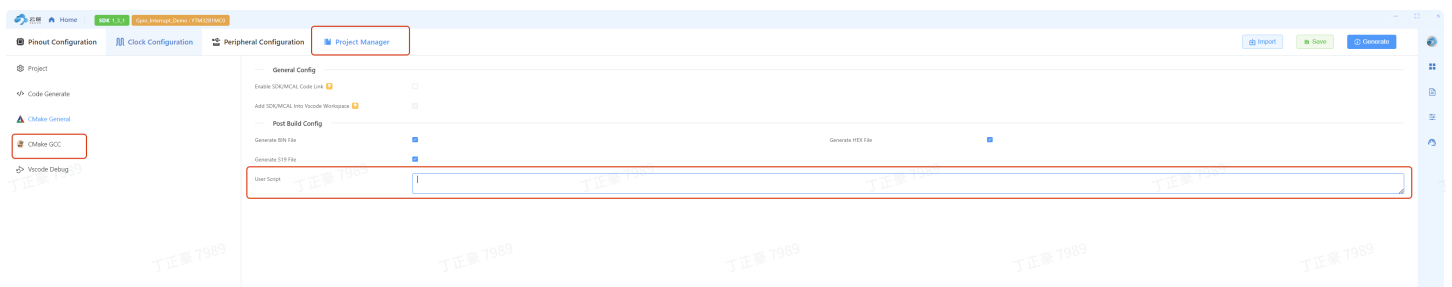


YCT 添加bat脚本

1、添加脚本位置

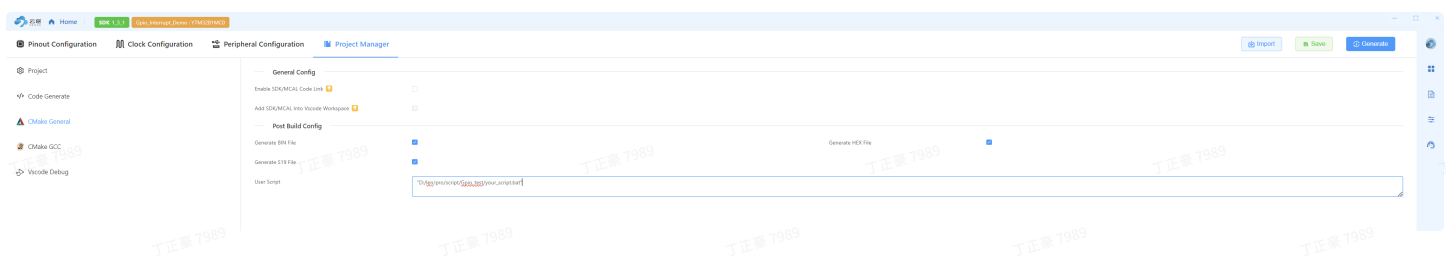
- SDK位置



MCAL位置同理

2、语法规则

1. 如果只是在windows下使用的bat文件，在YCT里只需要在User Script框 指定脚本所在目录



代码块

```
1 "D:/len/pro/script/Gpio_test/your_script.bat"
```

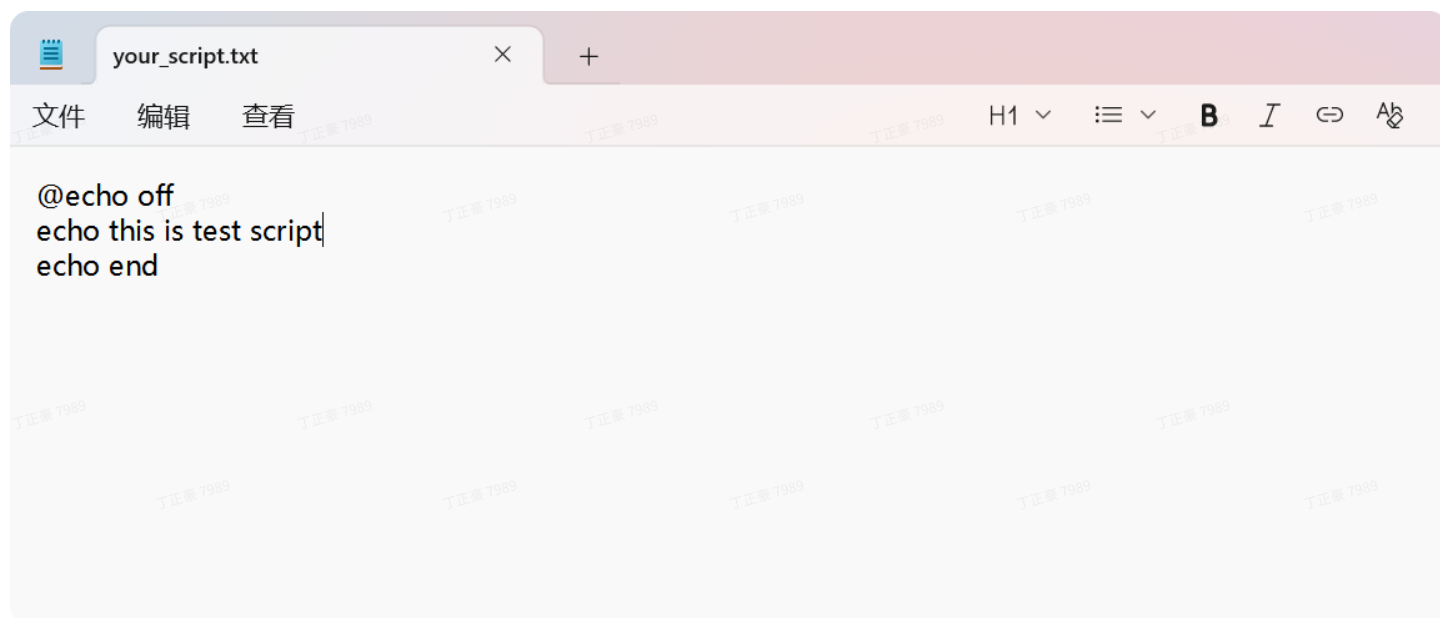
- 实际会在CMakeLists.txt文件末尾生成对应的框架，然后增加上对应的脚本地址。也可以直接在文件末尾增加类似结构。

代码块

```
1 # POST BUILD
2 add_custom_command(
3     TARGET ${project_elf} POST_BUILD
4     WORKING_DIRECTORY ${CMAKE_SOURCE_DIR}
5     COMMAND "D:/len/pro/script/Gpio_test/your_script.bat"
6 )
```

3、脚本实际执行测试

- 实际脚本内容



- 之后点击执行VScode最底端cmake生成按钮，
- 测试结果

文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行(R) 终端(T) 帮助(H) Gpio_Interrupt_Demo (工作区)

资源管理器

- GPIO_INTERRUPT_DEMO (工作区)
 - Gpio_test
 - .cache
 - .vscode
 - app
 - board
 - build
 - cmake
 - CMSIS
 - EWARM
 - middleware
 - platform
 - rtos
 - CMakeLists.txt
 - Gpio_Interrupt_Demo.code-workspace
 - Gpio_Interrupt_Demo.yct
 - ReadMe.md
 - your_script.txt

Gpio_test > CMakeLists.txt

```
101 add_custom_target(genlist
102     DEPENDS ${project_elf}
103     COMMAND ${CMAKE_OBJDUMP} --source --all-headers --demangle --line-numbers --wide ${CMAKE_CURRENT_BINARY_DIR}/${CMAKE_PROJECT_NAME}
104 )
105
106
107
108 # yct file as depend
109 file(GLOB yct_files "${CMAKE_SOURCE_DIR}/*.yct")
110 if(yct_files)
111     list(APPEND CMAKE_CONFIGURE_DEPENDS ${yct_files})
112 endif()
113
114 # POST_BUILD
115 add_custom_command(
116     TARGET ${project_elf} POST_BUILD
117     WORKING_DIRECTORY ${CMAKE_SOURCE_DIR}
118     COMMAND "D:/len/pro/script/Gpio_test/your_script.bat"
119 )
120 # USER CODE BEGIN others
121
122 # USER CODE END others
```

问题 输出 调试控制台 终端 窗口 MEMORY XRTOS

```
[build] [4/30 86% :: 0.096] building C object CMakeFiles/GENERATED_SDK_TARGET.dir/platform/drivers/src/uart/uart_driver.c.o
[build] [27/30 90% :: 1.013] Linking C static library libUTILITY_PRINT.a
[build] [28/30 93% :: 1.160] Linking C static library libGENERATED_SDK_TARGET.a
[build] [29/30 96% :: 1.299] Linking C static library libGENERATED_CONFIG_TARGET.a
[build] [30/30 100% :: 5.941] Linking C executable Gpio_Interrupt_Demo.elf
[build] text data bss dec hex filename
[build] 22672 44 2148 24864 6120 Gpio_Interrupt_Demo.elf
[build] this is test script
[build] end
[driver] 生成元串: 00:00:00:105
[build] 生成已完成, 退出代码为 0
```

大纲
时间轴
生成