

HPM6750

HPM6750EVKMINI 用户使用手册

此用户手册适用 **HPM6750** 系列高性能微控制器：

HPM6750IVM1、HPM6730IVM1
HPM6450IVM1、HPM6430IVM1
HPM6120IVM1、HPM6110IVM1
HPM6758IVM1



目录

第一章 HPM6750EVKMINI 简介	4
第二章 硬件电路	6
2.1 电路模块介绍	6
2.1.1 系统架构	6
2.1.2 电源	7
2.1.3 SDRAM	7
2.1.4 FLASH	7
2.1.5 LCD 接口	7
2.1.6 CAM 接口	7
2.1.7 USB 接口	7
2.1.8 TF 卡接口	7
2.1.9 WIFI 接口	8
2.1.10 DEBUG	8
2.1.11 按键	8
2.1.12 BOOT 拨码开关设置	8
2.1.13 电池座	8
2.1.14 扩展 IO 接口	8
第三章 HPM SDK 快速上手指南	11
3.1 简介	11
3.2 环境以及依赖	11
3.3 开发工具	11
3.4 使用 sdk_env/Segger Embedded Studio For RISC-V 快速指南	11
3.4.1 准备环境	11
3.4.2 创建工程	13
3.4.3 编译与运行工程	15
3.5 更新 sdk_env 中的 SDK/toolchain 指南	23
3.5.1 更新 sdk_env 中的 SDK	23
3.5.2 更新 sdk_env 中的 toolchain	23
第四章 RT-Thread Studio 快速上手指南	25

4.1 简介	25
4.2 环境以及依赖	25
4.2.1 安装板级支持包	25
4.2.2 安装编译器工具链	26
4.2.3 安装调试器支持包	26
4.3 新建工程	27
4.4 编译和调试	27
4.4.1 编译	28
4.4.2 调试	28
第五章 版本信息	30
第六章 免责声明	31

表格目录

1.1 主要器件位号对应器件功能名称	5
2.1 启动配置表	8
2.2 P1 连接器列表	9
2.3 P2 连接器列表	10
5.1 版本信息	30

图片目录

1.1 顶层器件位置图	4
1.2 底层器件位置图	4
2.1 HPM6750EVKMINI 硬件设计框图	7
3.1 安装 FTDI 驱动	12
3.2 查看 Windows 设备管理器	13
3.3 打开 sdk prompt	13
3.4 查询 SDK 支持的开发板	14
3.5 构建目标板工程	14
3.6 构建目标板 flash_xip 工程	15
3.7 generate_project 帮助	15
3.8 Segger Embedded Studio 打开 hello_world 工程	16
3.9 Segger Embedded Studio 选择编译类型	16
3.10 Segger Embedded Studio 编译 hello_world 工程	17
3.11 Segger Embedded Studio 配置串口	18
3.12 Segger Embedded Studio 打开串口	19
3.13 Segger Embedded Studio 运行 hello_world 工程	20
3.14 Segger Embedded Studio 运行 hello_world	20
3.15 Segger Embedded Studio 程序执行控制	21
3.16 Segger Embedded Studio 数据查看窗口	21
3.17 Segger Embedded Studio 选择寄存器组	22
3.18 Segger Embedded Studio 帮助	22
3.19 更新 SDK	23
3.20 拷贝 toolchain	24
3.21 更新 start_cmd.cmd 中 TOOLCHAIN_NAME	24
4.1 安装 HPM6750EVKMINI 板级支持包	25
4.2 安装 RISC-V GCC 工具链	26
4.3 安装 OpenOCD-HPMicro 调试器支持包	27
4.4 创建新的 RT-Thread 工程	27
4.5 编译新的 RT-Thread 工程	28
4.6 RT-Thread 启动信息	28
4.7 运行 RT-Thread 工程	29

第一章 HPM6750EVKMINI 简介

HPM6750EVKMINI 板的器件位置如图 1.1，图 1.2。表 1.1 给出了器件位置对应器件的名称。

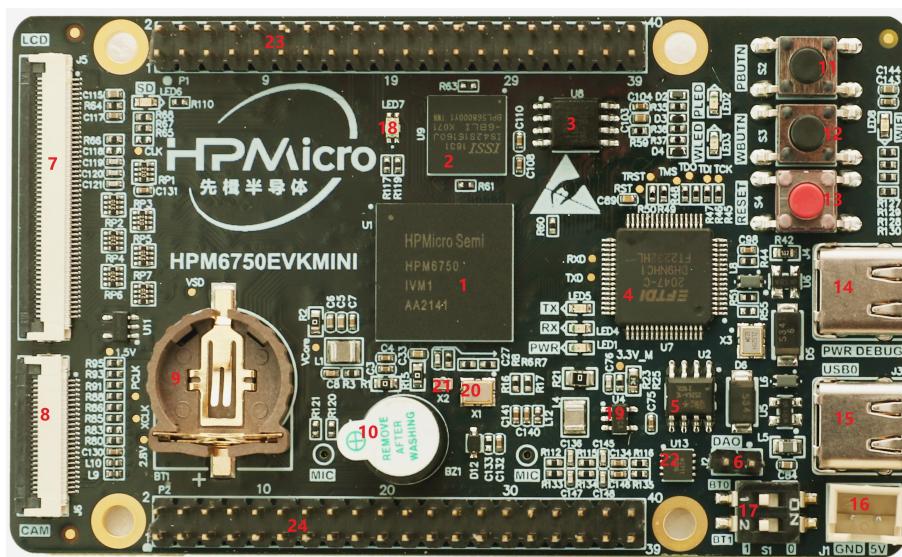
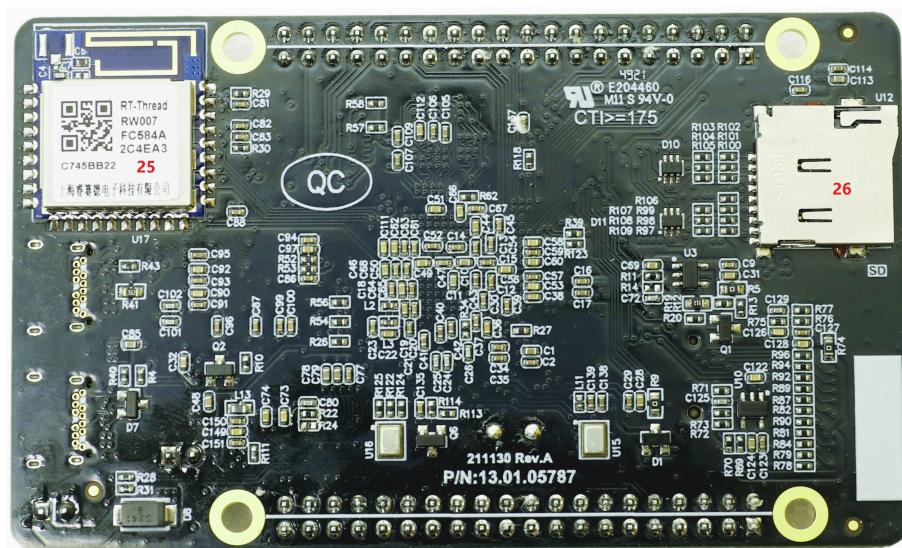


图 1.1: 顶层器件位置图



序号	名称	序号	名称
1	HPM6750	2	SDRAM
3	FLASH	4	DEBUG 芯片
5	USB 输出保护芯片	6	DAO 接口
7	LCD 连接器	8	CAM 连接器
9	电池座	10	蜂鸣器
11	PBTN 按键	12	WBTN 按键
13	RESET 按键	14	DEBUG TYPICAL 接口
15	USB0 TYPICAL 接口	16	5V 电源输入接口
17	BOOT 配置拨码开关	18	三色状态指示灯
19	DCDC 芯片	20	24MHZ 晶体
21	32.768K 晶体	22	数字功放
23	扩展接口 P1	24	扩展接口 P2
25	WIFI 模块	26	TF 卡插座

表 1.1: 主要器件位号对应器件功能名称

第二章 硬件电路

HPM6750EVKMINI 电源适配器供电不能超过 5.5V，如超过会可能会损坏板上器件。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

HPM6750EVKMINI 系统架构如图 2.1。

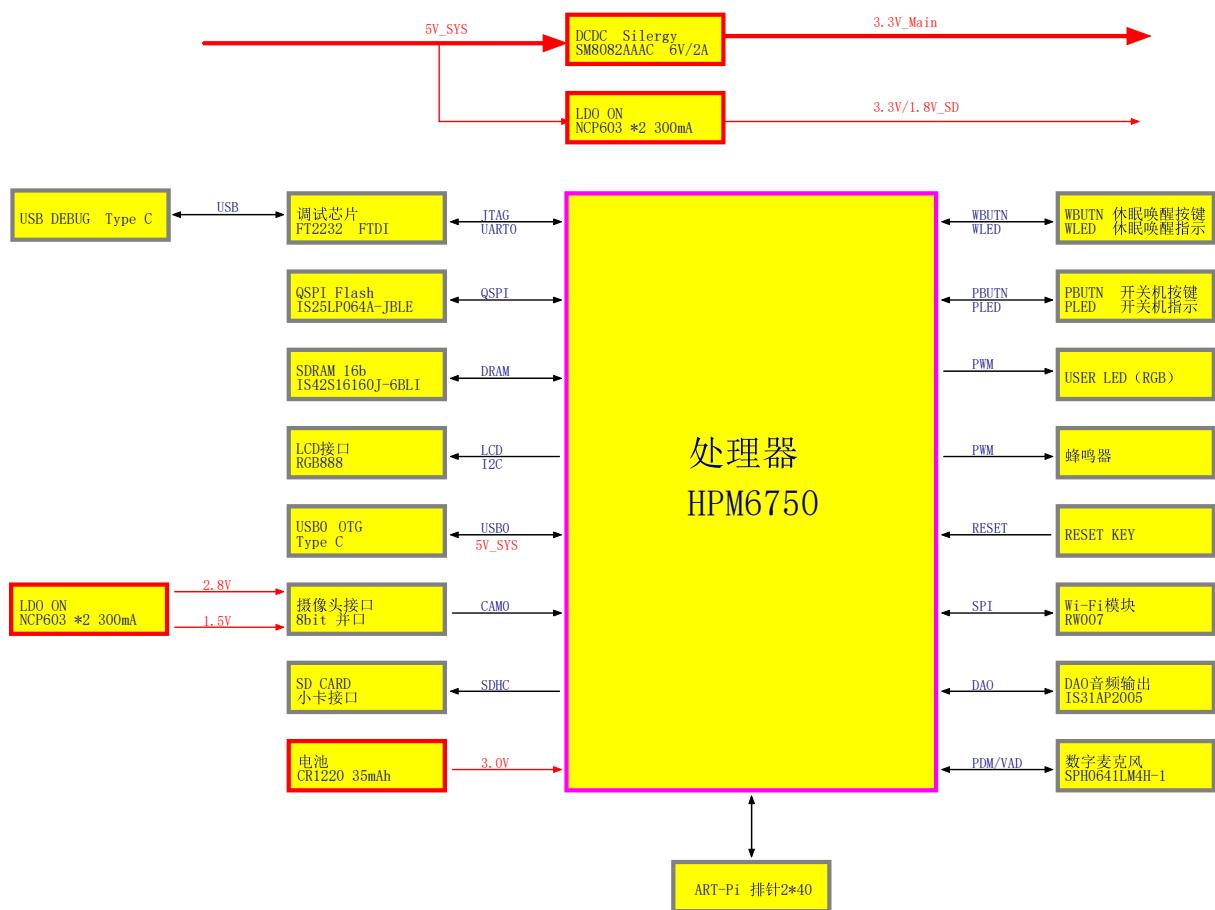


图 2.1: HPM6750EVKMINI 硬件设计框图

2.1.2 电源

HPM6750EVKMINI 具有多种供电方式，可以选择通过任意 USB Type C 或 5V 电源适配器接口来为整板供电。当系统外接 7 寸 LCD 屏时，由于背光耗电比较大，有可能单一 Type C 的 USB 接口供电不足，此时需要接多个 USB 接口供电或者使用 5V 电源适配器接口为整板供电。板上提供板载纽扣电池 CR1220 接口，为 MCU 电池备份域供电。

2.1.3 SDRAM

U9 是 HPM6750EVKMINI 板上的 SDRAM 器件，16 位数据线，容量 256Mb，时钟速度 166MHz，封装 BGA (8mmx8mm)。型号 IS42S16160J-6BLI。

2.1.4 FLASH

U8 是 HPM6750EVKMINI 板上的 NOR FLASH 器件，4 位数据线，容量 64Mb，时钟速度 104MHz，封装 SOP8 208mil。型号 IS25LP064A-JBLE。

2.1.5 LCD 接口

J5 是 HPM6750EVKMINI 板上的 LCD 接口，通过该接口实现显示功能。板上不提供背光功能，背光电路需要在接口板上实现。我们有配套 LCD 屏子板供应。若自己选配 LCD 屏，请参照原理图确认屏的接口信号定义，需要严格匹配，否则导致 PCBA 或屏的损坏。

2.1.6 CAM 接口

J6 是 HPM6750EVKMINI 板上的 CAM 接口，通过该接口实现 CAM 的图像传输功能。直接支持 OV7725 和 OV5640 模组。如果接其他 CAM 模块，先确保信号和电源接口跟原理图相匹配。

2.1.7 USB 接口

J3 是 HPM6750EVKMINI 板上的 USB 接口，连接器类型是 Type C。支持 USB 2.0 OTG。

2.1.8 TF 卡接口

U12 是 HPM6750EVKMINI 板上的 TF 卡接口。

2.1.9 WIFI 接口

U17 是 HPM6750EVKMINI 板上的 WIFI 模块。通过 SPI 接口访问。

2.1.10 DEBUG

J4 是 HPM6750EVKMINI 板上的 DEBUG 接口，连接器类型是 Type C。通过 U7:FT2232HL 实现 USB 到 JTAG 和 UART 的转换。用户只需要连接 USB 接口即可访问芯片 JTAG 接口和 UART0 接口。

2.1.11 按键

S2、S3、S4 分别是 EVK 板上对应的 PBUTN、WBUTN、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，PIBUTN 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。

用户通过 RESET 按键对 MCU 进行外部复位。

2.1.12 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 2.1。

S1 拨码开关 [1:0]		启动模式	说明
OFF	OFF	XPI NOR 启动	从连接在 XPIO/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 2.1: 启动配置表

2.1.13 电池座

BT1 是 HPM6750EVKMINI 板上的电池座，对应电池型号 1220，电压 3V。

2.1.14 扩展 IO 接口

扩展 IO 接口包括 P1、P2 两个连接器接口。P1 与树莓派是兼容，并且在树莓派基础上，还增加了 CAN 复用功能。P1、P2 的信号列表如表 2.2 和表 2.3。

引脚名	功能名	连接器编号		功能名	引脚名
	3.3V	1	2	5.0V	
PB13	I2C3.SDA	3	4	5.0V	
PB14	I2C3.SCL	5	6	GND	
PE25	PE25	7	8	URT13.TXD	PZ09
	GND	9	10	URT13.RXD	PZ08
PE26	PE26	11	12	I2S0.BCLK	PF06
PB11	I2C0.SCL	13	14	GND	
PB10	I2C0.SDA	15	16	PE29	PE29
	3.3V	17	18	URT14.TXD	PZ11
PB22	SPI2.MOSI	19	20	GND	
PB25	SPI2.MISO	21	22	URT14.RXD	PZ10
PB21	SPI2.SCLK	23	24	SPI2.CSN	PB24
	GND	25	26	PF0	PF0
PF1	PF1	27	28	PF2	PF2
PB27	PWM0.P4	29	30	GND	
PF4	PF4	31	32	PWM0.P5	PB26
PE28	URT6.TXD	33	34	GND	
PF09	I2S0.FCLK	35	36	URT6.RXD	PE27
PF03	I2S0.MCLK	37	38	I2S0.RXD1	PF05
	GND	39	40	I2S0.TXD1	PF07

表 2.2: 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		
PE14	ETH1.TXEN	3	4	3.3V	
PE19	ETH1.TXD0	5	6	GND	
PE31	CAN1.TXD	7	8	ETH1.TXD1	PE17
	GND	9	10	CAN1.RXD	PE30
PE21	ADC0/1/2.VINP7	11	12	ETH1.MDC	PD11
PE22	ADC0/1/2.VINP8	13	14	GND	
PE23	ADC0/1/2.VINP9	15	16	ADC0/1/2.VINP10	PE24
	3.3V	17	18	ETH1.RXD0	PE20
PE18	ETH1.RXD1	19	20	GND	
PE15	ETH1.RXDV	21	22	ETH1.REFCLK	PE16
PD16	PD16	23	24	ETH1.MDIO	PD14
	GND	25	26	ETH1.RST	PD15
	USB1_P	27	28	USB1_N	
PB29	SPI3.CSN	29	30	GND	
PC03	SPI3.MISO	31	32	SPI3.SCLK	PC02

PB30	SPI3.MOSI	33	34	GND	
PD19	PD19	35	36	PD20	PD20
PD23	PD23	37	38	PD24	PD24
	GND	39	40	PD25	PD25

表 2.3: P2 连接器列表

第三章 HPM SDK 快速上手指南

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。

3.4 使用 `sdk_env`/Segger Embedded Studio For RISC-V 快速指南

3.4.1 准备环境

1. 下载并安装 Segger Embedded Studio For RISC-V (按照默认选项安装即可)。
2. 下载并解压 `sdk_env.zip`。

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

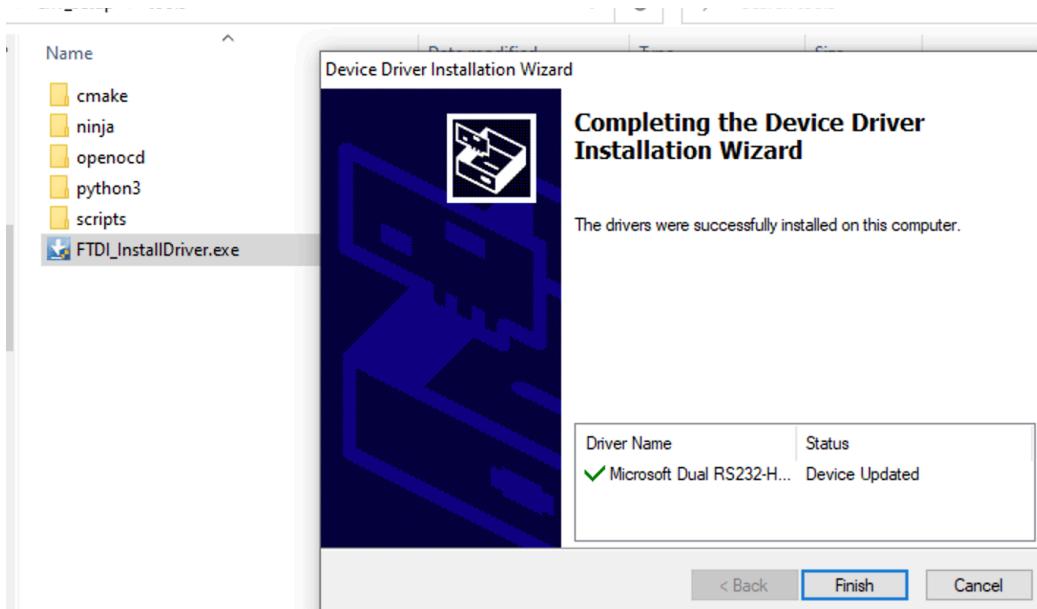


图 3.1: 安装 FTDI 驱动

正确安装驱动后，使用 USB type-C 线缆将开发板 DEBUG 接口连接到 PC，以 hpm6750evkmini 为例，连接 J4 到 PC，在 Windows 设备管理器中可以看到一个 USB Serial Port 以及一个 Dual RS232-HS，如图所示：

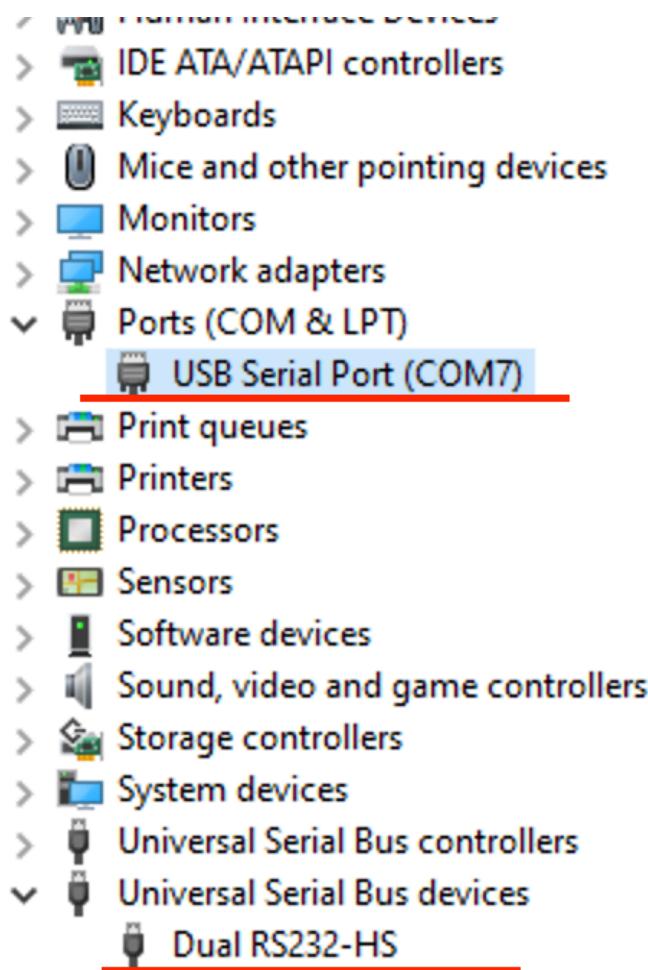


图 3.2: 查看 Windows 设备管理器

3.4.2 创建工程

- 双击打开 sdk_env 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，将会看到如下类似提示：

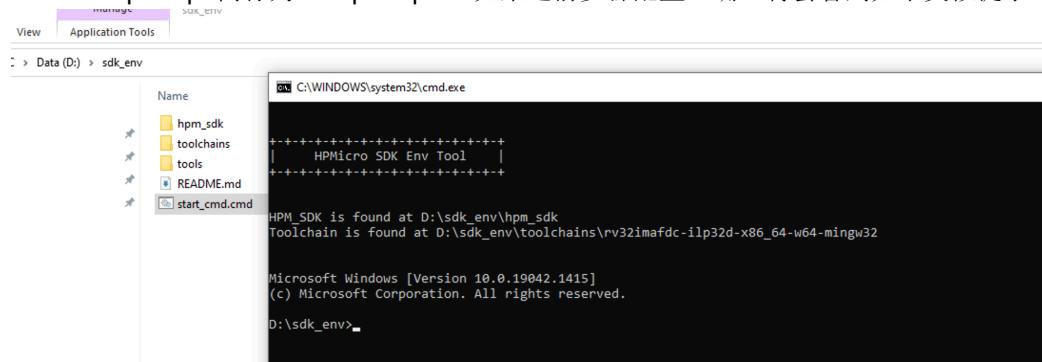


图 3.3: 打开 sdk prompt

- 运行如下命令，查询 SDK 支持的开发板

```
> generate_project -list
```

```

C:\Windows\system32\cmd.exe
+-----+
| HPMicro SDK Env Tool |
+-----+
HPM_SDK is found at D:\sdk_env\hpm_sdk
Toolchain is found at D:\sdk_env\toolchains\rv32imadc-ilp32d-x86_64-w64-mingw32
复制了 128 个文件
"openocd config file copied to C:\Users\Davis\AppData\Roaming from tools\openocd\tcl"

Microsoft Windows [版本 10.0.19042.1466]
(c) Microsoft Corporation。保留所有权利。

D:\sdk_env>generate_project -list
hpm6750evk
hpm6750evkmini

D:\sdk_env>

```

图 3.4: 查询 SDK 支持的开发板

3. 在 **sdk prompt** 中切换路径至 **SDK** 具体的一个示例程序目录, 以 **hello_world** 为例:

```
> cd %HPM_SDK_BASE%\samples\hello_world
```

4. 运行如下命令为目标板 (以 **hpm6750evkmini** 为例) 构建工程, 若构建成功, 将看到如下图类似提示。

在当前目录下将生成名为 **hpm6750evkmini_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**

```
> generate_project -b hpm6750evkmini
```

```

C:\Windows\system32\cmd.exe
D:\sdk_env\hpm_sdk\samples\hello_world>generate_project -b hpm6750evkmini
hpm6750evkmini
-- Application: D:/sdk_env/hpm_sdk/samples/hello_world
-- Board: hpm6750evkmini
-- Found toolchain: gnu (D:/sdk_env/toolchains/rv32imadc-ilp32d-x86_64-w64-mingw32)
-- The C compiler identification is GNU 11.1.0
-- The CXN compiler identification is GNU 11.1.0
-- The ASM compiler identification is GNU
-- Found assembler: D:/sdk_env/toolchains/rv32imadc-ilp32d-x86_64-w64-mingw32/bin/riscv32-unknown-elf-gcc.exe
-- Segger linker script: D:/sdk_env/hpm_sdk/soc/HPM6750/toolchains/segger/ram_ifc
-- Segger Embedded Studio Project: D:/sdk_env/hpm_sdk/samples/hello_world/hpm6750evkmini_build/segger_embedded_studio/hello_world.emProject
-- Configuring done
-- Generating done
-- Build files have been written to: D:/sdk_env/hpm_sdk/samples/hello_world/hpm6750evkmini_build

D:\sdk_env\hpm_sdk\samples\hello_world>

```

图 3.5: 构建目标板工程

Note: **generate_project** 可以生成多种工程类型, 如 **flash_xip** (最后链接完成后的应用中, **text** 将被链接到 **nor flash** 地址空间原地执行, **data** 将被链接到 SoC 片内 RAM 地址空间)

```

C:\WINDOWS\system32\cmd.exe
Name
hpm_sdk
toolchains
tools
README.md
start_cmd.cmd

HPM_SDK is found at D:\sdk_env\hpm_sdk
ToolChain is found at D:\sdk_env\toolchains\rv32imafdc-ilp32d-x86_64-w64-mingw32

Microsoft Windows [Version 10.0.19042.1415]
(c) Microsoft Corporation. All rights reserved.

D:\sdk_env>cd hpm_sdk\samples\hello_world\
D:\sdk_env\hpm_sdk\samples\hello_world>generate_project -b hpm6750evkmini -t flash_xip
-- Application: D:/sdk_env/hpm_sdk/samples/hello_world
-- Board: hpm6750evkmini
-- Found toolchain: gnu (D:\sdk_env\toolchains\rv32imafdc-ilp32d-x86_64-w64-mingw32)
-- 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: D:/sdk_env/toolchains/rv32imafdc-ilp32d-x86_64-w64-mingw32/bin/riscv32-unknown-elf-gcc.exe
-- Segger linker script: D:/sdk_env/hpm_sdk/soc/HPM6750/toolchains/segger/flash_xip.icf
-- Segger Embedded Studio Project: D:/sdk_env/hpm_sdk/samples/hello_world/hpm6750evkmini_build/segger_embedded_studio/hello_world.emProject
-- Configuring done
-- Generating done
-- Build files have been written to: D:/sdk_env/hpm_sdk/samples/hello_world/hpm6750evkmini_build

D:\sdk_env\hpm_sdk\samples\hello_world>

```

图 3.6: 构建目标板 flash_xip 工程

Note: 当调试 flash 目标时，建议把启动配置（具体请参考表 2.1）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。

Note: 更多 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
Name
hpm_sdk
toolchains
tools
README.md
start_cmd.cmd

HPM_SDK is found at D:\sdk_env\hpm_sdk
ToolChain is found at D:\sdk_env\toolchains\rv32imafdc-ilp32d-x86_64-w64-mingw32

Microsoft Windows [Version 10.0.19042.1415]
(c) Microsoft Corporation. All rights reserved.

D:\sdk_env>generate_project -h
generate_project [-f] [-b board] [-a] [-l] [-h]
  -f: force clean already existed build directory
  -b: board: specify board for project generation
  -a: generate projects for all supported boards
  -l: list all supported boards
  -t: type: specify build type: debug/release/flash_xip/flash_xip_release/flash_sdram_xip/flash_sdram_xip_release
  -h: show this text

D:\sdk_env>

```

图 3.7: generate_project 帮助

3.4.3 编译与运行工程

1. 使用 Segger Embedded Studio 打开构建的 hello_world.emProject 工程。

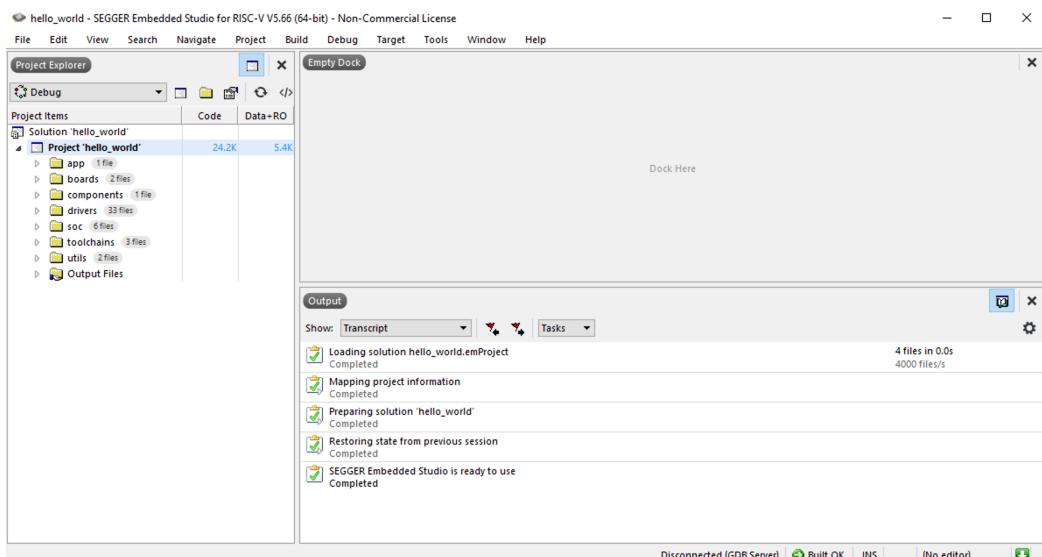


图 3.8: Segger Embedded Studio 打开 hello_world 工程

2. 点击 Build > Set Active Build Configuration > Debug/Release 选择编译类型。

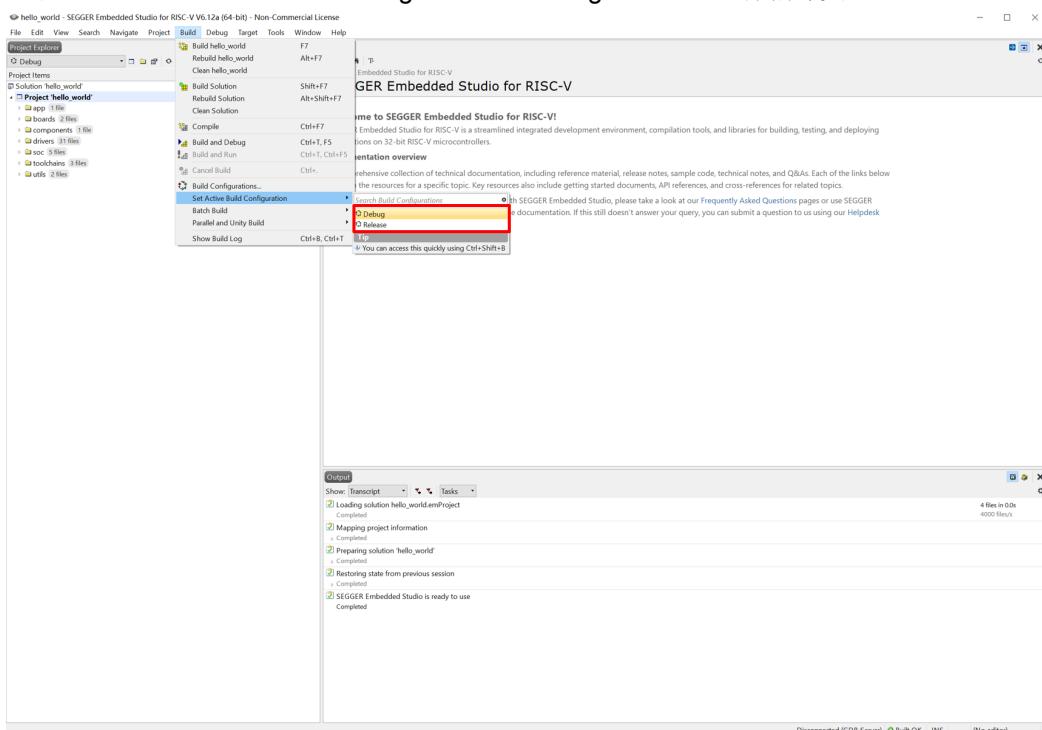


图 3.9: Segger Embedded Studio 选择编译类型

3. 点击 Build > Build project_name 编译工程。

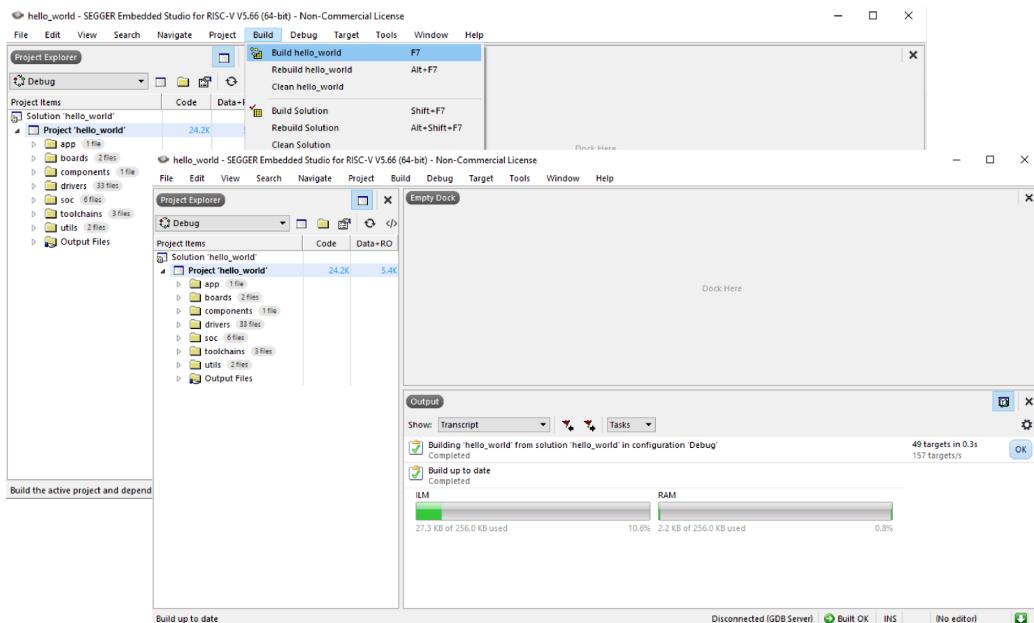


图 3.10: Segger Embedded Studio 编译 hello_world 工程

4. 点击 Tools > Terminal Emulator > Properties 在弹出的窗口中选择实际使用的串口并设置串口参数，配置完成后关闭弹出窗口。串口参数需配置如下：
 - (a) 115200 baud rate
 - (b) No parity
 - (c) 8 data bits
 - (d) 1 stop bit

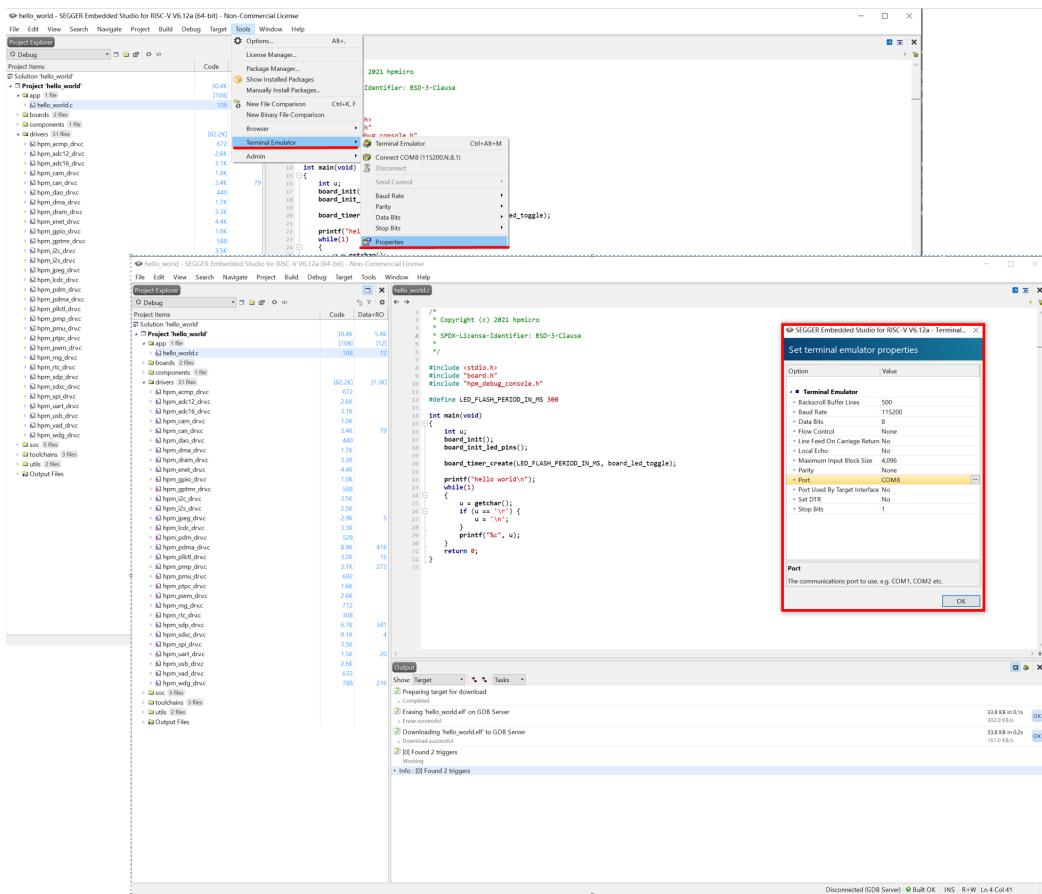


图 3.11: Segger Embedded Studio 配置串口

5. 点击 Tools > Terminal Emulator > Connect COMx, 接着点击 Tools > Terminal Emulator > Terminal Emulator 打开串口窗口。

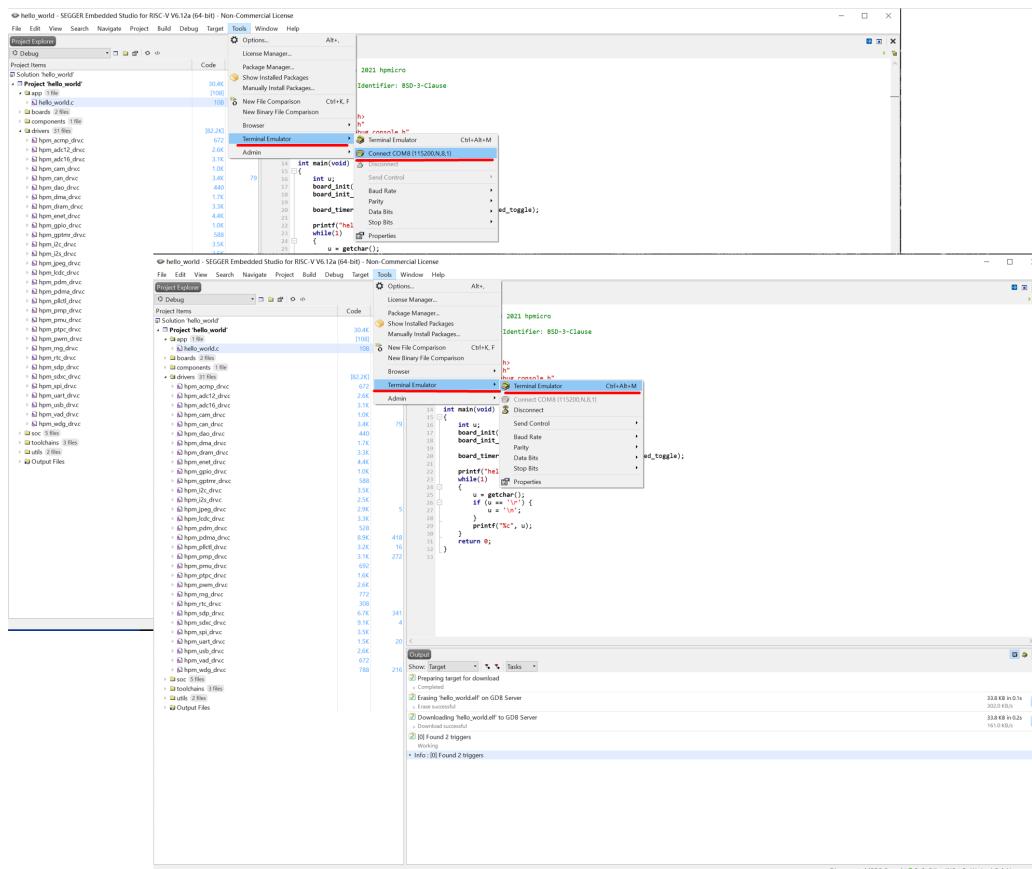


图 3.12: Segger Embedded Studio 打开串口

6. 点击 Debug > Go 开始调试程序，程序会停在 main 函数入口。

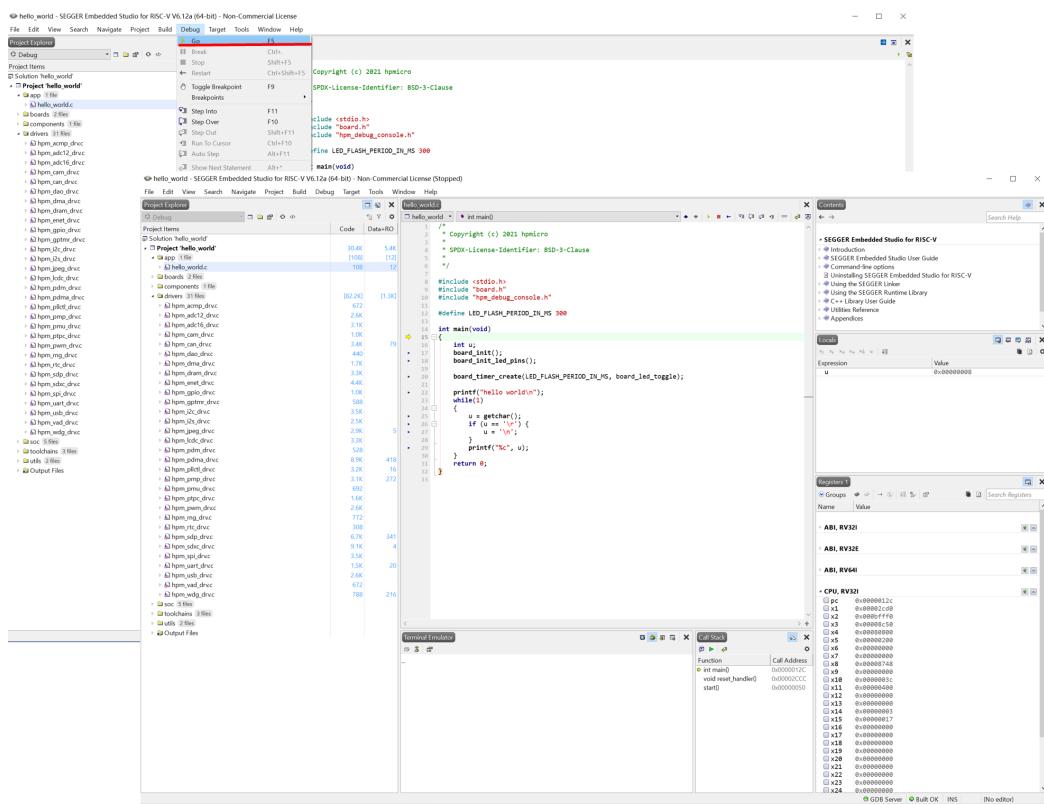


图 3.13: Segger Embedded Studio 运行 hello_world 工程

7. 点击 Continue Execution 按钮运行程序，串口窗口会输出程序运行信息。

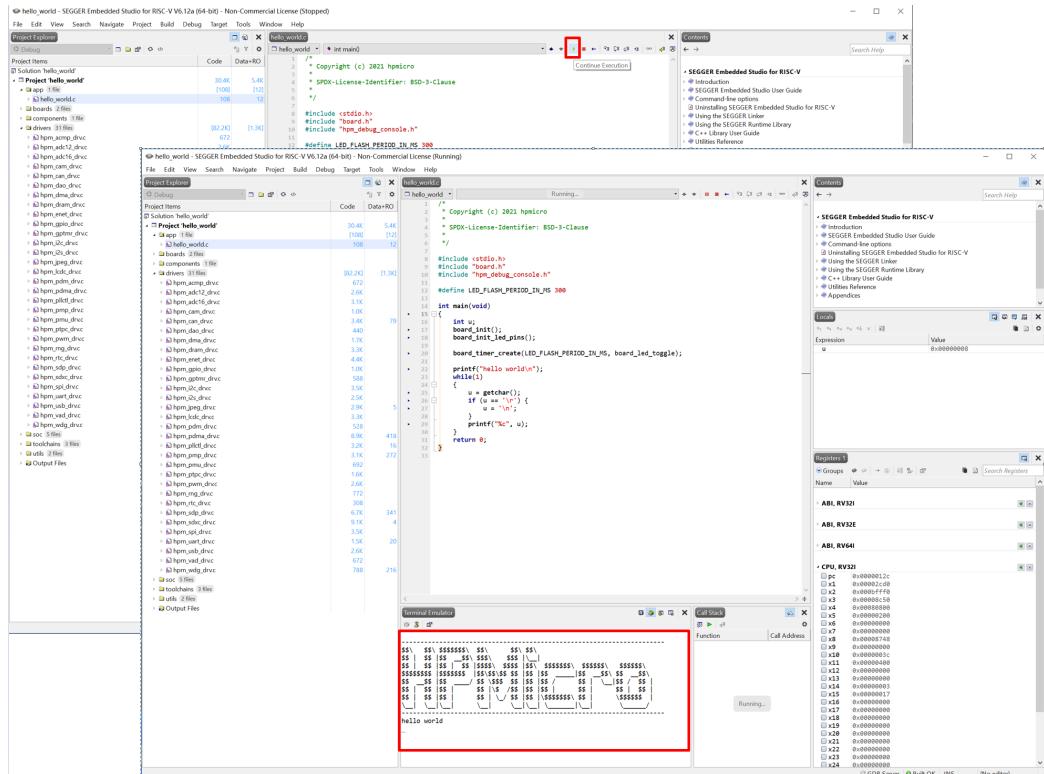


图 3.14: Segger Embedded Studio 运行 hello_world

8. Segger Embedded Studio For RISC-V IDE 支持丰富的调试操作，常用调试行为如下：

(a) 通过程序执行控制按钮或者 Debug 选项的下拉菜单控制代码运行。

Tips: 设置断点可通过直接双击希望设置断点的代码行的左侧区域（在行号前）来实现。

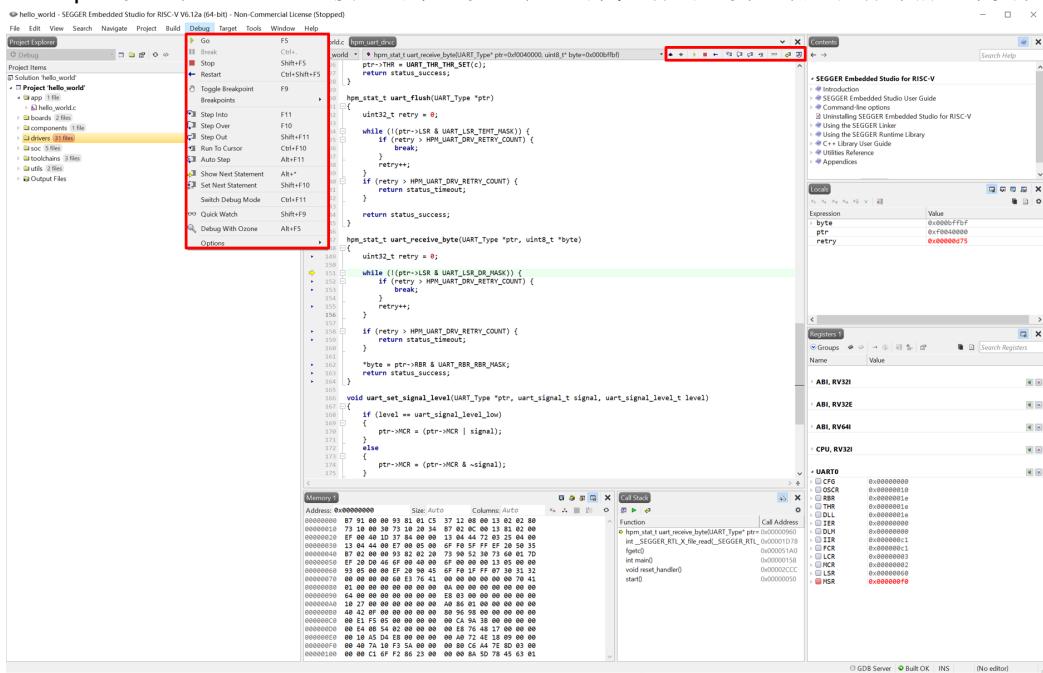


图 3.15: Segger Embedded Studio 程序执行控制

(b) 通过 view 选项的下拉菜单打开变量、寄存器、内存和汇编代码等窗口查看数据。

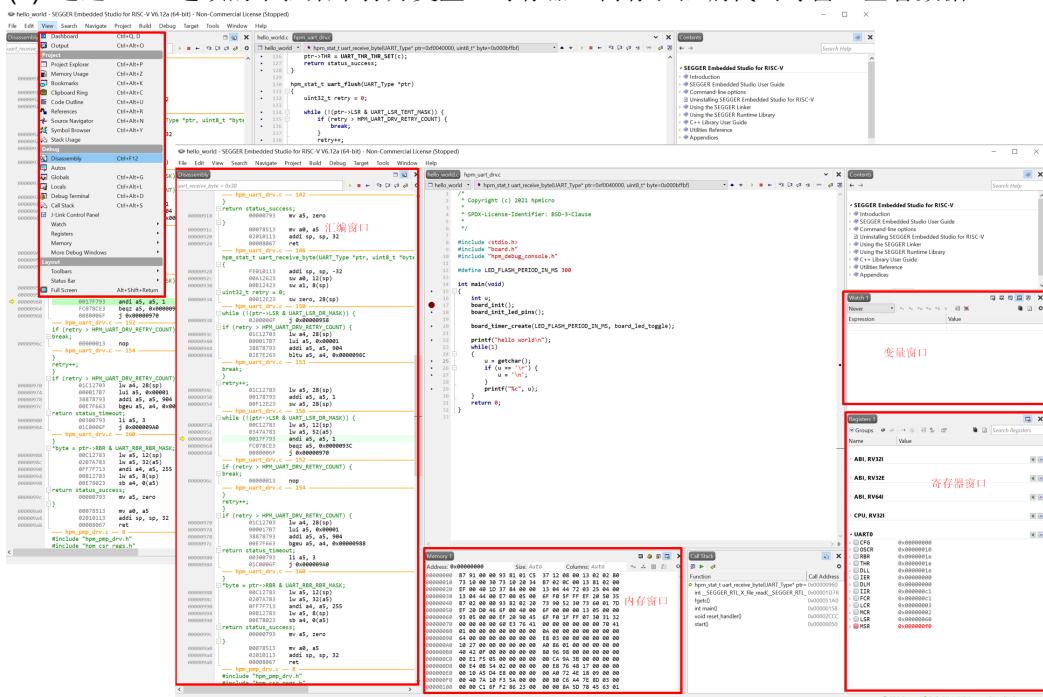


图 3.16: Segger Embedded Studio 数据查看窗口

查看外设寄存器时，需要先添加该外设寄存器组到寄存器显示窗口，可通过在寄存器窗口工具栏上点击 Group 按钮，从弹出的菜单中选择要显示的寄存器组实现。

Note: 在查看外设寄存器前请确保该外设的时钟已经打开，否则会造成 Segger Embedded Studio For RISC-V 运行出错。

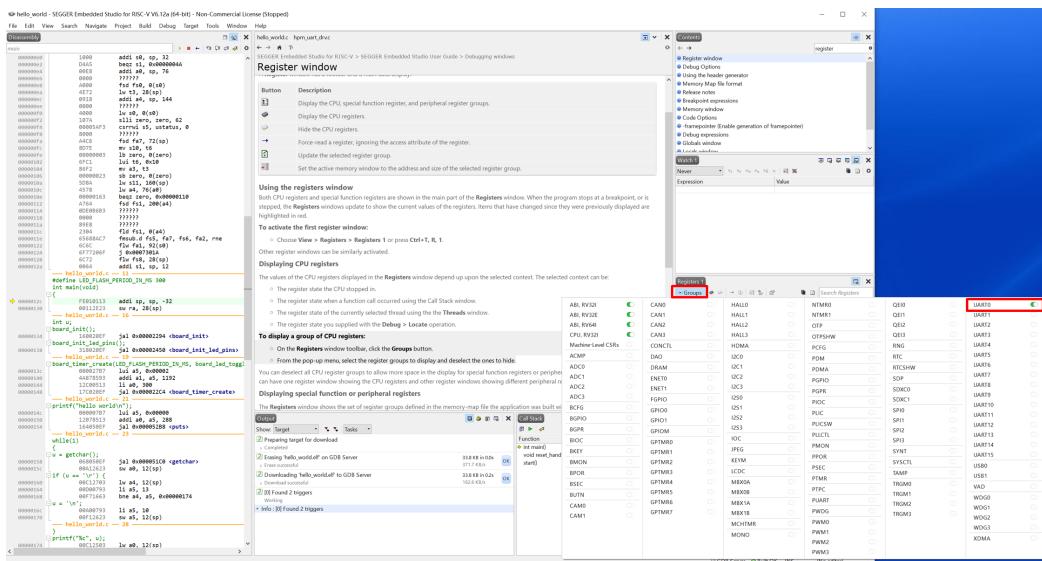


图 3.17: Segger Embedded Studio 选择寄存器组

(c) 点击 Help > SEGGER Embedded Studio Help 查看更多调试操作。

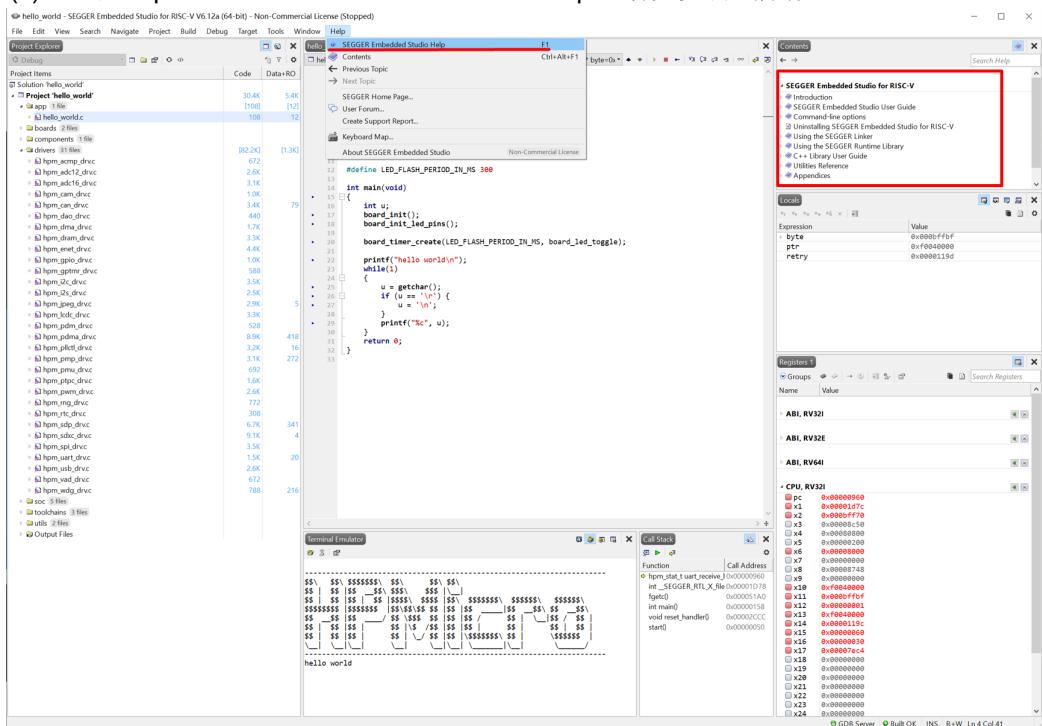


图 3.18: Segger Embedded Studio 帮助

3.5 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。

3.5.1 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。

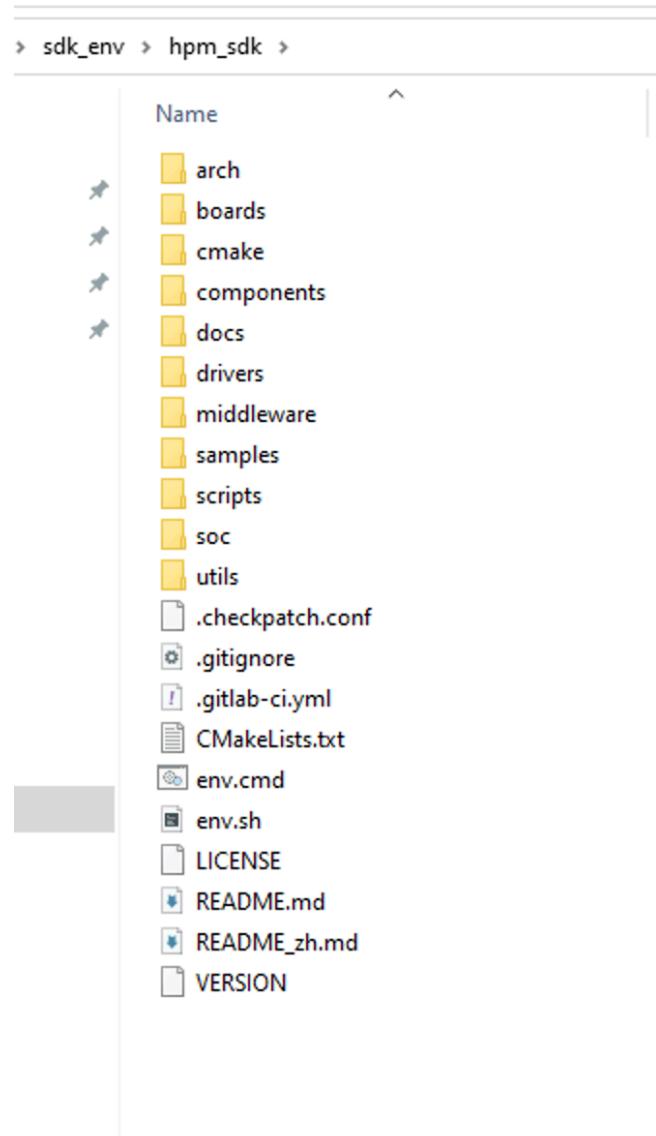


图 3.19: 更新 SDK

3.5.2 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。

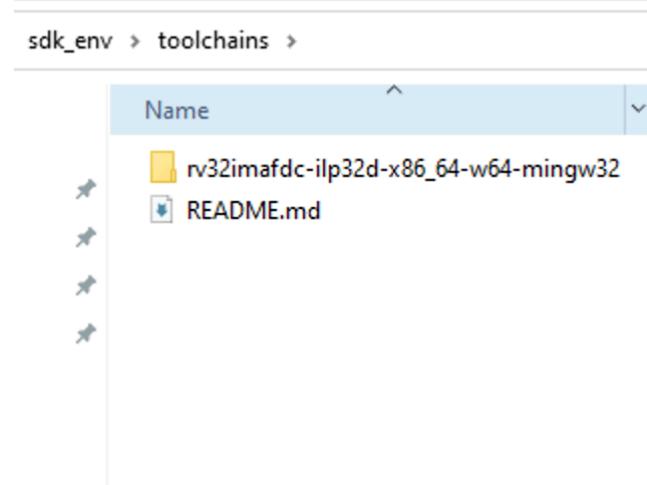


图 3.20: 拷贝 toolchain

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

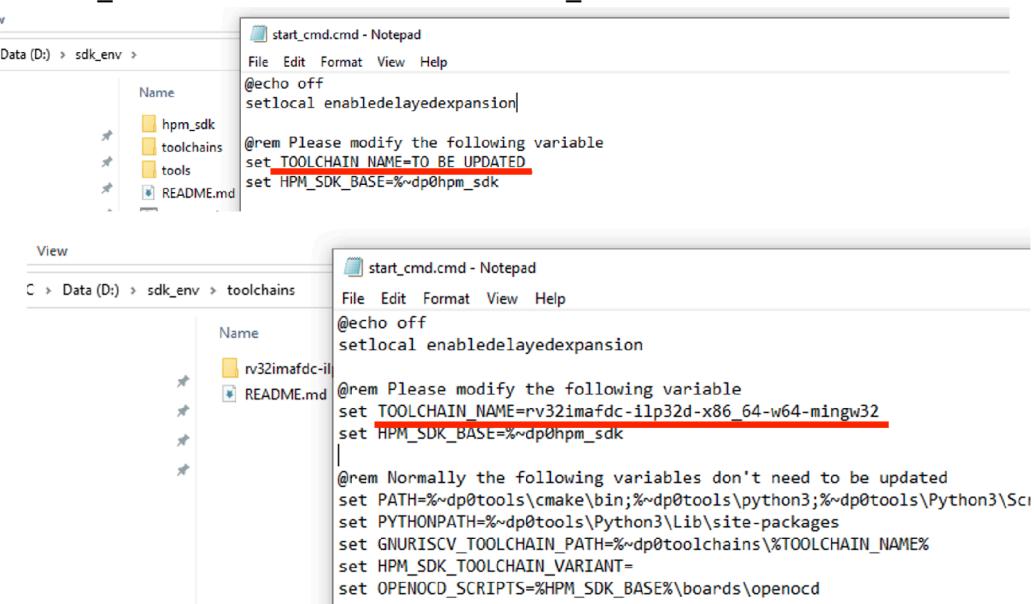


图 3.21: 更新 start_cmd.cmd 中 TOOLCHAIN_NAME

第四章 RT-Thread Studio 快速上手指南

4.1 简介

本章基于 HPM6750EVKMINI 开发板简要介绍 RT-Thread Studio 的快速上手步骤。

4.2 环境以及依赖

用户需要在 RT-Thread Studio 中安装如下包来使能对 HPM6750EVKMINI 软件开发的支持。

- 板级支持包 (Board Support Package)
- 编译器工具链 (Toolchain Support Package)
- 调试器支持包 (Debugger Support Package)。

4.2.1 安装板级支持包

点击如下图红色圈出“SDK Manager”图标，在弹出的窗口中找到“Board_Support_Packages”，并在项的子菜单下面找到“HPMicro”，选中 HPM6750EVKMINI，从下拉列表中选择最新的版本，如 0.2.1，点击“Install 1 packages”完成安装。

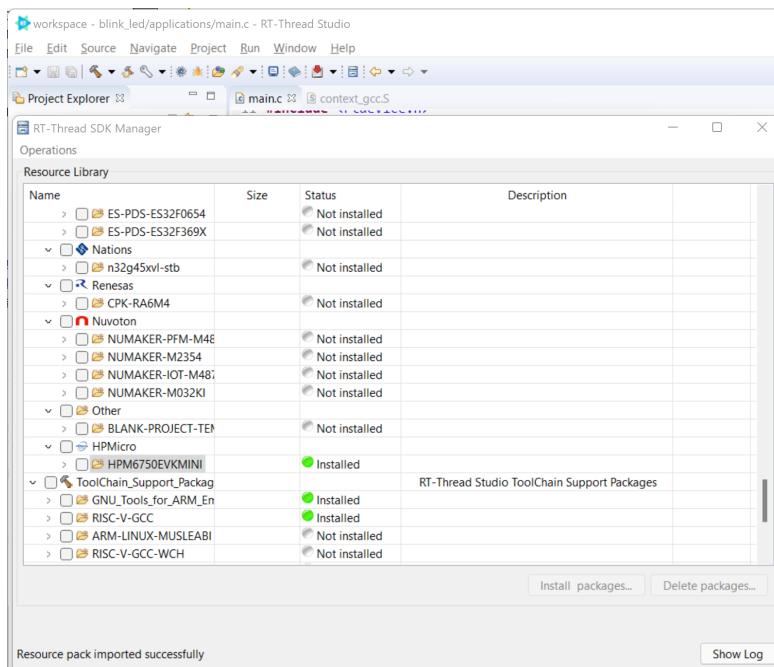


图 4.1: 安装 HPM6750EVKMINI 板级支持包

4.2.2 安装编译器工具链

点击如下图红色圈出“SDK Manager”图标，在弹出的窗口中找到“Toolchain_Support_Packages”，并在项的子菜单下面找到“RISC-V GCC”，选中最新的版本，如“10.1.0”，点击“Install 1 packages”完成安装。

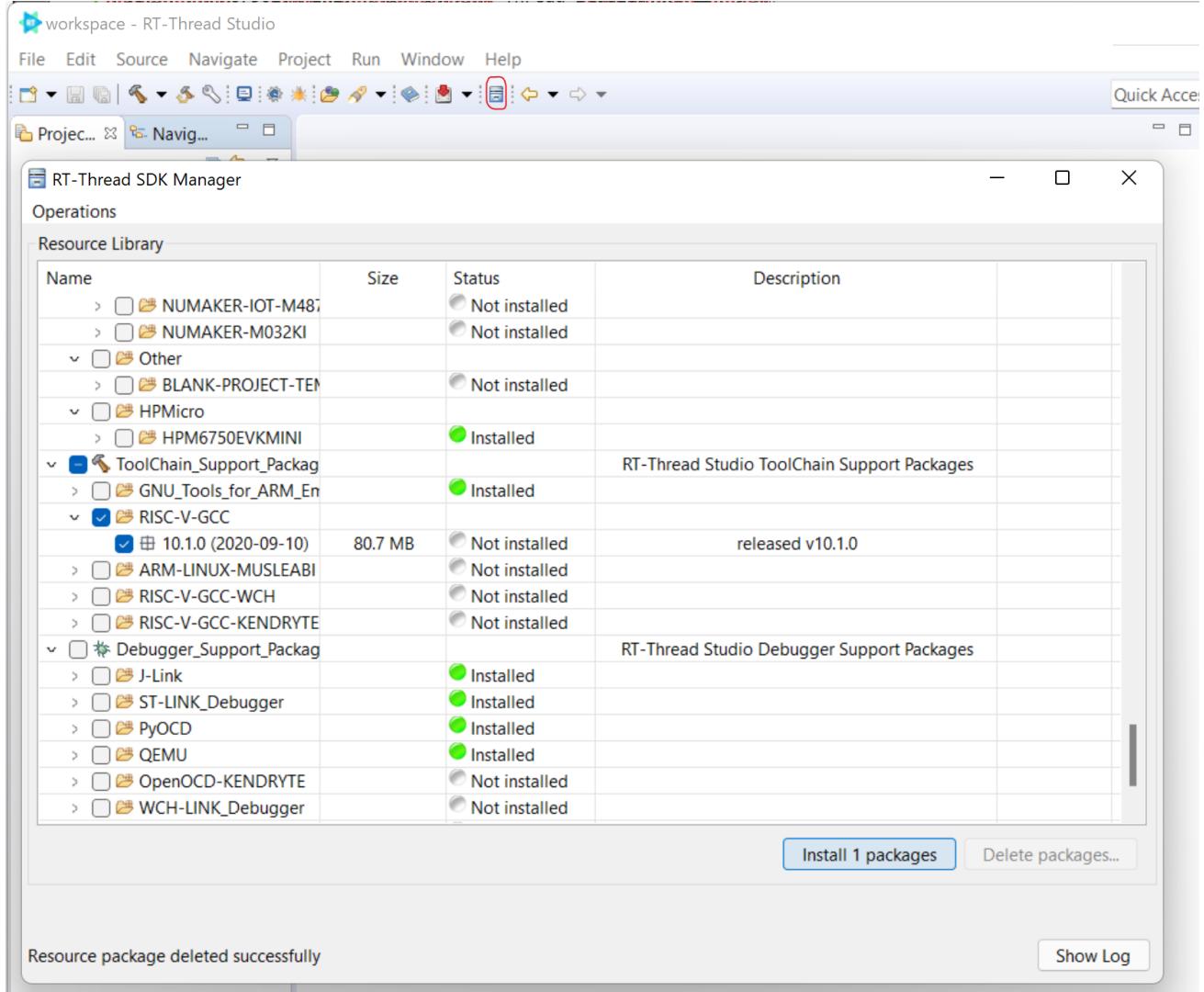


图 4.2: 安装 RISC-V GCC 工具链

4.2.3 安装调试器支持包

点击如下图红色圈出“SDK Manager”图标，在弹出的窗口中找到“Debugger_Support_Packages”，并在项的子菜单下面找到“OpenOCD-HPMicro”，选中最新的版本，如“0.2.0”，点击“Install 1 packages”完成安装。安装完调试器支持包后，用户可从 RT-Thread Studio 安装目录下的“repo/Extract/Debugger_Support_Packages /HPMicro/OpenOCD-HPMicro/<version>/tool”目录下找到“FTDI_InstallDriver.exe”，双击该应用程序完成驱动的安装。

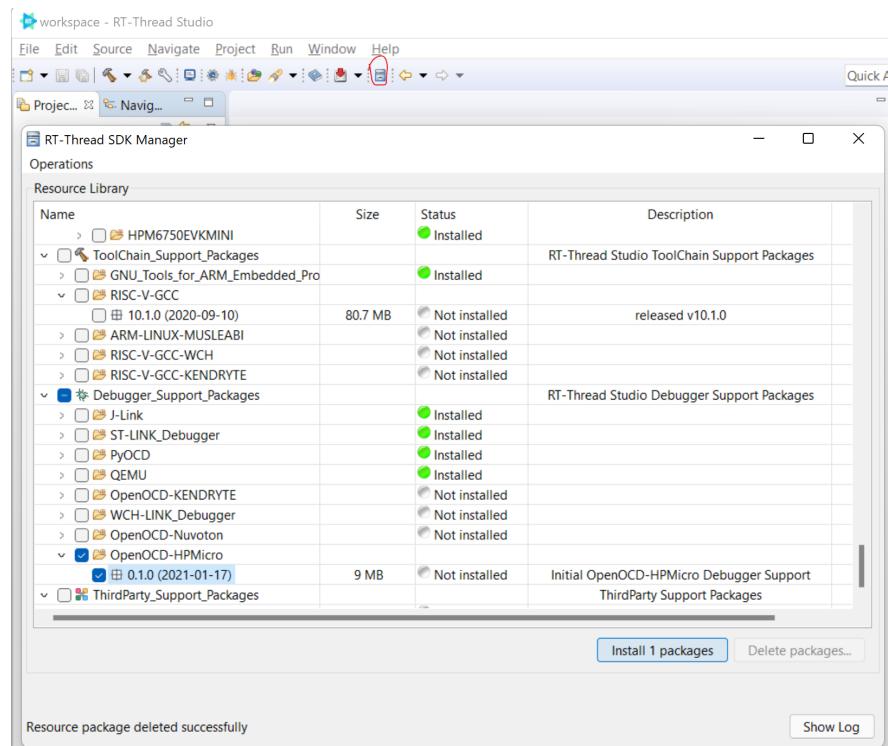


图 4.3: 安装 OpenOCD-HPMicro 调试器支持包

4.3 新建工程

在 Project Explorer 空白处点击鼠标右键，在弹出的菜单中依次点击“New”->“Project”，选中“RT-Thread Project”，再点击 Next，在弹出的窗口中选中“Based On Board”，出弹出如下窗口。填入工程的名称，如“blink_led”，点击“Finish”完成工程的创建。

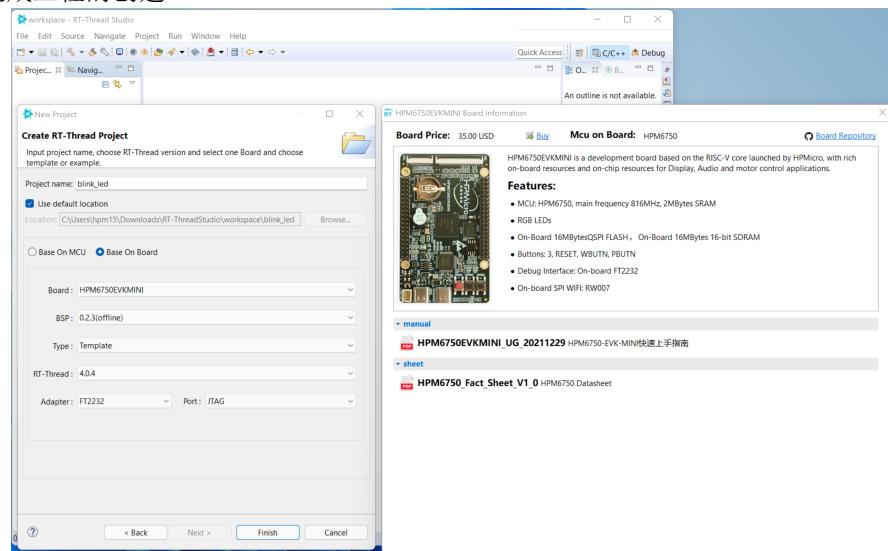


图 4.4: 创建新的 RT-Thread 工程

4.4 编译和调试

4.4.1 编译

点击如下图所示的编译图标，完成 blink_led 工程的编译。

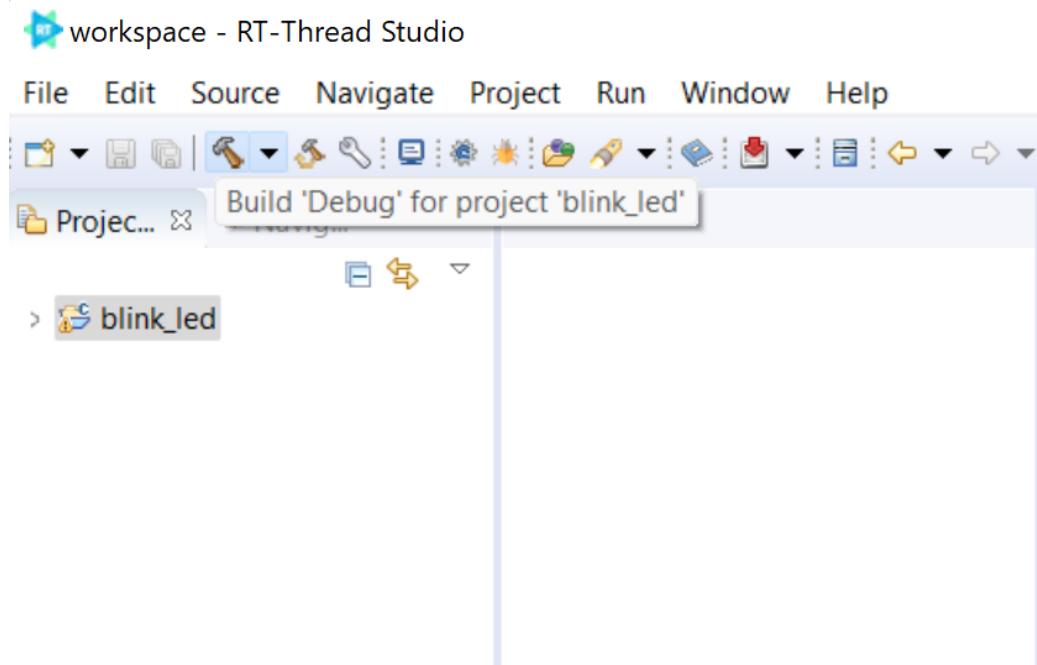


图 4.5: 编译新的 RT-Thread 工程

4.4.2 调试

1. 将板子 BOOT PIN 调到 BT0: OFF, BT1:OFF 的位置
2. 将 USB TypeC 线的 C 口插入到板子的“PWR DEBUG”端口，A 口插入到电脑的 USB 口。当板子的 LED7（三色灯）亮起时，说明 USB 连接正常。
3. 打开支持终端的串口助手，如 putty 或者 MobaXterm，连接板载的 USB 转串口端口。
4. 按”Ctrl + F5”或者点击[编译新 RT-Thread 工程](#)所示的虫子图标，程序会停在 main 函数入口处，同时串口终端会打印如下内容：

```

  \ | /
- RT -      Thread Operating System
  / | \      4.0.4 build Jan 14 2022 16:38:13
  2006 - 2021 Copyright by rt-thread team
msh />
  
```

图 4.6: RT-Thread 启动信息

5. 按 F5 或者下图所示的图标标记程序继续执行，此时，板载 RGB LED 灯会依次闪烁。

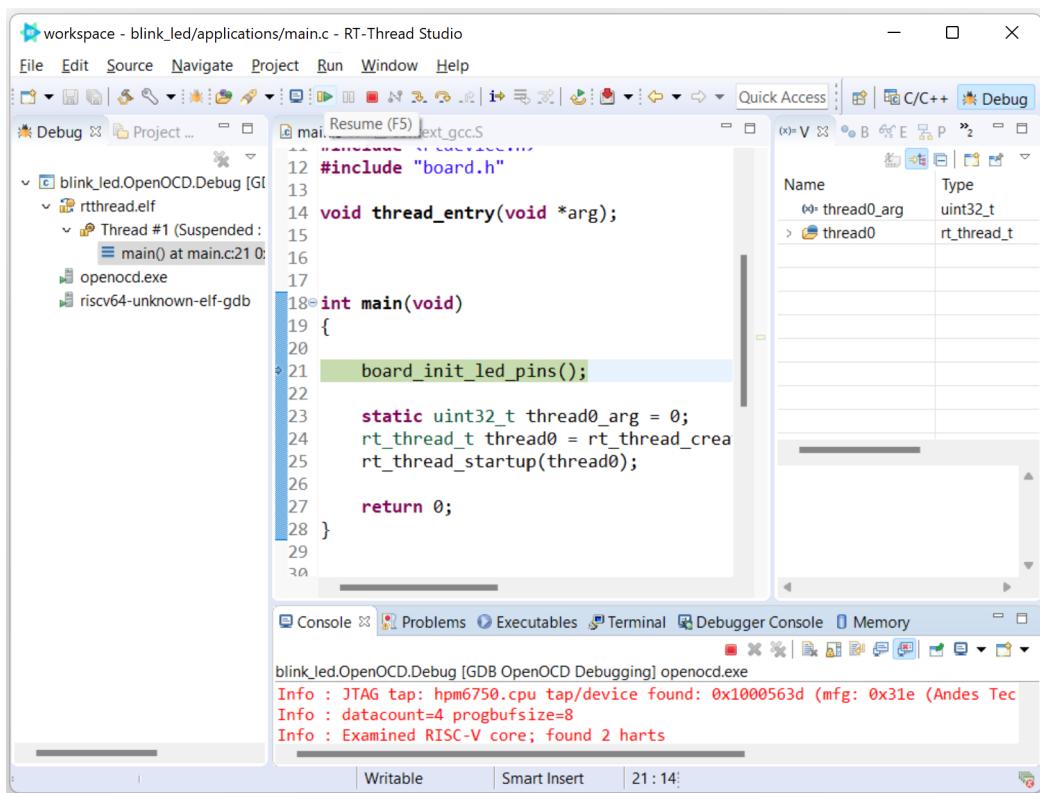


图 4.7: 运行 RT-Thread 工程

第五章 版本信息

日期	版本	描述
Rev0.1	2022/01/04	内部版初版发布。
Rev0.2	2022/01/18	增加 RT-Thread Studio 快速上手指南章节
Rev0.3	2022/01/25	更改 HPM SDK 快速上手指南章节名称，更换 FTDI 驱动安装截图（winusb）。
Rev0.4	2022/02/08	添加更多详细描述到 HPM SDK 快速上手指南章节。

表 5.1: 版本信息

第六章 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。