

HPM 软件快速上手指南

目录

第一章 HPM SDK 快速上手指南	2
1.1 简介	2
1.2 环境以及依赖	2
1.3 开发工具	2
1.4 使用 sdk_env/Segger Embedded Studio For RISC-V 快速指南	2
1.4.1 准备环境	2
1.4.2 创建工程	4
1.4.3 编译与运行工程	6
1.5 更新 sdk_env 中的 SDK/toolchain 指南	14
1.5.1 更新 sdk_env 中的 SDK	14
1.5.2 更新 sdk_env 中的 toolchain	14
第二章 RT-Thread Studio 快速上手指南	16
2.1 简介	16
2.2 环境以及依赖	16
2.2.1 安装板级支持包	16
2.2.2 安装编译器工具链	17
2.2.3 安装调试器支持包	17
2.3 新建工程	18
2.4 编译和调试	18
2.4.1 编译	19
2.4.2 调试	19
第三章 免责声明	21
3.1 版本信息	22

第一章 HPM SDK 快速上手指南

1.1 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。

1.2 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。

1.3 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。

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

1.4.1 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。

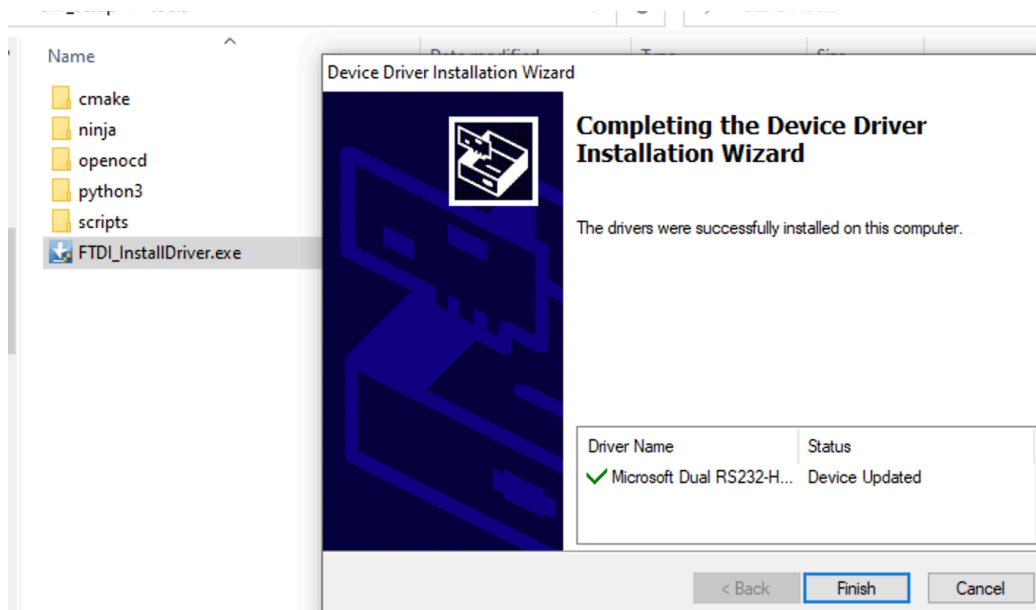


图 1.1: 安装 FTDI 驱动

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：

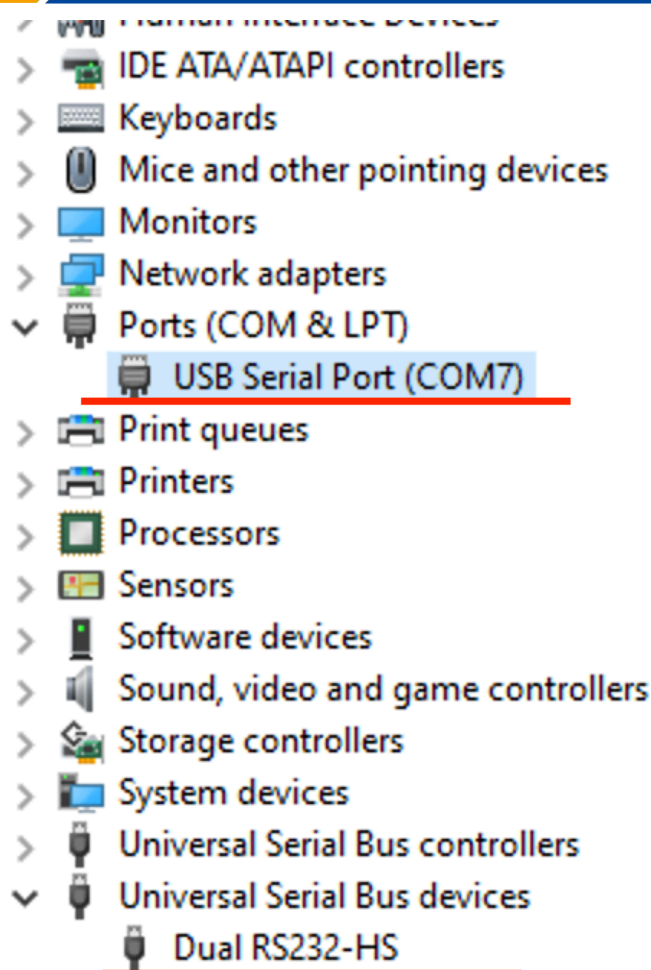
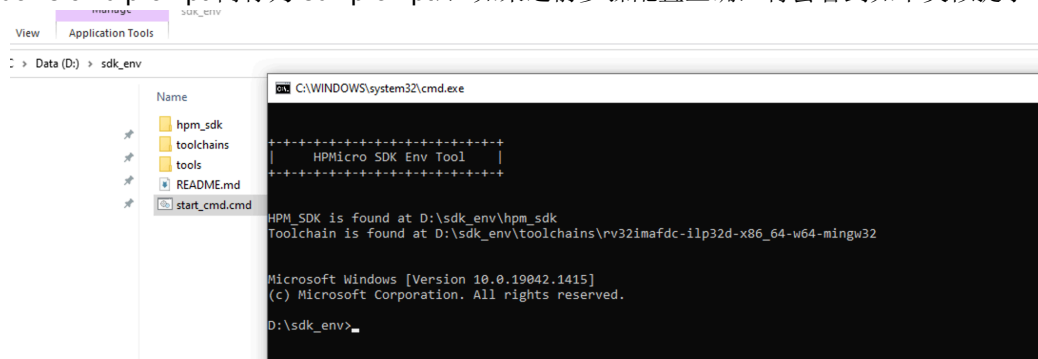


图 1.2: 查看 Windows 设备管理器

1.4.2 创建工程

1. 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 1.3: 打开 `sdk prompt`

2. 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 1.4: 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 CXX 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.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 1.5: 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）

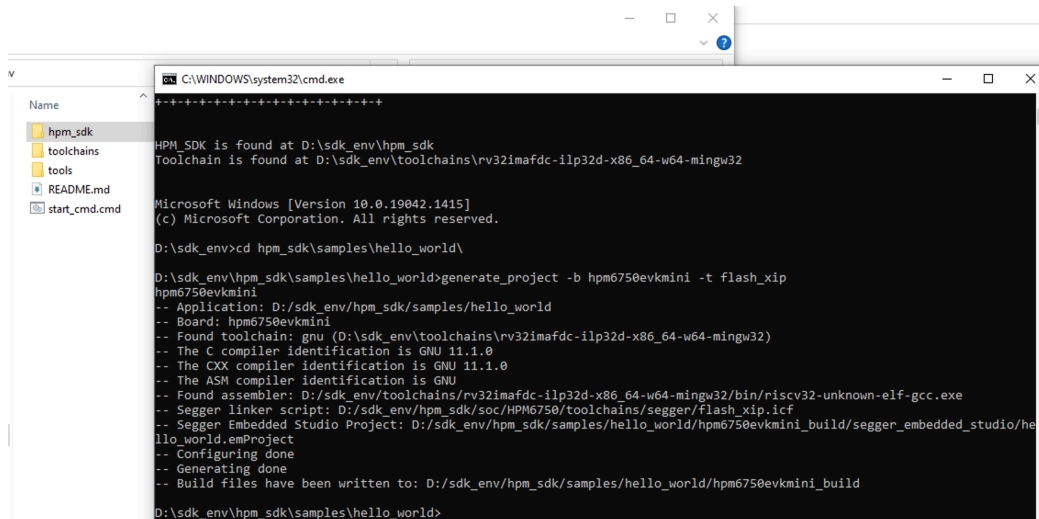


图 1.6: 构建目标板 flash_xip 工程

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

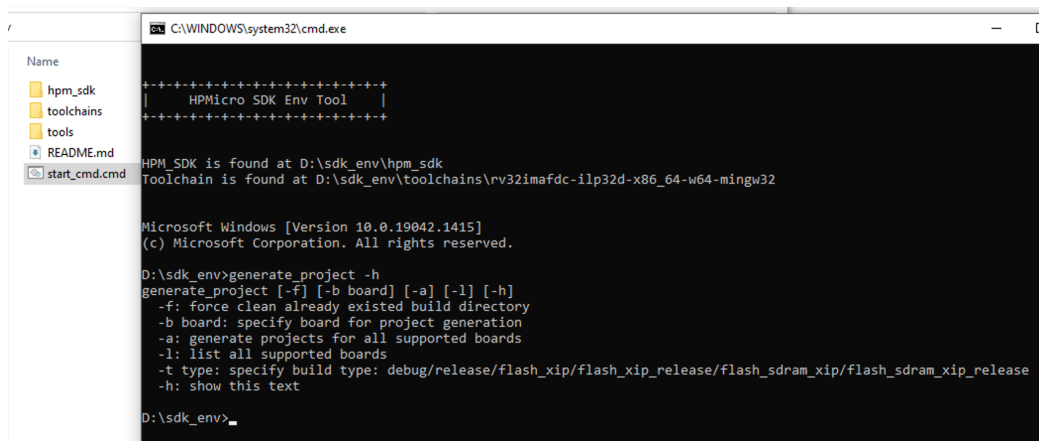


图 1.7: generate_project 帮助

1.4.3 编译与运行工程

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

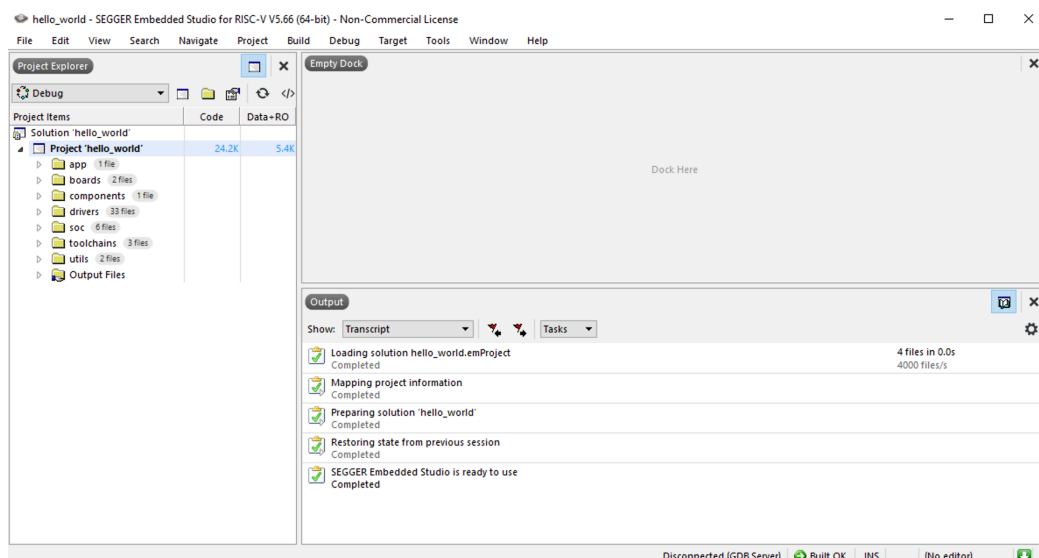


图 1.8: Segger Embedded Studio 打开 hello_world 工程

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

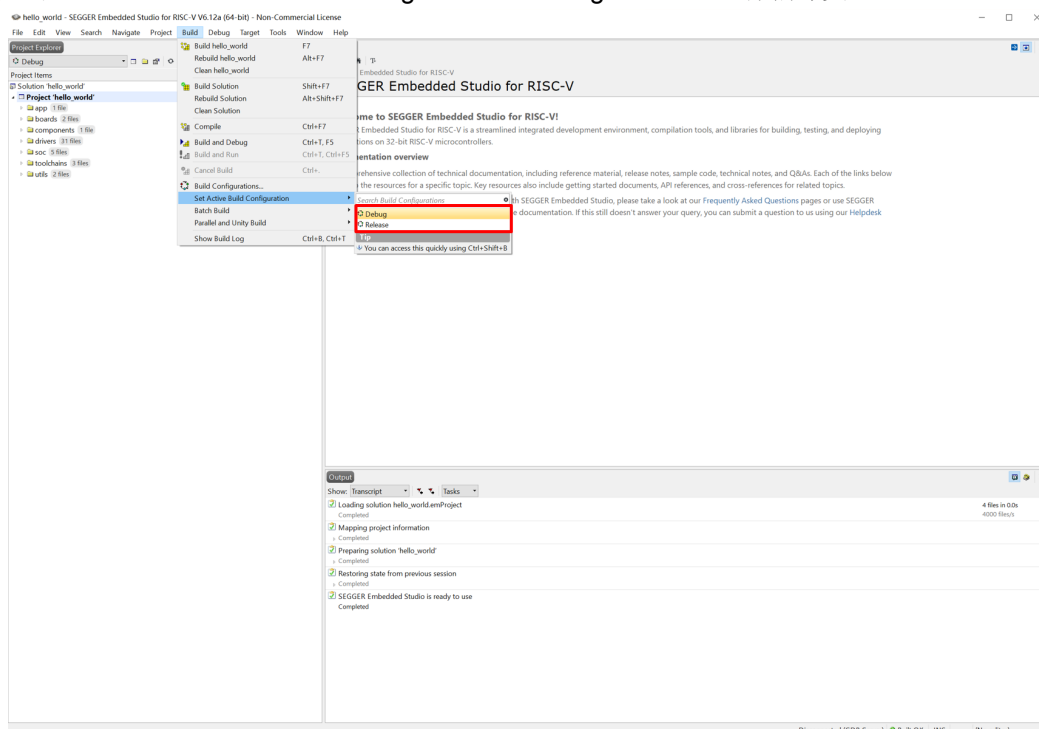


图 1.9: Segger Embedded Studio 选择编译类型

3. 点击 Build > Build project_name 编译工程。

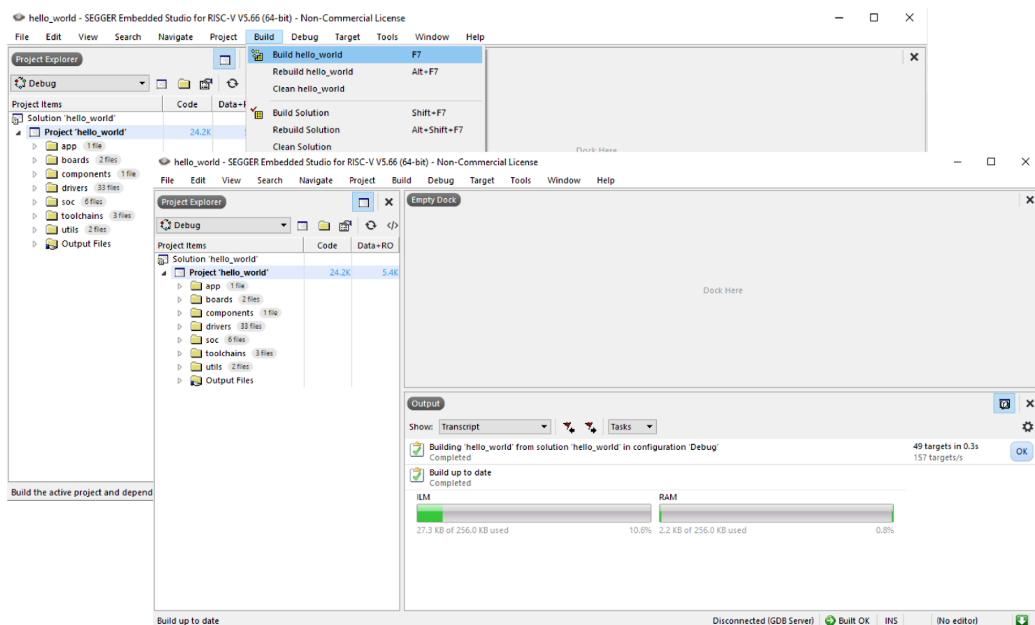


图 1.10: Segger Embedded Studio 编译 hello_world 工程

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

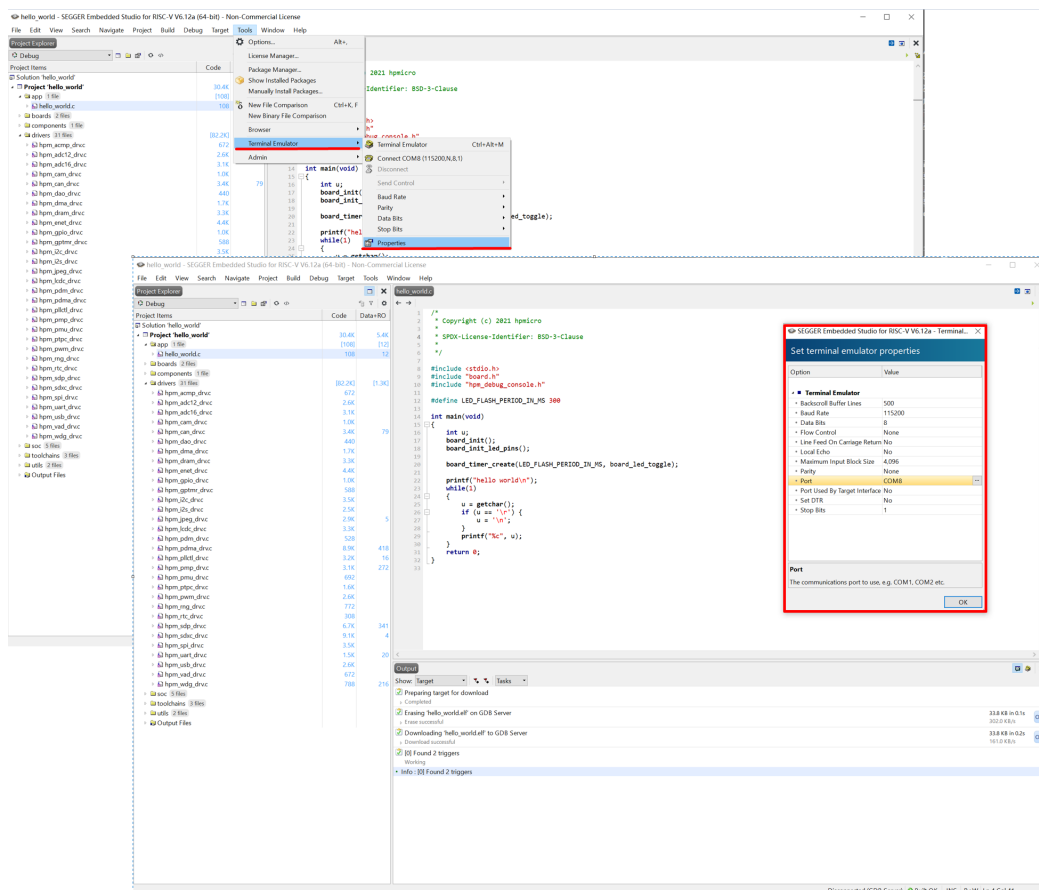


图 1.11: Segger Embedded Studio 配置串口

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

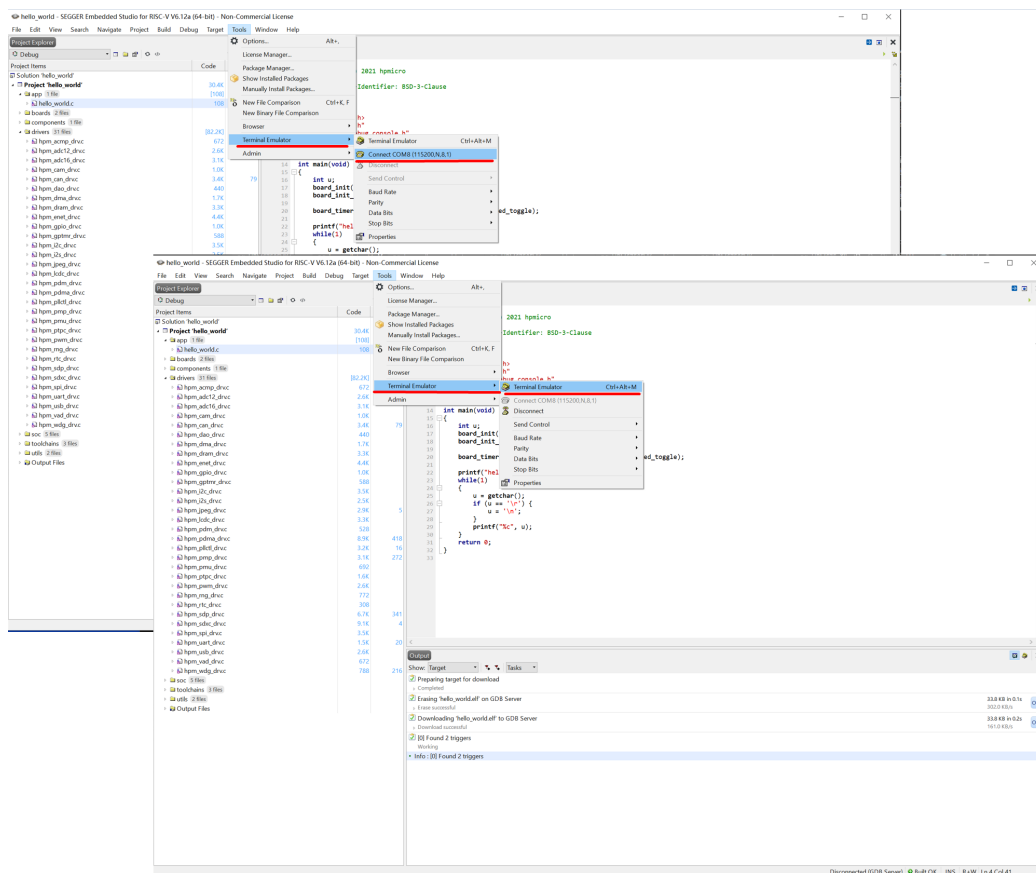


图 1.12: Segger Embedded Studio 打开串口

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

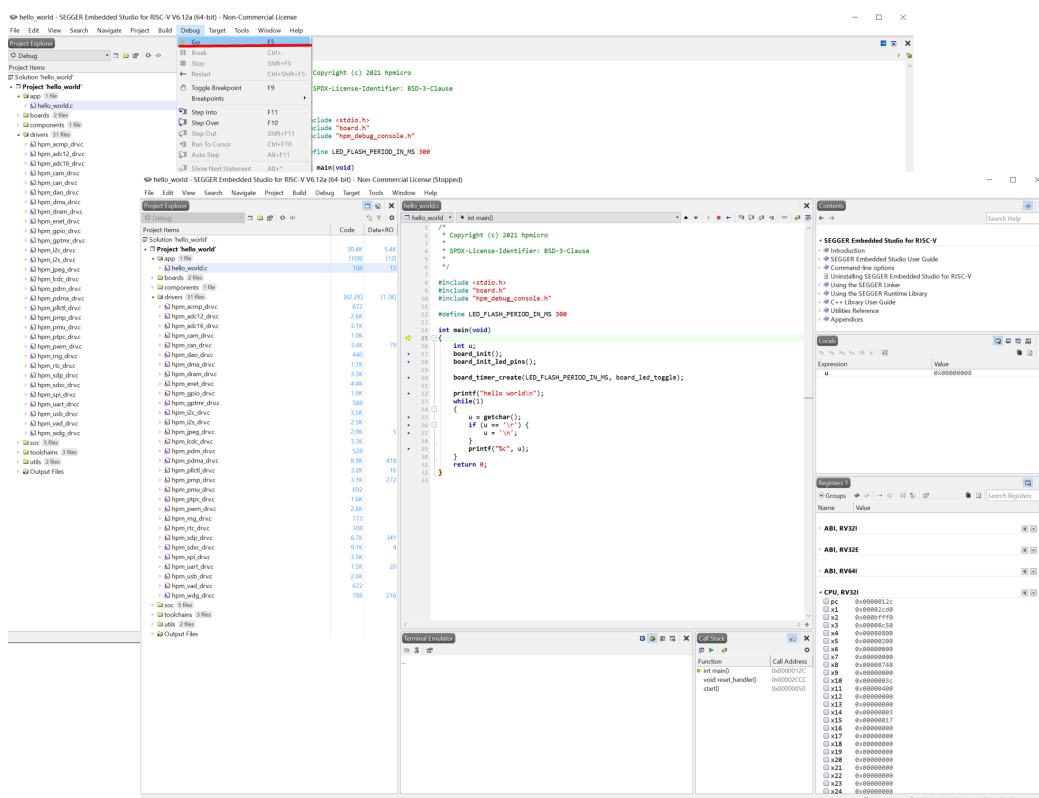


图 1.13: Segger Embedded Studio 运行 hello_world 工程

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

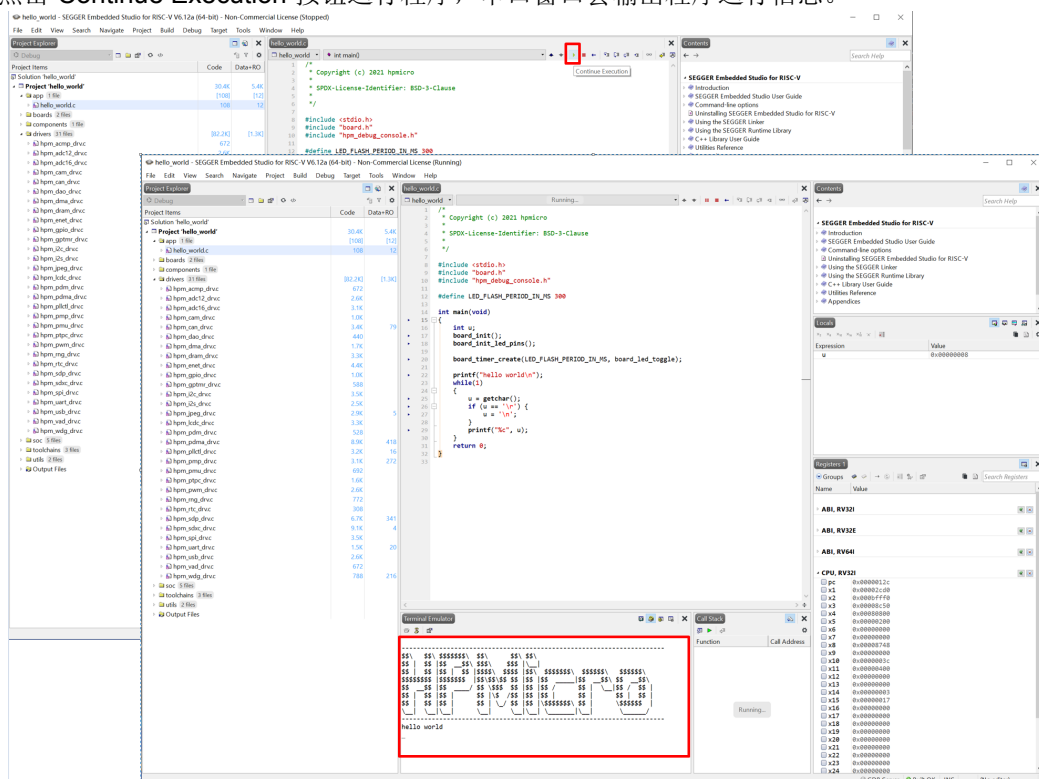


图 1.14: Segger Embedded Studio 运行 hello_world

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。

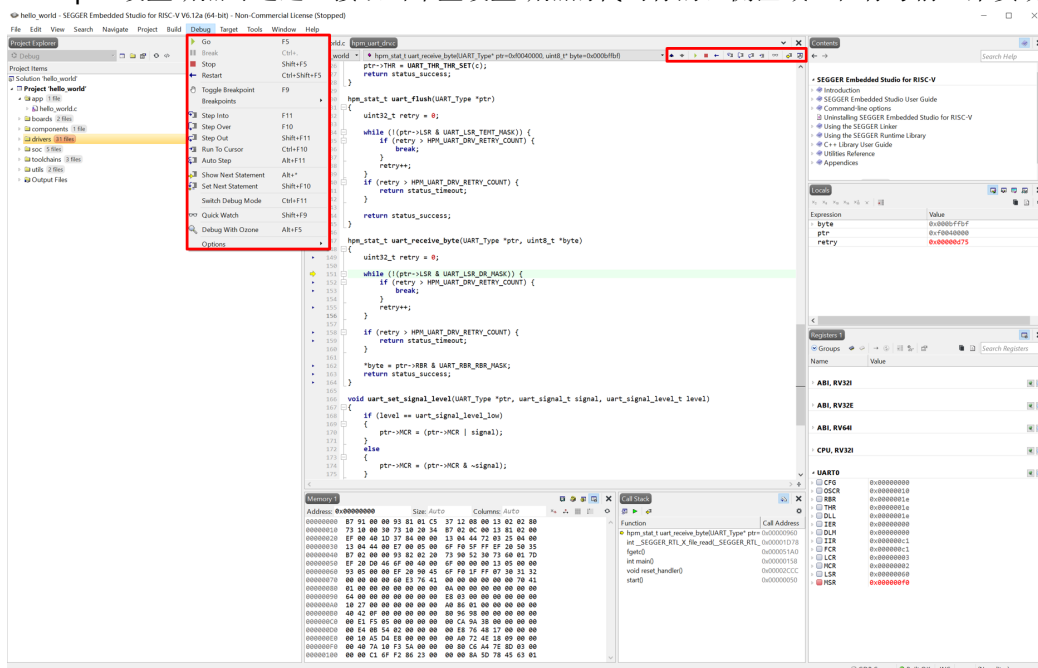


图 1.15: Segger Embedded Studio 程序执行控制

(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。

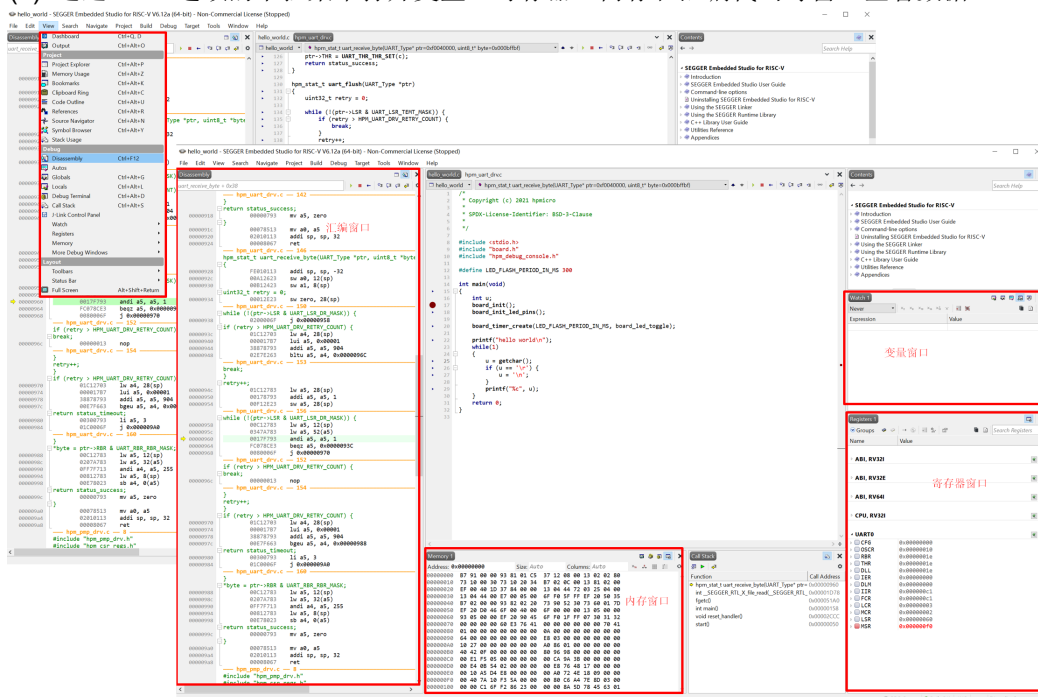


图 1.16: Segger Embedded Studio 数据查看窗口

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。

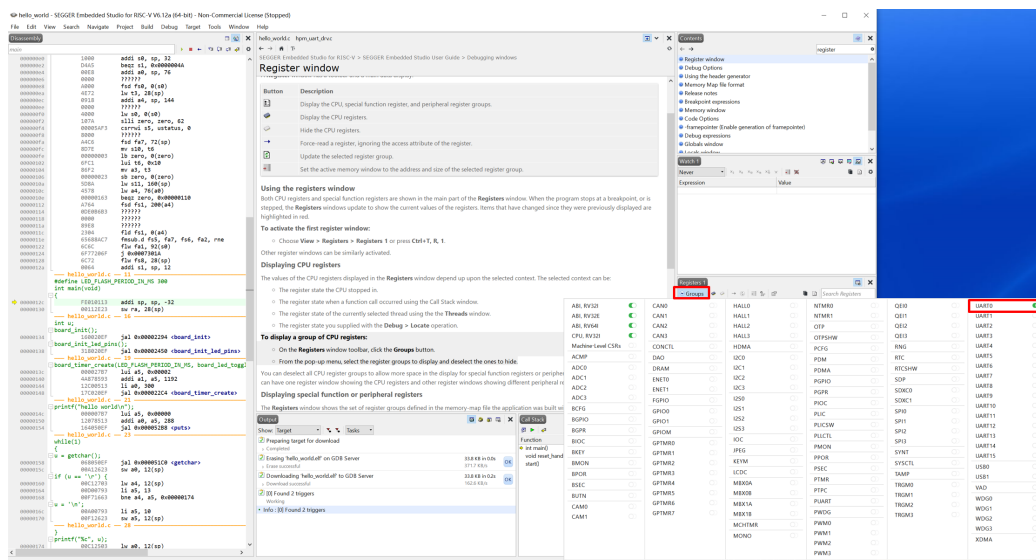


图 1.17: Segger Embedded Studio 选择寄存器组

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

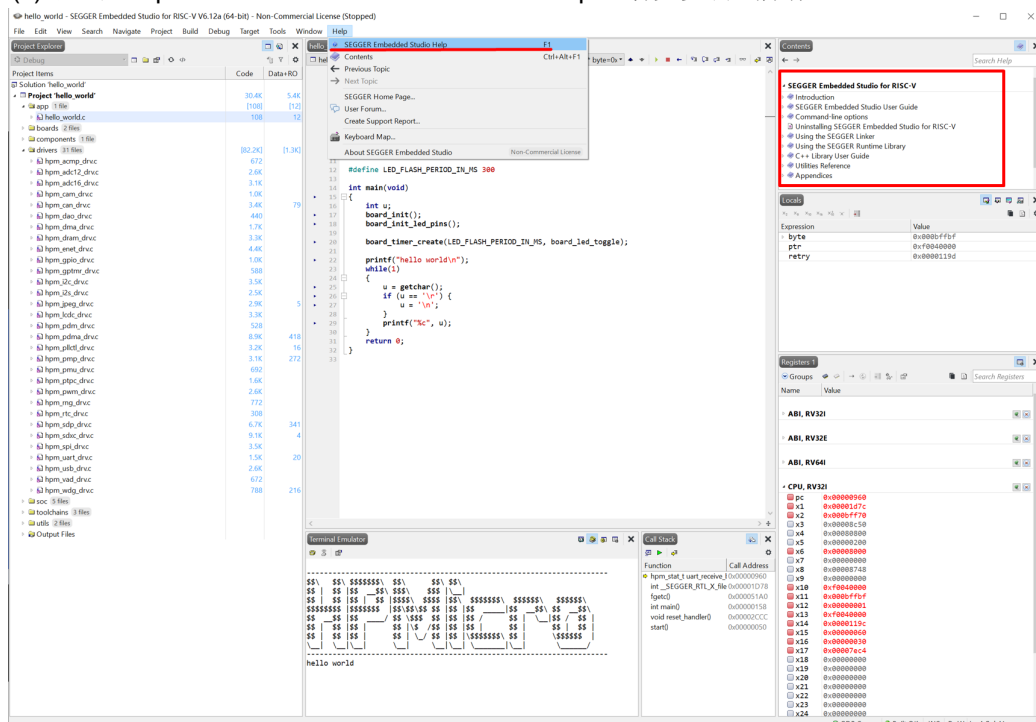


图 1.18: Segger Embedded Studio 帮助

1.5 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。

1.5.1 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。

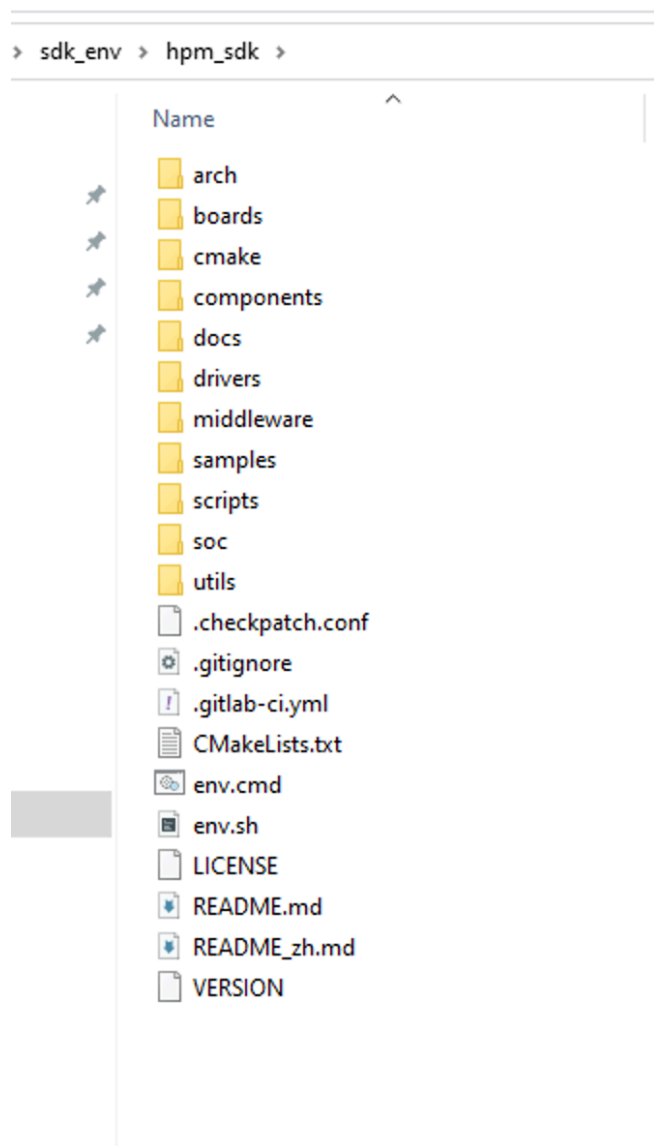


图 1.19: 更新 SDK

1.5.2 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。

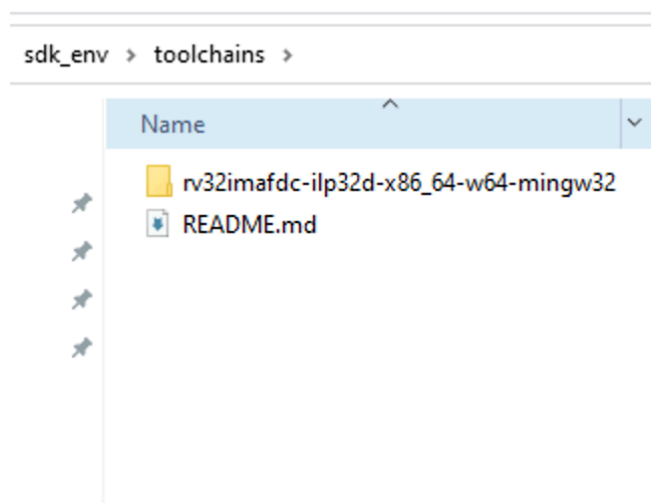


图 1.20: 拷贝 toolchain

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

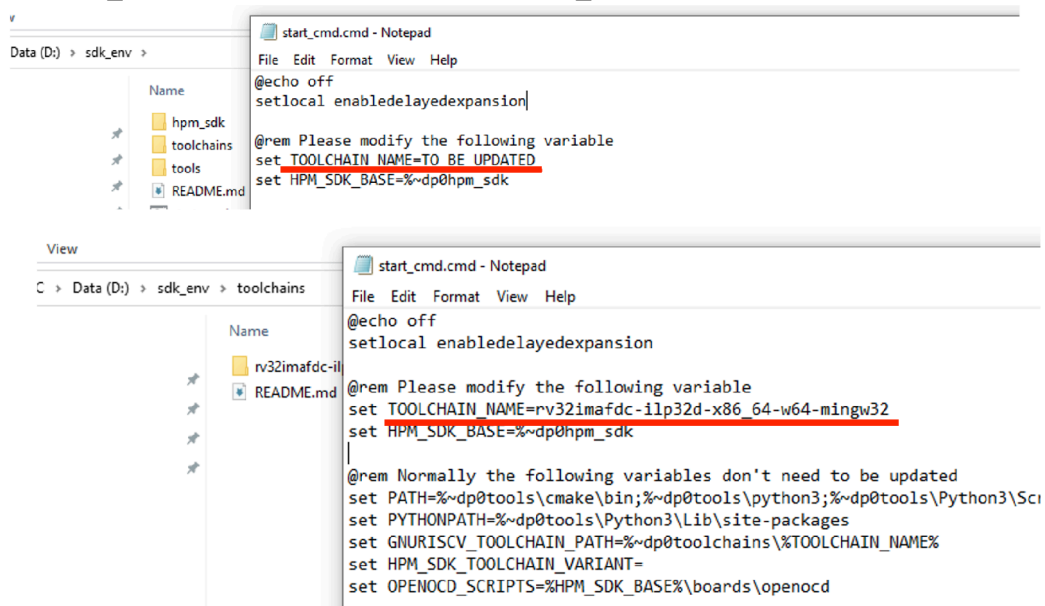


图 1.21: 更新 start_cmd.cmd 中 TOOLCHAIN_NAME

第二章 RT-Thread Studio 快速上手指南

2.1 简介

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

2.2 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)。

2.2.1 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。

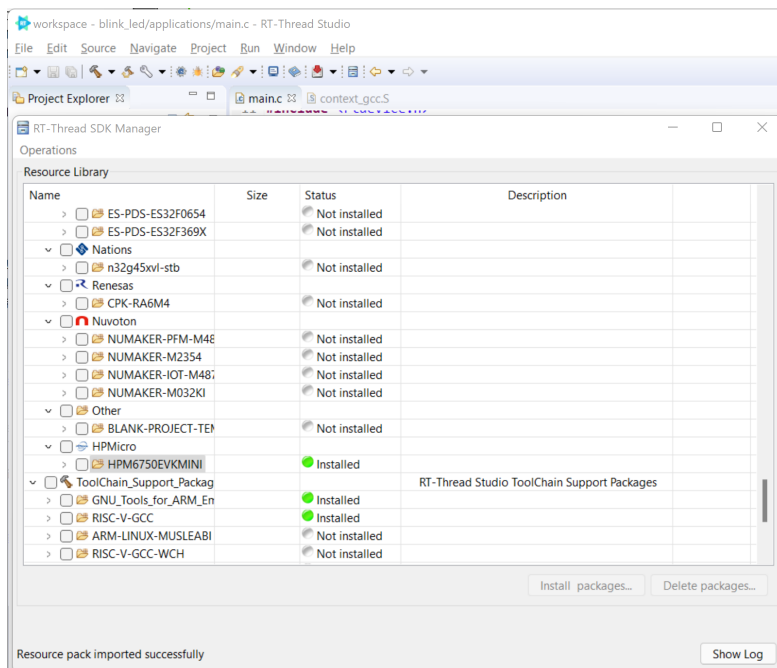


图 2.1: 安装 HPM6750EVKMINI 板级支持包

2.2.2 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。

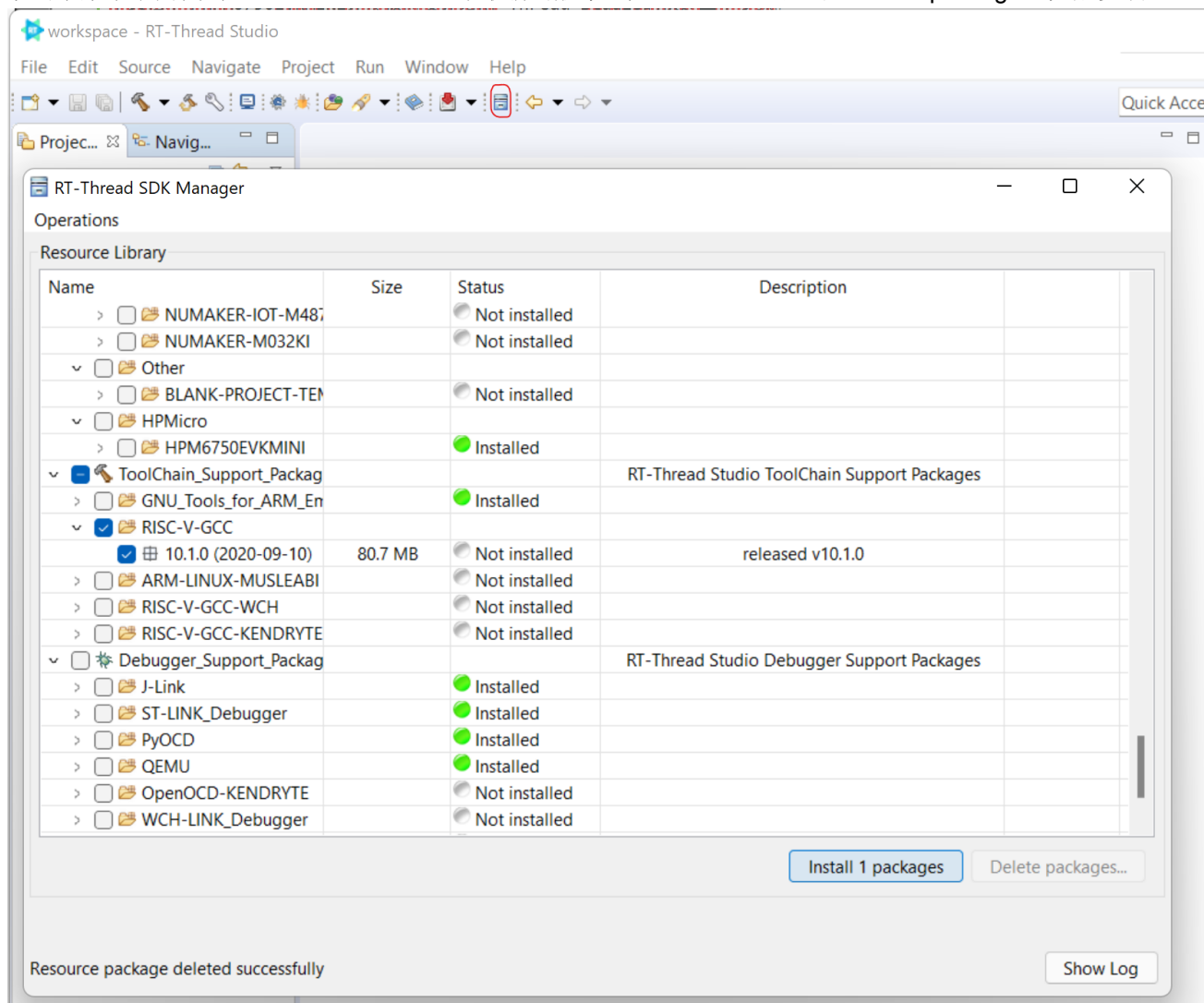


图 2.2: 安装 RISC-V GCC 工具链

2.2.3 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。

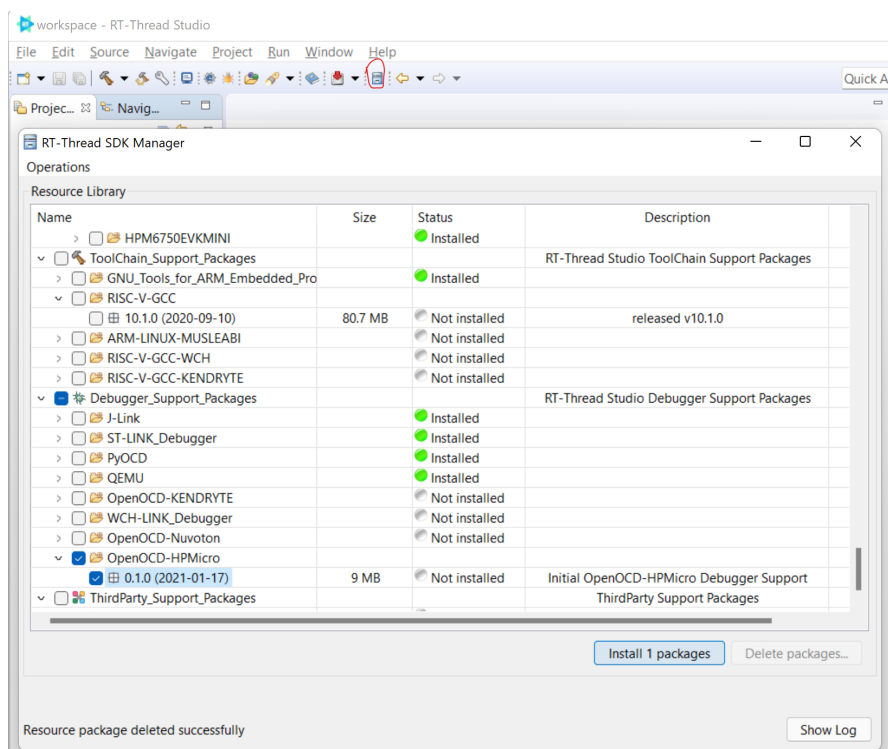


图 2.3: 安装 OpenOCD-HPMicro 调试器支持包

2.3 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。

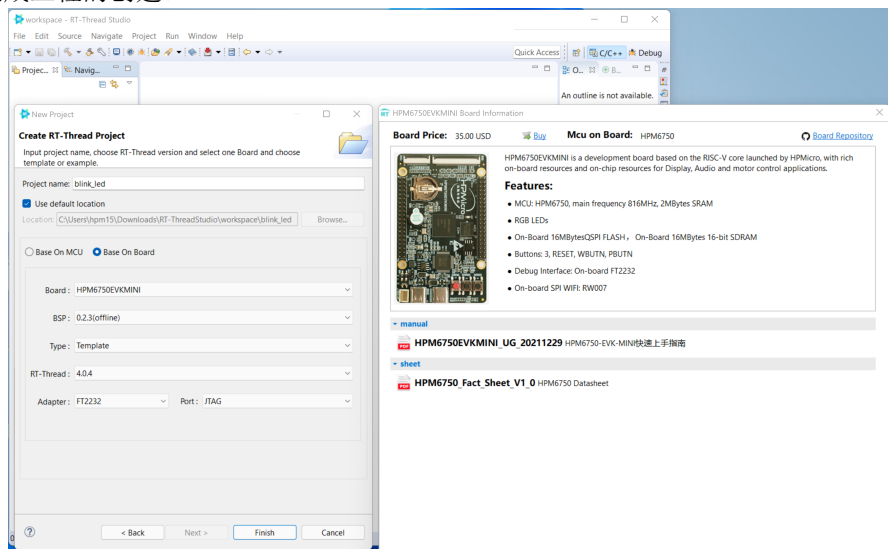


图 2.4: 创建新的 RT-Thread 工程

2.4 编译和调试

2.4.1 编译

点击如下图所示的编译图标，完成 blink_led 工程的编译。

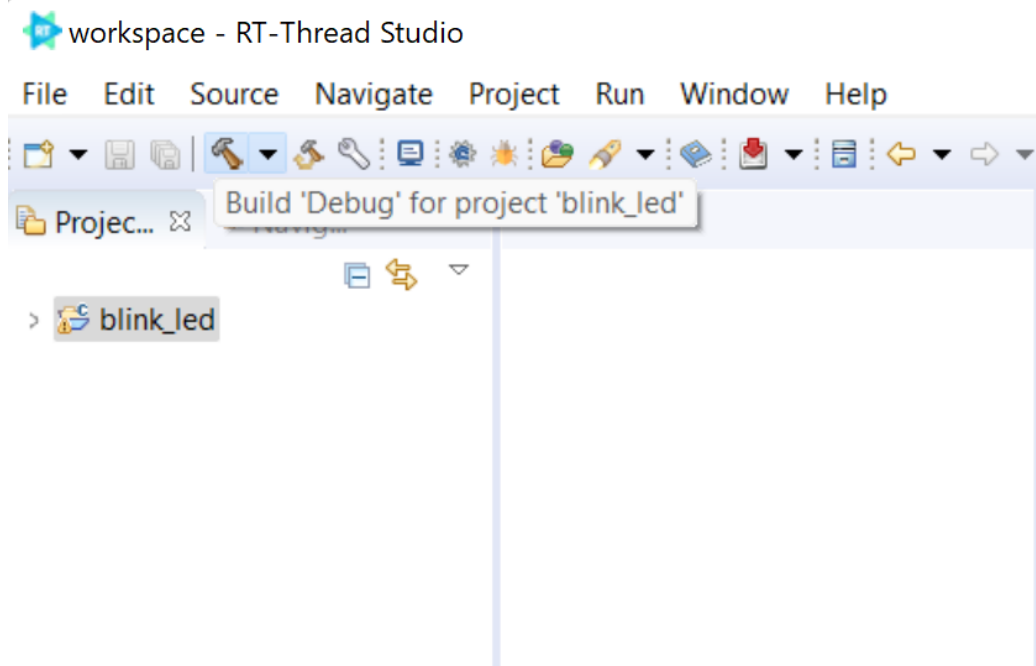


图 2.5: 编译新的 RT-Thread 工程

2.4.2 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 2.6: RT-Thread 启动信息

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

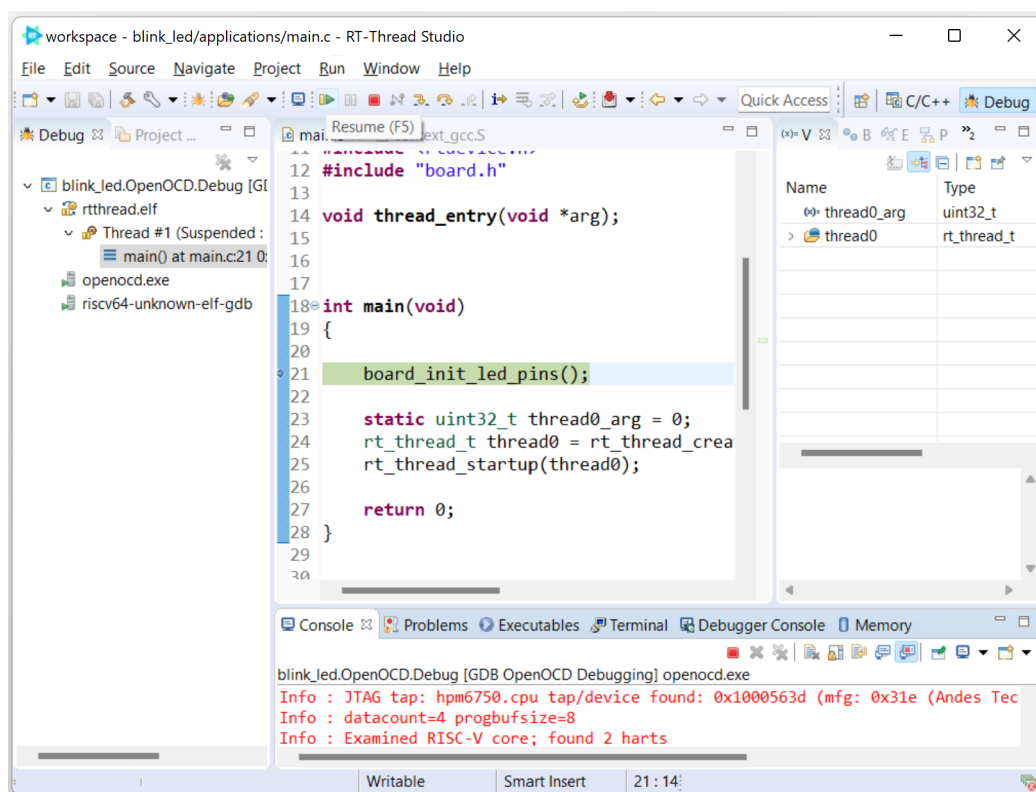


图 2.7: 运行 RT-Thread 工程

第三章 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。

3.1 版本信息

日期	版本	描述
Rev0.1	2022/07/26	初版发布。

表 3.1: 版本信息