

YTM32B1MD2x HCU Demo 使用说明

简述

YTM32B1MD2x 支持非对称算法 ECC-192/256/384 以及 RSA-1024/2048/3072/4096

特封出静态库，下面给出个简单示例

Demo 中展示

- AES-ECB 使用软件密钥进行加密
- AES-CMAC 使用软件密钥进行签名与验签，使用硬件密钥进行签名与验签
- SHA-256 的签名与验签
- RSA-1024 使用软件密钥进行加密与解密
- RSA-4096 使用硬件密钥进行加密与解密
- ECC-256 的基础运算，点加与点乘运算
- ECDSA 公钥生成，签名生成，以及签名认证

发布清单

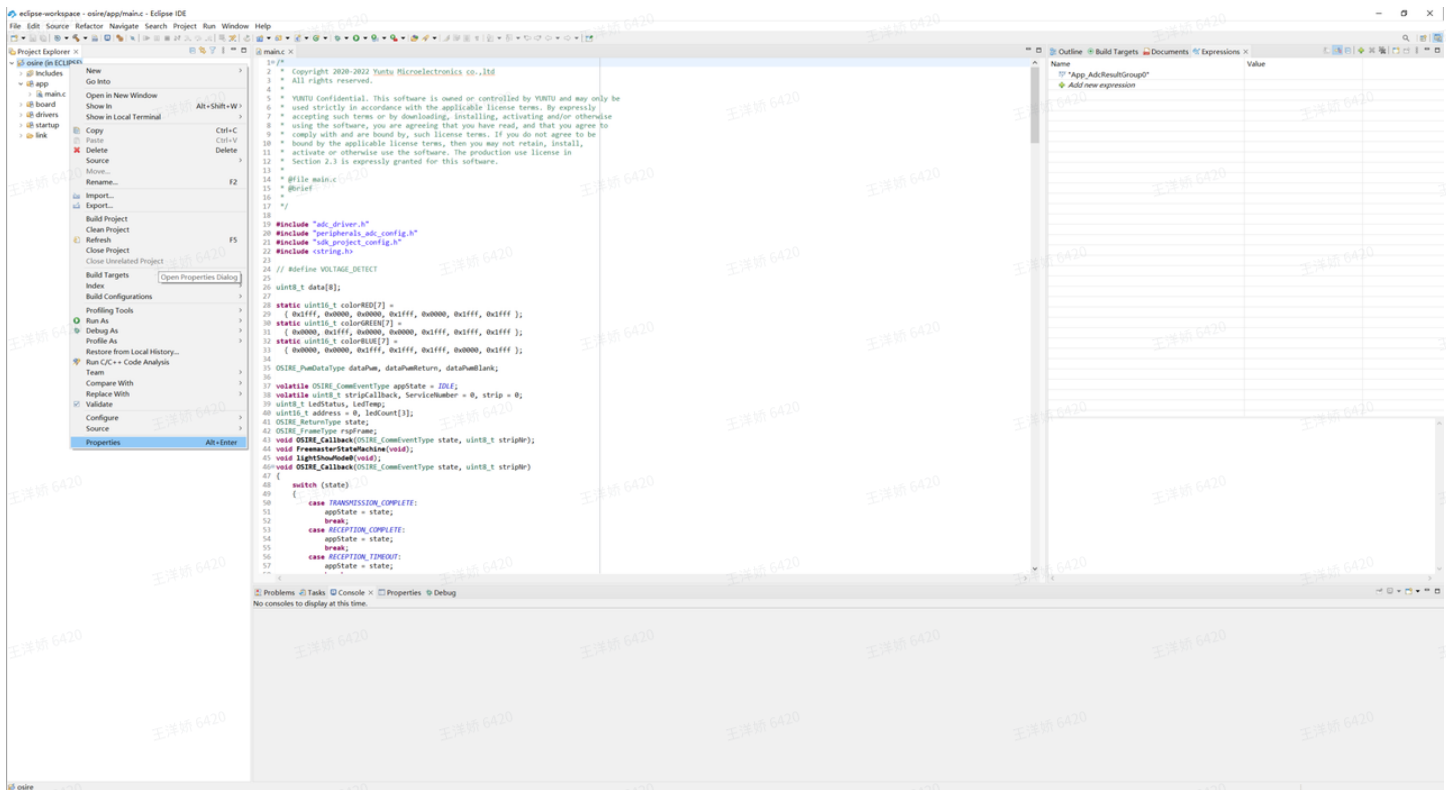
- HCU demo 应用程序 (双击 HCU_Demo.yct 生成工程)
- lib_hcu.a (HCU 的驱动包，适用于 GCC 与 YuntuIDE 开发环境)
- mdk_hcu.lib (HCU 的驱动包，适用于 KEIL 开发环境)
- iar_hcu.a (HCU 的驱动包，适用于 IAR 开发环境)

常见问题

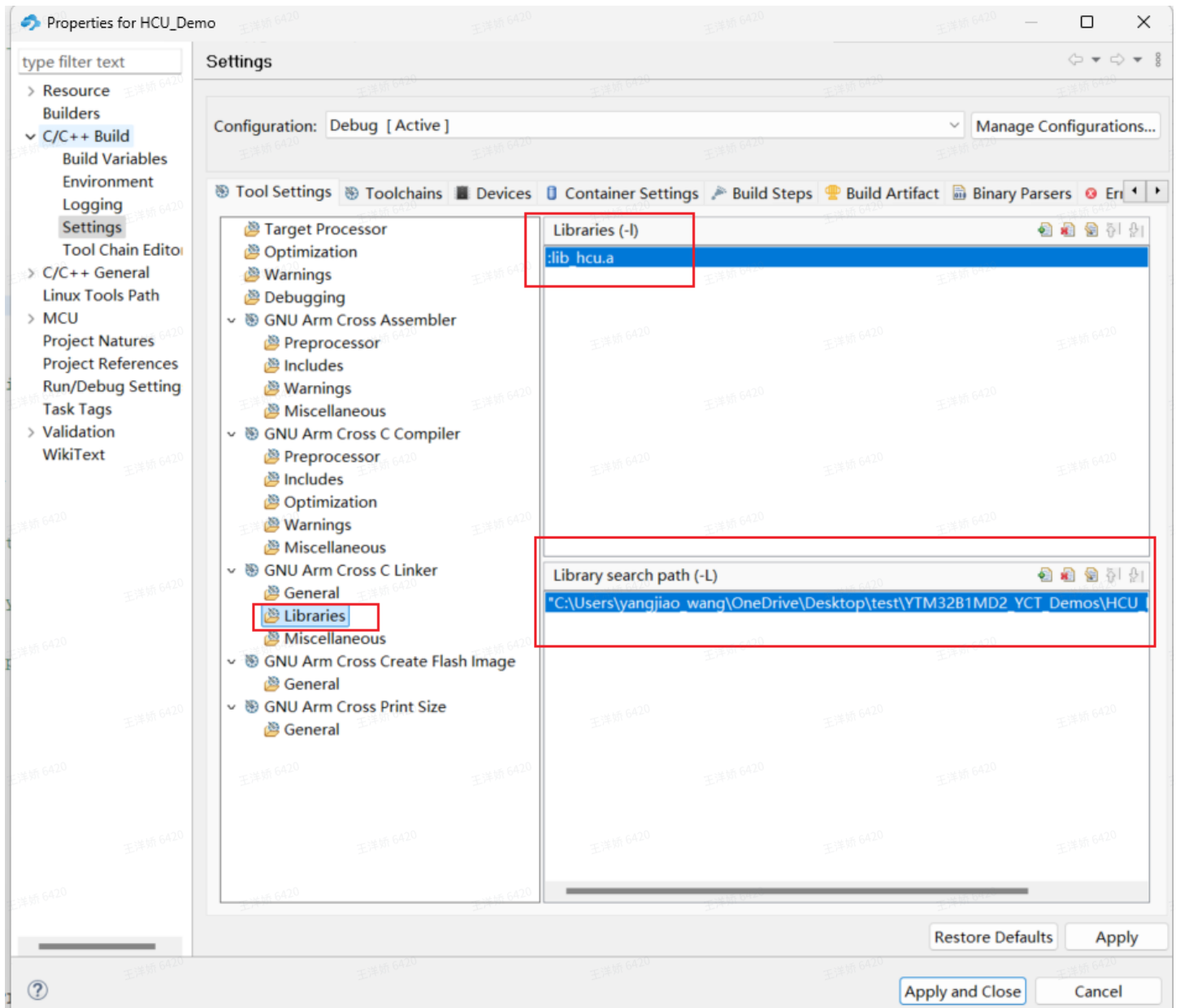
YuntuIDE 添加库

在 YuntuIDE 加载完 OSIRE 的示例后，请注意下面几条提示。

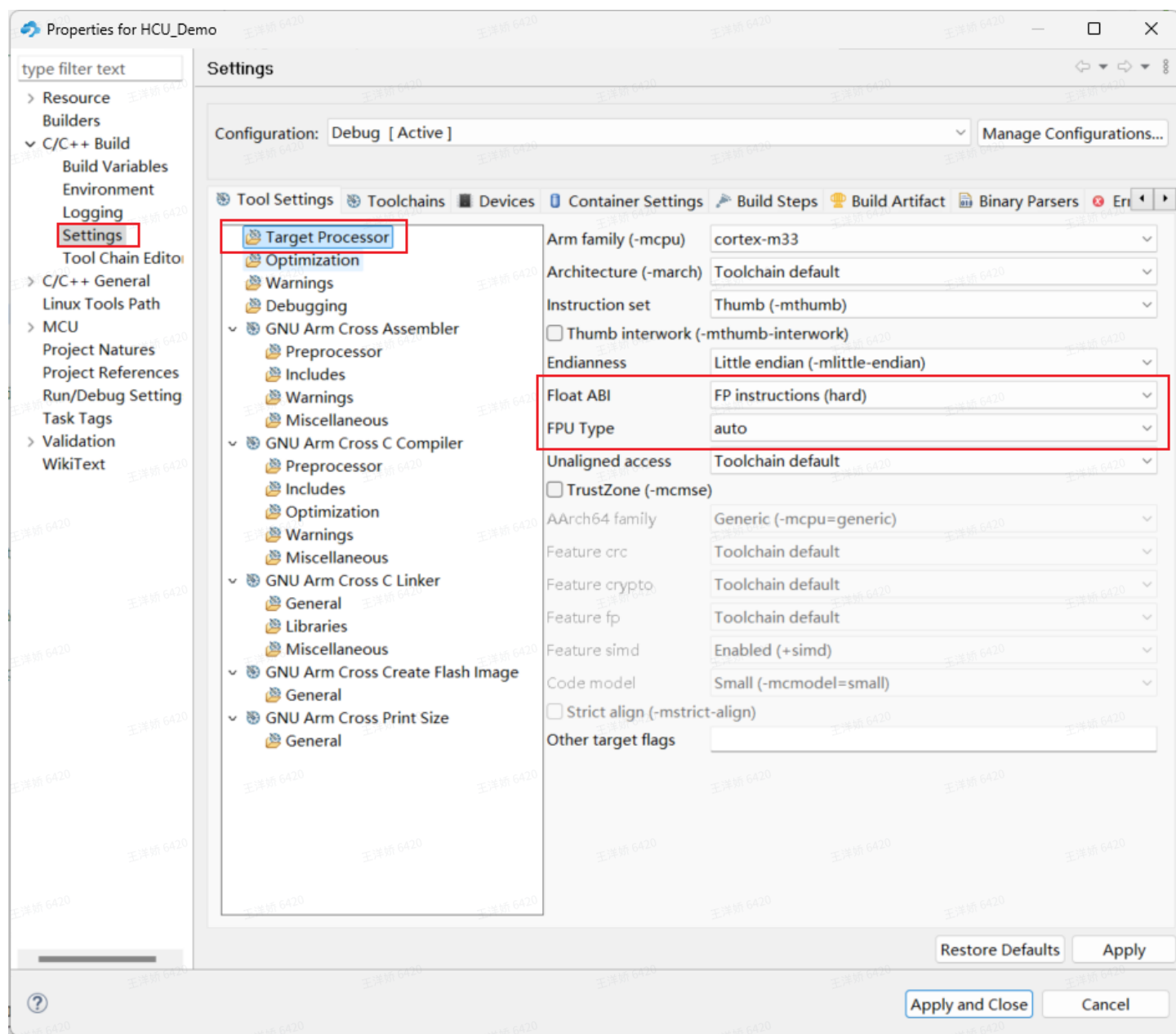
右键工程并点击 **Properties** 来打开工程属性窗口。



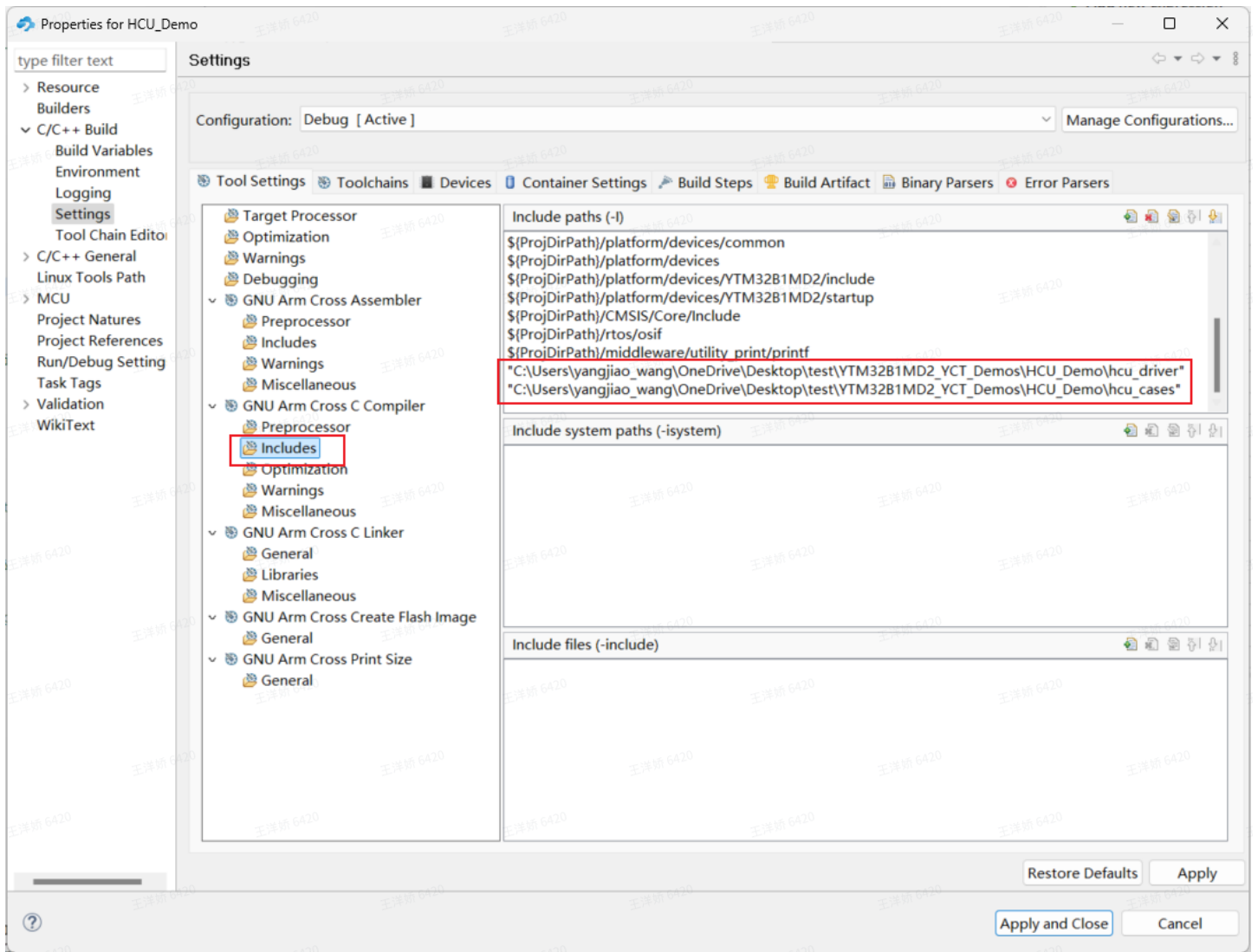
然后点击 **C/C++ Build** 来展开并点击 **Setting**, 并选择 **Libraries**。Libraries(-l) 里的内容是实际的库文件, 例如, :lib_hcu.a, Library search path(-L) 里的内容是库所在的路径。



配置 FPU 属性，选择硬件 FPU，配置如下



添加路径，增加 hcu_driver 与 hcu_cases 目录

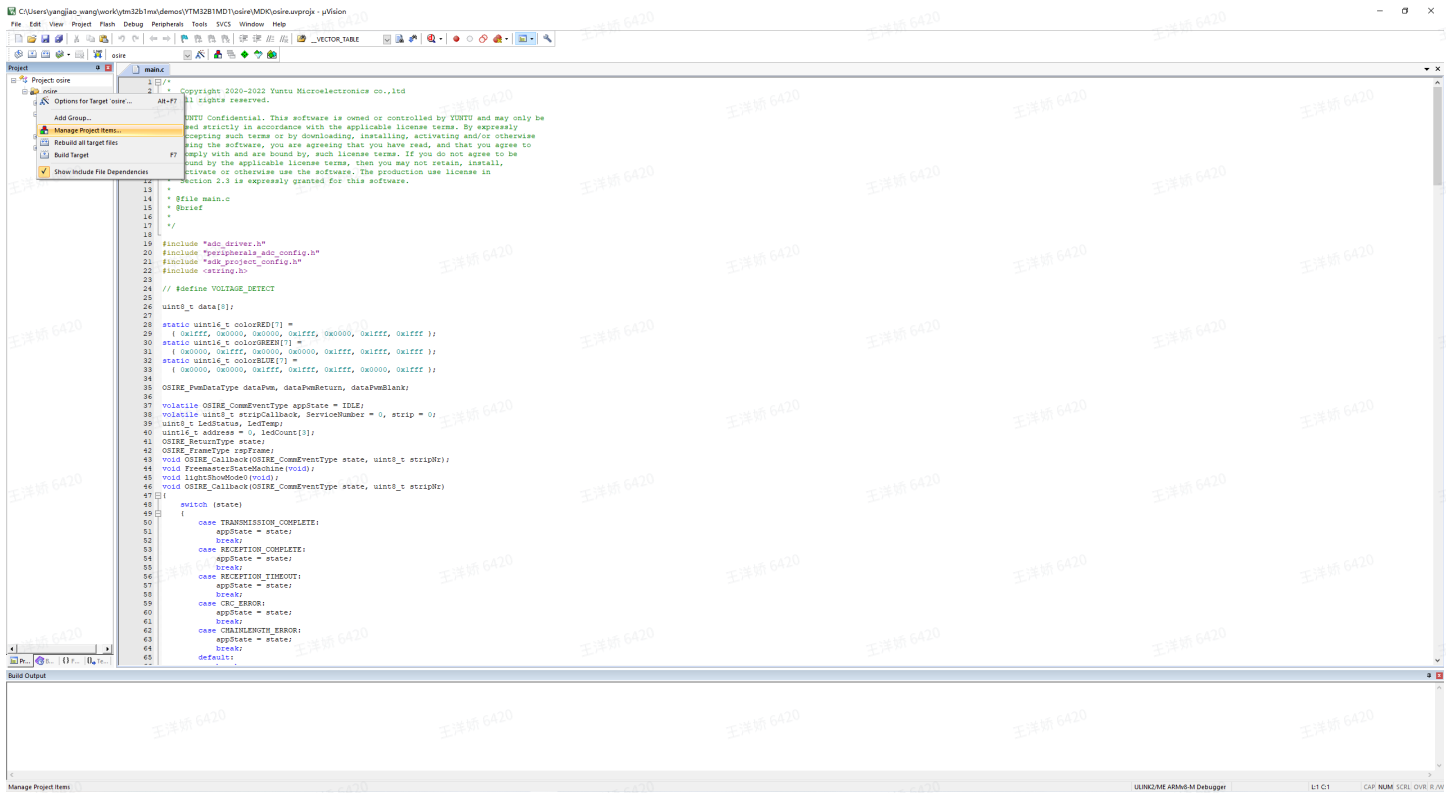


随后正常编译即可。

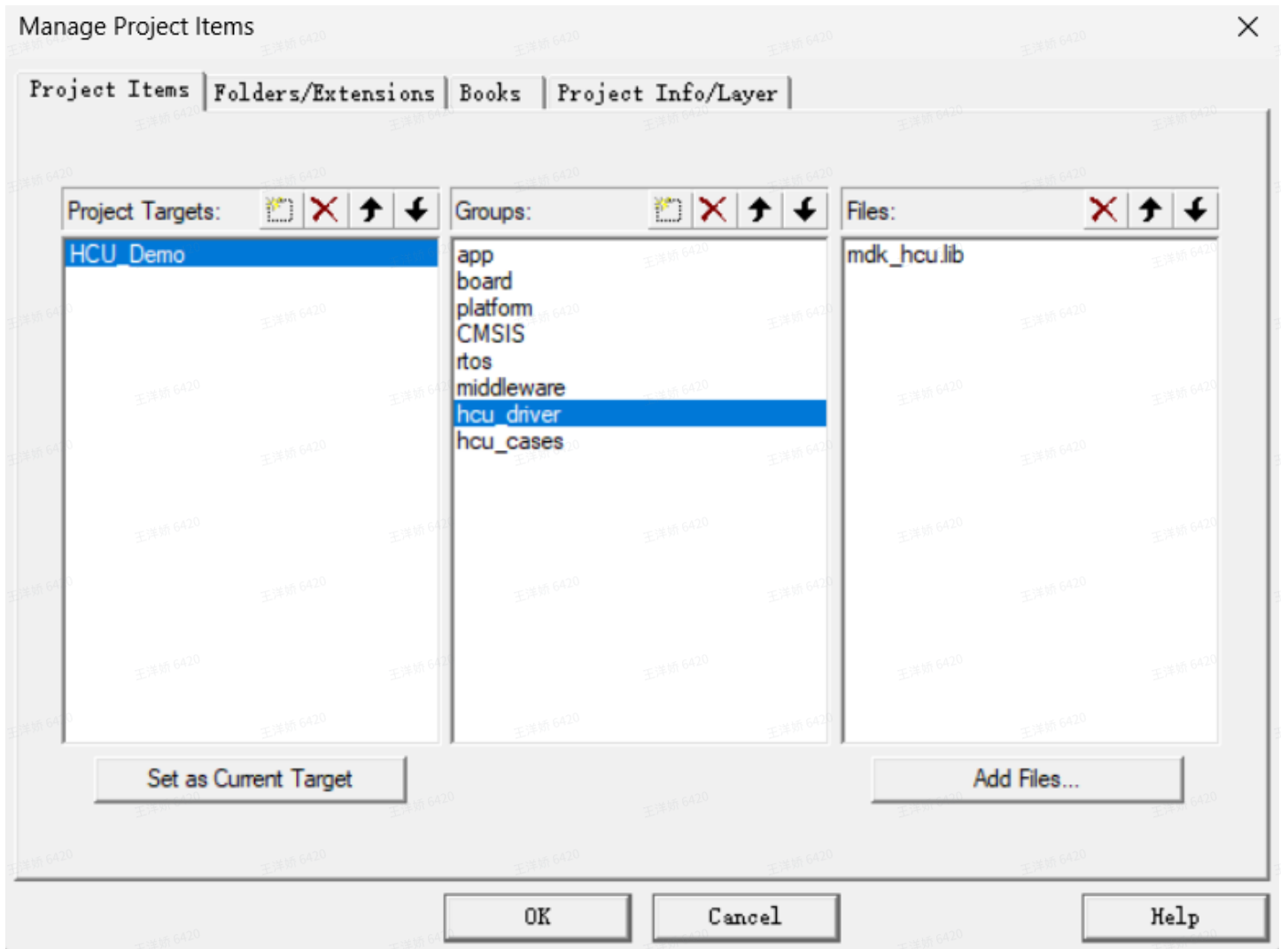
KEIL添加库

在 KEIL 加载完 HCU 的示例后，请注意下面几条提示。

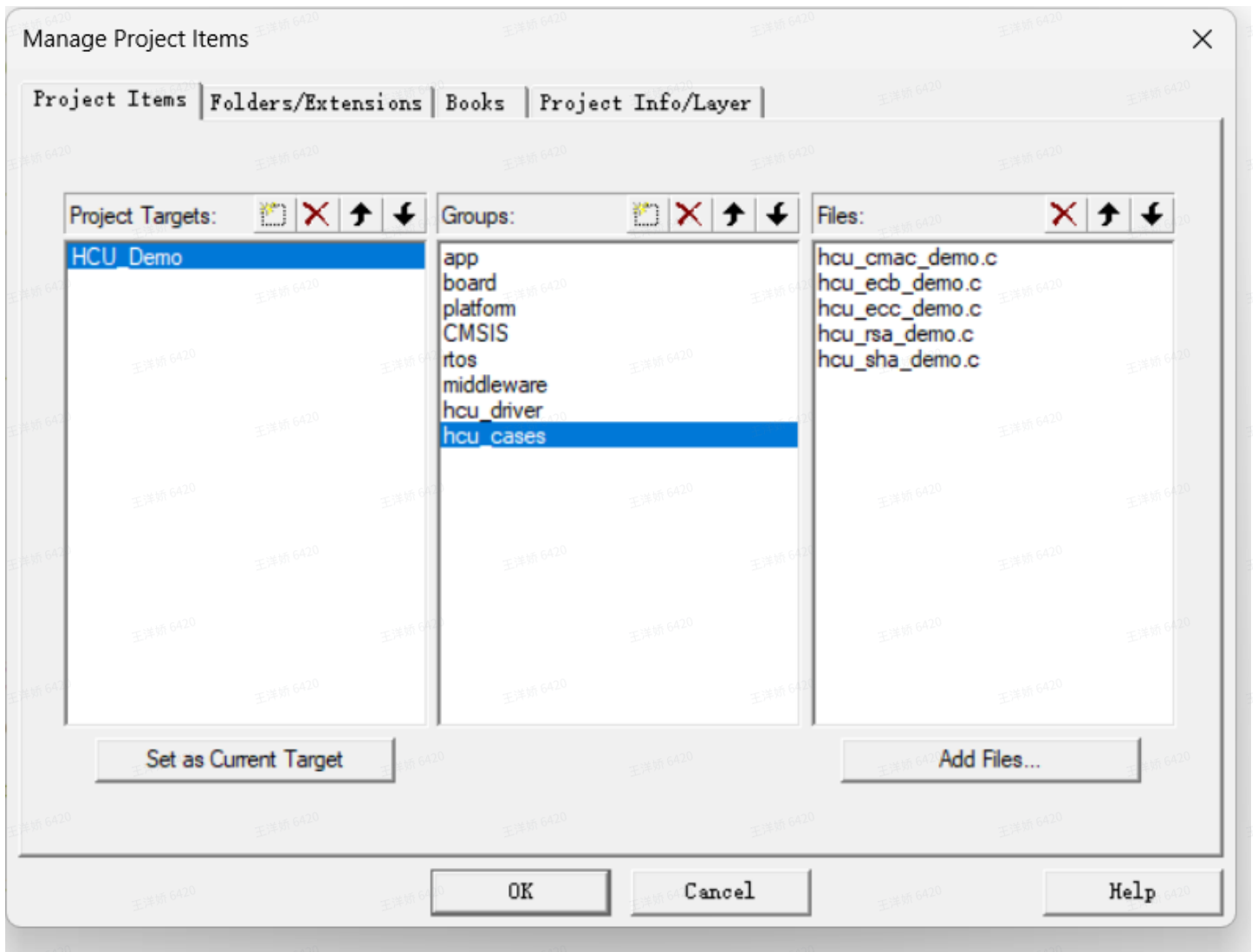
右键工程并点击 **Manage Project Items** 来打开工程文件配置窗口



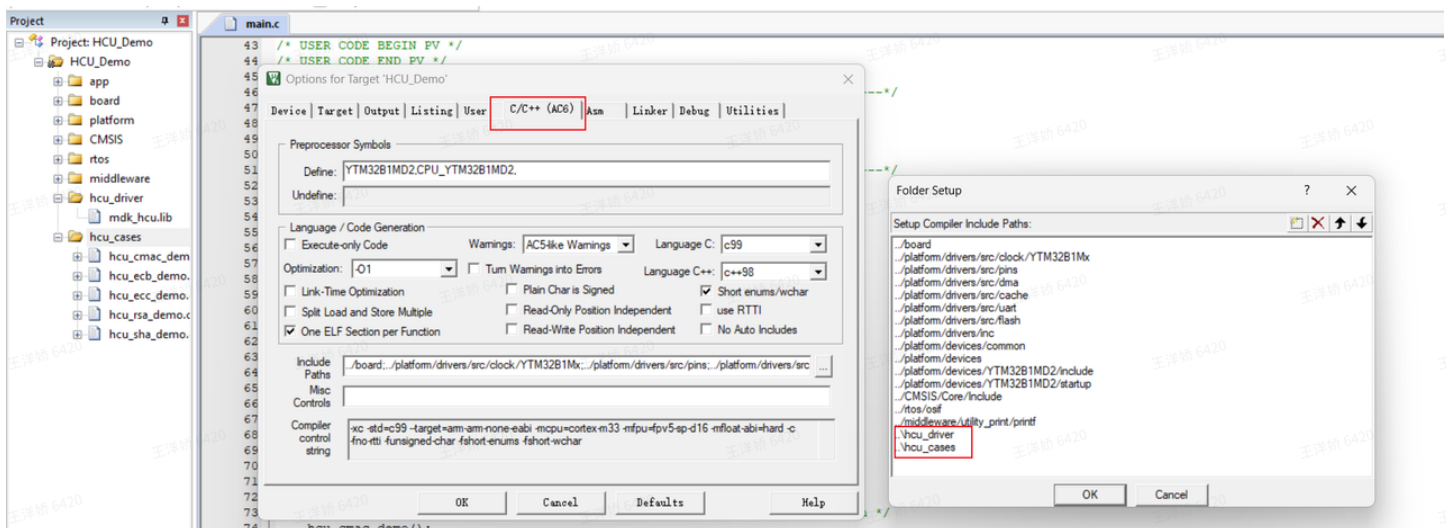
添加 hcu_driver 文件夹，并添加 mdk_hcu.lib



添加 hcu_cases 文件夹，并添加各个例程



添加编译路径，增加 **hcu_driver** and **hcu_cases**

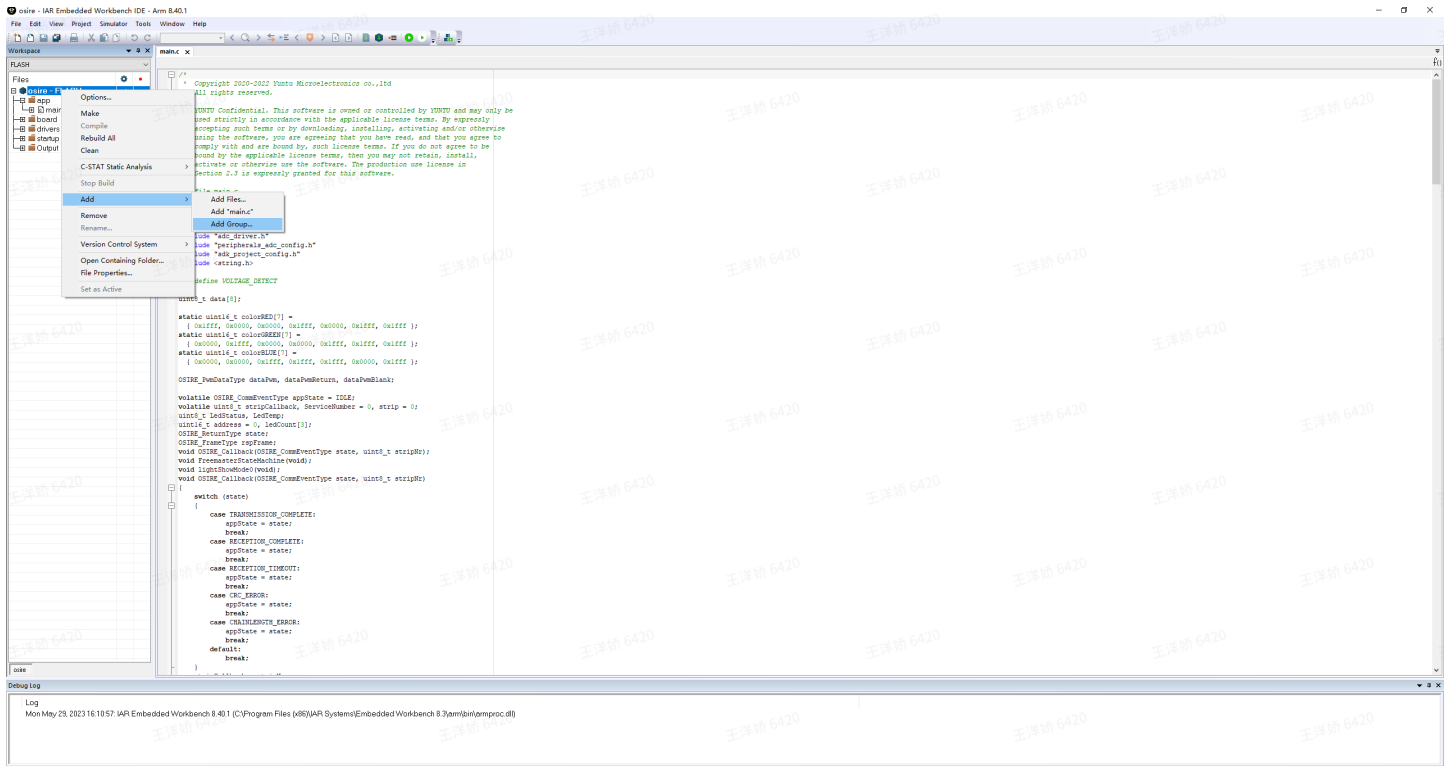


随后正常编译即可。

IAR添加库

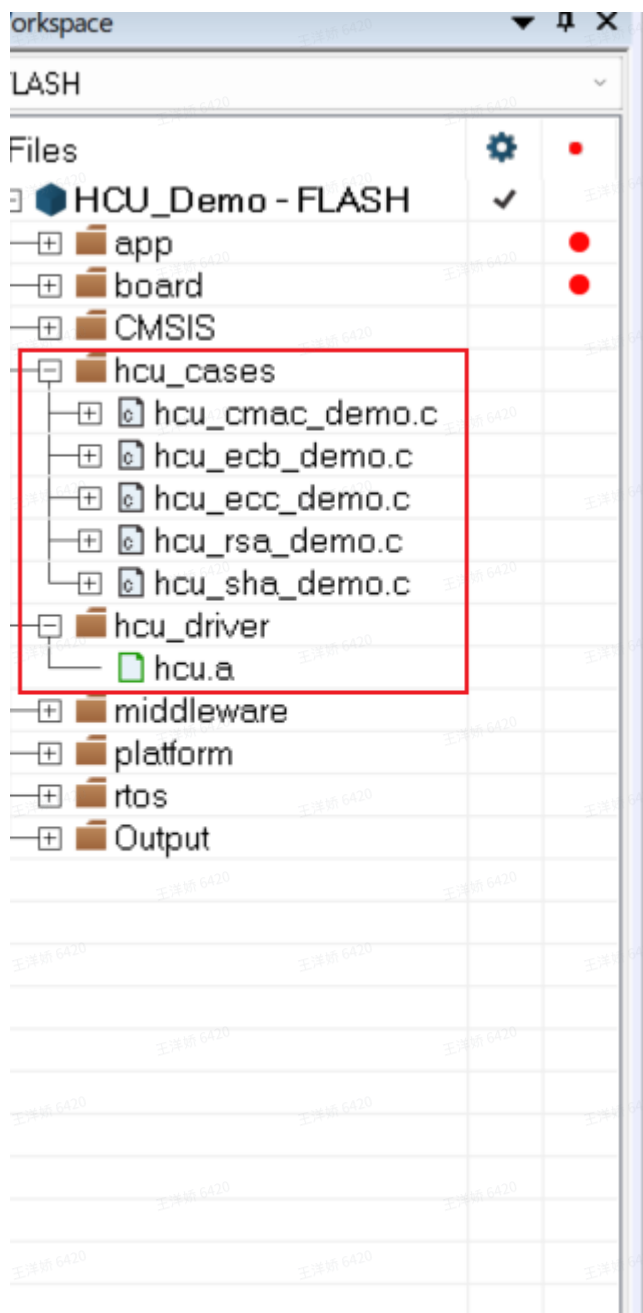
在 IAR 中加载完 HCU 的示例后，请注意下面几条提示。

右键工程并点击 Add，并选择 Add Group，添加 hcu_driver 的目录与 hcu_cases 的目录。

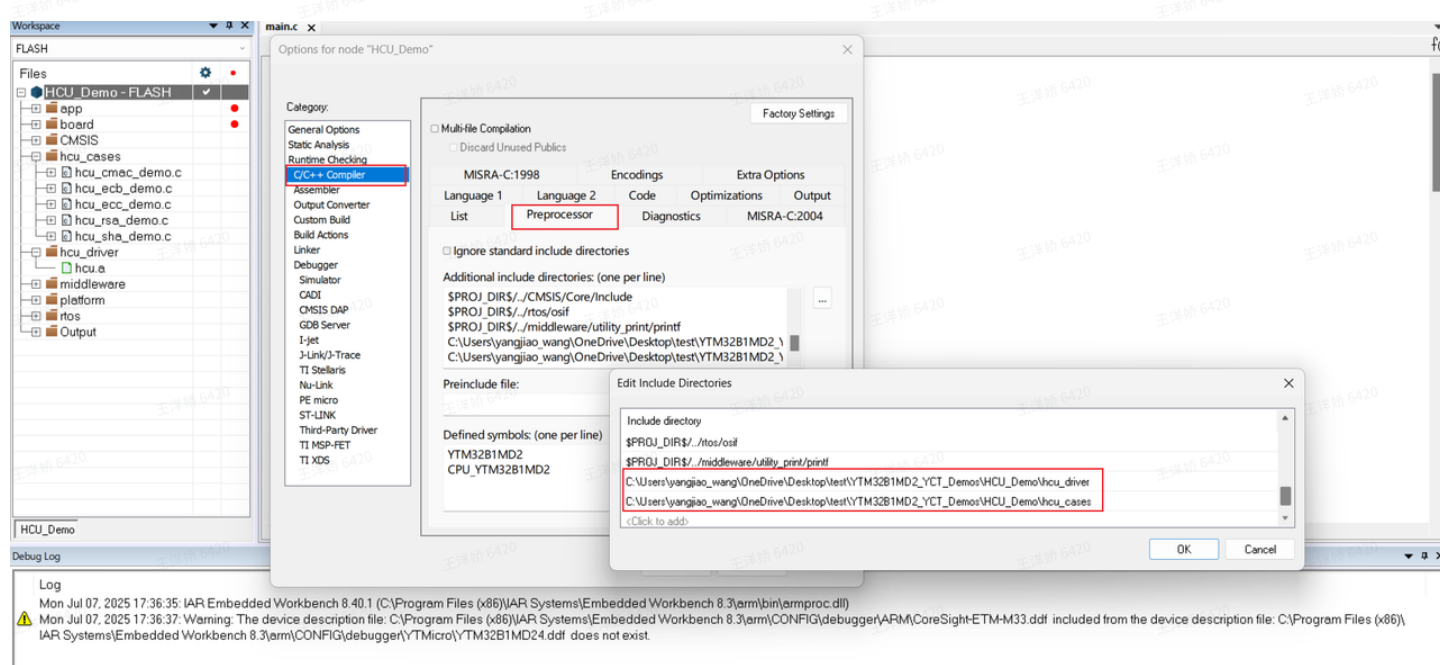


右键 **hcu_driver** Group，并选择 **Add Files** 添加 **iar_hcu.a**。

右键 **hcu_cases** Group，并选择 **Add Files** 添加各个demo例程。



添加编译路径，增加 hcu_driver and hcu_cases



随后正常编译即可

CMakeGCC

通过使用文件夹内的 CMakeLists.txt ，直接编译。

```
iar_hcu.a  M CMakeLists.txt X
HCU_Demo > M CMakeLists.txt
22 # include(...)
23 # USER CODE END include
24
25 set(project_elf HCU_Demo.elf)
26 add_executable(${project_elf} app/main.c)
27 #add app as include path
28 target_include_directories(${project_elf} PRIVATE app)
29 #add all source files in app folder
30 file(GLOB dir_sources "app/*.c" "app/*.cpp" "app/*.s")
31 if(dir_sources)
32     foreach(src ${dir_sources})
33         target_sources(${project_elf} PRIVATE ${src})
34     endforeach()
35 endif()
36
37 # USER CODE BEGIN add_executable
38 # target_include_directories()
39 # target_sources(${project_elf} PRIVATE ..)
40 target_include_directories(${project_elf} PRIVATE "hcu_driver")
41 target_include_directories(${project_elf} PRIVATE "hcu_cases")
42 file(GLOB dir_cases "hcu_cases/*.c")
43 if(dir_cases)
44     foreach(src ${dir_cases})
45         target_sources(${project_elf} PRIVATE ${src})
46     endforeach()
47 endif()
48 # USER CODE END add_executable
49
50 configcore(${project_elf} ${CMAKE_SOURCE_DIR})
51
52
53 # USER CODE BEGIN target_compile_definitions
54 # target_compile_definitions(...)
55 # USER CODE END target_compile_definitions
56
57 target_compile_definitions(${project_elf} PUBLIC
58     "YTM32B1MD2"
59     "CPU_YTM32B1MD2"
60 )
61
62 # USER CODE BEGIN target_compile_options
63 # target_compile_options(...)
64 # USER CODE END target_compile_options
65
66
67 target_link_libraries(${project_elf} "-Wl,--whole-archive" GENERATED_CONFIG_TARGET GENERATED_SDK_TARGET UTILITY_PRINT "-Wl,--no-whole-archive")
68
69 # USER CODE BEGIN target_link_libraries
70 # target_link_libraries(...)
71 target_link_libraries(${project_elf} -L${CMAKE_SOURCE_DIR}/hcu_driver -l_hcu)
72 # USER CODE END target_link_libraries
73
```