新生培训

TALK:

c/c++、cmake、gcc、embed...

2025.11.2

---

who am I?

github: woquchonglang
新框架开发者之一
工程组电控

---

what is embed?

```cpp
#include <print>

int main() {
    std::println("hello world");
}
```

这段代码能在单片机运行吗？
如果能，那会怎么运行？

---

vscode不是编译器！

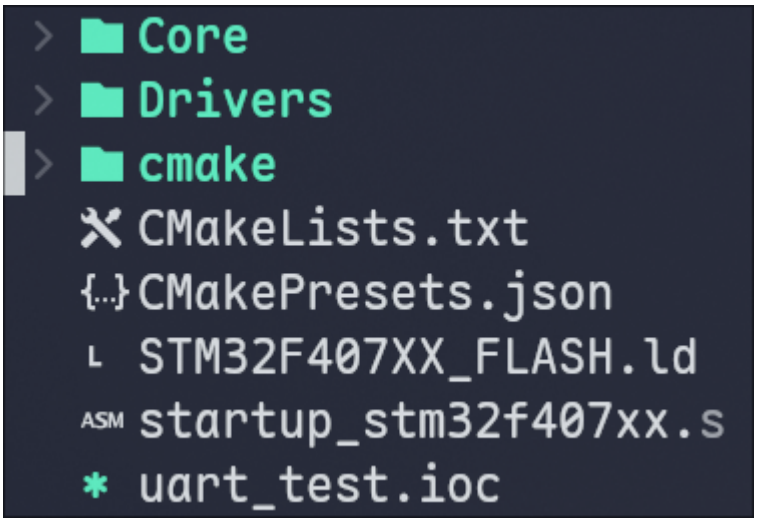
vscode只是一个文本编译器

编译器 | 链接器

在嵌入式中常用的
编译器是
arm-none-eabi-gcc
链接用的是
arm-none-eabi-ld

---

cubemx 生成的文件树



---

cmake

A Powerful Software Build System

```
g++ -o out main.cpp
```

```
cmake_minmum_required(VERSION 3.0)
project(main)
add_executable(out main.cpp)
```

```
cmake -B build -G Ninja
ninja -C build
```

- 通常用于模块化配置或共享代码。

```
cmake_minimum_required(VERSION 3.22)
# Enable CMake support for ASM and C languages
enable_language(C ASM)
# STM32CubeMX generated symbols (macros)
set(MX_Defines_Syms
   USE_HAL_DRIVER
   STM32F070x6
    $<$<CONFIG:Debug>:DEBUG>
)

# STM32CubeMX generated include paths
set(MX_Include_Dirs
    ${CMAKE_SOURCE_DIR}/Core/Inc
    ${CMAKE_SOURCE_DIR}/Drivers/STM32F0xx_HAL_Driver/Inc
    ${CMAKE_SOURCE_DIR}/Drivers/STM32F0xx_HAL_Driver/Inc/Legacy
    ${CMAKE_SOURCE_DIR}/Drivers/CMSIS/Device/ST/STM32F0xx/Include
    ${CMAKE_SOURCE_DIR}/Drivers/CMSIS/Include
)

# STM32CubeMX generated application sources
set(MX_Application_Src
    ${CMAKE_SOURCE_DIR}/Core/Src/main.c
```

```
    ${CMAKE_SOURCE_DIR}/Core/Src/gpio.c
    ${CMAKE_SOURCE_DIR}/Core/Src/adc.c
    ${CMAKE_SOURCE_DIR}/Core/Src/i2c.c
    ${CMAKE_SOURCE_DIR}/Core/Src/stm32f0xx_it.c
    ${CMAKE_SOURCE_DIR}/Core/Src/stm32f0xx_hal_msp.c
    ${CMAKE_SOURCE_DIR}/Core/Src/sysmem.c
    ${CMAKE_SOURCE_DIR}/Core/Src/syscalls.c
    ${CMAKE_SOURCE_DIR}/startup_stm32f070x6.s
)

# STM32 HAL/LL Drivers
set(STM32_Drivers_Src
    ${CMAKE_SOURCE_DIR}/Core/Src/system_stm32f0xx.c
    ${CMAKE_SOURCE_DIR}/Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_adc.c
    ${CMAKE_SOURCE_DIR}/Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_adc_ex.c
    ${CMAKE_SOURCE_DIR}/Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_rcc.c
    ${CMAKE_SOURCE_DIR}/Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_rcc_ex.c
    ${CMAKE_SOURCE_DIR}/Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal.c
    ${CMAKE_SOURCE_DIR}/Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_i2c.c
    ${CMAKE_SOURCE_DIR}/Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_i2c_ex.c
    ${CMAKE_SOURCE_DIR}/Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_gpio.c
    ${CMAKE_SOURCE_DIR}/Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_dma.c
    ${CMAKE_SOURCE_DIR}/Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_cortex.c
    ${CMAKE_SOURCE_DIR}/Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_pwr.c
    ${CMAKE_SOURCE_DIR}/Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_pwr_ex.c
    ${CMAKE_SOURCE_DIR}/Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_flash.c
    ${CMAKE_SOURCE_DIR}/Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_flash_ex.c
    ${CMAKE_SOURCE_DIR}/Drivers/STM32F0xx_HAL_Driver/Src/stm32f0xx_hal_exti.c
)

# Drivers Midllewares



# Link directories setup
set(MX_LINK_DIRS

)
# Project static libraries
set(MX_LINK_LIBS
    STM32_Drivers

)
# Interface library for includes and symbols
add_library(stm32cubemx INTERFACE)
target_include_directories(stm32cubemx INTERFACE ${MX_Include_Dirs})
target_compile_definitions(stm32cubemx INTERFACE ${MX_Defines_Syms})

# Create STM32_Drivers static library
add_library(STM32_Drivers OBJECT)
target_sources(STM32_Drivers PRIVATE ${STM32_Drivers_Src})
target_link_libraries(STM32_Drivers PUBLIC stm32cubemx)


# Add STM32CubeMX generated application sources to the project
target_sources(${CMAKE_PROJECT_NAME} PRIVATE ${MX_Application_Src})

# Link directories setup
target_link_directories(${CMAKE_PROJECT_NAME} PRIVATE ${MX_LINK_DIRS})

# Add libraries to the project
target_link_libraries(${CMAKE_PROJECT_NAME} ${MX_LINK_LIBS})

# Add the map file to the list of files to be removed with 'clean' target
set_target_properties(${CMAKE_PROJECT_NAME} PROPERTIES ADDITIONAL_CLEAN_FILES
${CMAKE_PROJECT_NAME}.map)
```

```
# Validate that STM32CubeMX code is compatible with C standard
if((CMAKE_C_STANDARD EQUAL 90) OR (CMAKE_C_STANDARD EQUAL 99))
    message(ERROR "Generated code requires C11 or higher")
endif()
```

- 通常用于模块化配置或共享代码。

```
cmake_minimum_required(VERSION 3.22)

#
# This file is generated only once, and is not re-generated if converter is
# called multiple times.
#
# User is free to modify the file as much as necessary
#

# Setup compiler settings
set(CMAKE_C_STANDARD 11)
set(CMAKE_C_STANDARD_REQUIRED ON)
set(CMAKE_C_EXTENSIONS ON)

# Define the build type
if(NOT CMAKE_BUILD_TYPE)
  set(CMAKE_BUILD_TYPE "Debug")
endif()

# Set the project name
set(CMAKE_PROJECT_NAME 5VPOWER)

# Include toolchain file
include("cmake/gcc-arm-none-eabi.cmake")
# include("cmake/clang.cmake")

# Enable compile command to ease indexing with e.g. clangd
set(CMAKE_EXPORT_COMPILE_COMMANDS TRUE)

# Core project settings
project(${CMAKE_PROJECT_NAME})
message("Build type: " ${CMAKE_BUILD_TYPE})

# Enable CMake support for ASM and C languages
enable_language(C ASM)

# Create an executable object type
add_executable(${CMAKE_PROJECT_NAME})

# Add STM32CubeMX generated sources
add_subdirectory(cmake/stm32cubemx)

# Link directories setup
target_link_directories(
  ${CMAKE_PROJECT_NAME} PRIVATE # Add user defined library search paths
)

# Add sources to executable
target_sources(
  ${CMAKE_PROJECT_NAME} PRIVATE "USERCODE/adc_input.c"
                                # Add user sources here
)
```

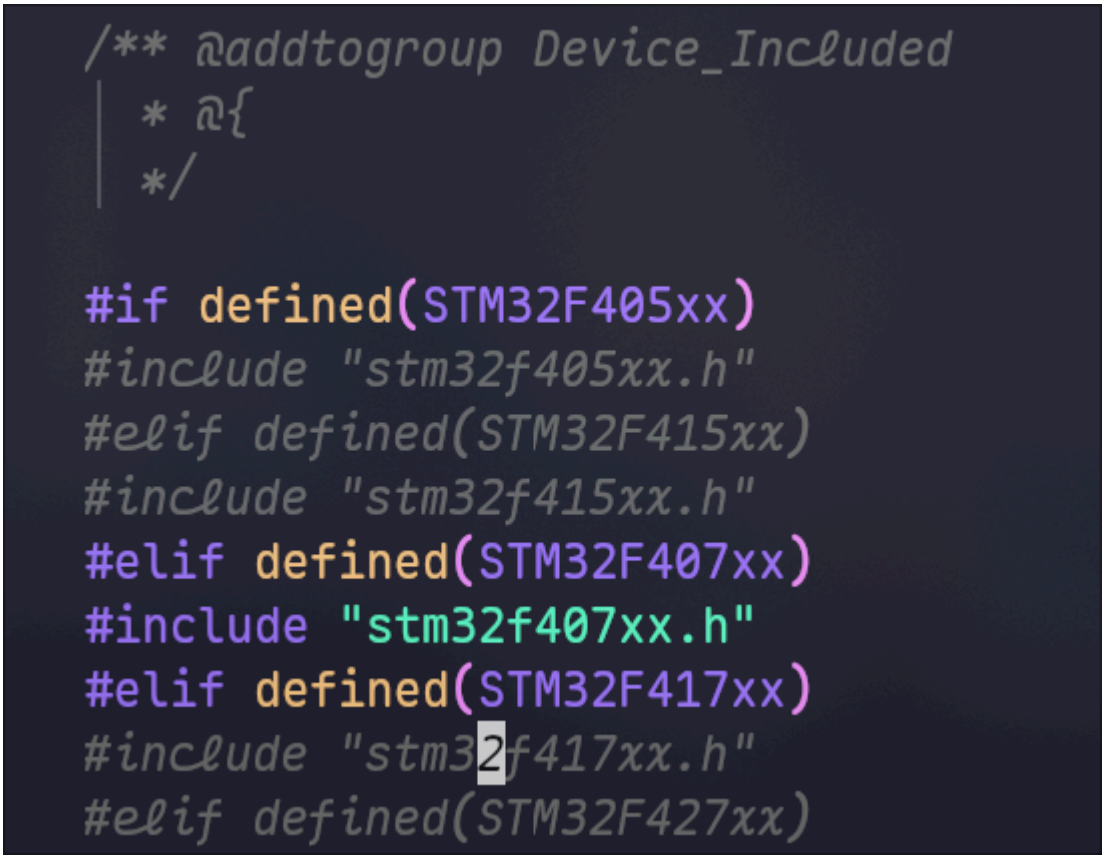
```
# Add include paths
target_include_directories(
  ${CMAKE_PROJECT_NAME} PRIVATE "USERCODE"
                                # Add user defined include paths
)

# Add project symbols (macros)
target_compile_definitions(
  ${CMAKE_PROJECT_NAME} PRIVATE # Add user defined symbols
)

# Add linked libraries
target_link_libraries(
  ${CMAKE_PROJECT_NAME} stm32cubemx # Add user defined libraries
)
```

- 可以包含变量、函数、宏等定义。

/Drivers/...



- 常用于将通用的 CMake 逻辑提取到单独的文件中，以便复用。

```
macro(FIND_INCLUDE_DIR result curdir)
  file(GLOB_RECURSE children "${curdir}/*.hpp" "${curdir}/*.h")
  set(dirlist "")
  foreach(child ${children})
    string(REGEX REPLACE "(.*)/.*" "\\1" LIB_NAME ${child})
    if(IS_DIRECTORY ${LIB_NAME})
```

```
      list(FIND dirlist ${LIB_NAME} list_index)

      if(${list_index} LESS 0)
        list(APPEND dirlist ${LIB_NAME})
      endif()
    endif()
  endforeach()
  set(${result} ${dirlist})
endmacro()
```

内存地址

```
/* Specify the memory areas */
MEMORY
{
RAM (xrw)      : ORIGIN = 0x20000000, LENGTH = 6K
FLASH (rx)      : ORIGIN = 0x8000000, LENGTH = 32K
}
```

```
Memory region         Used Size  Region Size  %age Used
           RAM:        1728 B         6 KB     28.12%
         FLASH:        8356 B        32 KB     25.50%
```

```
arm-none-eabi-readelf ./build/xxx.elf -h
arm-none-eabi-readelf ./build/xxx.elf -S
arm-none-eabi-readelf ./build/xxx.elf -s
```

烧录

```
openocd -f interface/cmsis-dap.cfg -f target/stm32f1x.cfg  -c 'program ./build/test.elf verify
reset exit'
```

程序运行

STM32F103xx_FLASH.ld

```
/* Entry Point */
ENTRY(Reset_Handler)
```

startup_stm32f103xx.s

```
/**
 * @brief  This is the code that gets called when the processor first
 *         starts execution following a reset event. Only the absolutely
 *         necessary set is performed, after which the application
 *         supplied main() routine is called.
 * @param  None
 * @retval : None
*/

  .section .text.Reset_Handler
  .weak Reset_Handler
  .type Reset_Handler, %function
Reset_Handler:
  ldr   r0, =_estack
  mov   sp, r0          /* set stack pointer */
```

```
/* Call static constructors */
  bl __libc_init_array
/* Call the application's entry point.*/
  bl main
```

外设与中断

```
/******************************************************************************
*
* The minimal vector table for a Cortex M0.  Note that the proper constructs
* must be placed on this to ensure that it ends up at physical address
* 0x0000.0000.
*
******************************************************************************/
   .section .isr_vector,"a",%progbits
  .type g_pfnVectors, %object
  .size g_pfnVectors, .-g_pfnVectors


g_pfnVectors:
  .word  _estack
  .word  Reset_Handler
  .word  NMI_Handler
  .word  HardFault_Handler
  .word  0
  .word  0
  .word  0
  .word  0
  .word  0
  .word  0
  .word  0
  .word  SVC_Handler
  .word  0
  .word  0
```

```
  .word  PendSV_Handler
  .word  SysTick_Handler
  .word  WWDG_IRQHandler                   /* Window WatchDog              */
  .word  0                                 /* Reserved                     */
  .word  RTC_IRQHandler                    /* RTC through the EXTI line    */
  .word  FLASH_IRQHandler                  /* FLASH                        */
  .word  RCC_IRQHandler                    /* RCC                          */
  .word  EXTI0_1_IRQHandler                /* EXTI Line 0 and 1            */
  .word  EXTI2_3_IRQHandler                /* EXTI Line 2 and 3            */
  .word  EXTI4_15_IRQHandler               /* EXTI Line 4 to 15            */
  .word  0                                 /* Reserved                     */
  .word  DMA1_Channel1_IRQHandler          /* DMA1 Channel 1               */
  .word  DMA1_Channel2_3_IRQHandler        /* DMA1 Channel 2 and Channel 3 */
  .word  DMA1_Channel4_5_IRQHandler        /* DMA1 Channel 4 and Channel 5 */
  .word  ADC1_IRQHandler                   /* ADC1                         */
  .word  TIM1_BRK_UP_TRG_COM_IRQHandler    /* TIM1 Break, Update, Trigger and Commutation */
  .word  TIM1_CC_IRQHandler                /* TIM1 Capture Compare         */
  .word  0                                 /* Reserved                     */
  .word  TIM3_IRQHandler                   /* TIM3                         */
  .word  0                                 /* Reserved                     */
  .word  0                                 /* Reserved                     */
  .word  TIM14_IRQHandler                  /* TIM14                        */
  .word  0                                 /* Reserved                     */
  .word  TIM16_IRQHandler                  /* TIM16                        */
  .word  TIM17_IRQHandler                  /* TIM17                        */
  .word  I2C1_IRQHandler                   /* I2C1                         */
  .word  0                                 /* Reserved                     */
  .word  SPI1_IRQHandler                   /* SPI1                         */
  .word  0                                 /* Reserved                     */
  .word  USART1_IRQHandler                 /* USART1                       */
  .word  USART2_IRQHandler                 /* USART2                       */
  .word  0                                 /* Reserved                     */
  .word  0                                 /* Reserved                     */
  .word  USB_IRQHandler                    /* USB                          */

/*******************************************************************************
*
* Provide weak aliases for each Exception handler to the Default_Handler.
* As they are weak aliases, any function with the same name will override
* this definition.
*
*******************************************************************************/

  .weak      NMI_Handler
  .thumb_set NMI_Handler,Default_Handler

  .weak      HardFault_Handler
  .thumb_set HardFault_Handler,Default_Handler

  .weak      SVC_Handler
  .thumb_set SVC_Handler,Default_Handler

  .weak      PendSV_Handler
  .thumb_set PendSV_Handler,Default_Handler

  .weak      SysTick_Handler
  .thumb_set SysTick_Handler,Default_Handler

  .weak      WWDG_IRQHandler
  .thumb_set WWDG_IRQHandler,Default_Handler

  .weak      RTC_IRQHandler
  .thumb_set RTC_IRQHandler,Default_Handler

  .weak      FLASH_IRQHandler
  .thumb_set FLASH_IRQHandler,Default_Handler
```

```
.weak      RCC_IRQHandler
.thumb_set RCC_IRQHandler,Default_Handler

.weak      EXTI0_1_IRQHandler
.thumb_set EXTI0_1_IRQHandler,Default_Handler

.weak      EXTI2_3_IRQHandler
.thumb_set EXTI2_3_IRQHandler,Default_Handler

.weak      EXTI4_15_IRQHandler
.thumb_set EXTI4_15_IRQHandler,Default_Handler

.weak      DMA1_Channel1_IRQHandler
.thumb_set DMA1_Channel1_IRQHandler,Default_Handler

.weak      DMA1_Channel2_3_IRQHandler
.thumb_set DMA1_Channel2_3_IRQHandler,Default_Handler

.weak      DMA1_Channel4_5_IRQHandler
.thumb_set DMA1_Channel4_5_IRQHandler,Default_Handler

.weak      ADC1_IRQHandler
.thumb_set ADC1_IRQHandler,Default_Handler

.weak      TIM1_BRK_UP_TRG_COM_IRQHandler
.thumb_set TIM1_BRK_UP_TRG_COM_IRQHandler,Default_Handler

.weak      TIM1_CC_IRQHandler
.thumb_set TIM1_CC_IRQHandler,Default_Handler

.weak      TIM3_IRQHandler
.thumb_set TIM3_IRQHandler,Default_Handler

.weak      TIM14_IRQHandler
.thumb_set TIM14_IRQHandler,Default_Handler

.weak      TIM16_IRQHandler
.thumb_set TIM16_IRQHandler,Default_Handler

.weak      TIM17_IRQHandler
.thumb_set TIM17_IRQHandler,Default_Handler

.weak      I2C1_IRQHandler
.thumb_set I2C1_IRQHandler,Default_Handler

.weak      SPI1_IRQHandler
.thumb_set SPI1_IRQHandler,Default_Handler

.weak      USART1_IRQHandler
.thumb_set USART1_IRQHandler,Default_Handler

.weak      USART2_IRQHandler
.thumb_set USART2_IRQHandler,Default_Handler

.weak      USB_IRQHandler
.thumb_set USB_IRQHandler,Default_Handler
```

Learning Resources

C++ 教程 | 菜鸟教程
cppreference.com

rust-embedded-book
HNUYueLuRM/basic_framework
RoboMaster/RoboRTS-Firmware