

HPM6800

HPM6800EVK 用户使用手册

适用于上海先楫半导体 HPM6800 系列高性能微控制器

目录

表格目录.....	3
第一章 HPM6800EVK 简介.....	5
第二章 硬件电路.....	7
2.1 电路模块介绍.....	7
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
3.6 更新 sdk_env 中的 SDK/toolchain 指南.....	28
3.6.1 更新 sdk_env 中的 SDK.....	28
3.6.2 更新 sdk_env 中的 toolchain.....	28
3.7 版本信息.....	31
第四章 免责声明.....	32

表格目录

表 1：主要器件位号对应器件功能名称.....6

表 2：启动模式..... 11

表 3：P2连接器列表..... 11

表 4：版本信息..... 31

图片目录

图 1: 顶层器件位置图.....	5
图 2: 底层器件位置图.....	6
图 3: HPM6800EVK 硬件设计框图.....	7
图 4: 安装 FTDI 驱动.....	14
图 5: 查看 Windows 设备管理器.....	15
图 6: sdk_env创建工程方式.....	15
图 7: 打开 sdk prompt.....	16
图 8: 构建目标板工程.....	16
图 9: 构建目标板 flash_xip 工程.....	17
图 10: generate_project 帮助.....	18
图 11: Segger Embedded Studio hello_world 工程.....	18
图 12: Segger Embedded Studio 打开 hello_world 工程.....	18
图 13: Segger Embedded Studio 编译 hello_world 工程.....	19
图 14: Segger Embedded Studio 调试 hello_world 工程.....	19
图 15: Segger Embedded Studio 配置串口.....	20
图 16: Segger Embedded Studio 连接串口.....	20
图 17: Segger Embedded Studio 打开串口.....	21
图 18: Segger Embedded Studio 运行 hello_world.....	21
图 19: start_gui 工具.....	22
图 20: GUI project generator工具操作界面.....	22
图 21: GUI project generator生成hello_world工程.....	23
图 22: hello_world工程.....	23
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.....	25
图 26: 使用cmsis-dap 调试器GDB Server配置.....	25
图 27: J-Link驱动下载.....	26
图 28: J-Link驱动安装.....	26
图 29: Target Connection 设置为J-Link.....	26
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: start_gui.exe更新TOOLCHAIN完成.....	30

第一章 HPM6800EVK 简介

HPM6800EVK 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。

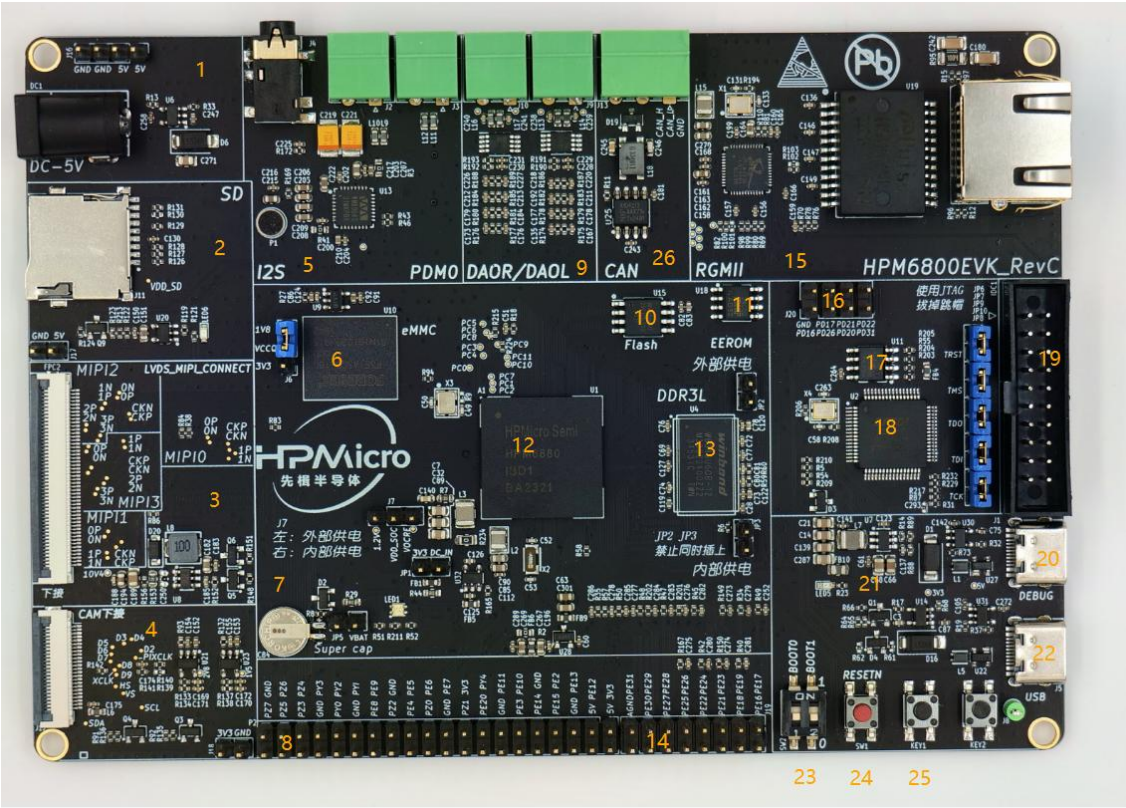


图 1：顶层器件位置图

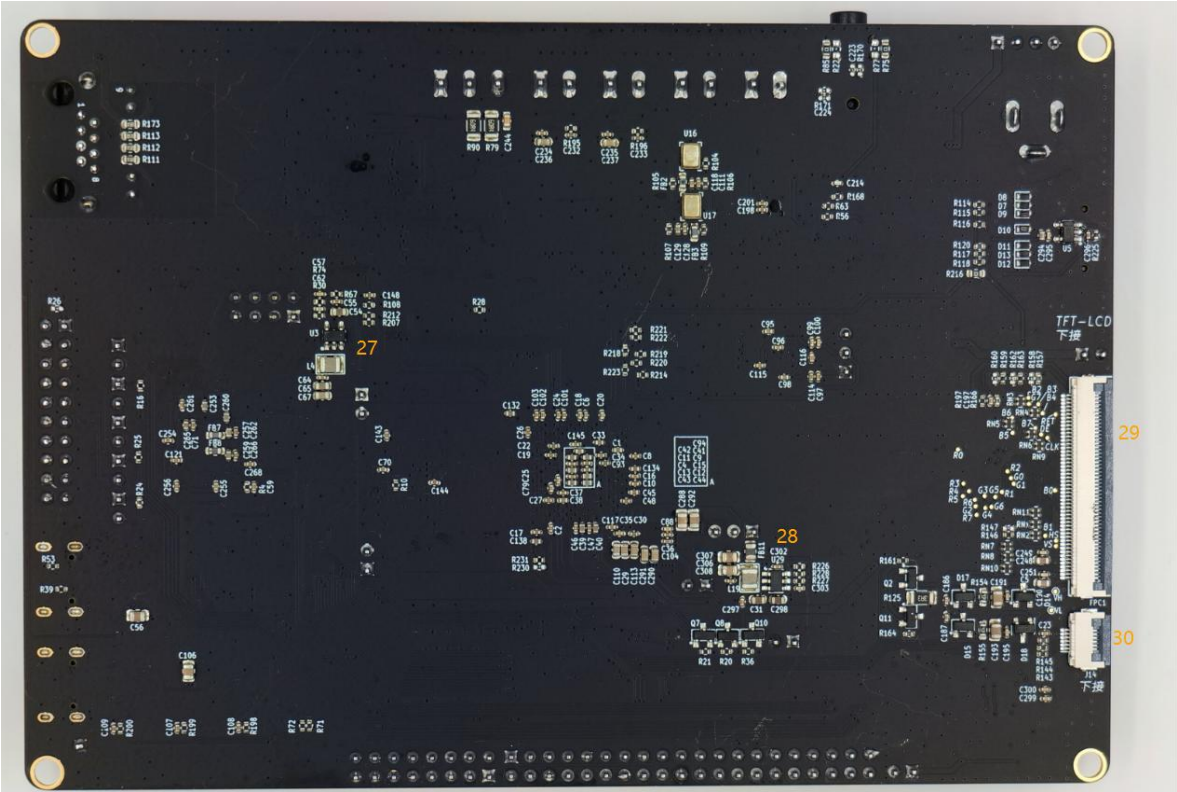


图 2：底层器件位置图

序号	名称	序号	名称
1	DC-5V电源插口	2	T-Flash
3	LVDS/MIPI 接口	4	DVP CAM 接口
5	CODEC + PDM	6	eMMC
7	超级电容	8	扩展排针P1
9	DAOR/DAOL	10	FLASH
11	EEROM	12	HPM6800
13	DDR3L	14	ADC 接口
15	RGMII PHY	16	I/O 引出
17	FT2232HL配置EEPROM	18	FT2232
19	JTAG DEBUG接口	20	FT2232 DEBUG 接口
21	3.3主电源	22	USB 接口
23	BOOT 拨码开关	24	RESET 按键
25	用户按键	26	CAN 芯片
27	DDR3 外部供电芯片	28	HPM6800 1.2V 外部供电
29	RGB LCD接口	30	DVP CAM TOUCH接口

表 1：主要器件位号对应器件功能名称

第二章 硬件电路

HPM6800EVK 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

HPM6800EVK 系统架构如图 3。

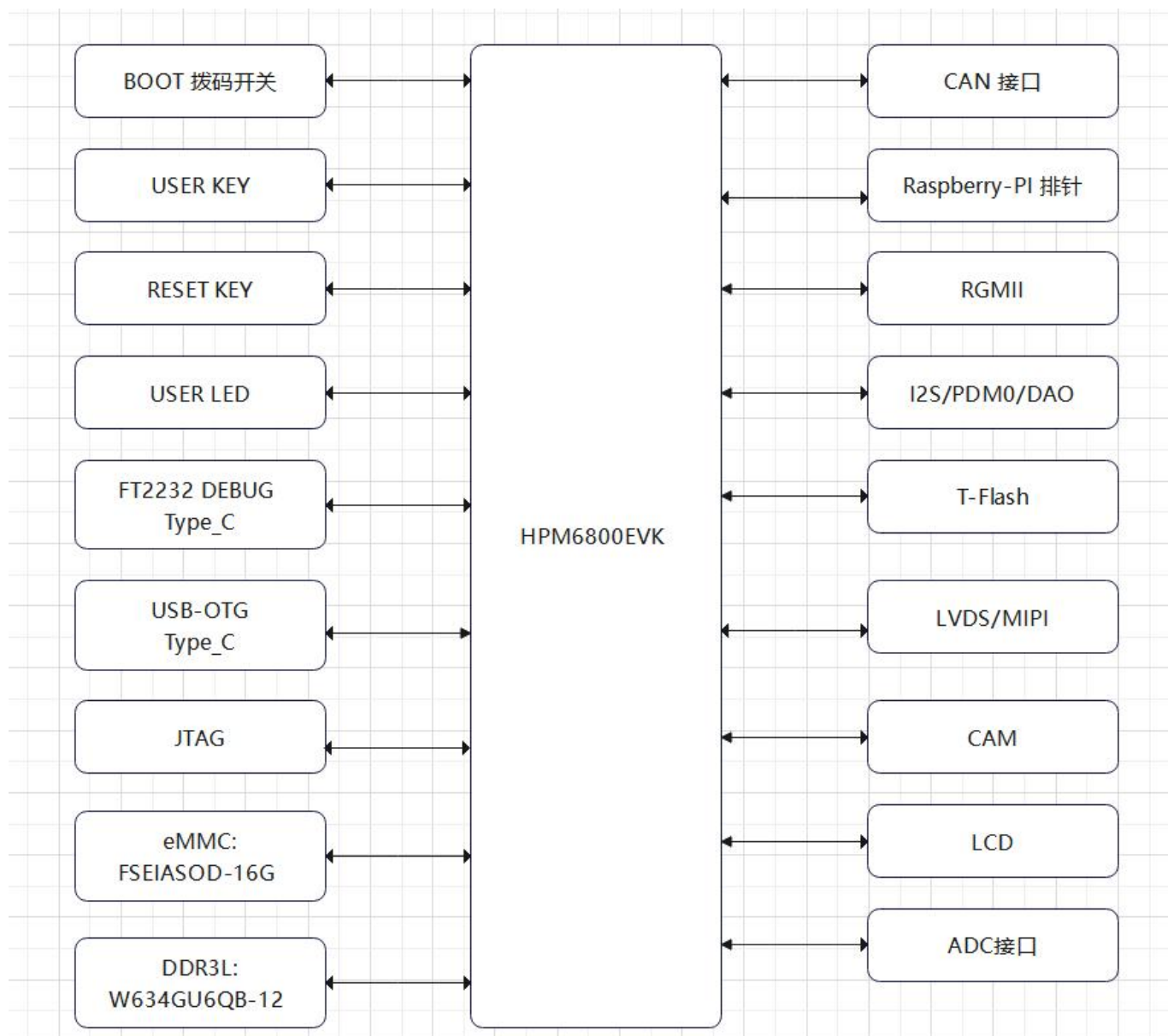


图 3：HPM6800EVK 硬件设计框图

HPM6800EVK

2.1.2 电源

HPM6800EVK 供电方式如下

- 外接DC5V输入 备用电源接口，USB电源供电不足时候可以使用该接口。
- 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
- HPM6800 内核电源1.2V供电，该EVK 可以选择使用内部和外部供电两种，使用板子前先插上J7调帽。

注：用户在使用大功率器件时，例如LCD屏，喇叭等。需要注意USB的带载能力。

2.1.3 USER LED

HPM6800EVK 板载功能型LED如下

LED1 三色RGB

LED5 电源指示

2.1.4 USB 接口

J5 是 HPM6800EVK 板上的 USB0 接口，连接器类型是 Type-C。支持 USB 2.0 OTG。同时支持USB串行启动和ISP，即通过USB给芯片下载bin文件，下载工具通过官网获取。

2.1.5 DEBUG接口

HPM6800EVK 提供两种DEBUG接口，默认是连接J1 DEBUG 接口到PC，通过 U2 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U15 FT2232HL切换到JTAG直连。

2.1.6 TF接口

HPM6800EVK 板载T-Flash接口。

硬件设计上将SD卡电源与信号电源分别供电，IO总线支持3.3V/1.8V切换，支持高速T卡。

2.1.7 按键

HPM6800EVK 板载3个按键，其中红色按键是RESET。其余2个是用户自定义按键。

2.1.8 网口

HPM6800EVK 板载千兆网口，网口芯片为RTL8211E-VB-CG。

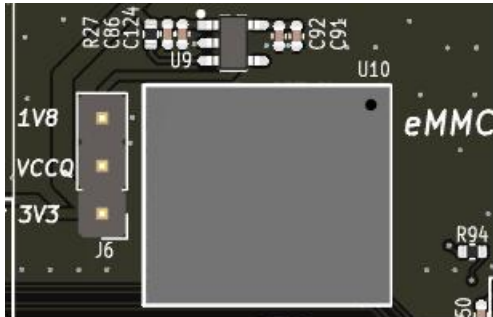
HPM6800EVK

2.1.9 数据存储

HPM6800EVK 板载单颗QSPI接口 NOR FLASH, 容量 128Mb, 型号W25Q128JVS1Q

HPM6800EVK 板载单颗EEPROM, 容量 2Kb, 型号24C02。

HPM6800EVK 板载单颗eMMC, 型号FSEIAS0D-16G, 支持eMMC 5.1 高速模式, 使用前先插上eMMC供电插针J6, 可以选择1.8V和3.3V, VCCQ电源同时给HPM6800 EMMC信号电源bank供电, 以保证信号电平一致。



注：受到市场供应链的影响，HPM6800EVK上的存储器件，可能发生变化。先楫半体会确保所使用的器件，性能不会受到影响。

2.1.10 DDR3L

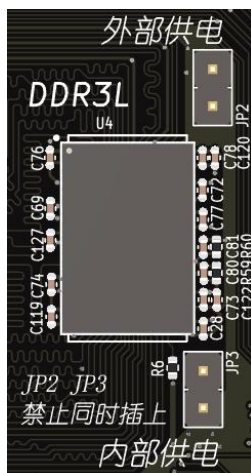
HPM6800EVK 板载1颗16bit DDR3L, 容量 4Gmb, 型号 W634GU6QB-12。

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

如图所示：JP2 是DDR外部供电，默认板子上电后输出1.4V电压，插上该插座即可给DDR3L供电。

JP3是6800内部DDR输出电源，DDR3L颗粒供电可以选择由6880芯片供电和外部供电两种，通过软件设置内部供电是否打开，及输出电压大小。

注：两者不要同时供电，即JP2和JP3不要同时上电，会造成电源短路。



注：受到市场供应链的影响，HPM6800EVK上的存储器件，可能发生变化。先楫半体会确保所使用的器件，性能不会受到影响。

2.1.11 音频接口

HPM6800EVK 支持音频输入输出

- Audio Codec (WM8960)
 - 3.5mm 耳麦 (默认美标, 可以修改电阻该为国标)
 - 左, 右声道 (1W@8Ω 每声道)
 - Mic
- DA0
 - 2路 3W@4Ω
- Digital Mic
 - 2 * SPH0641LU4H

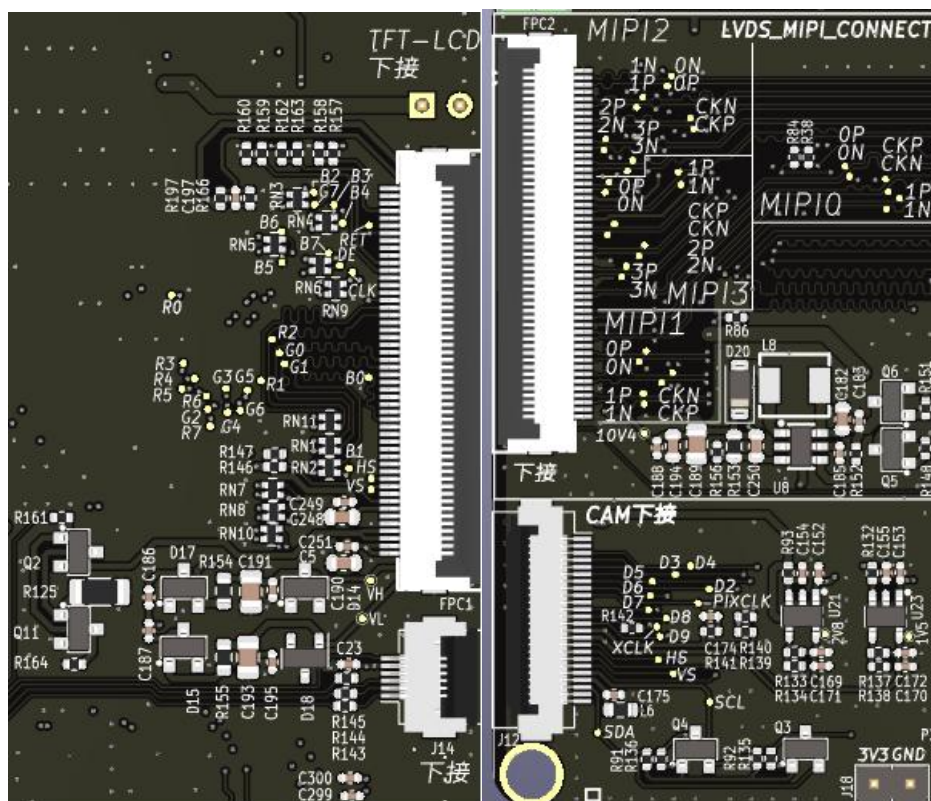
注: 外置喇叭均由5V供电, 因此用户在使用时要确保5V的带载能力

2.1.12 视频接口

HPM6800EVK 支持视频输入输出, 包括DVP接口摄像头、RGB LCD屏, LVDS屏, MIPI屏, MIPI摄像头。

顶层TOP

底层BOTTOM



对上面接口做简要介绍:

- J12摄像头接口, DVP接口 (OV5640), 该接口已经提供CAM相关电源 (1.5v, 1.8v, 2.8v), 翻盖下接
- J14触摸屏接口, 翻盖下接
- FPC1 RGB LCD接口, 该接口放置在板子背面, 接口支持背光板子自带LCD升压电路, 翻盖下接
- FPC2 接口上包含LVDS, MIPI接口, 用户可以根据该接口定义外接自己需要的屏幕和摄像头。翻盖下接。

HPM6800提供了丰富的视频接口, 用户可以根据需求, 设计相应的子板, 进行视频开发。

2.1.13 ADC 接口

J19 是 HPM6800EVK 板载ADC接口，所有6880的ADC引脚均已经连接到该接插件。

2.1.14 B00T 拨码开关设置

芯片默认是通过 拨码开关设置对应 B00T_MODE[1:0]=[PA03:PA02] 引脚选择启动模式，配置如表 2 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	在系统编程 (ISP)/串行启动	从 UART0/USB0 上烧写固件, OTP, 或从UART0/USB0启动
ON	OFF	在系统编程 (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 2 启动模式

2.1.15 扩展 I0 接口

HPM6800EVK 板载一排扩展 I0，即排针 P2。接口机械尺寸与RASPBerry-PI兼容。P2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	
PE12	SDA1	3	4	5. 0V	
PE13	SCL1	5	6	GND	
PE2	GP I07	7	8	TXD	PE15
	GND	9	10	RXD	PE14
PE10	GP I00	11	12	GP I01	PE3
PE11	GP I02	13	14	GND	
PY4	GP I03	15	16	GP I04	PE20
	3. 3V	17	18	GP I05	PZ1
PE7	MOSI	19	20	GND	
PE6	MISO	21	22	GP I06	PZ0
PE5	SCLK	23	24	CE0	PE4
	GND	25	26	CE1	PZ2
PE9	SDA0	27	28	SCL0	PE8
PY1	GP I021	29	30	GND	
PY2	GP I022	31	32	GP I026	PY0
PY3	GP I023	33	34	GND	
PZ4	GP I024	35	36	GP I027	PZ3
PZ6	GP I025	37	38	GP I028	PZ5
	GND	39	40	GP I029	PZ7

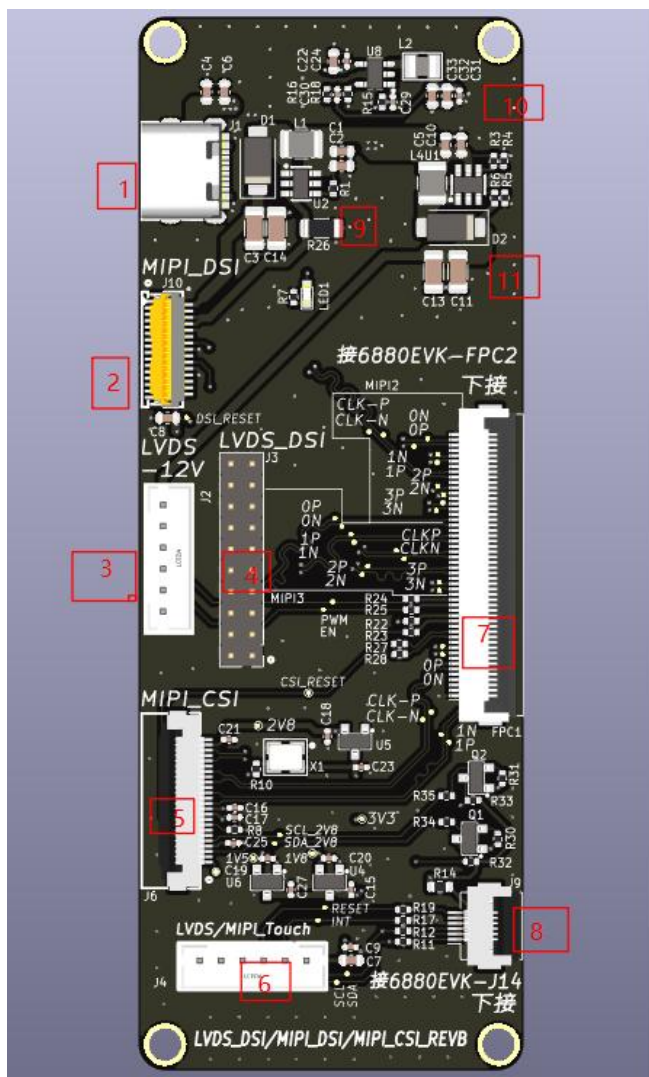
表 3: P2 连接器列表

注：P2机械尺寸和管脚功能与RASPBerry-PI兼容，用户如果想驱动RASPBerry-PI扩展板时请先查看扩展板与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。

HPM6800EVK

2.1.16 显示子板

HPM6800EVK开发板套件中同时提供一个显示转接板，上面包含单路LVDS_DSI, MIPI_DSI, MIPI_CSI, 三个接口。可以单独使用屏幕，或者配合摄像头一起使用，同时转接板也提供相应的背光升压电路。由于显示设备的接口各异，用户可以根据需要，设计相应的显示子板。



- 转接板各功能介绍：

1. USB 供电引脚，主要给显示屏背光升压电路供电，如果两个屏幕都打开背光，耗电在1A左右
2. MIPI DSI接口，MIPI屏型号：CC10128007-31C。
3. LVDS 背光插座，12V供电，电源板子提供。
4. LVDS 信号接口，注意插座1脚跟板子1脚对应。翻盖下接，LVDS型号：CC10128007-20A
5. MIPI CSI接口，板子提供Cam需要的电源，翻盖下接MIPI CSI型号：OV5640
6. 触摸屏I2C配置接口
7. LVDS/MIPI与开发板连接接口，连接6800EVK FPC2接口（正面），
8. MIPI CSI I2C配置接口，与HPM6800EVK J14（背面）连接。
9. MIPI 背光升压电路，带载电压12v，空载30V。
10. 3v3 电源

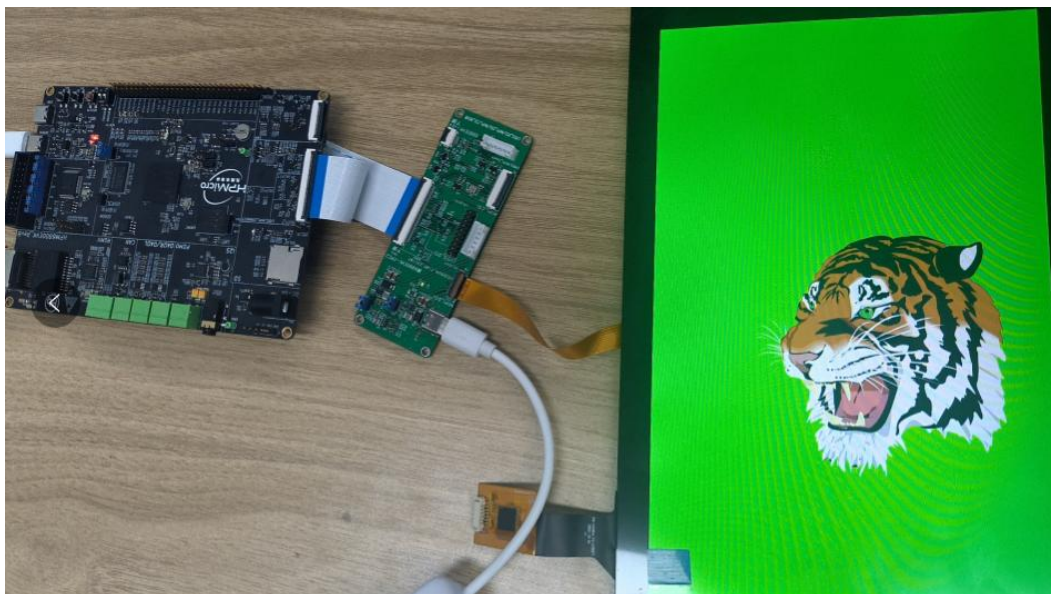
HPM6800EVK

11. LVDS 背光升压电路接。电压12v。

需要的线缆包括：

1. FPC50P, 0.5MM间距, 同向, 15CM
2. FPC6P, 0.5MM间距, 反向, 15CM

转接板MIPI LCD和EVK连接示意图：



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_v1.4.0 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_v1.4.0\tools\FTDI_InstallDriver.exe 以安装可用于调试的 FT2232 驱动。

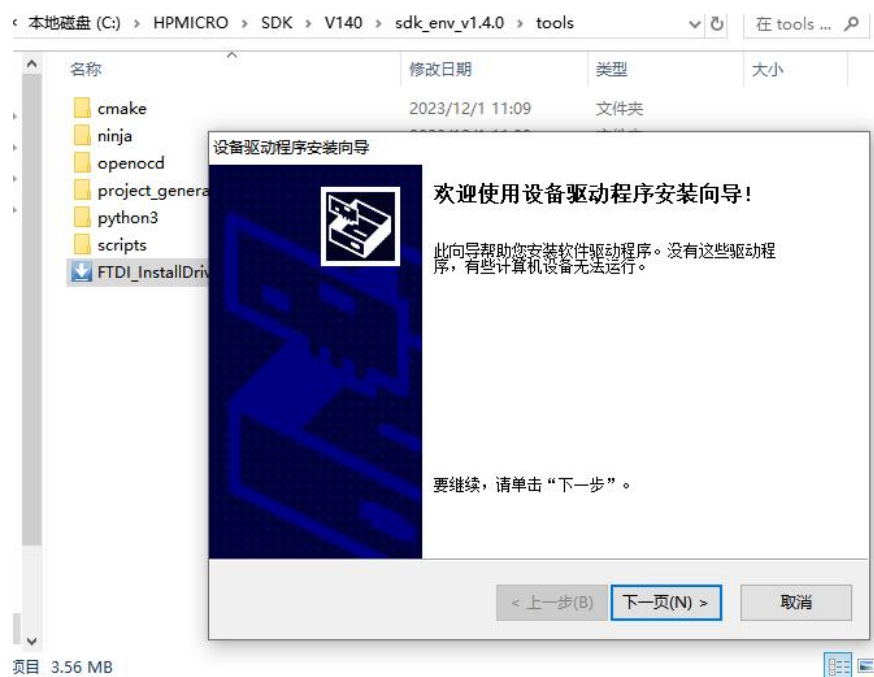


图 4：安装 FTDI 驱动

正确安装驱动后，使用 USB Type-C 线缆将 HPM6800EVK 上的 J1 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：

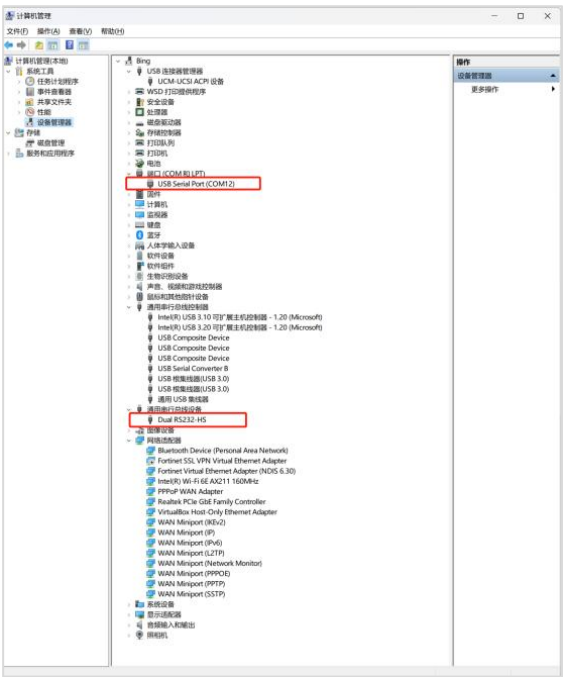


图 5：查看 Windows 设备管理器

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。

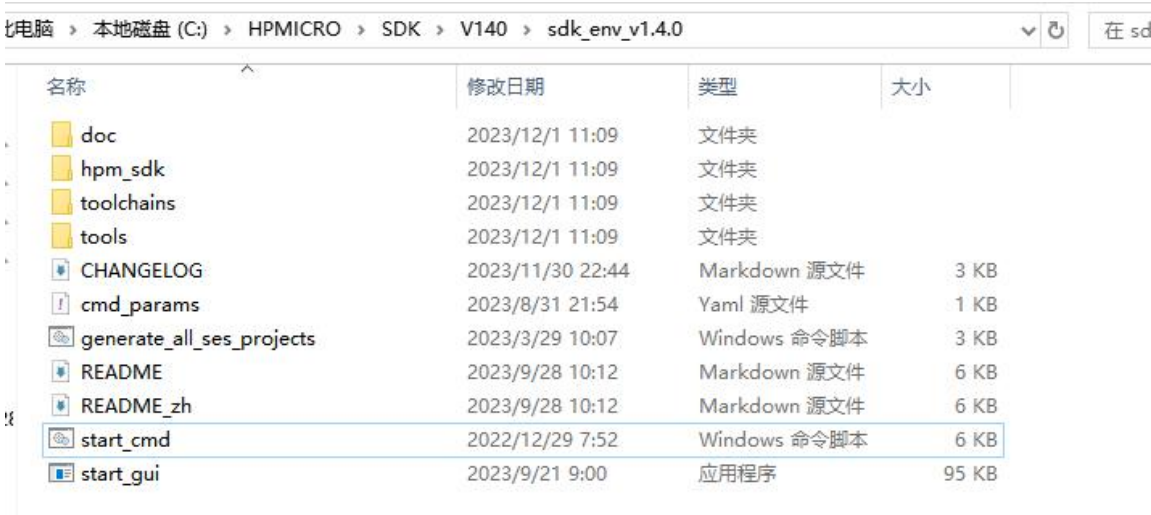


图 6：sdk_env创建工程方式

以命令行工具为例，双击打开 sdk_env_v1.4.0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。

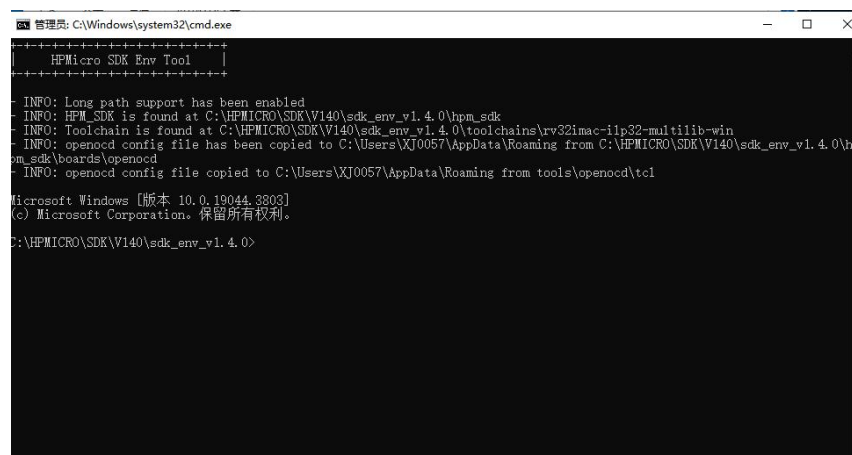


图 7：打开 sdk prompt

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 HPM6800EVK 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 hpm6800evk -f
```

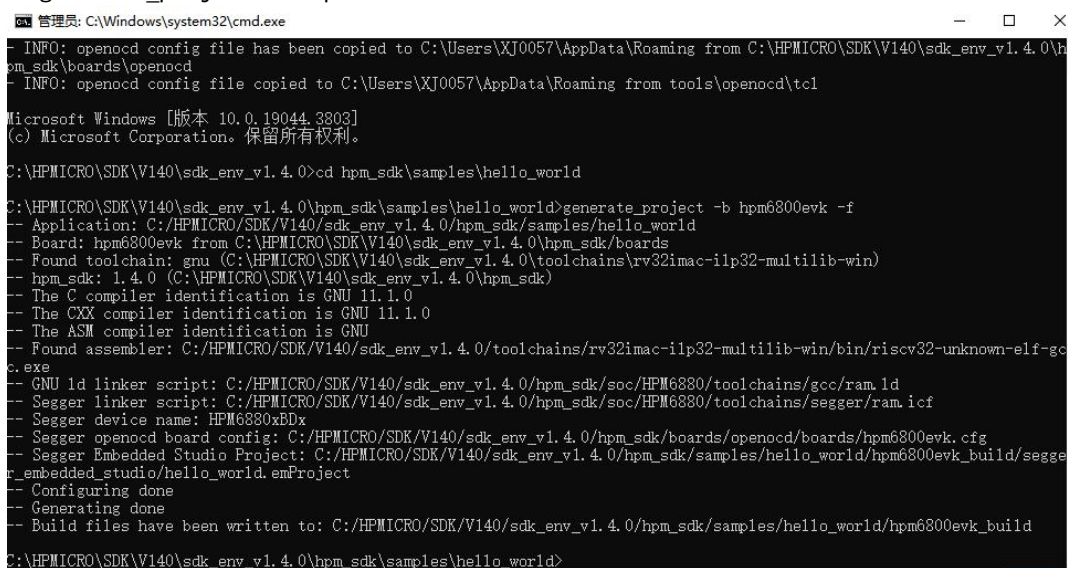


图 8：构建目标板工程

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。

HPM6800EVK

```
管理: C:\Windows\system32\cmd.exe
-- Segger linker script: C:/HPMICRO/SDK/V140/sdk_env_v1.4.0/hpm_sdk/soc/HPM6880/toolchains/segger/ram.icf
-- Segger device name: HPM6880xBDx
-- Segger openocd board config: C:/HPMICRO/SDK/V140/sdk_env_v1.4.0/hpm_sdk/boards/openocd/boards/hpm6800evk.cfg
-- Segger Embedded Studio Project: C:/HPMICRO/SDK/V140/sdk_env_v1.4.0/hpm_sdk/samples/hello_world/hpm6800evk_build/segge
r_embedded_studio/hello_world.emProject
-- Configuring done
-- Generating done
-- Build files have been written to: C:/HPMICRO/SDK/V140/sdk_env_v1.4.0/hpm_sdk/samples/hello_world/hpm6800evk_build

C:\HPMICRO\SDK\V140\sdk_env_v1.4.0\hpm_sdk\samples\hello_world>generate_project -b hpm6800evk -f -t flash_xip
-- Application: C:/HPMICRO/SDK/V140/sdk_env_v1.4.0/hpm_sdk/samples/hello_world
-- Board: hpm6800evk from C:\HPMICRO\SDK\V140\sdk_env_v1.4.0\hpm_sdk\boards
-- Found toolchain: gnu (C:/HPMICRO/SDK/V140/sdk_env_v1.4.0/toolchains/rv32imac-ilp32-multilib-win)
-- hpm_sdk: 1.4.0 (C:/HPMICRO/SDK/V140/sdk_env_v1.4.0/hpm_sdk)
-- 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/V140/sdk_env_v1.4.0/toolchains/rv32imac-ilp32-multilib-win/bin/riscv32-unknown-elf-gc
c.exe
-- GNU ld linker script: C:/HPMICRO/SDK/V140/sdk_env_v1.4.0/hpm_sdk/soc/HPM6880/toolchains/gcc/flash_xip.ld
-- Segger linker script: C:/HPMICRO/SDK/V140/sdk_env_v1.4.0/hpm_sdk/soc/HPM6880/toolchains/segger/flash_xip.icf
-- Segger device name: HPM6880xBDx
-- Segger openocd board config: C:/HPMICRO/SDK/V140/sdk_env_v1.4.0/hpm_sdk/boards/openocd/boards/hpm6800evk.cfg
-- Segger Embedded Studio Project: C:/HPMICRO/SDK/V140/sdk_env_v1.4.0/hpm_sdk/samples/hello_world/hpm6800evk_build/segge
r_embedded_studio/hello_world.emProject
-- Configuring done
-- Generating done
-- Build files have been written to: C:/HPMICRO/SDK/V140/sdk_env_v1.4.0/hpm_sdk/samples/hello_world/hpm6800evk_build

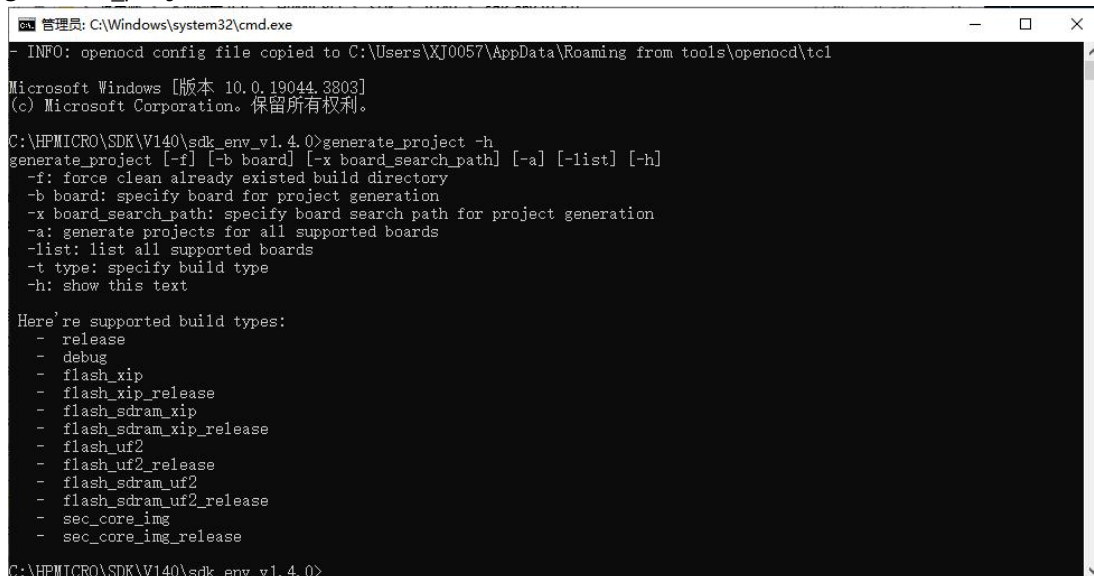
C:\HPMICRO\SDK\V140\sdk_env_v1.4.0\hpm_sdk\samples\hello_world>
```

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 4）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



```

C:\HPMICRO\SDK\V140\sdk_env_v1.4.0>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_sdram_xip
- flash_sdram_xip_release
- flash_uf2
- flash_uf2_release
- flash_sdram_uf2
- flash_sdram_uf2_release
- sec_core_img
- sec_core_img_release

C:\HPMICRO\SDK\V140\sdk_env_v1.4.0>
  
```

图 10: generate_project 帮助

- 当前目录下将生成名为 hpm6800evk_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。



图 11: Segger Embedded Studio hello_world 工程

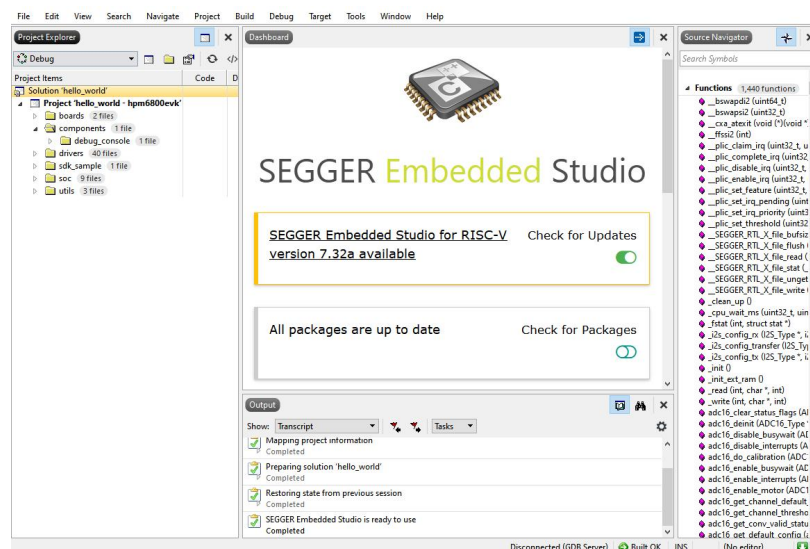


图 12: Segger Embedded Studio 打开 hello_world 工程

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

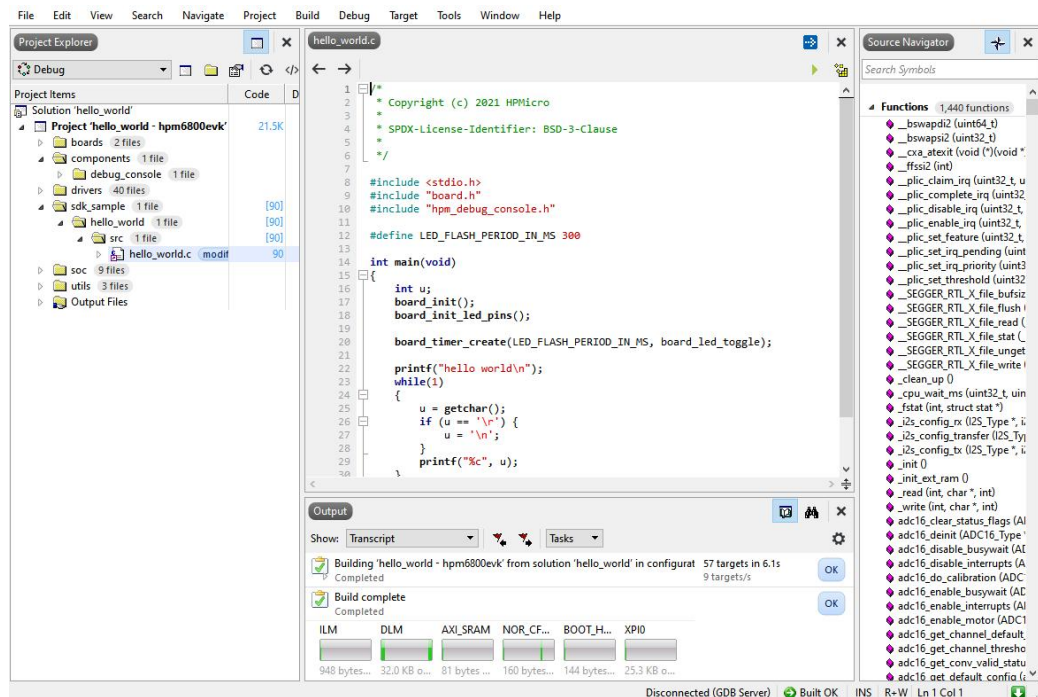


图 13: Segger Embedded Studio 编译 hello_world 工程

10. 使用 Segger Embedded Studio 进行 hello_world 调试。

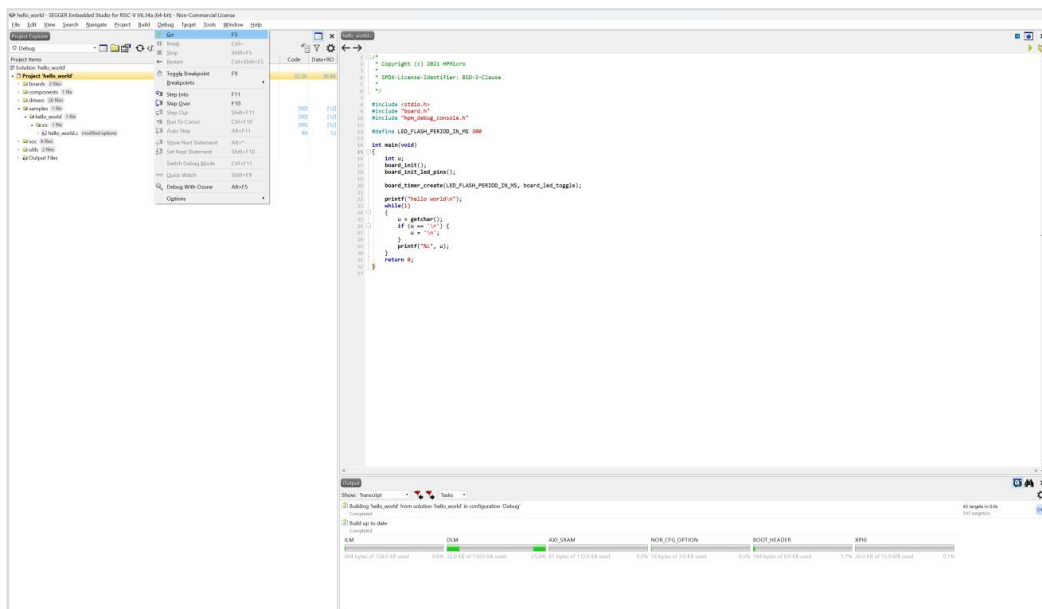


图 14: Segger Embedded Studio 调试 hello_world 工程

11. 在 Segger Embedded Studio 中配置串口。

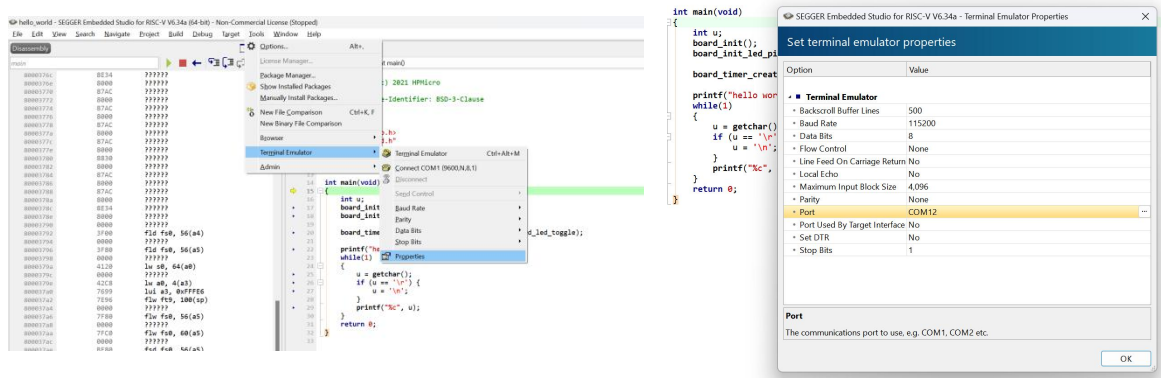


图 15: Segger Embedded Studio 配置串口

12. 在 Segger Embedded Studio 中连接串口。

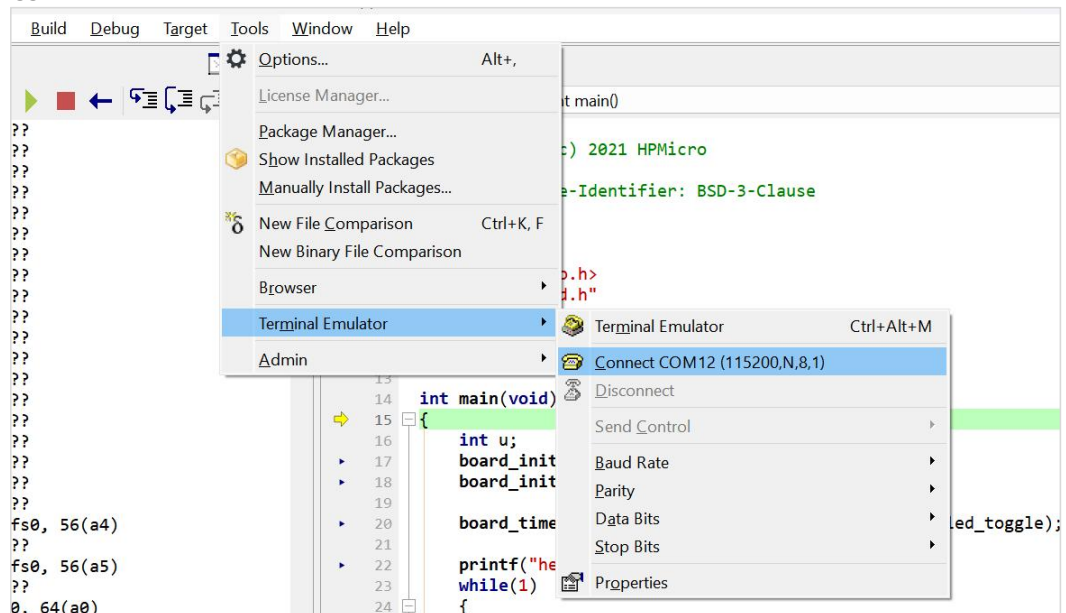


图 16: Segger Embedded Studio 连接串口

13. 在 Segger Embedded Studio 中打开串口。

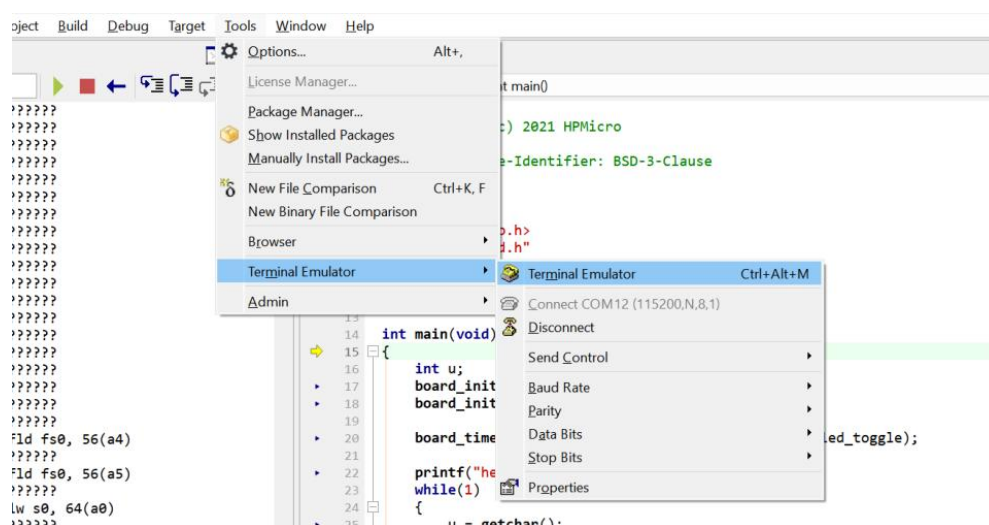


图 17: Segger Embedded Studio 打开串口

14. 运行 hello_world。

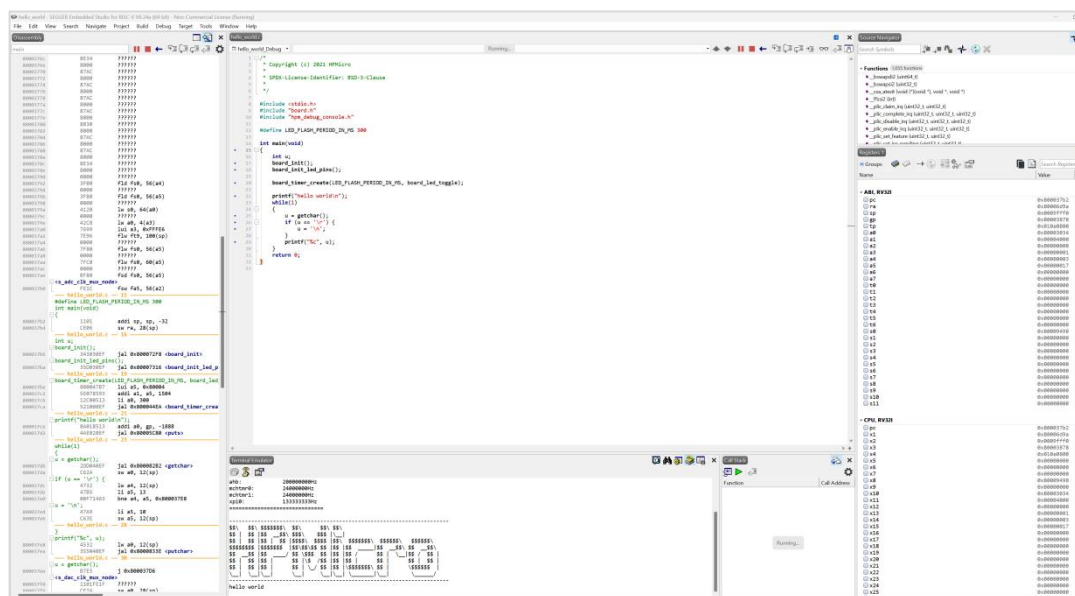


图 18: Segger Embedded Studio 运行 hello_world

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

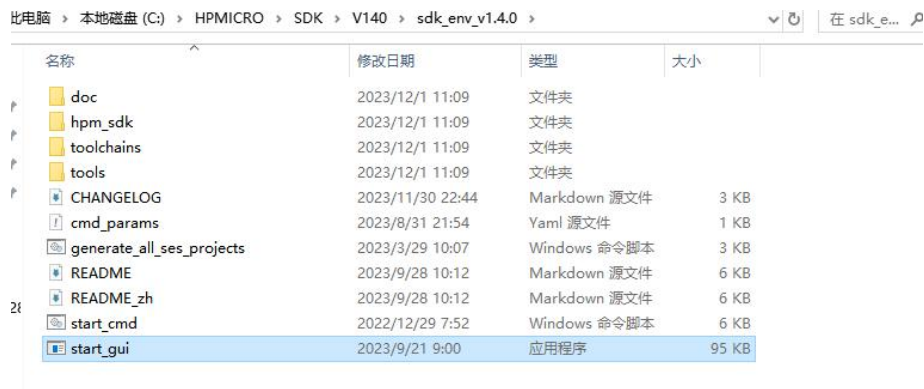


图 19: start_gui 工具

16. 双击打开start_gui.exe。

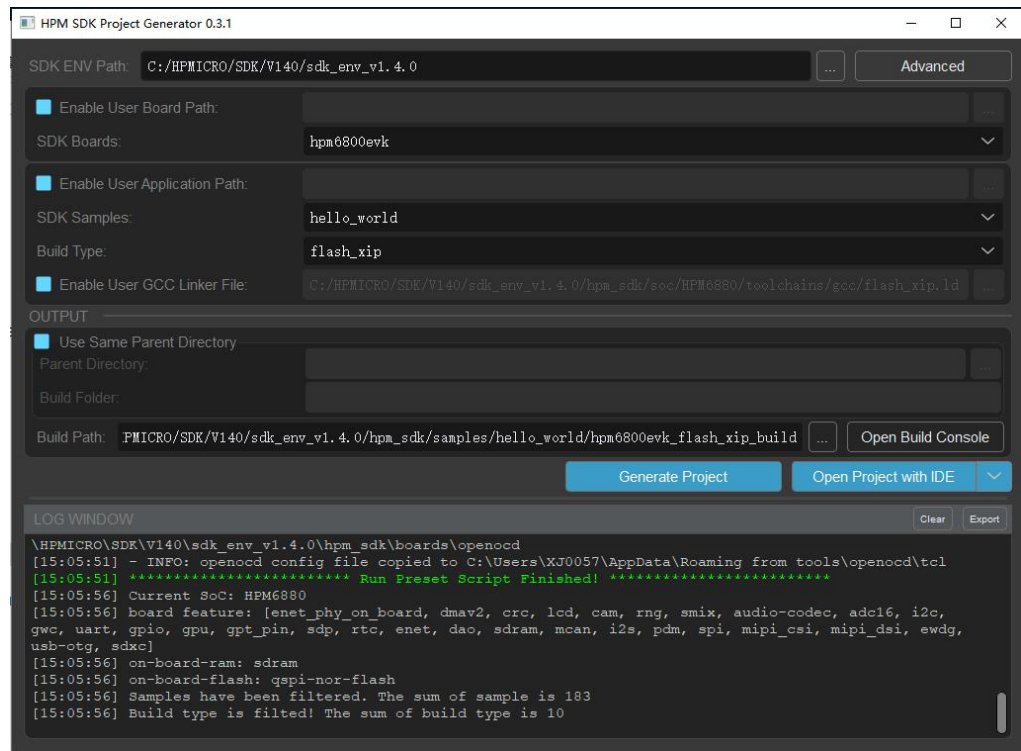


图 20: GUI project generator工具操作界面

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”hpm6800evk”，在”TYPE”下拉列表中选择”flash_xip”。点击”Generate Project”按钮，即可生成flash_xip类型的hello_world工程。如图 21所示。

HPM6800EVK

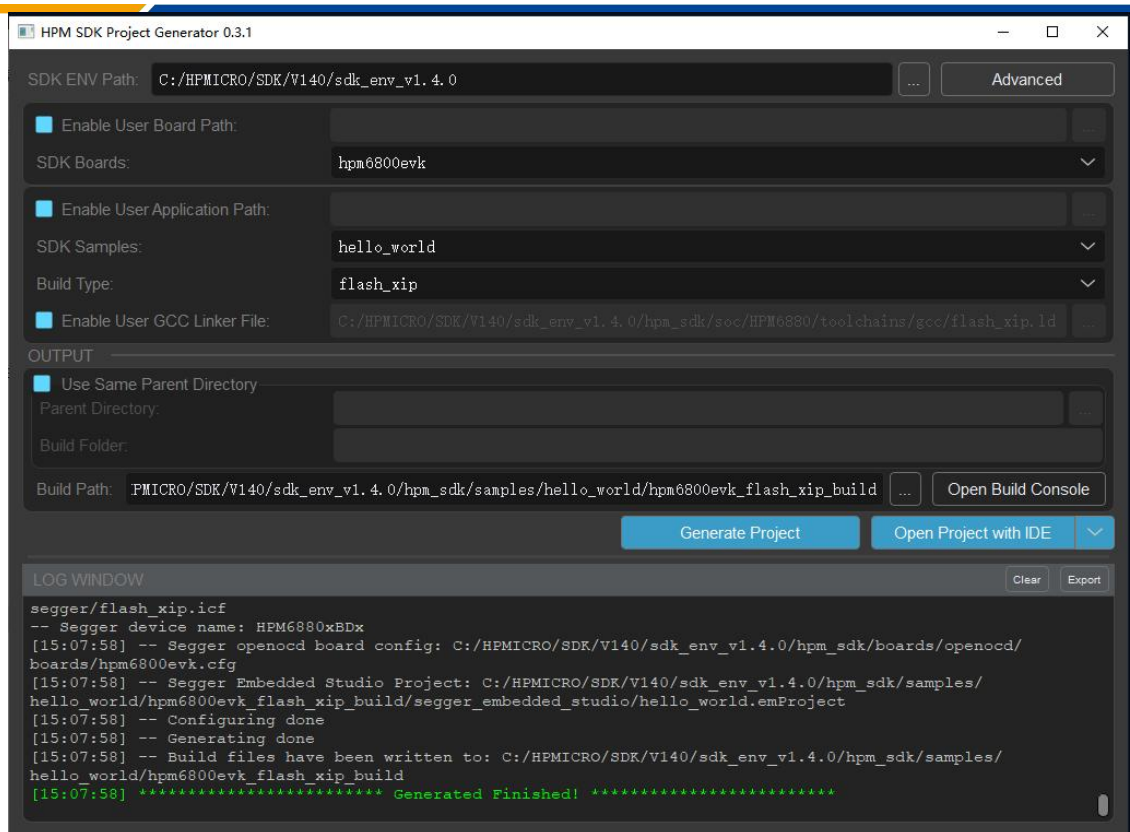


图 21: GUI project generator生成hello_world工程

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

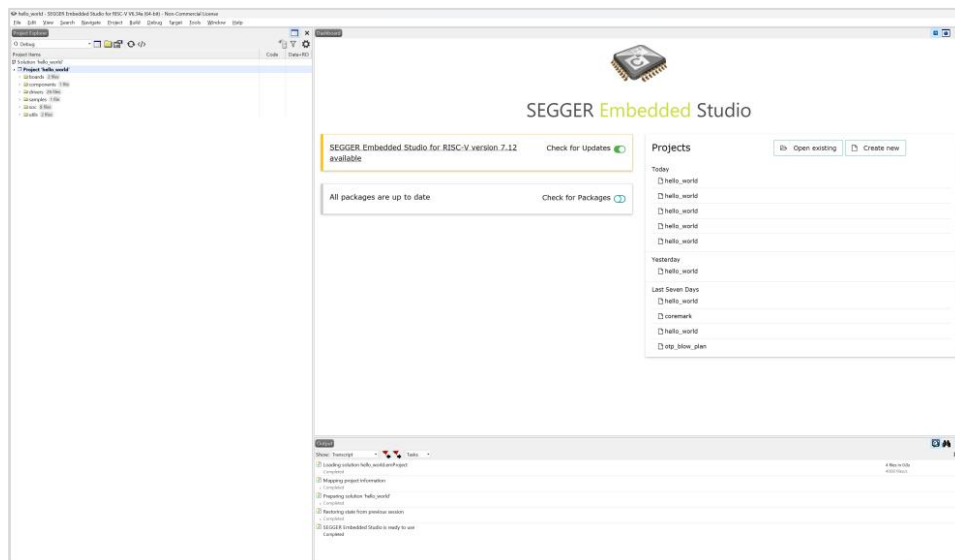


图 22: hello_world工程

3.5 调试出错常见原因

1. FT2232驱动没有正确安装

HPM6800EVK 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 and 串口驱动是否正确：一个USB Serial Port，一个Dual RS232-HS。

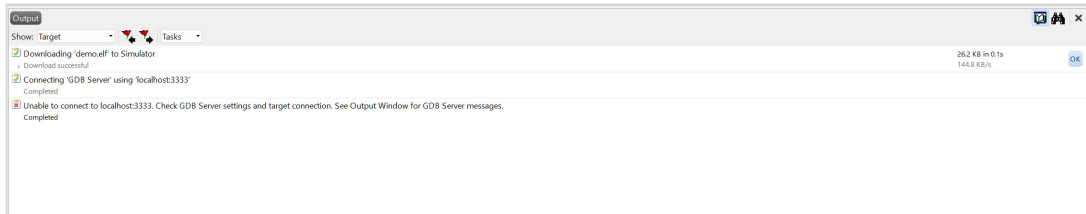


图 23: GDB Server连接失败

（请注意，当HPM6800EVK 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

HPM6800EVK 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：B00T0 = 0，B00T1=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：B00T0 = 0，B00T1=1，将芯片复位或者重新上电，之后再吧boot pin调整到：B00T0 =0，B00T1=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，HPM6800EVK 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。

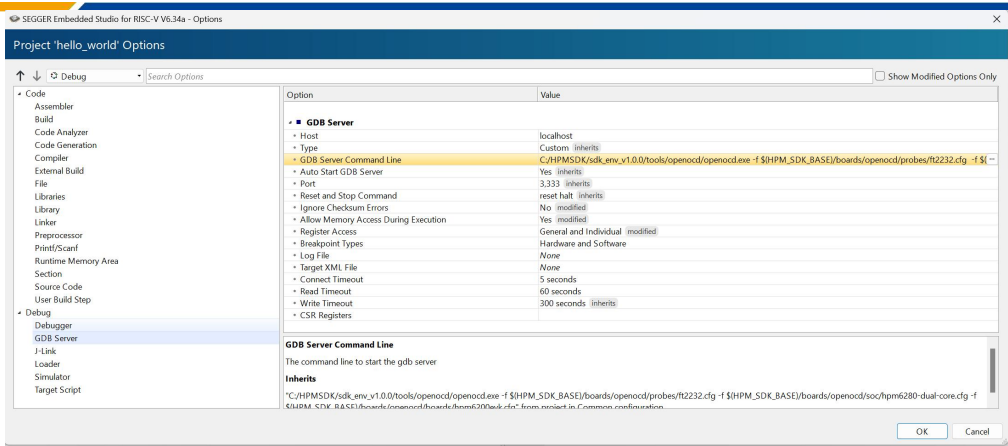


图 24：查看openocd配置

SDK默认配置如图 25所示，默认使用ft2232调试器。

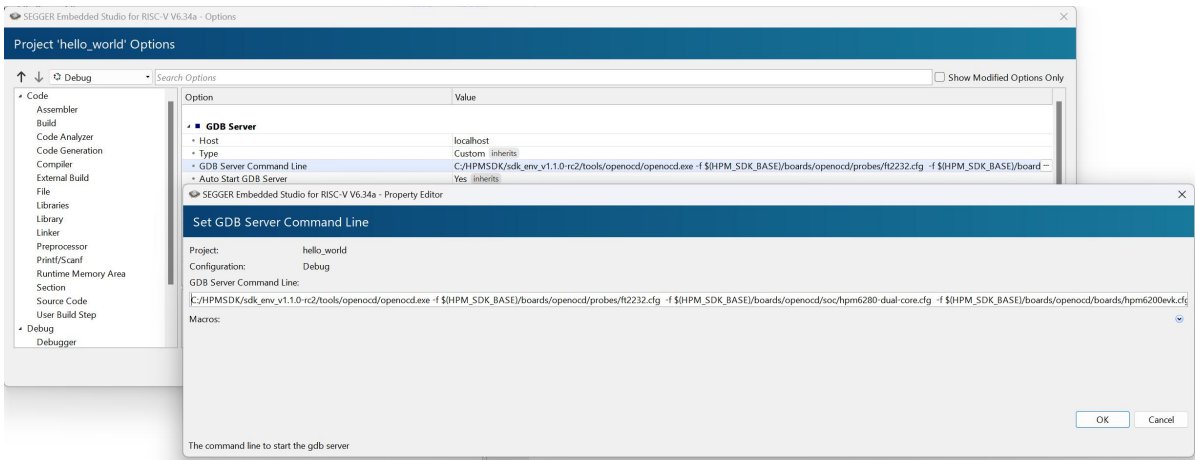


图 25：GDB Server默认配置

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。

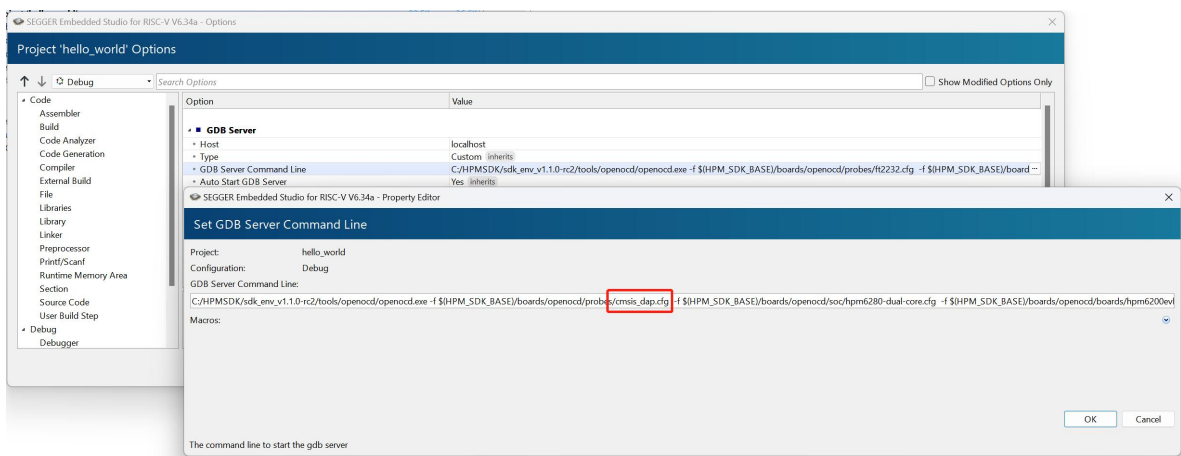


图 26：使用cmsis-dap 调试器GDB Server配置

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。

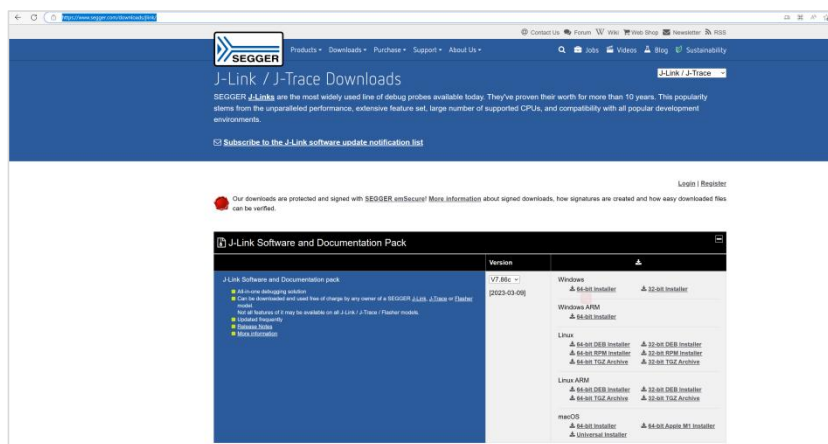


图 27: J-Link驱动下载

下载完成后安装J-Link驱动。

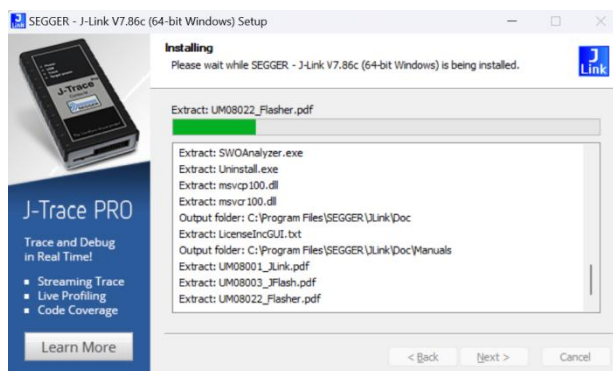


图 28: J-Link驱动安装

安装驱动完成后，正确连接J-Link JTAG接口到HPM6800EVK 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。

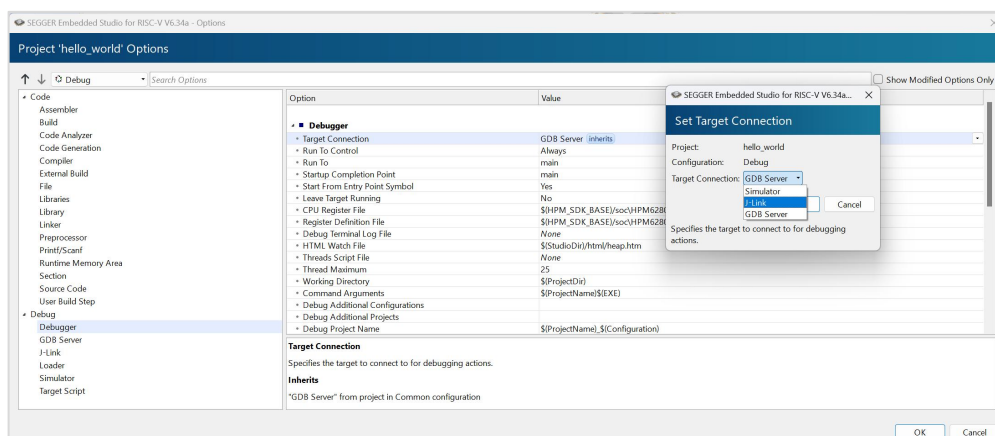


图 29: Target Connection 设置为J-Link

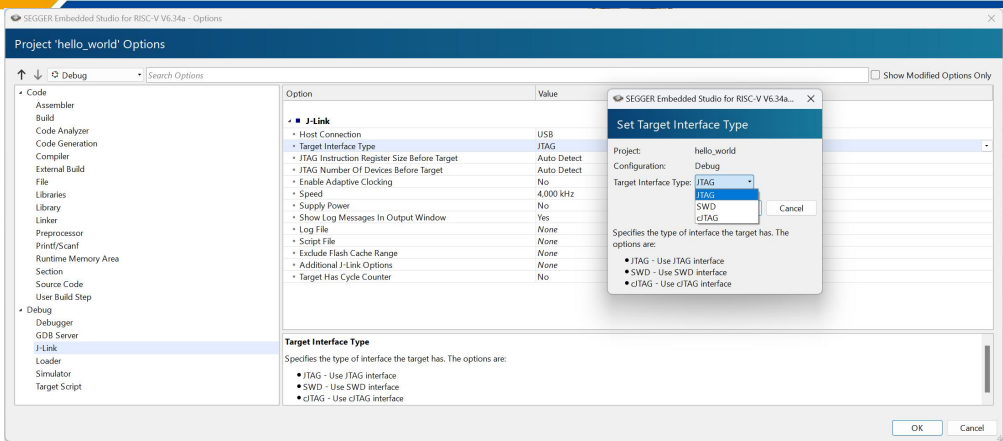


图 30: Target Interface Type 设置为 JTAG

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。

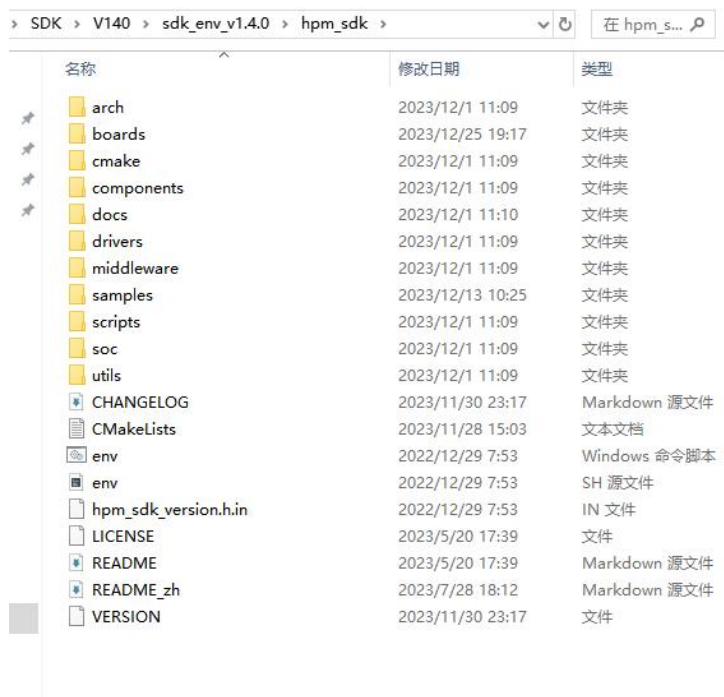


图 31：更新 SDK

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。



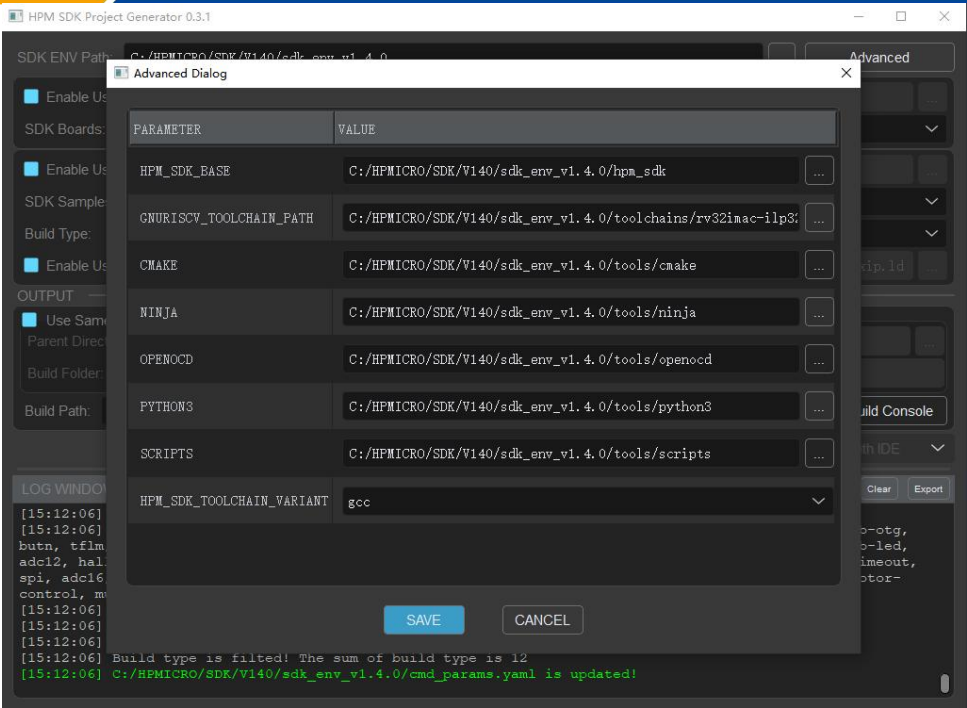


图 34: 更新 start_gui.exe 中 TOOLCHAIN_NAME

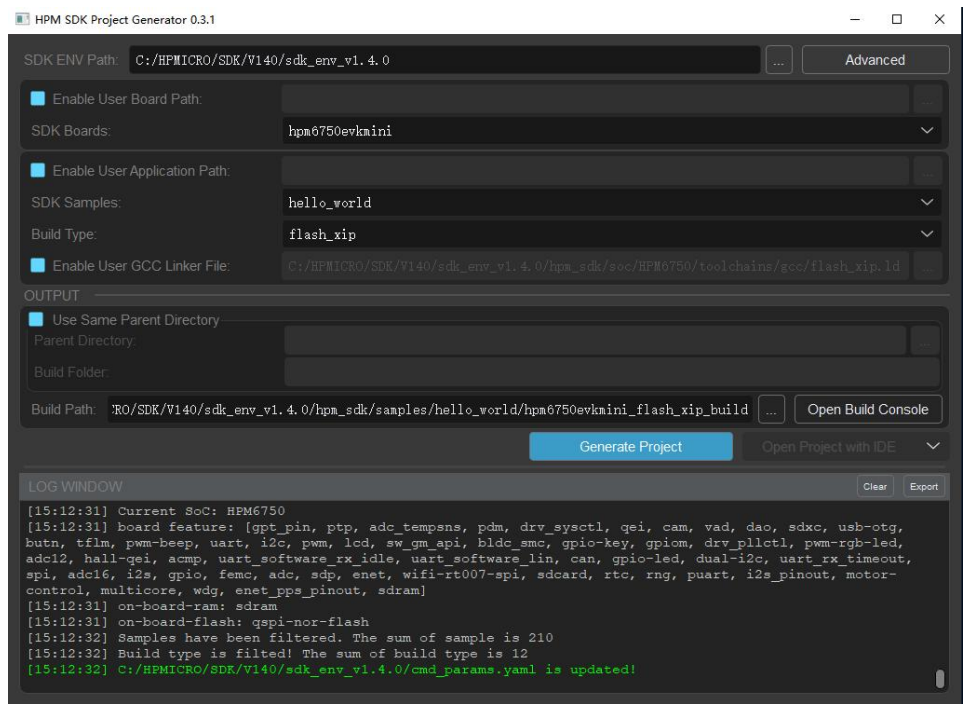


图 35: start_gui.exe更新TOOLCHAIN完成

3.7 版本信息

日期	版本	描述
Rev1. 0	2023/12/27	初版发布。

表 4：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。