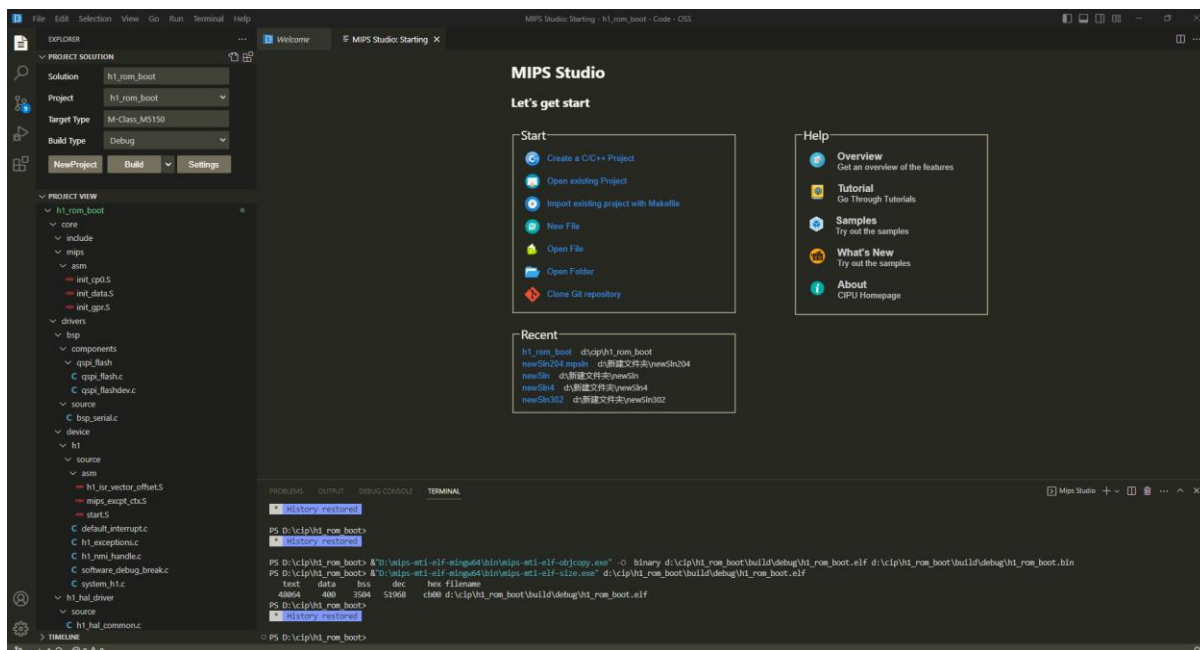


欢迎使用 MIPS Studio

Mips Studio VSCode 插件基本功能概述

Mips Studio 是基于 Microsoft 的 VSCode 上，提供对基于 MIPS MCU/MPU 硬件系统架构的集成开发调试功能。

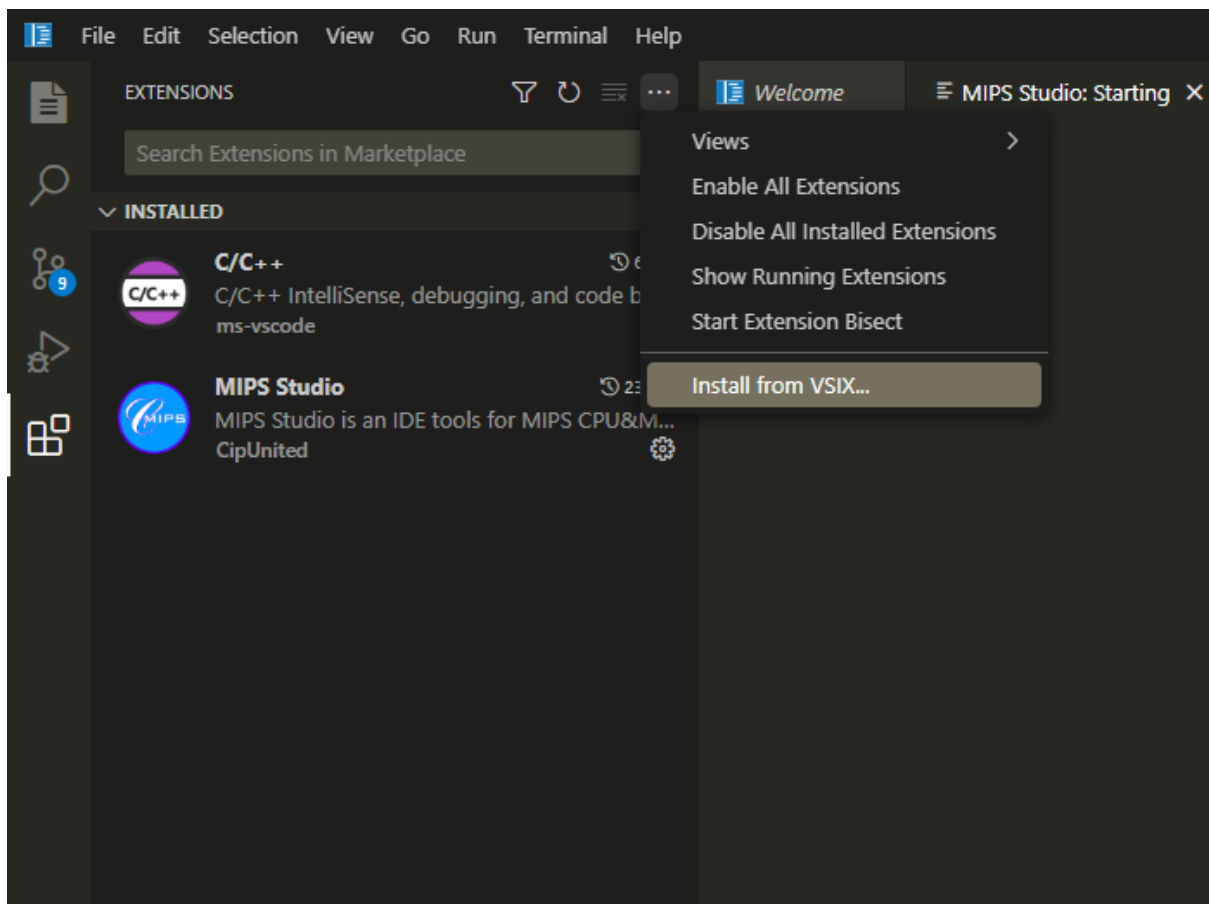
Mips Studio 依赖 Microsoft 的 C/C++ 插件。VSCode 版本要求 1.74.0 以上版本。



安装

目前发布的版本为基于修订后的设计文档重新设计而来的，版本从 0.27 开始。低于该版本的请先卸载。

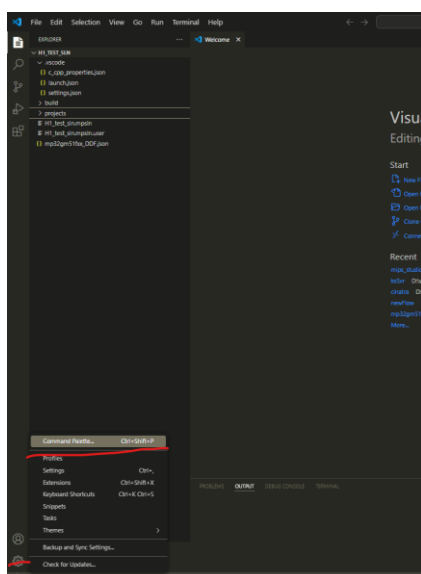
目前已发布到 Microsoft VSCode MarketPlace，建议通过 VSCode 的 Extensions 界面直接搜索后安装。



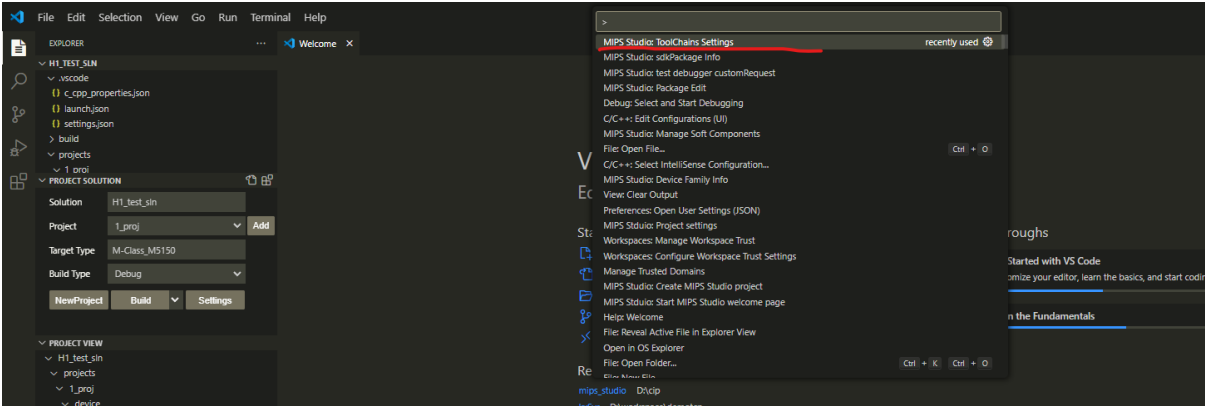
使用

配置工具链

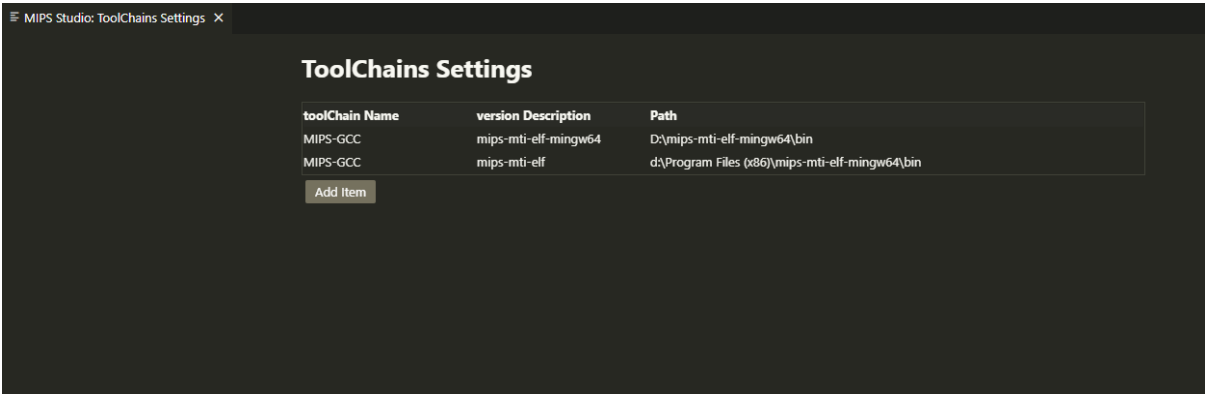
用户在 VSCode 中通过菜单进入 Command Palette..., 如下图所示



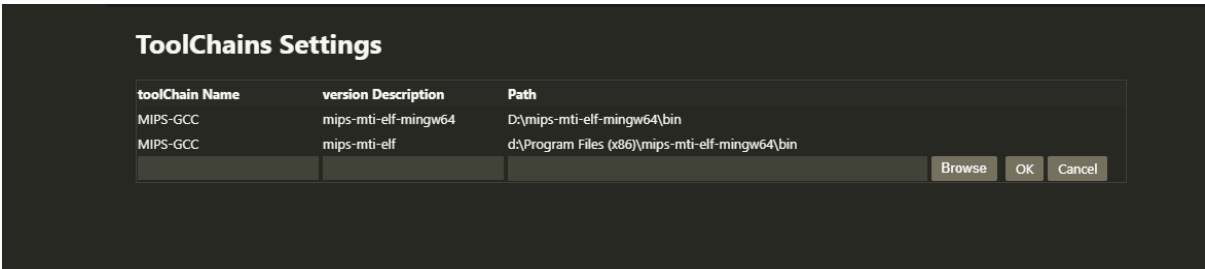
在 Command Palette 出现的输入框内输入 Mips Studio ToolChains Settings（输入的过程中下方选择列表会动态变化，可通过鼠标快速选择）。



进入 ToolChains Settings 界面后，



可通过点击 Add Item 按钮，增加 ToolChains 信息，



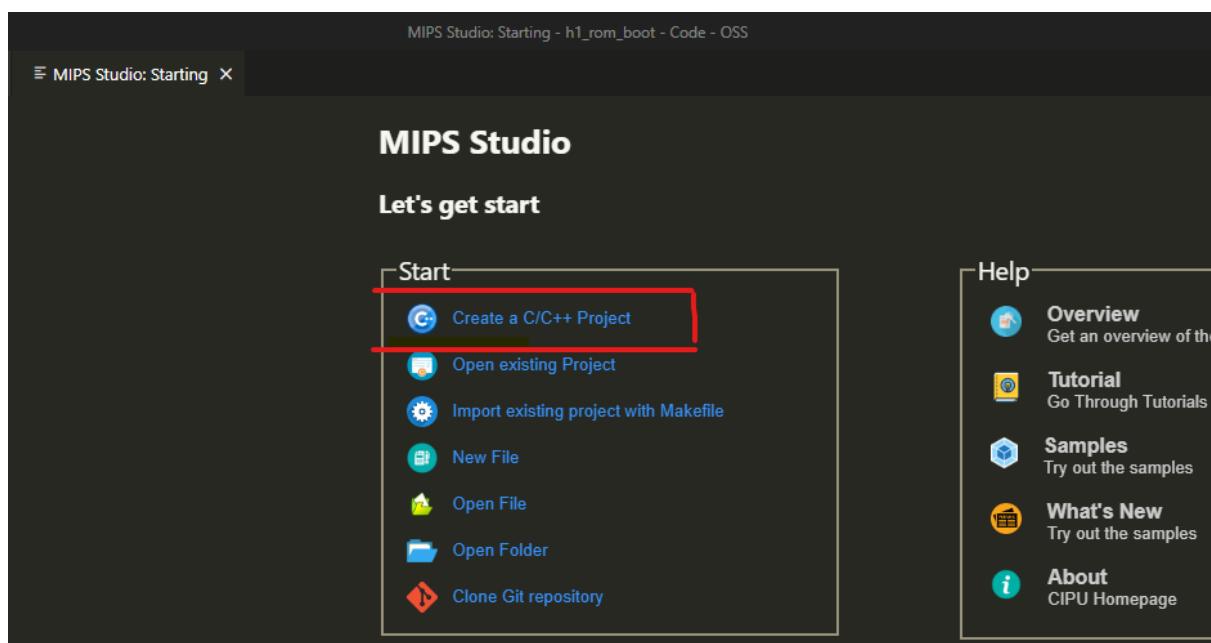
依次输入 Name，version，选择 MIPS GCC 对应的本地路径，点击 OK 即可。

用户可以根据自已的情况，配置多个不同的 MIPS GCC 版本，在不同的项目中可以切换使用。

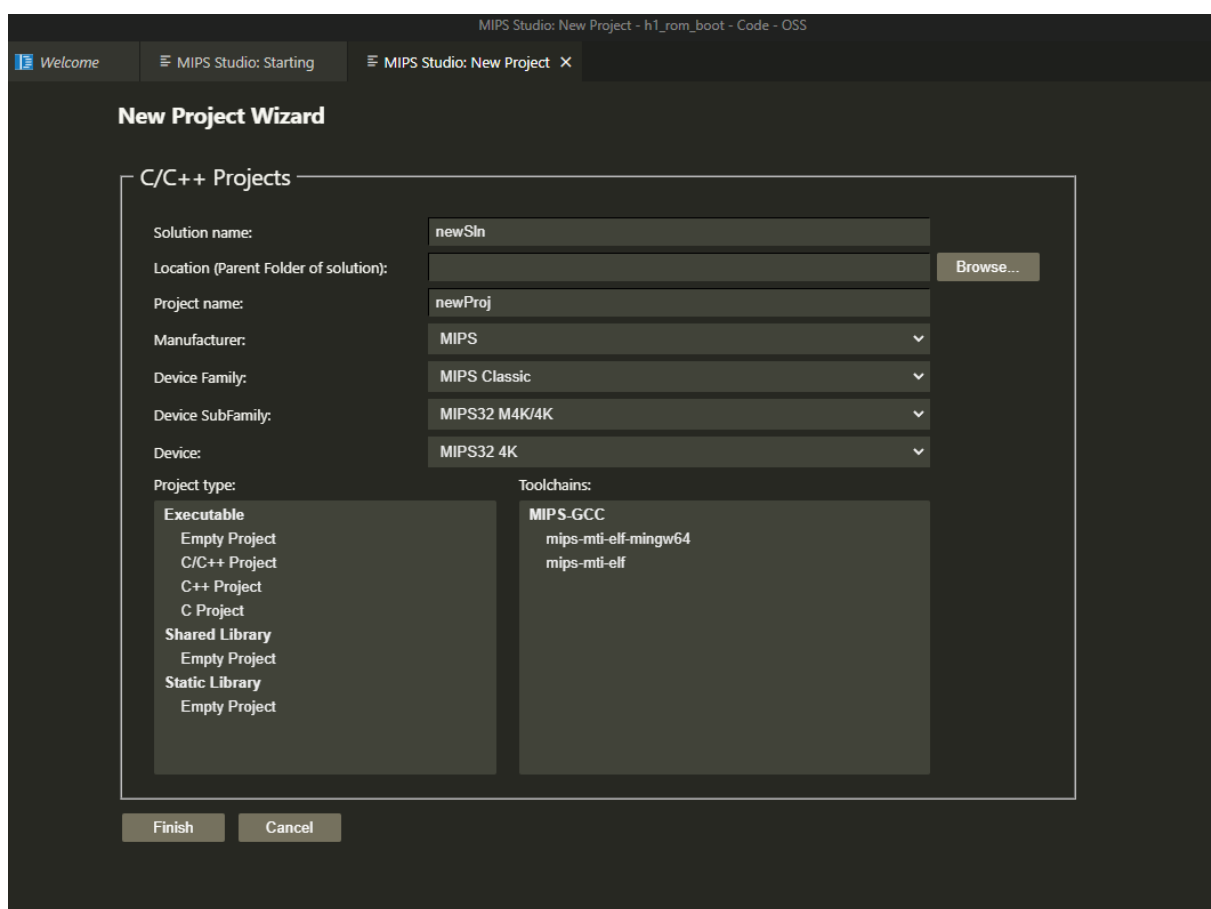
创建新项目

有两种操作来创建新代码项目。

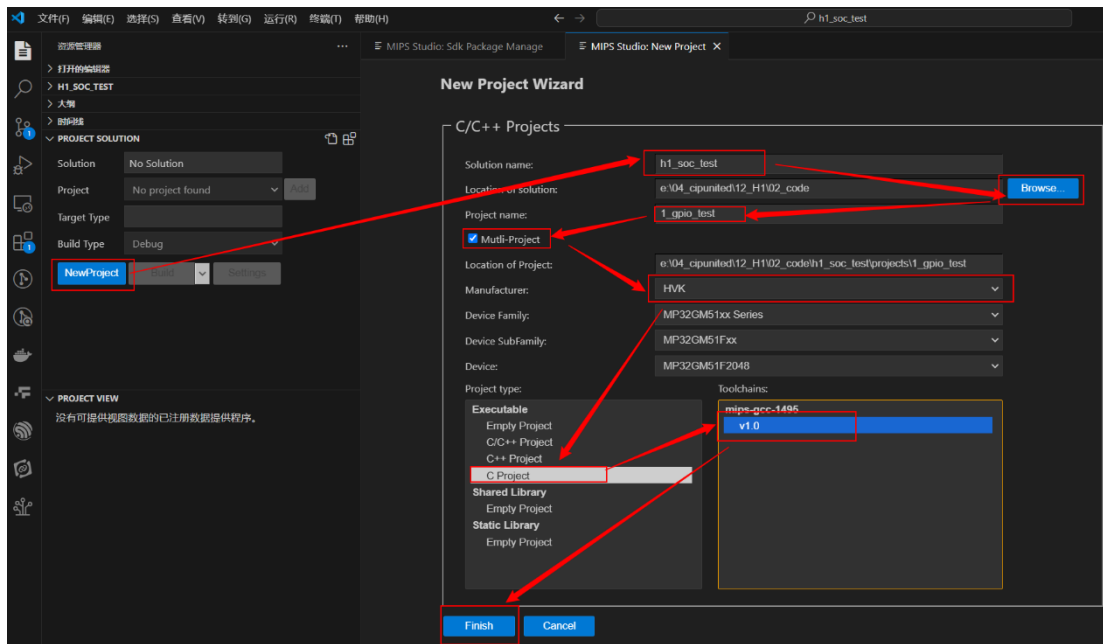
1. 在 Mips Studio 的开始页面点击 “Create Project”即可开始创建新项目。



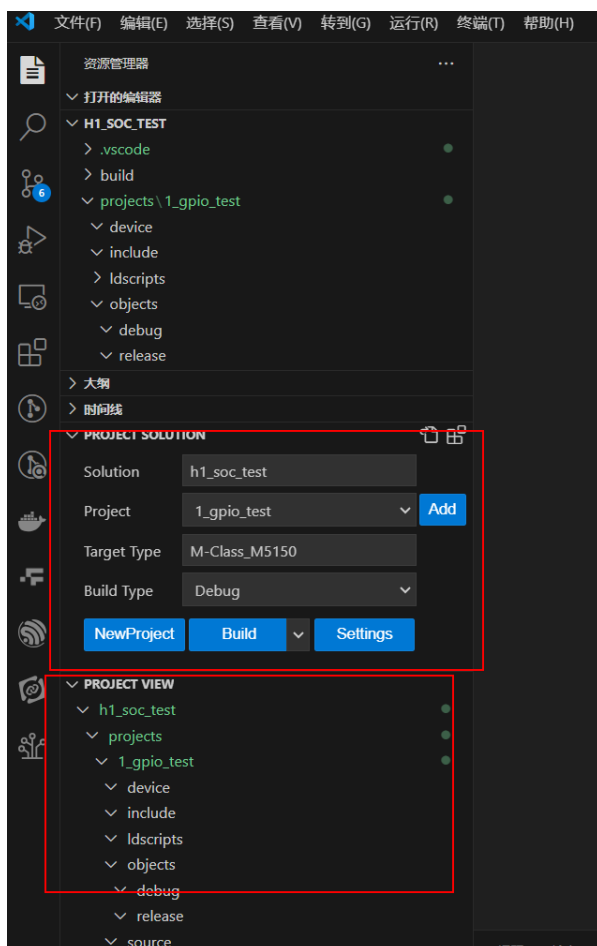
项目创建页面中完成必要的信息输入和选择后即可完成项目创建。



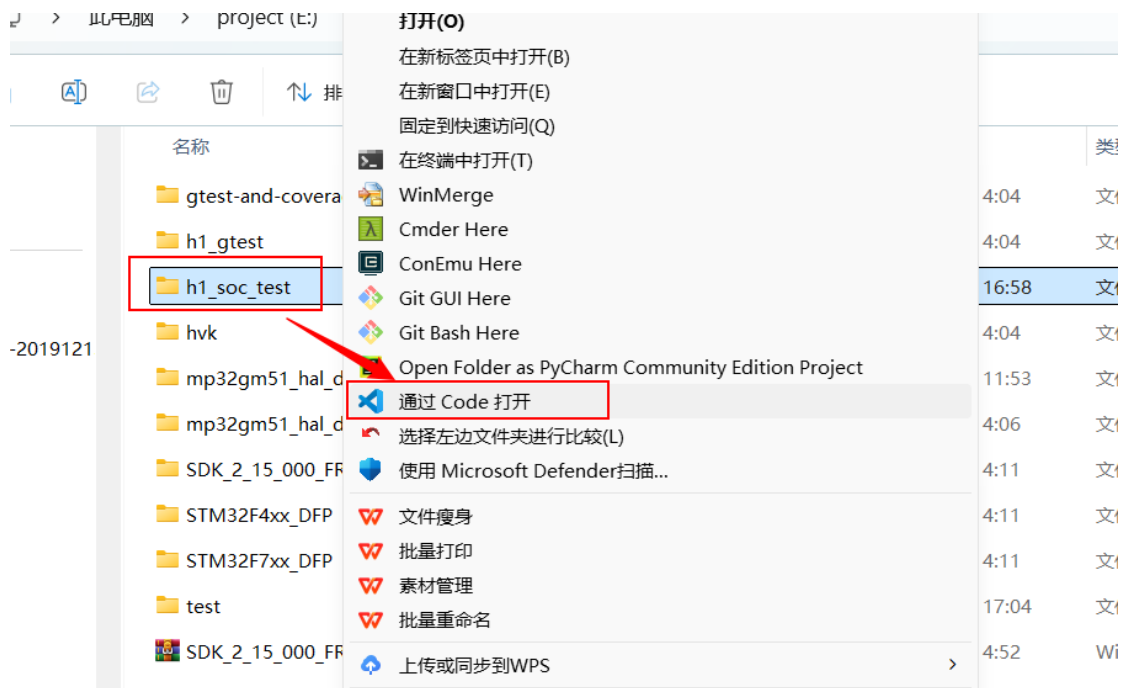
2. 在 Solution View 中点击 New Project 来创建。



建立完成后“PROJECT VIEW”和“PROJECT SOLUTION”会有相应的工程

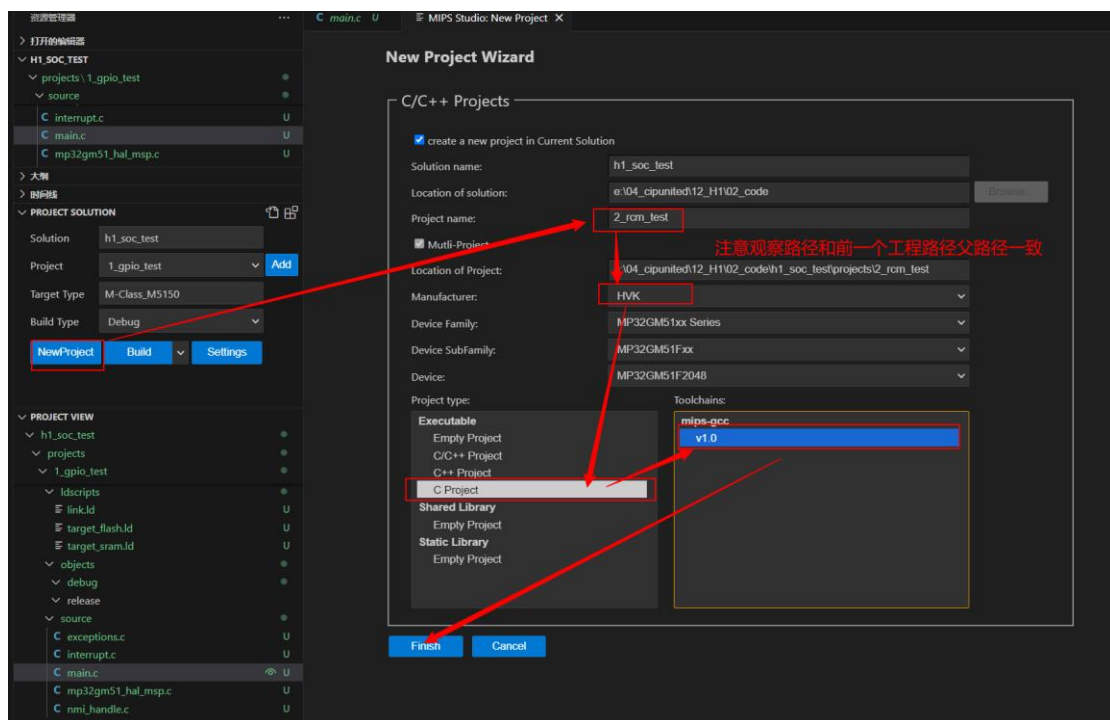


如果没有正常显示，可以重新打开 VSCode，加载新建的 Solution 的文件夹。如下图

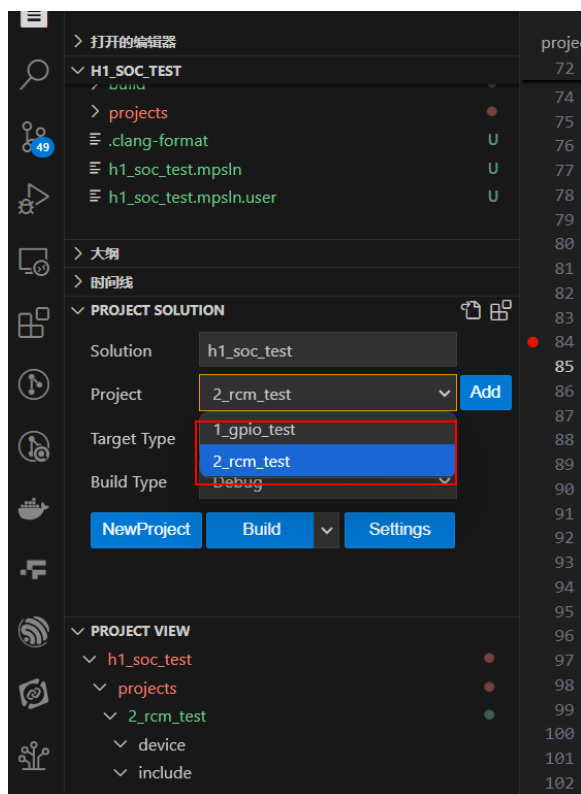


已有 Solution 增加 Project

在现有的 Solution 中增加一个新的 Project，如下图所示，勾选上 Create a new project in Current Solution，则创建的新 Project 添加到当前的 Solution 中。



在 Solution View 中可以看到两个工程，可以在工程之间进行切换（如下图所示）。

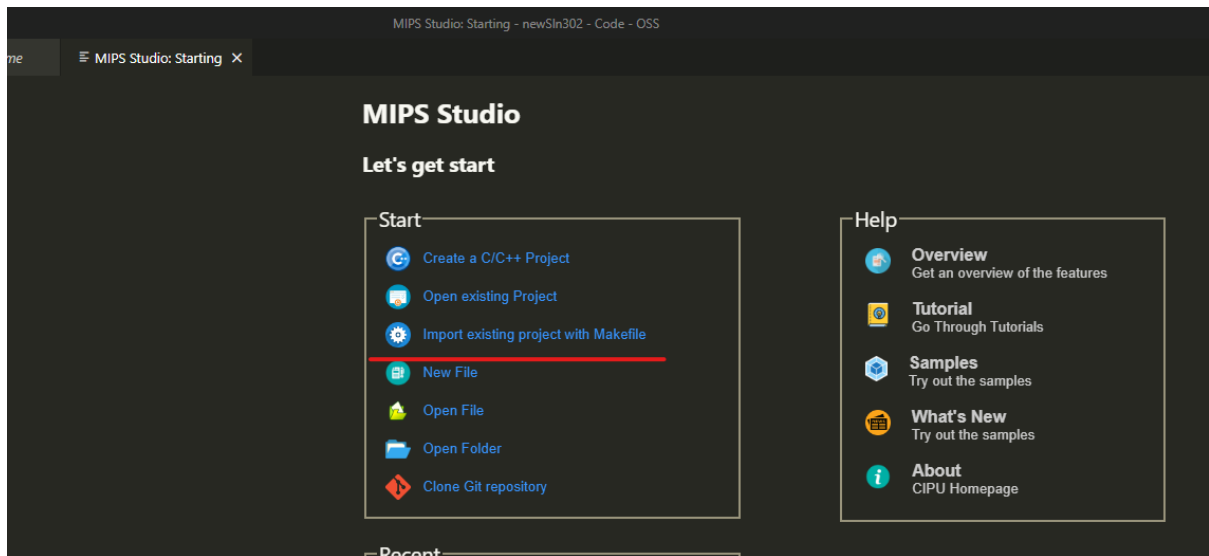


导入项目

目前仅提供对特定格式 makefile 的导入功能，本功能是为方便之前的代码项目导入使用。

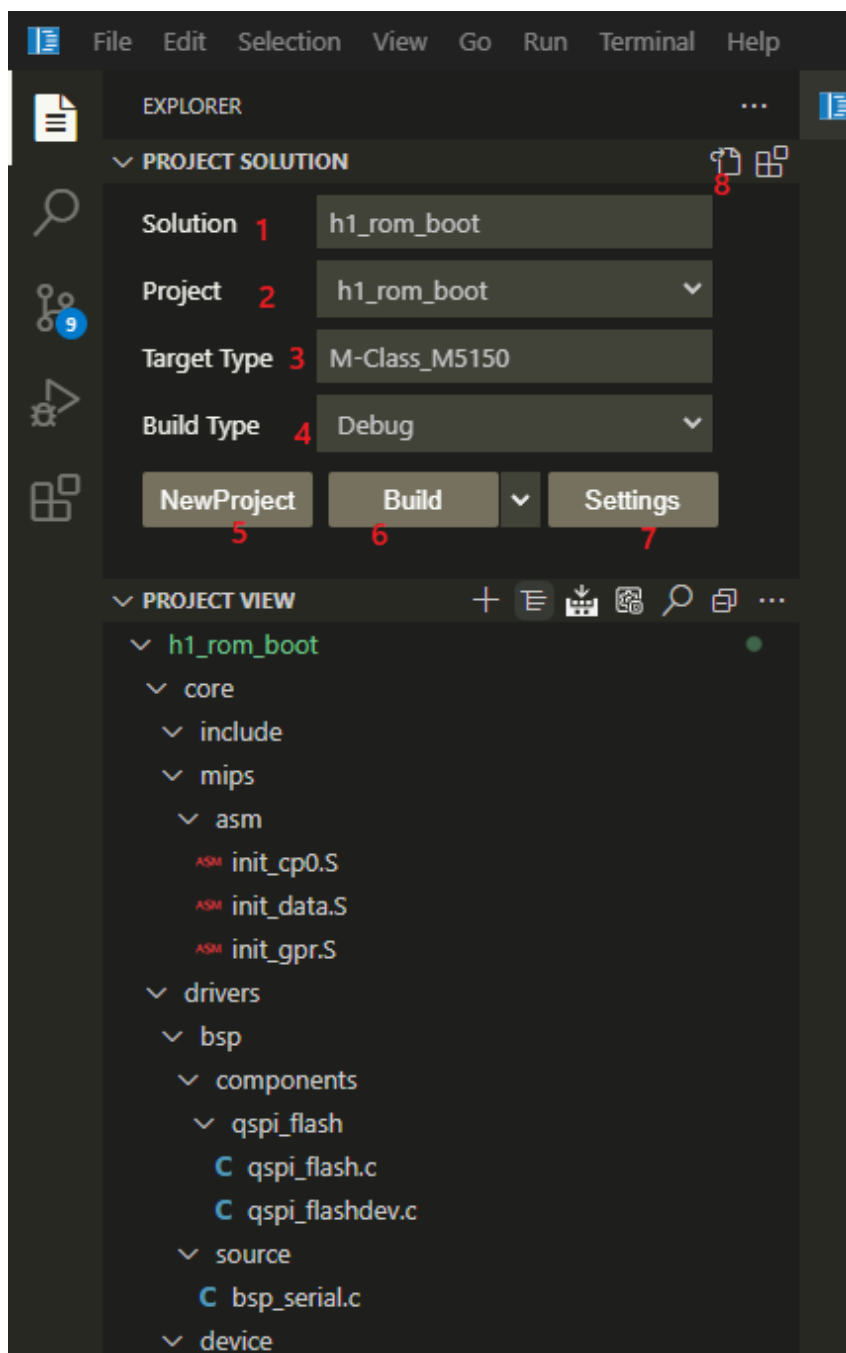
采用如下形式编译变量名称的 makefile。

```
INC_PATH
INCLUDES
CSOURCES
C_SRCS
ASOURCES
ASM_SRCS
LD_PATH
CFLAGS
```



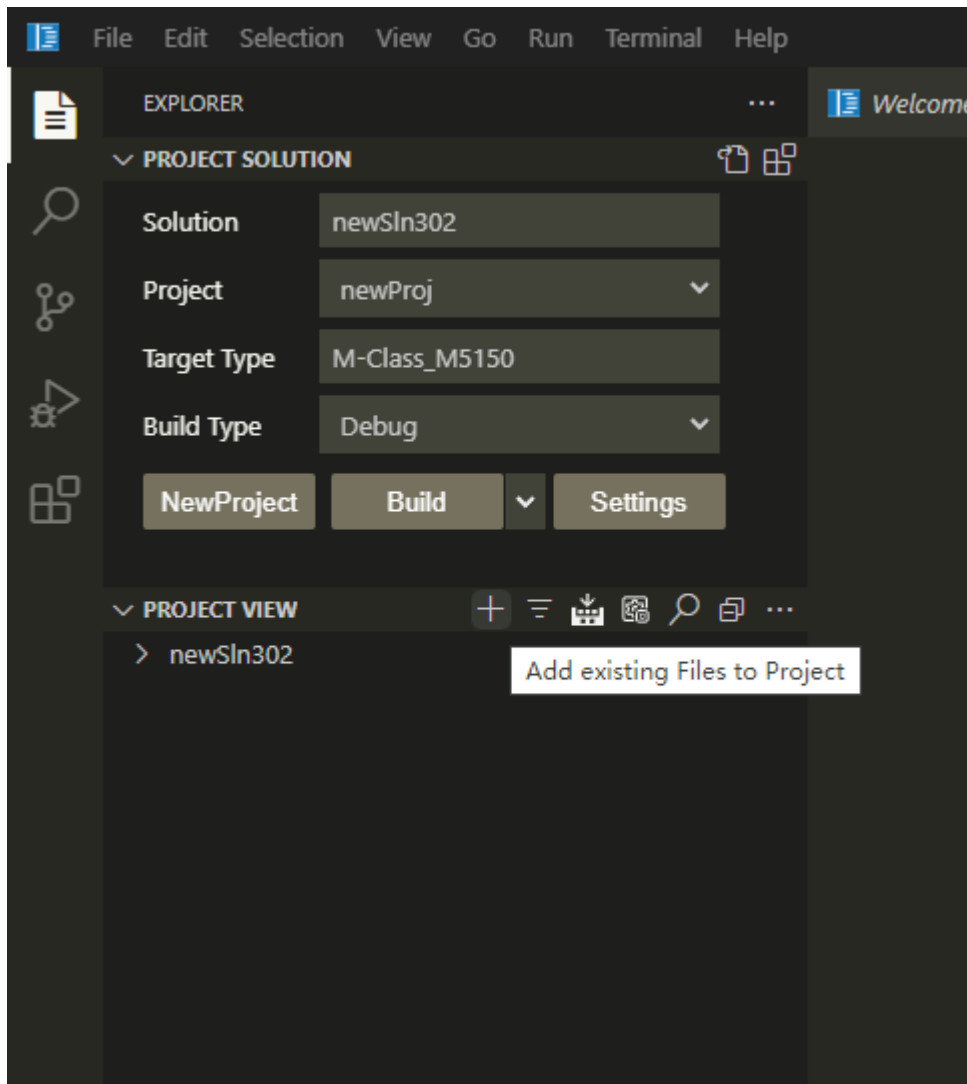
导入后的项目即可使用 MIPS Studio 进行项目管理和编译生成操作。

项目管理



序号	功能项	描述
1	Solution	显示多项目管理中解决方案名，解决方案名与解决方案目录名称应相同。只有一个 Project 的 Solution，Project 的名称可以与 Solution 名称相同。没有解决方案打开时显示 <i>"No Solution"</i> 。

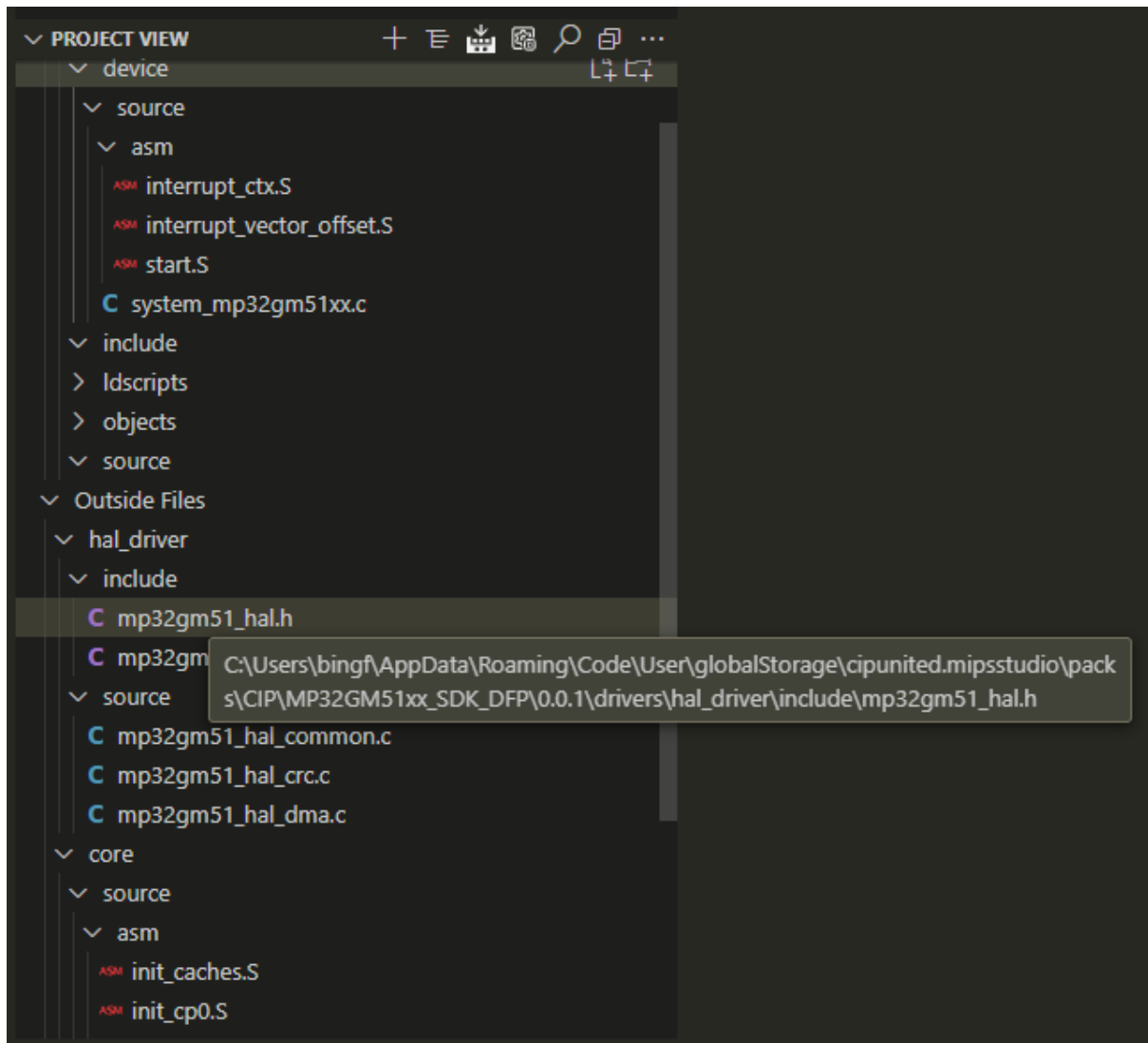
2	Project	显示当前激活的项目名称，点击下拉按钮可以切换到其它项目，当没有项目打开时显示“ <i>No project found</i> ”。
3	Target Type	显示当前项目中的目标处理器，目标处理器与“ <i>Target Processor</i> ”选项中选择当前“ <i>Target Processor</i> ”一致，当没有项目打开时此处空白。
4	Build Type	用于选择当前项目输出的结果文件，此处定义了两种类型：“ <i>Debug</i> ”和“ <i>Release</i> ”，当没有项目打开时此处空白。
5	New Project	单击按钮，当鼠标点击时打开“ <i>New Project</i> ”页面，用于新建一个项目。
6	Build	单击按钮，当鼠标点击时开始项目构建，并输出相应的文件和信息。build 按钮还可以“ <i>Clean Build</i> ”、“ <i>Only Compile</i> ”等
7	Settings	单击按钮，当鼠标点击时打开“ <i>ProjectSettings</i> ”页面，用于项目配置
8	Open MIPS Start	单击按钮，放到工具栏，当鼠标点击时打开“ <i>Start</i> ”页面。



在 ProjectView 的菜单项有如下几个功能项

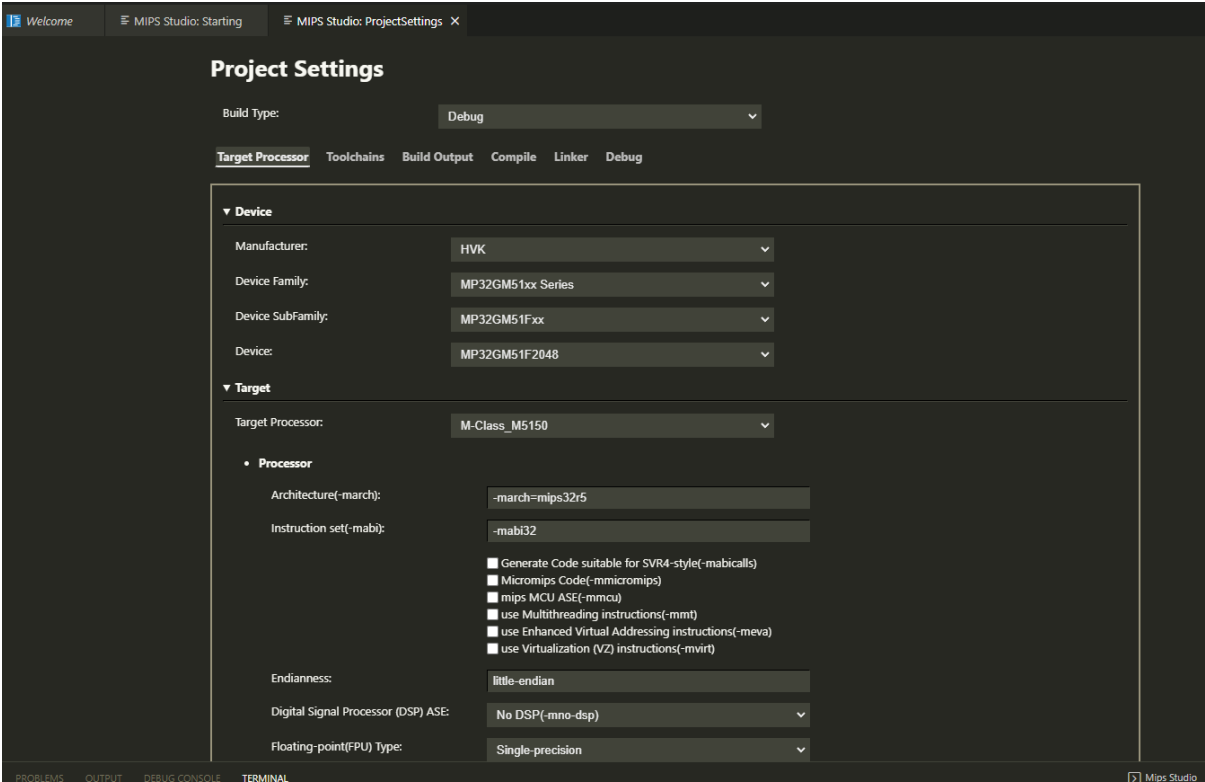
Add Existing File	单击按钮，放到“ <i>Project View</i> ”工具栏，当鼠标点击时打开系统 explorer，选择要添加的文件
Build Project	单击按钮，放到“ <i>Project View</i> ”工具栏，单击时直接构建当前项目，功能与“ <i>Project solution -> Build</i> ”相同。
Search in Project	单击按钮，将在当前项目文件中进行查找

对于由 SDK 软件组件导入等情况加入到当前项目，但文件路径在当前 solution 路径之外的，在 ProjectView 中以 Outside Files 的形式显示，如下图所示。



在该处的菜单项中，export makefile 以及 flash download 功能尚未完善，请知悉。

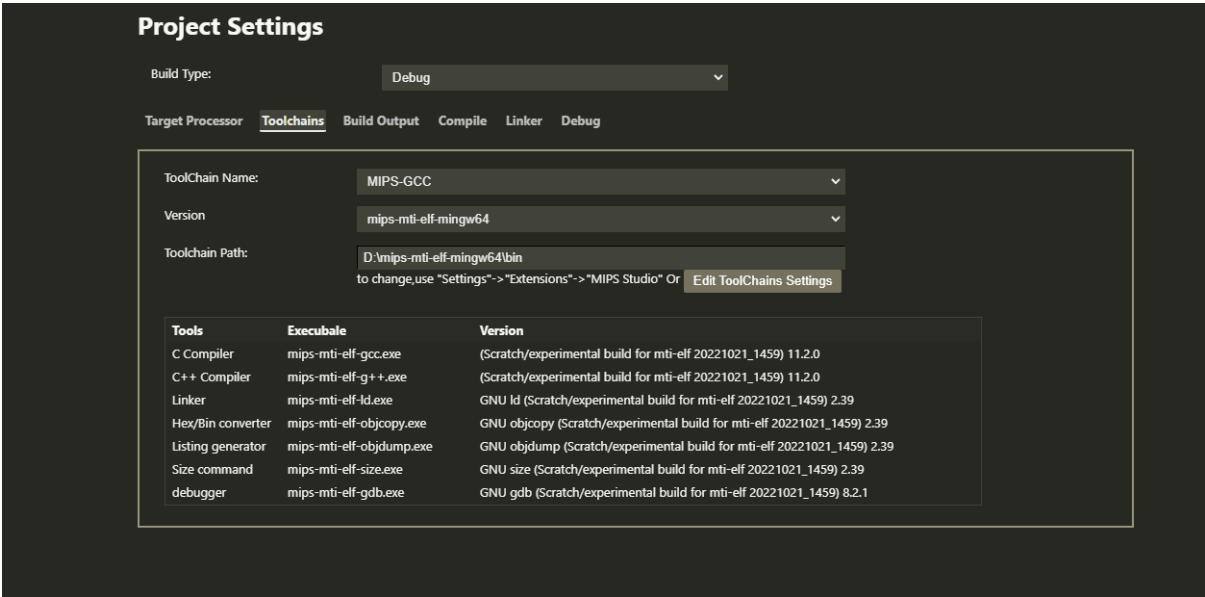
项目选项设置



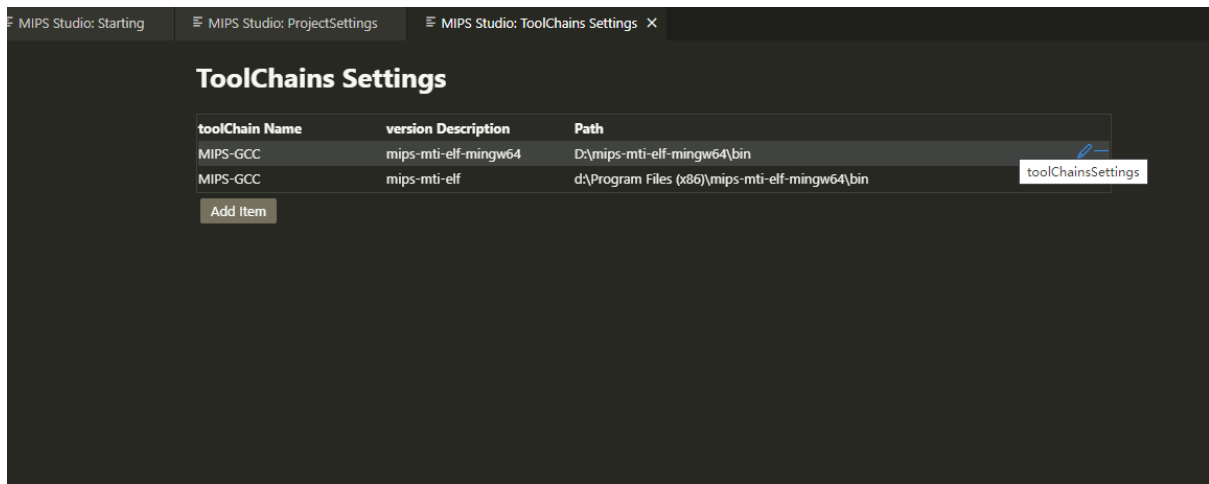
分别对项目的目标器件、选用工具链、输出、编译选项、链接选项以及 debug 选项进行设置。

编译生成

在项目选项设置中 ToolChains 选项页面中可以对编译工具链进行设置。

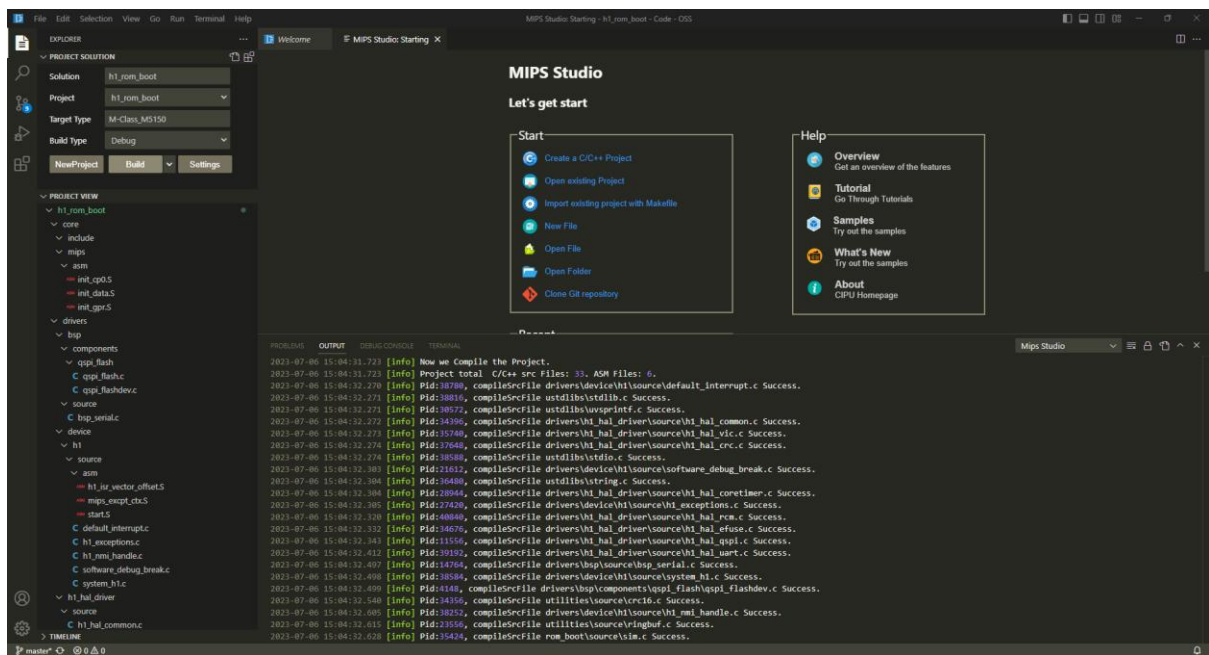


初始状态下，需要点击“Edit ToolChains Settings”进行配置



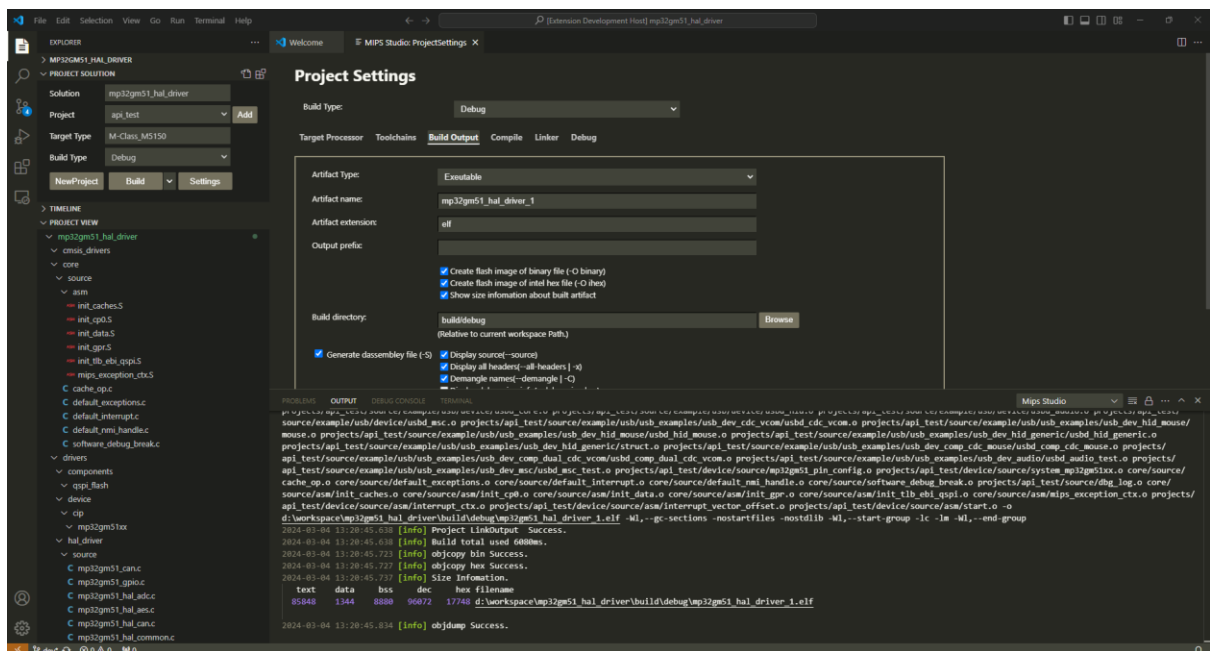
通过点击 Solution View 上的 Build 按钮，即可完成项目的编译生成，

【Important】 在 0.65 版本后，为提高编译生成的速度，项目在 VSCode 打开后仅首次 build 做全量编译，后续仅对变动的文件做编译，当用户修改了项目编译选项配置后，会再次做全量编译。



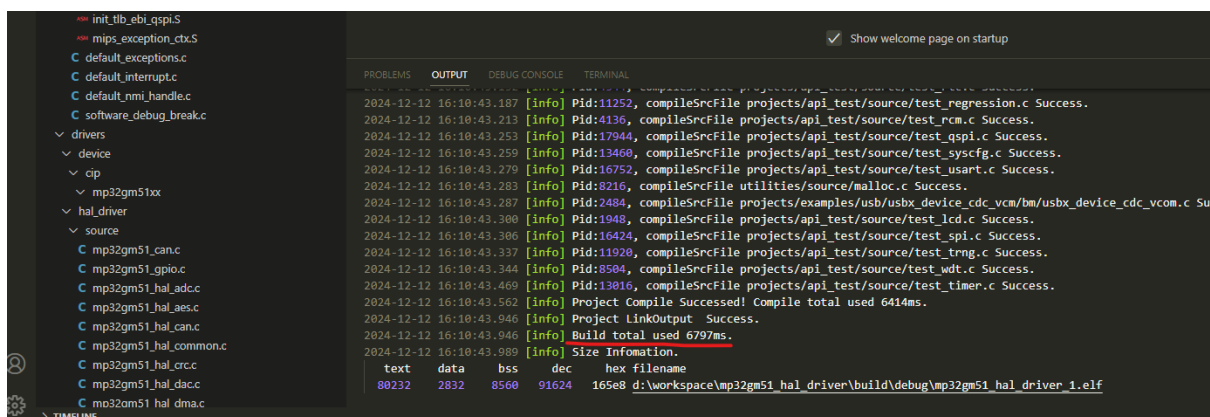
编译过程结果在 Output 窗口中输出显示，

对于 BuildOut 选项中设置了同步生成 bin/hex，显示 size 等选项，将在 Output 窗口进行信息显示。



【Important】因为采用了多进程编译，当项目文件较多时，会同时启动多个 gcc 来进行编译，如电脑上安装有防病毒软件或其他占用 CPU 资源较多的软件时，编译时电脑的 CPU 可能会占用率居高不下，如无法关闭其他软件，请耐心等待编译完成，或者手动关闭其他消耗 CPU 的软件后再编译。举例说明，AMD R7 5700U 16G 内存设备，207 个.c/.S 源文件，开启微软的电脑管家时，未开启其他软件，编译耗时 23~30s，CPU 平均占用 90%，关闭电脑管家时（关闭进程 MSPCManager），编译耗时降到 7.5s。

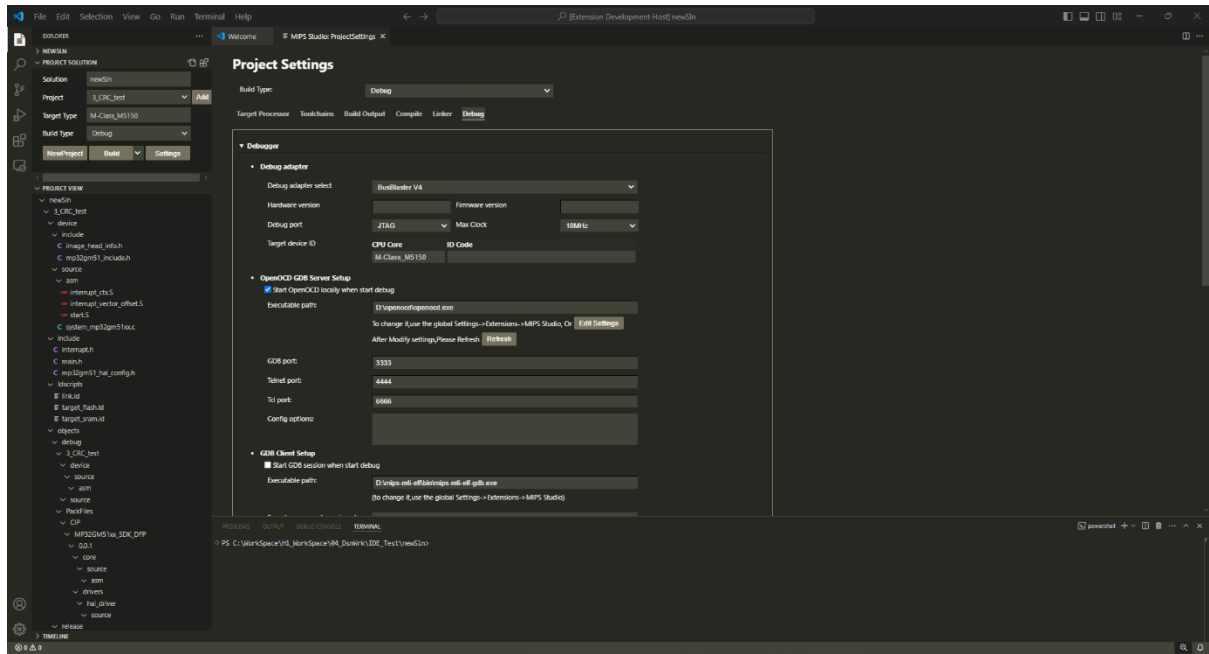
编译完成后，会对编译的.c/.S 做头文件依赖检查并生成缓存，根据用户修改的文件情况，做增量编译，从而加快大型项目的编译速度。在生成缓存的过程中，可能也会大量占用系统 CPU 资源，这个时间一般小于编译时间。编译耗时如下图所示提示。



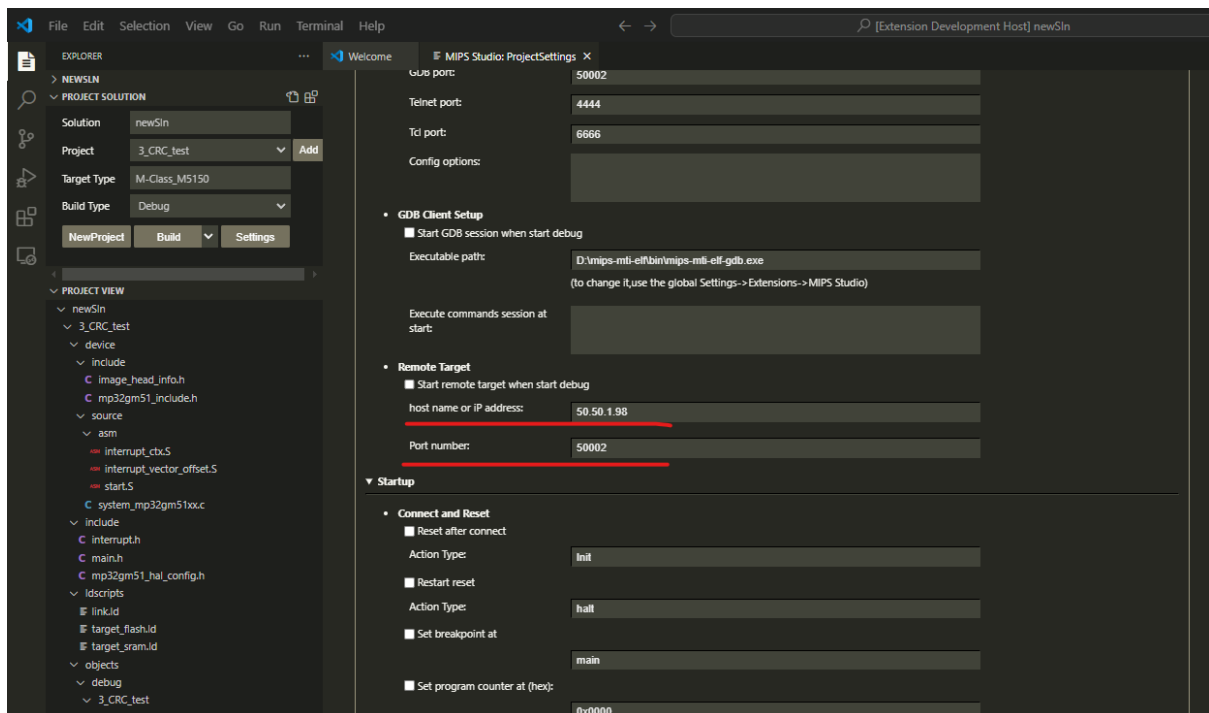
Debug 调试

本功能基于 VSCode 以及 The C/C++ extension 提供的通用调试功能，支持通过 GDB 与 OpenOCD 连接的 SOC 目标设备（真实硬件或者 FPGA）进行调试。

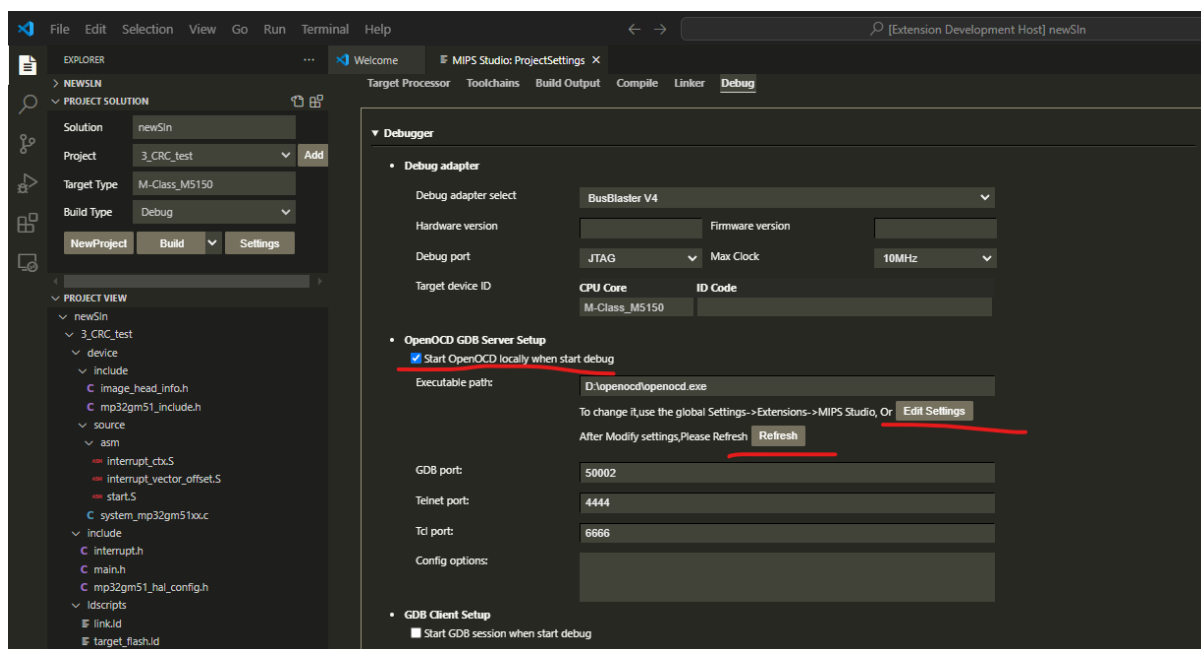
通过对当前项目的 Debug 选项进行设置。



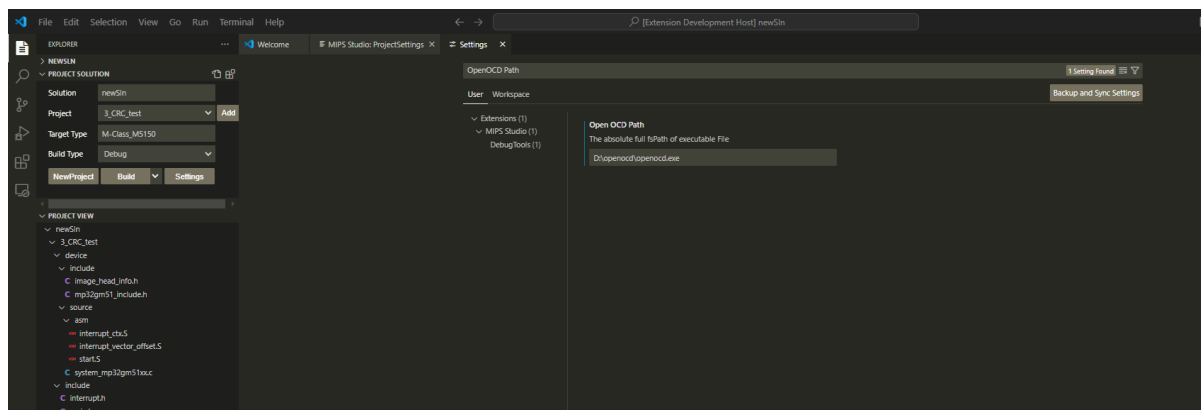
支持 Debugging Remote Programs，设置好远端目标机器 IP 和端口即可。



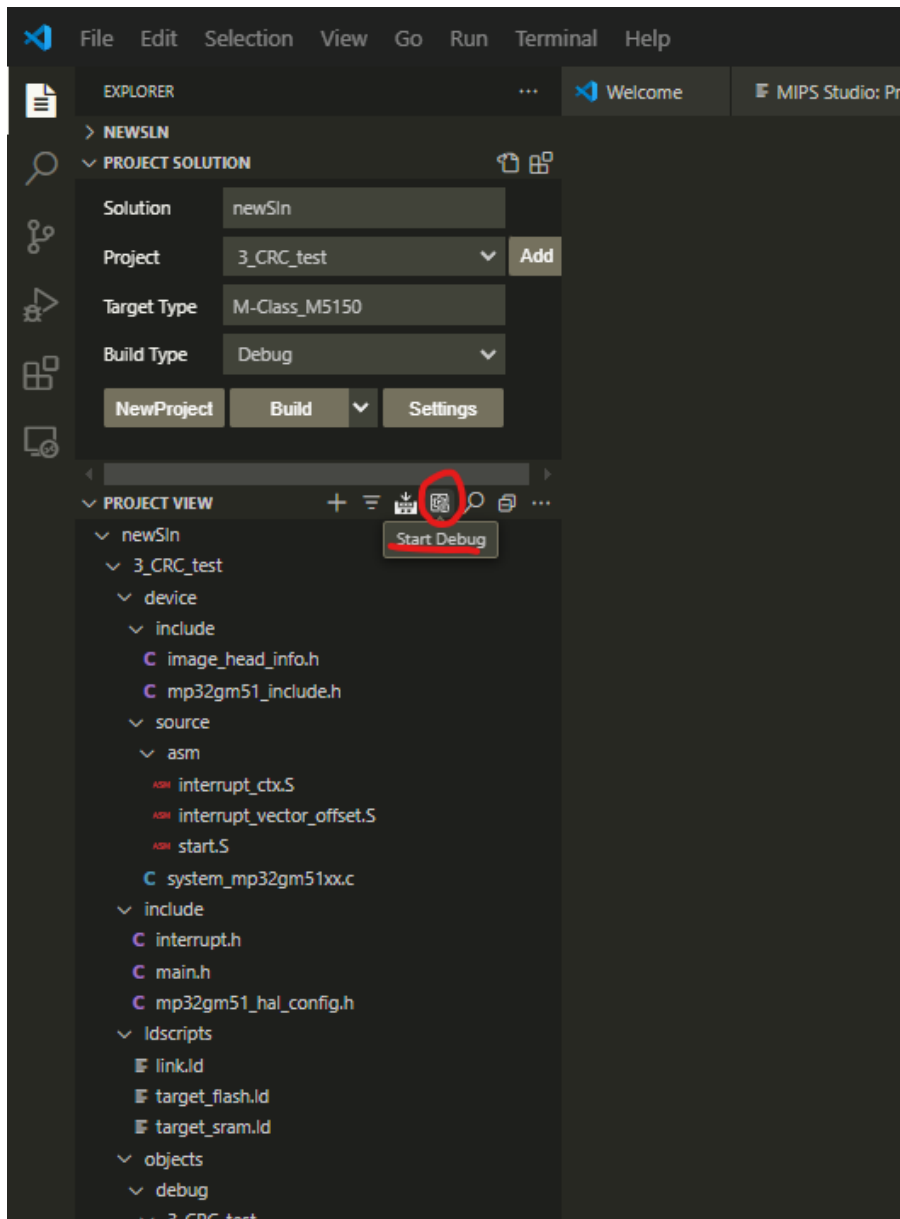
支持本机调试时同步启动 OpenOCD 连接目标芯片开发板，如下图所示，勾选上 Start OpenOCD When start debug。



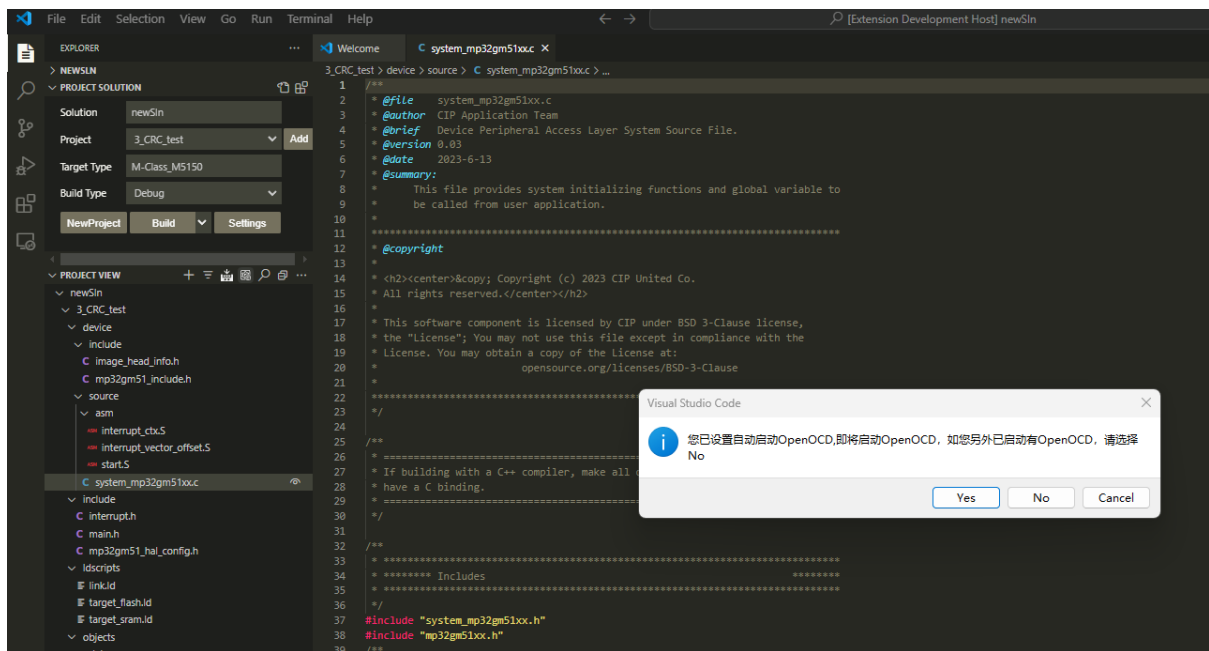
OpenOCD 的可执行文件 (exe) 的路径需要在 VSCode 中设置好，可以通过 VSCode 的设置->User->Extension->MIPS Studio 然后找到 OpenOCD Path，或者点击下图中的 Edit Settings 按钮即可进入（如下图所示）。



自动启动 OpenOCD 的调试，在 Project View 中点击 Start Debug 按钮即可。



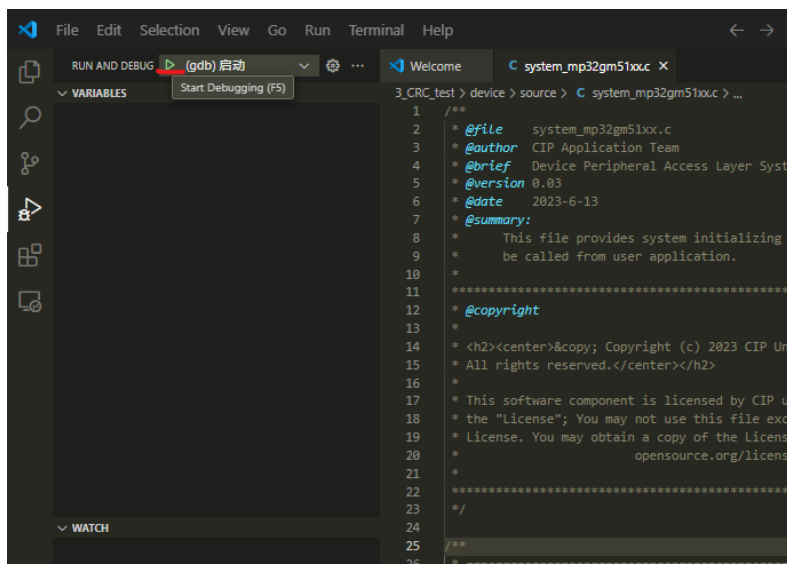
点击 Start Debug 之后，会弹出如下的提示框，提示用户是否已启动过 OpenOCD，



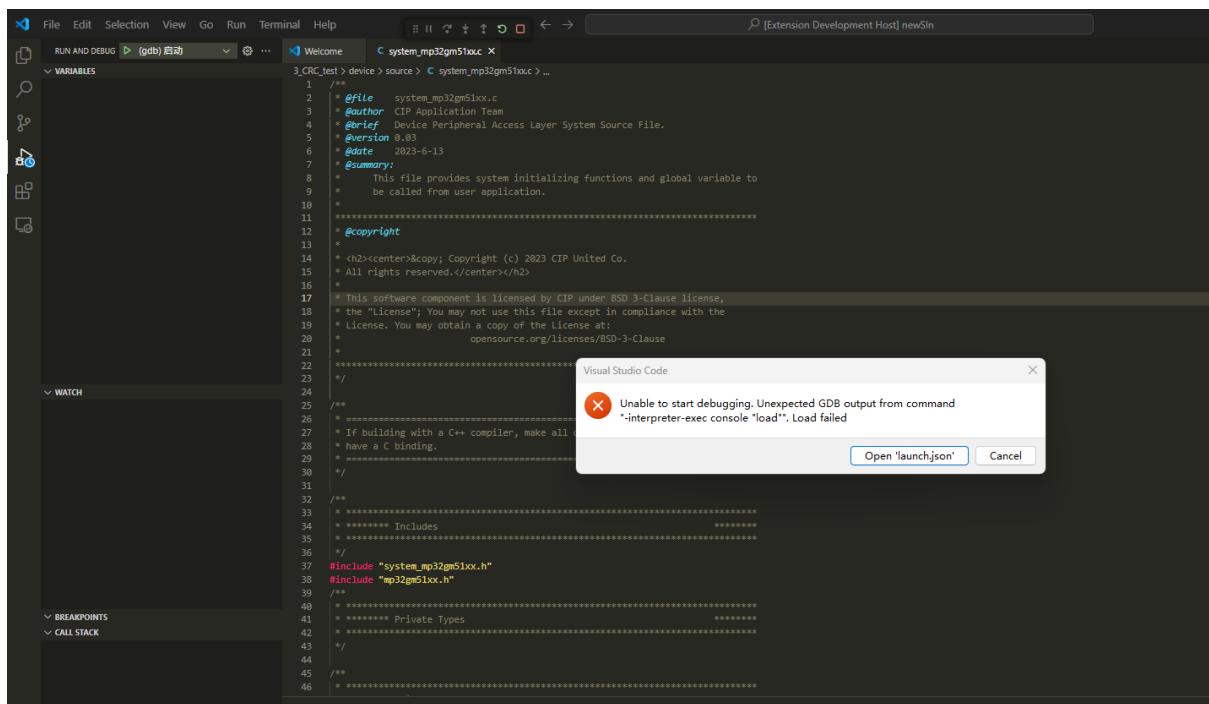
用户选择 Cancel 则不做任何动作（不启动调试），用户选择 No 则本次不再启动 OpenOCD（因为重复启动会导致失败），启动 GDB 连接 OpenOCD 进行调试。

用户选择 Yes 则按照用户在项目设置里设定的端口等参数启动 OpenOCD，之后启动 GDB 连接 OpenOCD 进行调试。

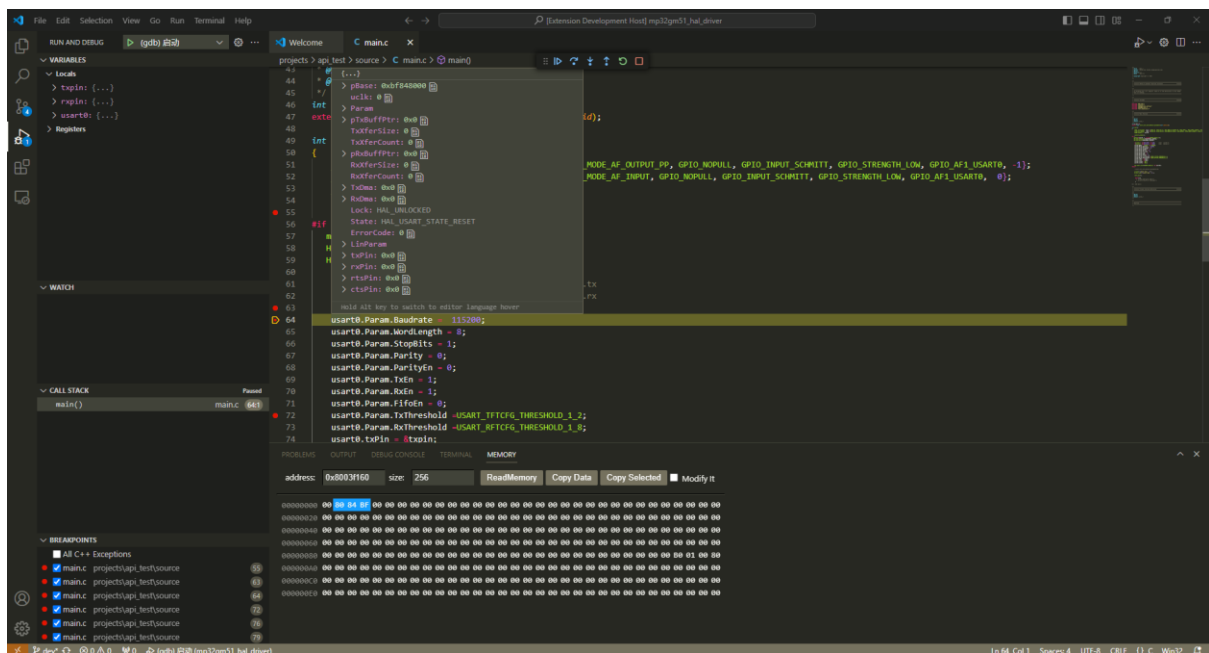
用户电脑上已经启动 OpenOCD 之后，代码编译生成 elf 后，每次重新开启调试直接按 F5 快捷键或者点击 VSCode 调试窗口的按钮都可以，不需要每次重新启动 OpenOCD。



当用户的 OpenOCD 没有连接好目标芯片开发板硬件时，会报 load 失败等错误信息（如下图），需要用户检查 OpenOCD 连接硬件。

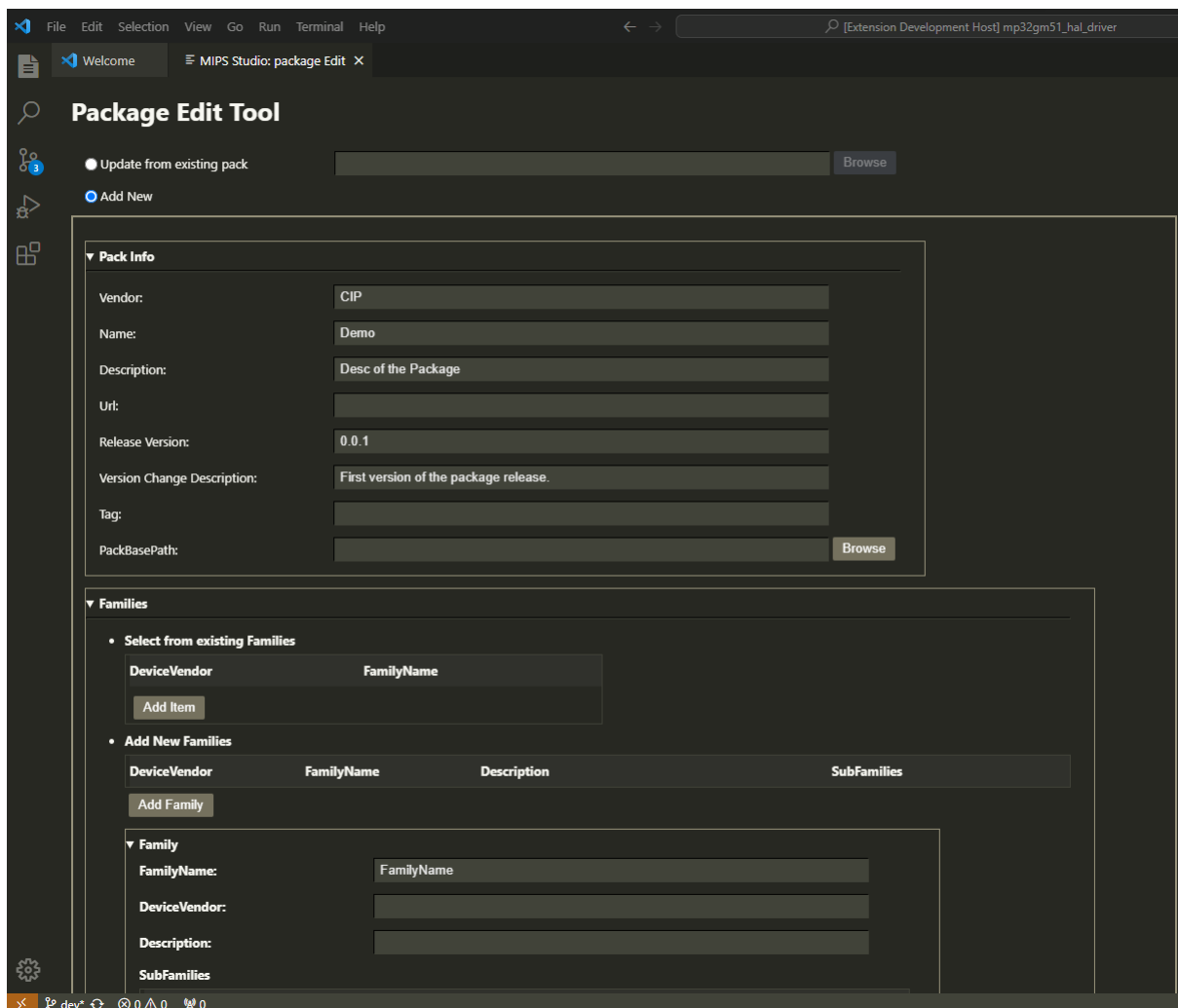


可以图形化的逐行代码进行调试了，变量、Watch、调用堆栈这些都有，另外 MIPS Studio 还提供了内存查看和修改功能，开始愉快的调试工作吧：)



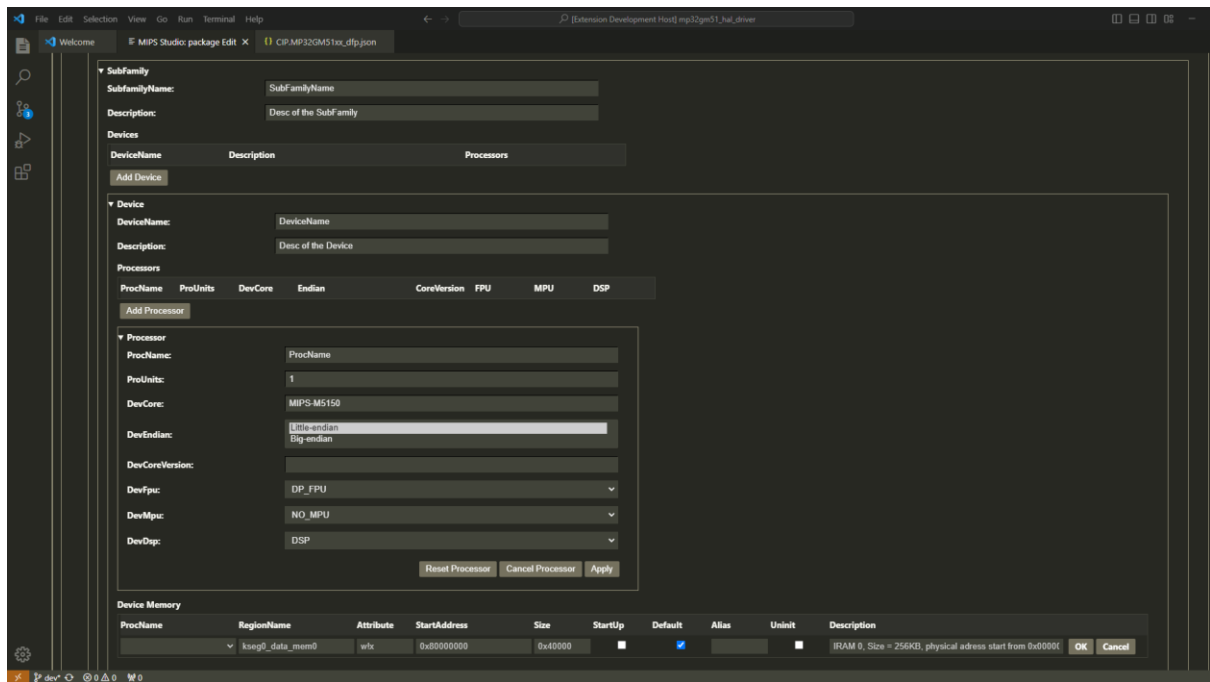
Package 生成

本项功能应仅适合作为 SDK 组件包提供者的厂商来使用，创建 SDK Package 并发布提供给用户使用。

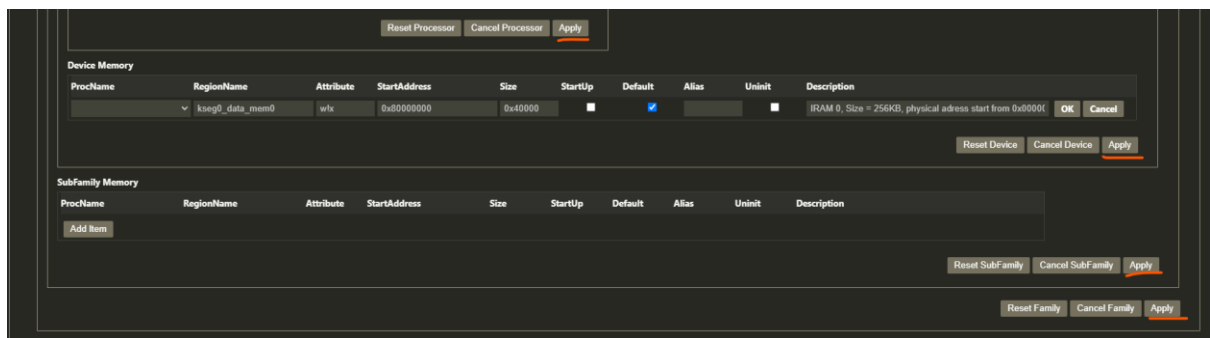


Add New --> Add Family -->Add SubFamily-->Add Device-->Add Processor

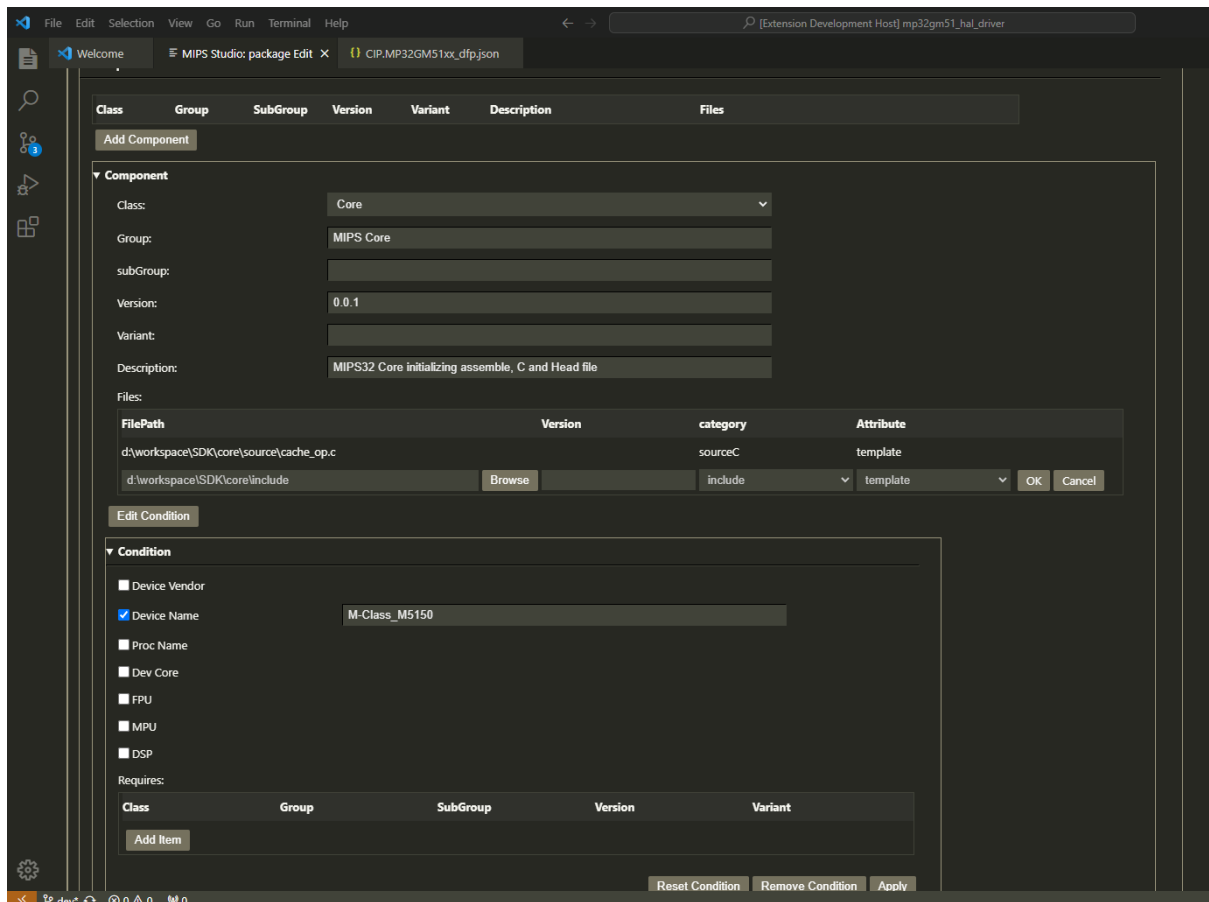
-->Add Device Memroy



须按完整 Family 添加，逐层 Apply。



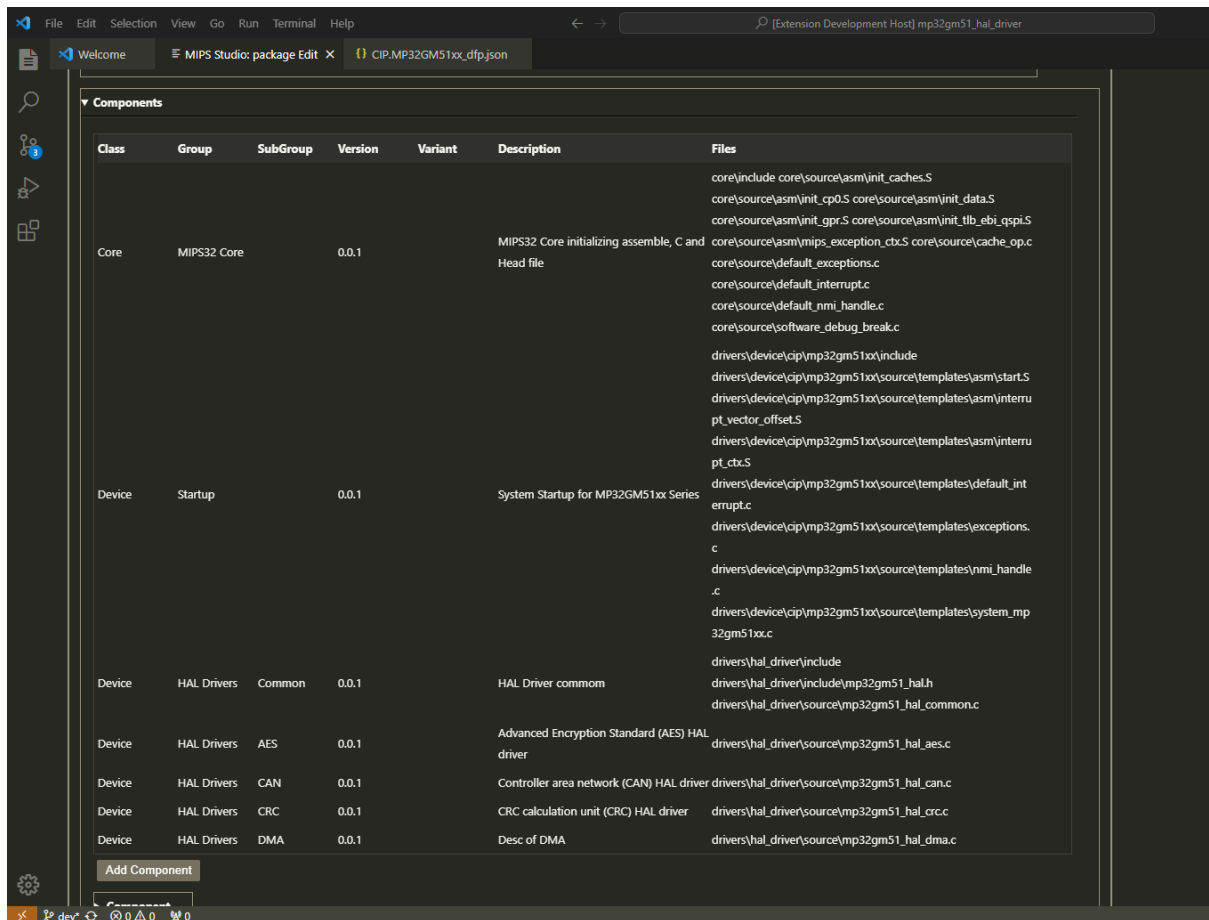
然后可以进行 Component 的添加。



逐一添加 component 和对应的 condition,

需要说明的是, files 里的文件应该都在前面 PackageRootPath 的文件夹路径下。

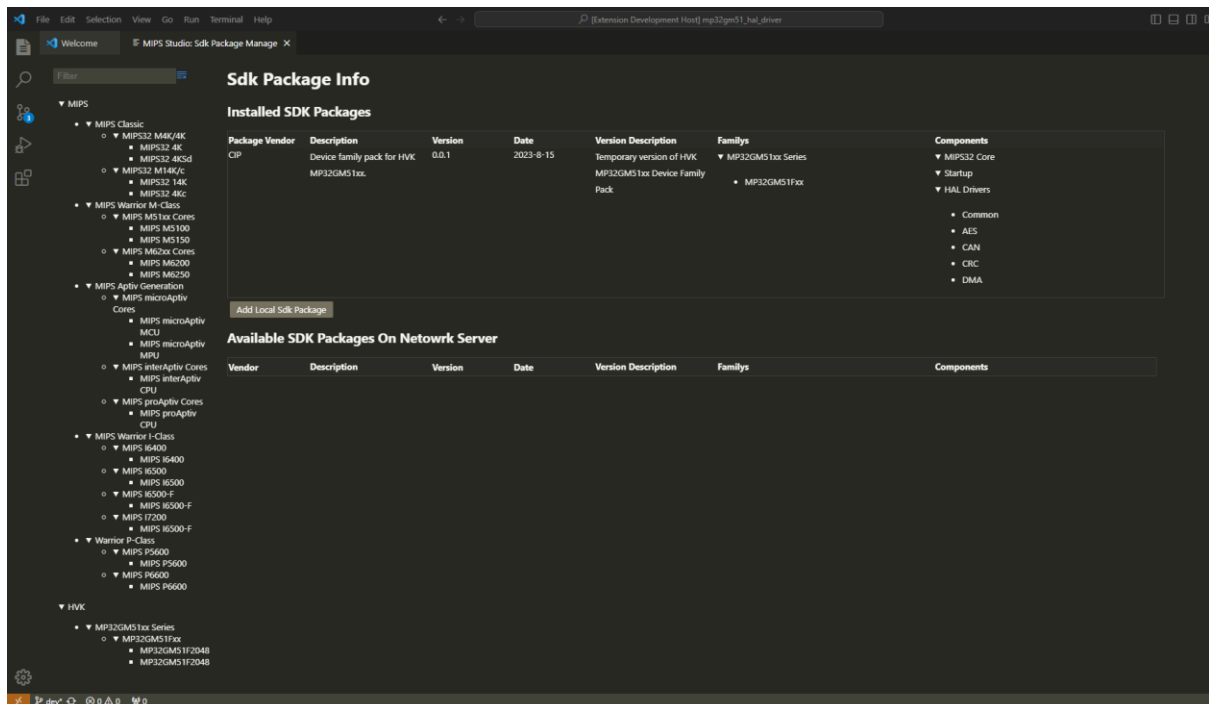
可完成得到如下所示的组件包部分:



最后，点击页面最下方的 Generate DFP 和 Generate Package 将在前面指定的 PackageRootPath 内完成 Package 的生成（选定的文件都会被打包进去）。



Package 安装管理



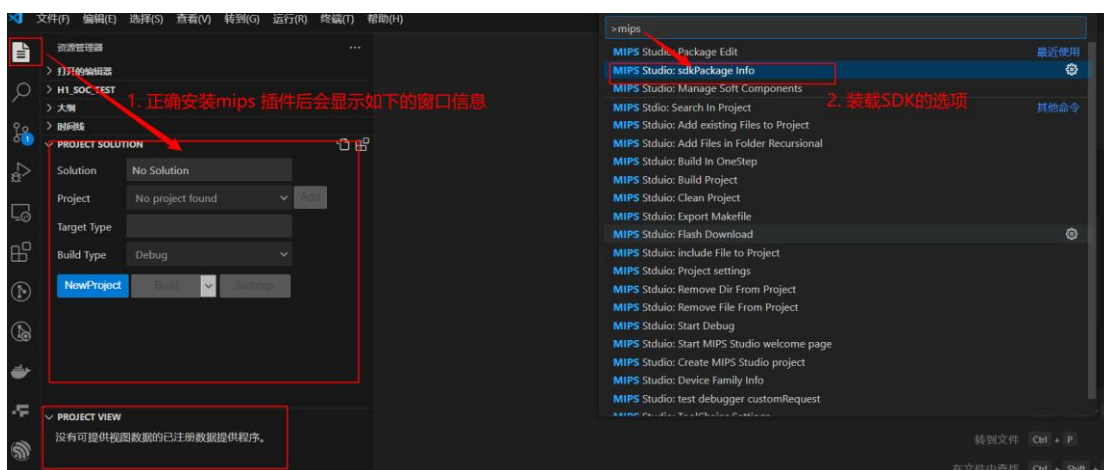
点击 Add Local SDK Package 后选择由厂商提供的 SDK Package。

获取 SDK

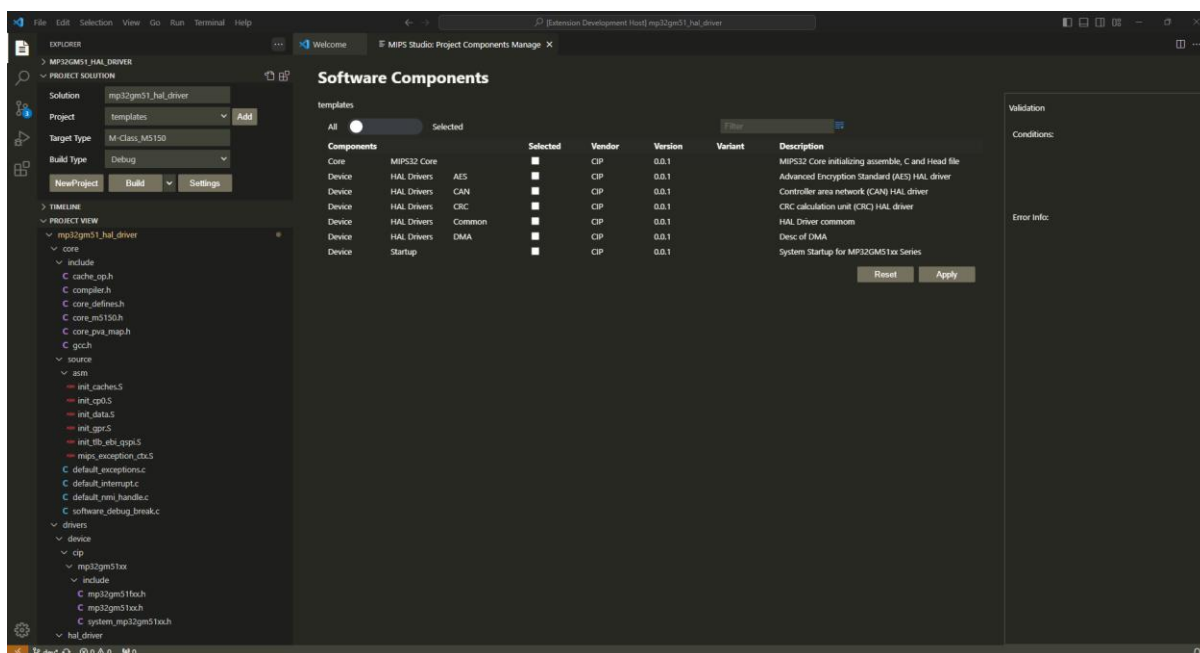
后续会支持根据用户选择的器件，查询可用的 SDK Package 列表显示，用户可进行选择下载安装。目前是联系厂商提供对应的 SDK。

装载 SDK

1. 按下 Ctrl + Shift + P 配置 sdkPackage info



2. 加载获取的 SDK

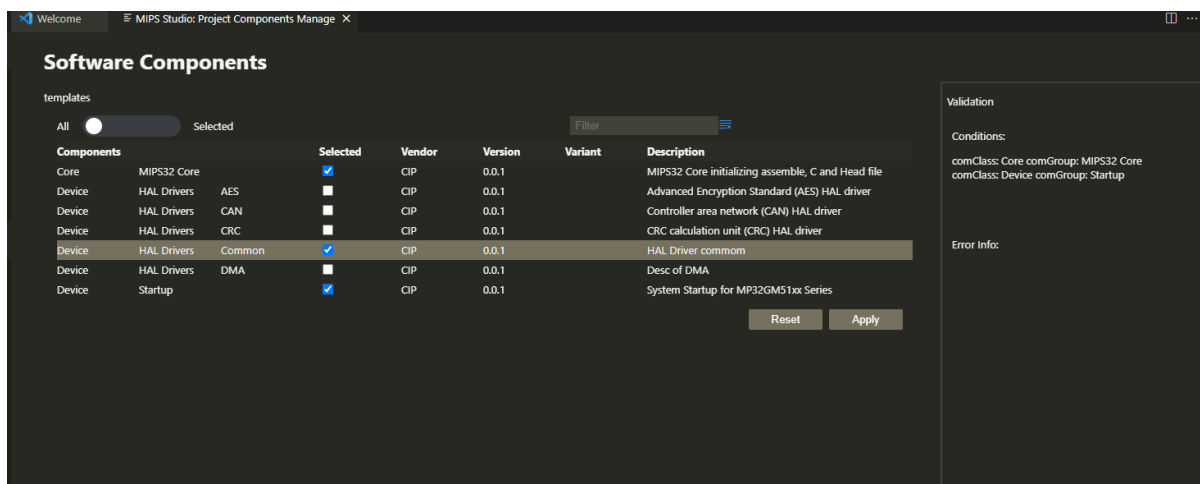


用户通过点击解决方案视图的右上角按钮，进入软件组件管理页面。

根据当前项目的器件信息，展示目前已安装的 SDK Package 中适用的软件组件。

用户可筛选查询，并可进行选择不同的组件。

基于软件组件的依赖项信息，用户选中某个组件功能时，将自动勾选所依赖的组件，如下图所示。



用户选择好软件组件后，点击 Apply 按钮，将自动将选中的组件加载到用户当前项目中（组件内的文件将 copy 到项目路径下）。

当由于某些原因（系统权限或其他未知问题）没有能完全将组件中的文件 Copy 到用户当前项目中时，会弹出对话框提示错误信息。

Framework 是必选的，其它的模块更具自己需求选择（图中是选择 GPIO + Framework）

