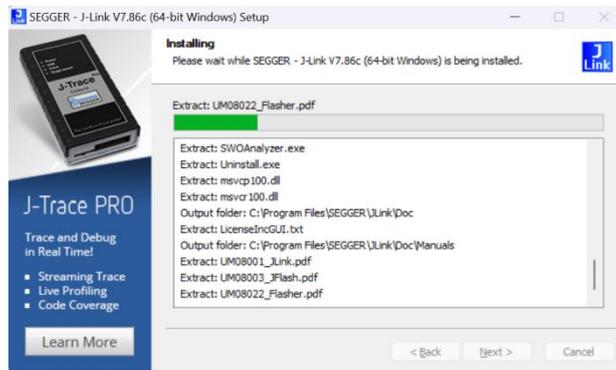


图 28: J-Link驱动安装

安装驱动完成后，正确连接J-Link JTAG接口到HPM6200EVK CN1 20pin 牛角插座，同时拔掉 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。

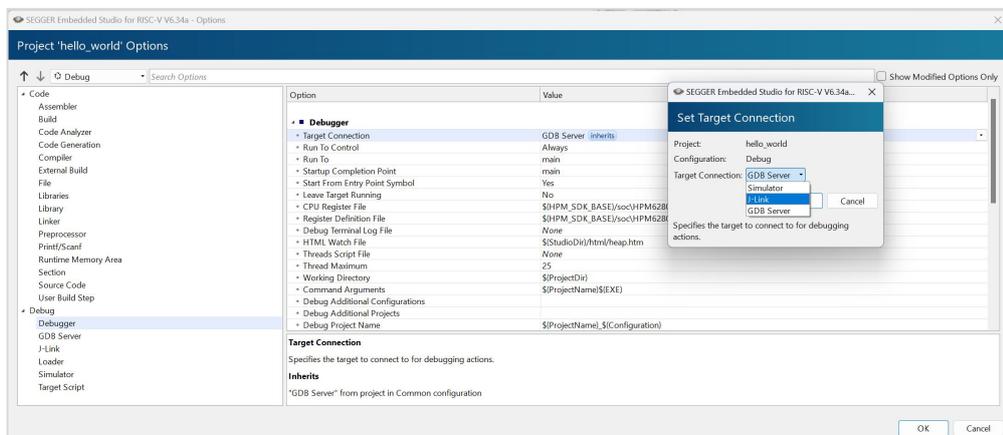


图 29: Target Connection 设置为J-Link

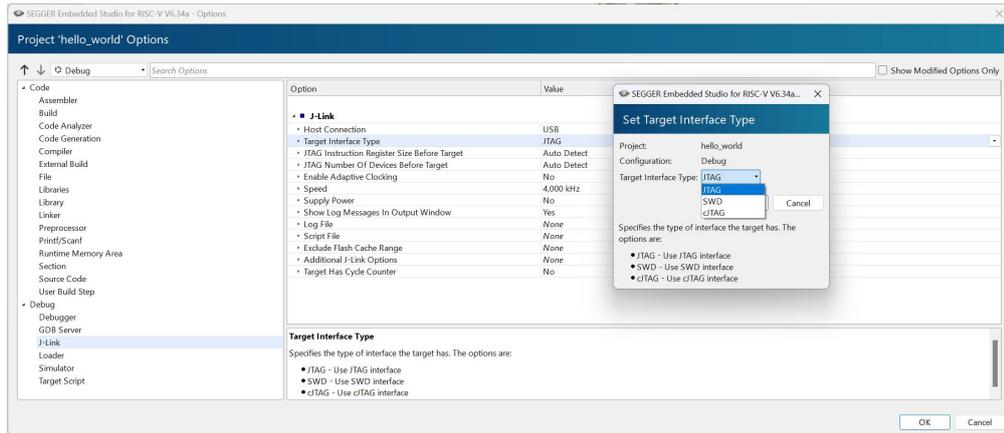


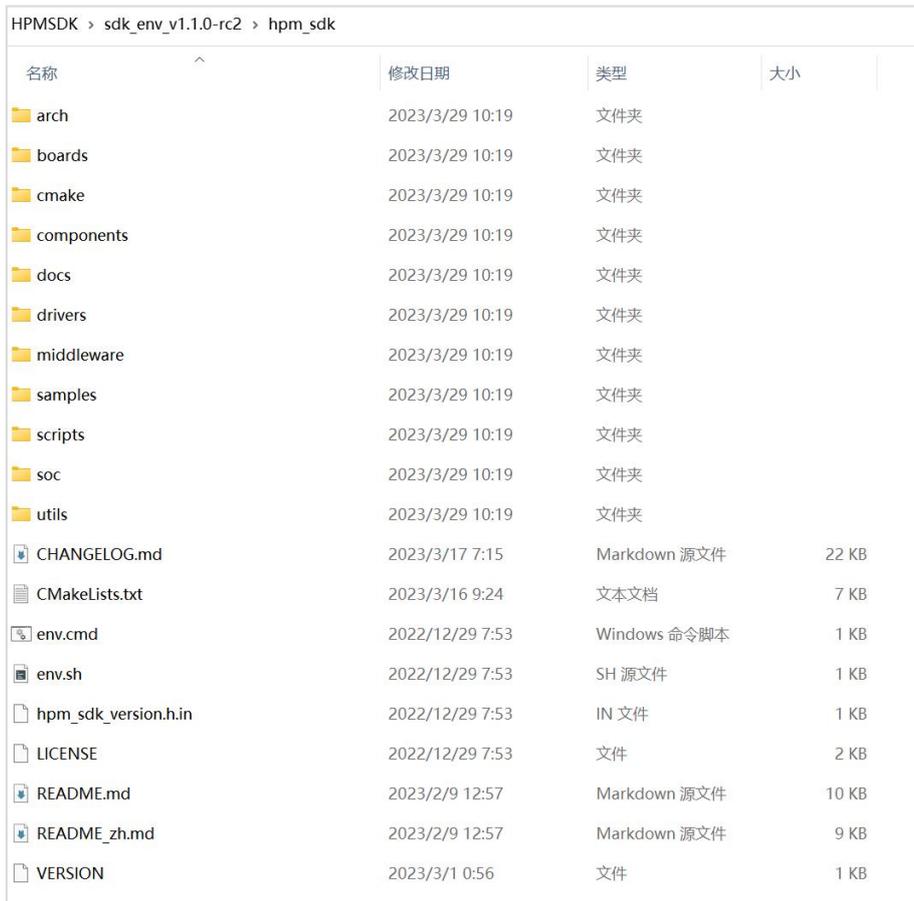
图 30: Target Interface Type设置为JTAG

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toolchain。

3.6.1 更新 sdk_env 中的 SDK

1. 下载hpm_sdk.zip后解压缩。
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。



| 名称 | 修改日期 | 类型 | 大小 |
|----------------------|-----------------|--------------|-------|
| arch | 2023/3/29 10:19 | 文件夹 | |
| boards | 2023/3/29 10:19 | 文件夹 | |
| cmake | 2023/3/29 10:19 | 文件夹 | |
| components | 2023/3/29 10:19 | 文件夹 | |
| docs | 2023/3/29 10:19 | 文件夹 | |
| drivers | 2023/3/29 10:19 | 文件夹 | |
| middleware | 2023/3/29 10:19 | 文件夹 | |
| samples | 2023/3/29 10:19 | 文件夹 | |
| scripts | 2023/3/29 10:19 | 文件夹 | |
| soc | 2023/3/29 10:19 | 文件夹 | |
| utils | 2023/3/29 10:19 | 文件夹 | |
| CHANGELOG.md | 2023/3/17 7:15 | Markdown 源文件 | 22 KB |
| CMakeLists.txt | 2023/3/16 9:24 | 文本文档 | 7 KB |
| env.cmd | 2022/12/29 7:53 | Windows 命令脚本 | 1 KB |
| env.sh | 2022/12/29 7:53 | SH 源文件 | 1 KB |
| hpm_sdk_version.h.in | 2022/12/29 7:53 | IN 文件 | 1 KB |
| LICENSE | 2022/12/29 7:53 | 文件 | 2 KB |
| README.md | 2023/2/9 12:57 | Markdown 源文件 | 10 KB |
| README_zh.md | 2023/2/9 12:57 | Markdown 源文件 | 9 KB |
| VERSION | 2023/3/1 0:56 | 文件 | 1 KB |

图 31: 更新 SDK

3.6.2 更新 sdk_env 中的 toolchain。

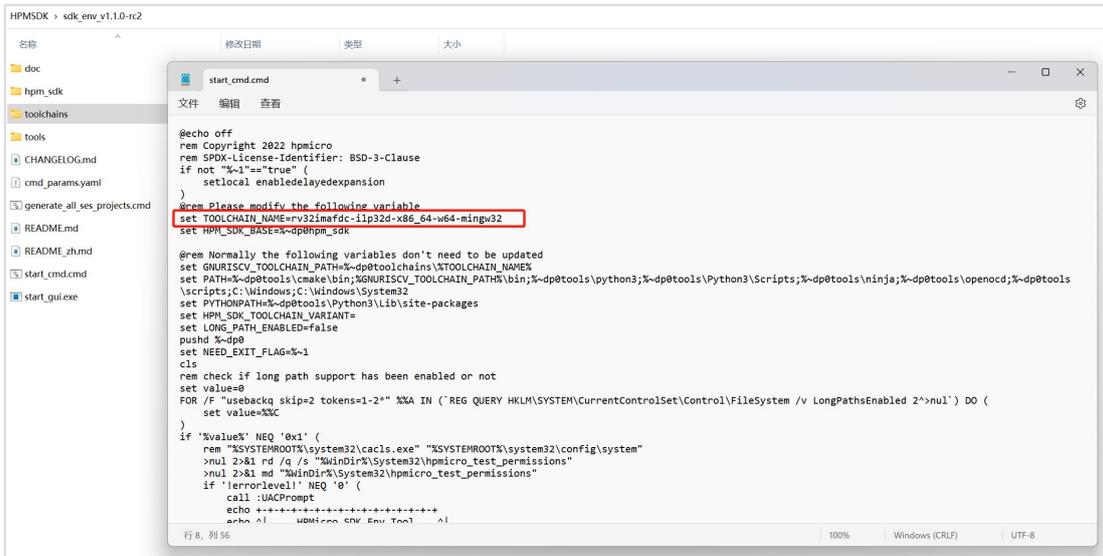
1. 下载 toolchain (以 rv32imafdc-ilp32d-x86_64-w64-mingw32.zip 为例)
2. 将解压后的 toolchain 放至 sdk_env\toolchains\ 目录下，确保可以在 sdk_env\toolchains\rv32imafdc-ilp32d-x86_64-w64-mingw32 目录中可以找到 bin 文件夹。



| 名称 | 修改日期 | 类型 | 大小 |
|--------------------------------------|-----------------|--------------|------|
| rv32imac-ilp32-multilib-win | 2023/3/29 10:20 | 文件夹 | |
| rv32imafdc-ilp32d-x86_64-w64-mingw32 | 2023/3/29 11:11 | 文件夹 | |
| README.md | 2022/12/29 7:52 | Markdown 源文件 | 1 KB |

图 32: 拷贝 toolchain

3. 编辑 start_cmd.cmd, 更新环境变量 TOOLCHAIN_NAME。



```
@echo off
rem Copyright 2022 hpmicro
rem SPDX-License-Identifier: BSD-3-Clause
if not "%1"=="true" (
    setlocal enabledelayedexpansion
)
@rem Please modify the following variable
set TOOLCHAIN_NAME=rv32imafdc-ilp32d-x86_64-w64-mingw32
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
set value=0
FOR /F "usebackq skip=2 tokens=1-2*" %A IN ("REG QUERY HKLM\SYSTEM\CurrentControlSet\Control\FileSystem /v LongPathsEnabled 2>>nul") DO (
    set value=%A
)
if '%value%' NEQ '0x1' (
    rem "%SYSTEMROOT%\system32\cacls.exe" "%SYSTEMROOT%\system32\config\system"
    >nul 2>&1 rd /q /s "%windir%\system32\hpmicro_test_permissions"
    >nul 2>&1 md "%windir%\system32\hpmicro_test_permissions"
    if '!errorlevel!' NEQ '0' (
        call :UACPrompt
        echo ++++++
        echo |>
    )
)
echo |>
```

图 33: 更新 start_cmd.cmd 中 TOOLCHAIN_NAME

4. 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_v1.1.0\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。

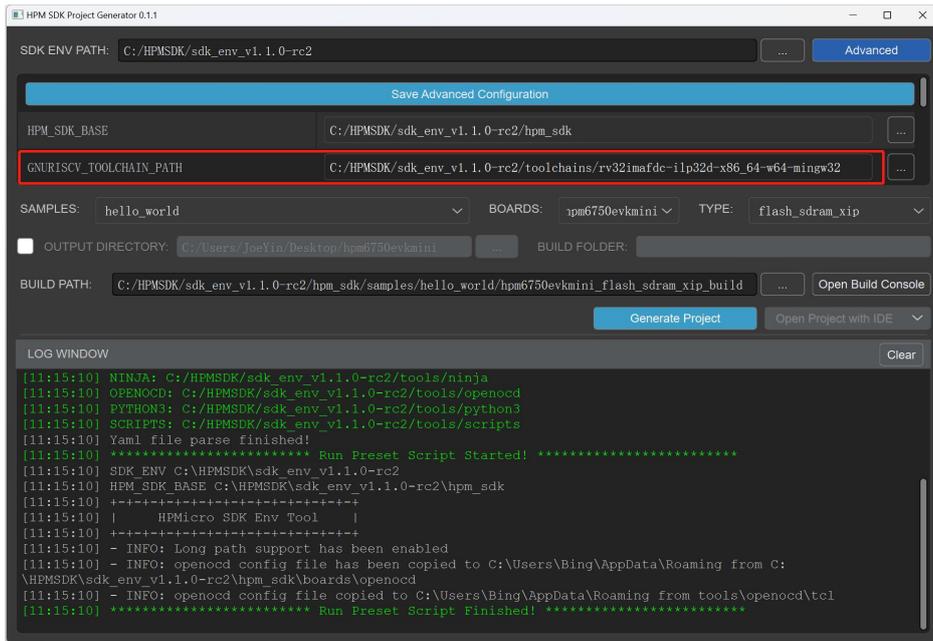


图 34: 更新 start_gui.exe 中 TOOLCHAIN_NAME

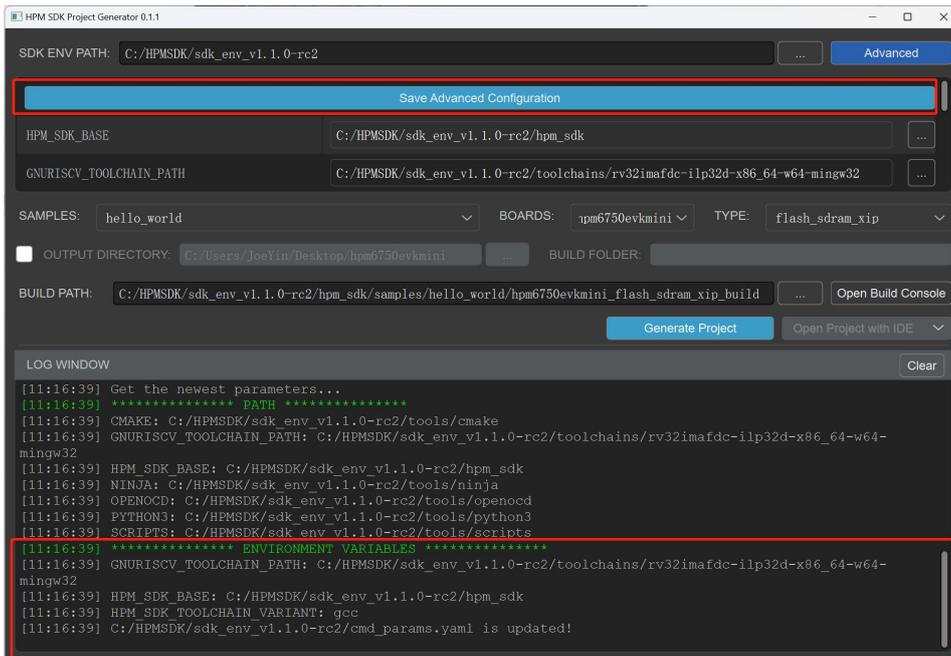


图 35: start_gui.exe更新TOOLCHAIN完成

3.7 版本信息

| 日期 | 版本 | 描述 |
|--------|------------|-------|
| Rev1.0 | 2023/03/29 | 初版发布。 |

表 6: 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.hpimicro.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。