



# 基于 STM32Cube MX 开发的 TencentOS-Tiny 软件包 TencentOS-Tiny software package based on STM32Cube MX

2021/11/10

东南大学

导师：汪礼超

学员：崔林威

Email: 1797878653@qq.com

<https://github.com/OpenAtomFoundation/TencentOS-tiny>

## 摘要

腾讯物联网操作系统（TencentOS tiny）是腾讯面向物联网领域开发的实时操作系统，具有低功耗，低资源占用，模块化，可裁剪等特性。TencentOS tiny 提供了最精简的 RTOS 内核，内核组件可裁剪可配置，可灵活移植到多种终端 MCU 上；基于 RTOS 内核提供了 COAP/MQTT/TLS/DTLS 等常用物联网协议栈及组件，方便用户快速接入腾讯云物联网通信 IoT Hub；同时，为物联网终端厂家提供一站式软件解决方案，方便各种物联网设备快速接入腾讯云，可支撑智慧城市、智能穿戴、车联网等多种行业应用。

为了减少开发人员移植 TencentOS tiny 到 STM32 系列单片机上的开发时间，研究了 **STM32Cube MX 软件包的制作**，基于 STM32 PackCreator 完成了 TencentOS Tiny 软件包的封装，并在 STM32Cube IDE 和 MDK-ARM v5 上完成了软件包的**移植测试**，从而用户在安装本软件包后，能够使用 pack 在 STM32Cube MX 上直接生成适合不同 MCU 的 TencentOS Tiny 工程。

# 目录

1、STM32 Cube MX 软件包介绍 .....	4
1.1 软件包简介.....	4
1.2 软件包制作.....	6
1.2.1 软件包开发过程.....	6
1.2.2 PDSC 文件的编写.....	6
1.2.3 生成软件包.....	错误!未定义书签。
2、TencentOS-tiny 软件包 .....	10
2.1 软件包内容.....	10
2.2 软件包安装.....	10
3、软件包测试 .....	12
3.1 ARM 内核移植 TencentOS tiny 软件包.....	12
3.2 STM32 不依赖裸机工程移植 .....	15
3.3 单片机裸机工程移植.....	18
4、总结 .....	错误!未定义书签。
5、开发参考 .....	25
6、附录-移植配置参考 .....	错误!未定义书签。
6.1 MDK5.14 版本移植到 ARM 内核 .....	错误!未定义书签。
6.1.1 Cortex-M0 内核移植.....	错误!未定义书签。
6.1.2 Cortex-M0+内核移植.....	错误!未定义书签。
6.1.3 Cortex-M3 内核移植.....	错误!未定义书签。
6.1.4 Cortex-M4 内核移植.....	错误!未定义书签。
6.1.5 Cortex-M7 内核移植.....	错误!未定义书签。
6.2 MDK5.14 版本移植到基于 ARM 内核的芯片 .....	错误!未定义书签。
6.2.1 移植到 stm32f103c8 芯片.....	错误!未定义书签。
6.2.2 移植到 stm32f767igt 芯片 .....	错误!未定义书签。
6.3 MDK5.30 和 MDK5.35 版本移植（Cortex-M0+、0、3、4、7 内核和芯片） .....	错误!未定义书签。
6.4 MDK5.30 和 MDK5.35 版本移植（Cortex-M23、33） .....	错误!未定义书签。
6.4.1 Cortex-M23 内核移植.....	错误!未定义书签。
6.4.2 Cortex-M33 内核移植.....	错误!未定义书签。

# 1、STM32 Cube MX 软件包介绍

## 1.1 软件包简介

在进行嵌入式软件开发时，ARM 为我们提供了软件包功能，能够将软件算法等模块进行集成封装，从而方便第三方用户使用。ARM 软件包能够为微控制器设备和开发板提供支持，包含软件组件（Software Component）如驱动程序和中间件，还可以包含示例项目和代码模板等，主要有以下类型的软件包：

(1) 器件系列包（Device Family Pack）：由硅供应商或工具供应商生成，为特定的目标微控制器创建软件应用提供支持；

(2) 板级支持包（Board Support Pack）：由电路板供应商发布，为安装在电路板上的外围硬件提供软件支持。

(3) CMSIS 软件包：由 ARM 提供，包括对 CMSIS 核心、DSP 和 RTOS 的支持；

(4) 中间件包（Middleware Pack）：由芯片供应商、工具供应商或第三方创建；通过提供对常用软件组件（如软件堆栈、特殊硬件库等）的软件集成，从而减少开发时间；

(5) 内部组件（In-house components）：由工具用户开发，用于内部或外部分发。

软件组件包括以下几部分：

(1) 源代码、库、头文件/配置文件和文档；

(2) 完整的示例项目，展示了软件组件的使用，可以下载并在评估硬件上执行；

(3) 代码模板，方便使用软件组件。

一个完整的软件包是一个 ZIP 文件，包含所有需要的软件库和文件，以及一个包含软件包所有信息的包描述文件（PDSC 文件），ARM 软件包的结构是在 CMSIS 中定义的 (<http://www.keil.com/CMSIS/Pack>)。

STM32 CubeMX 是 ST 公司推出了专门用于生成 STM32 的 HAL 代码的代码生成软件，可以通过可视化界面完成工程的配置，同时生成能够在 STM32 CubeMX、Keil 等软件中运行的工程。STM32 Cube MX 软件包是在 ARM 软件包的基础上，结合 ST 公司提供的软件包生成工具 STM32 PackCreator 制作的，利用这个软件我们可以很方便地以图形化界面的方式进行软件包的制作。

如下图所示，使用 STM32PackCreator 需要我们的操作系统安装 Java™ Runtime Environment (JRE)和 JavaFX™，目前 STM32CubeMX 已经更新到了 6.3 版本，但这个版本并没有集成 JavaFX，最为方便的方式是安装 STM32 CubeMX 6.2 版本，省去了 JavaFX 的配置问题。

### System requirements

STM32PackCreator requires a Java™ Runtime Environment (JRE) to execute.

- Starting with STM32CubeMX 6.2 (STM32PackCreator 3.2) there is no need to install a JRE.
- The JRE installed with STM32CubeMX is used to run STM32PackCreator.

- The JRE version constraints are:

- 64-bit version is mandatory.
- 32-bit version is not supported.

- JRE must support JavaFX™.

- The minimum JRE version is 1.8\_45 (known limitation with 1.8\_251).
- Version 11 is supported.
- Versions 7, 9, 10, 12, and upper are not supported.

- STMicroelectronics promotes the use of the following JREs:

- Oracle® JRE, subject to a license fee.
- Amazon Corretto™ JRE, no-cost solution based on OpenJDK. The JDK installer is recommended.
- STM32PackCreator operation is not guaranteed with other JREs.

CSDN @linwei\_Cui

图 1.1 STM32PackCreator 使用要求

具体步骤如下：

(1) 安装 STM32CubeMX6.2.0: [STM32CubeMX - STM32Cube 初始化代码生成器 - STMicroelectronics](#);



图 1.2 下载 STM32CubeMX6.2.0

(2) 双击安装目录下的 STM32PackCreator，此时会提示需要安装 JRE，点击确定，跳转到 <https://java.com/zh-CN/download/>，然后下载安装。

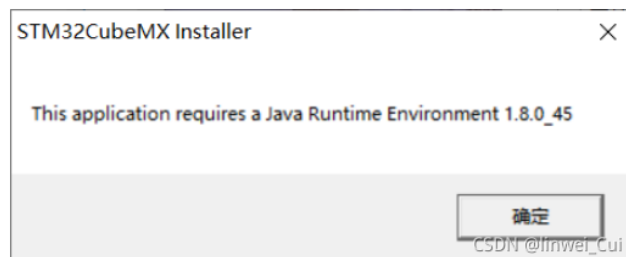


图 1.3 STM32PackCreator 使用前的配置

(3) 然后再次双击打开 STM32PackCreator.exe，STM32 PackCreator 的界面如下图所示。

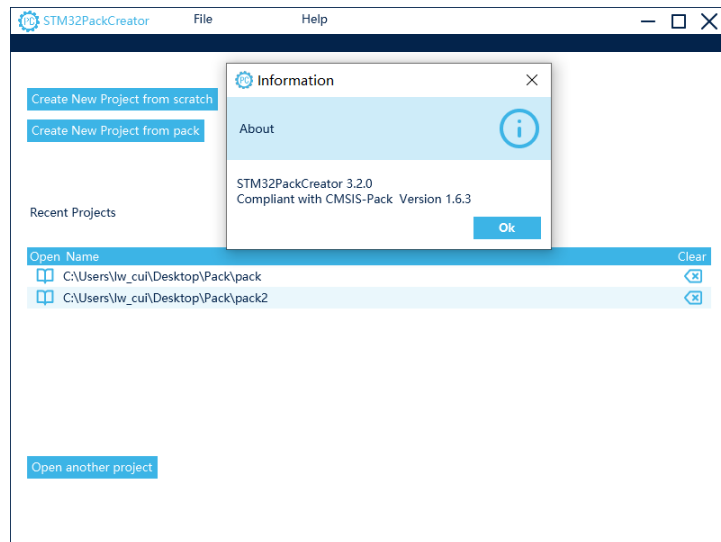


图 1.4 STM32 PackCreator 界面

## 1.2 软件包开发

### 1.2.1 软件包开发过程

软件包的开发过程相当于完成了一项产品的制作，因此引入产品生命周期管理（PLM）的概念，PLM 包括以下四个阶段：（1）概念的产生，基于软件包需求进行产品定义，并创建第一个功能原型；（2）设计，根据技术特征和要求，进行原型测试和产品的实施，通过广泛的测试验证产品的功能与规格；（3）发布，产品被制造出来并推向市场；（4）服务，对产品的维护，包括对客户的支持，最后不断优化，结束产品的周期。

在制作软件包时，主要面临以下几个过程：

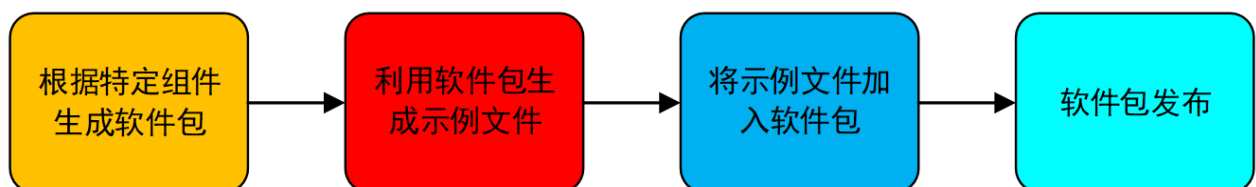


图 1.5 软件包开发流程

首先，根据特定组件生成软件包即根据需求将相应的头文件、库文件等软件组件利用 PDSC 文件进行组织，在组织完成后即可利用软件包生成工具生成对应版本的软件包，然后对新生成的软件包进行测试，给出示例测试程序，再将其包含如 PDSC 文件中，最后经测试完成后生成最终的软件包。

### 1.2.2 利用 STM32PackCreator 制作软件包

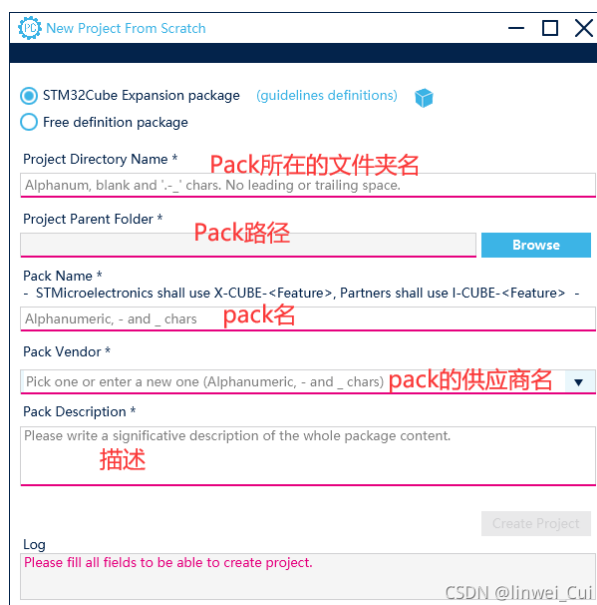
STM32PackCreator 能够通过图形化界面设计软件包，省去了我们手动编写 PDSC 文件和软件包生成脚本文件的步骤，使用起来非常方便快捷，以下是使用 STM32PackCreator 制作软件包的步骤。

(1) 打开 STM32 PackCreator 后，点击 File-new project from scratch。




图 1.6 生成新软件包

(2) 进行软件包的配置，包括软件包生成的文件夹、软件包的 vendor、name 和 description 属性。



**New Project From Scratch**

☒ STM32Cube Expansion package (guidelines definitions) 

☐ Free definition package

Project Directory Name \* **Pack所在的文件夹名**  
Alphanumeric, blank and '-' chars. No leading or trailing space.

Project Parent Folder \* **Pack路径** Browse

Pack Name \* **pack名**  
- STMicronics shall use X-CUBE-<Feature>, Partners shall use I-CUBE-<Feature> -  
Alphanumeric, - and \_ chars

Pack Vendor \* **pack的供应商名**  
Pick one or enter a new one (Alphanumeric, - and \_ chars)

Pack Description \* **描述**  
Please write a significative description of the whole package content.

Create Project

Log  
Please fill all fields to be able to create project.

CSDN @linwei\_Cui

图 1.7 软件包初始配置

(3) 配置完成后，可以看到有四个属性栏，他们分别代表软件包的基本配置、版本控制、组件内容和附加文件，如图 1.7 所示：

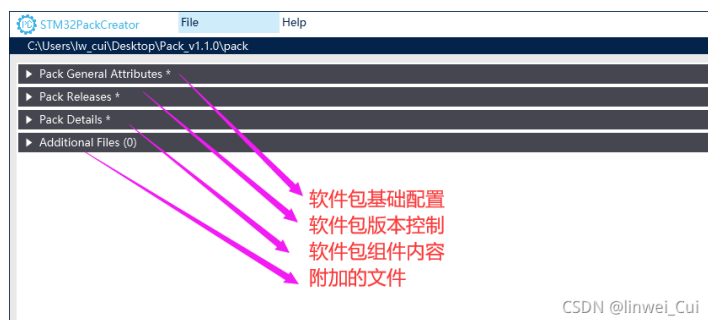
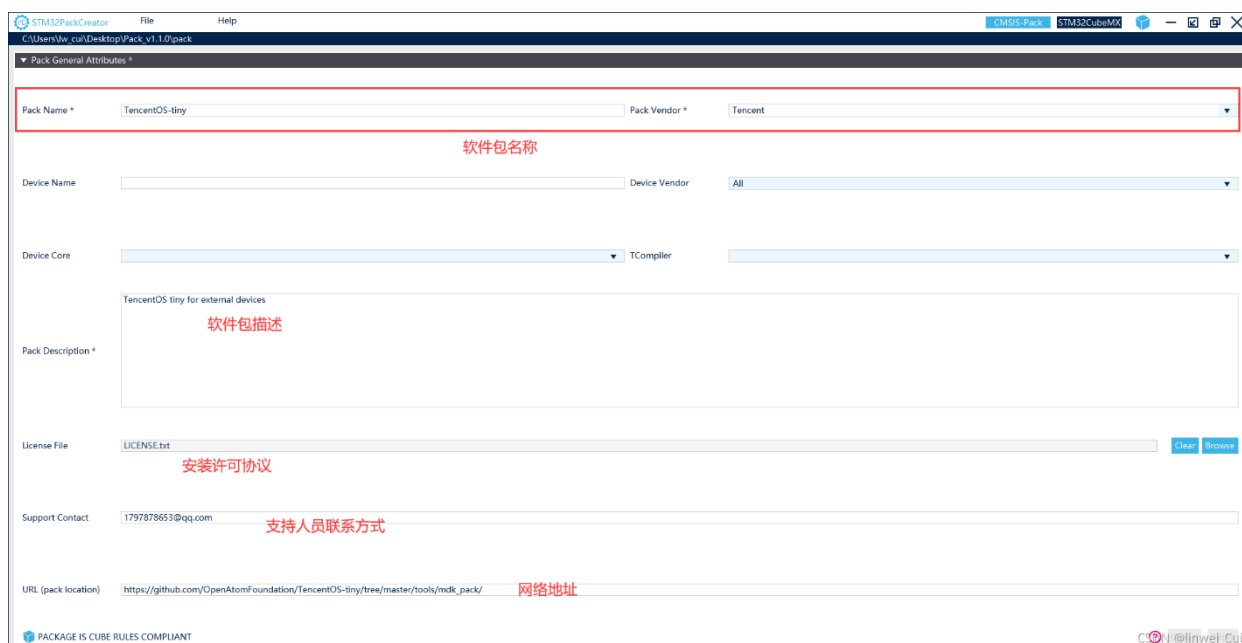


图 1.8 软件包属性栏

(4) 如下为制作 Tencent OS-tiny 软件包时的配置：



STM32PackCreator File Help CMSIS-Pack STM32CubeMX

C:\Users\lv\_cui\Desktop\Pack\_v1.1.0\pack

▼ Pack General Attributes \*

Pack Name \* **软件包名称** TencentOS-tiny Pack Vendor \* Tencent

Device Name Device Vendor All

Device Core TCompiler

Pack Description \* **软件包描述**  
TencentOS tiny for external devices

License File LICENSE.txt **安装许可协议** Clear Browse

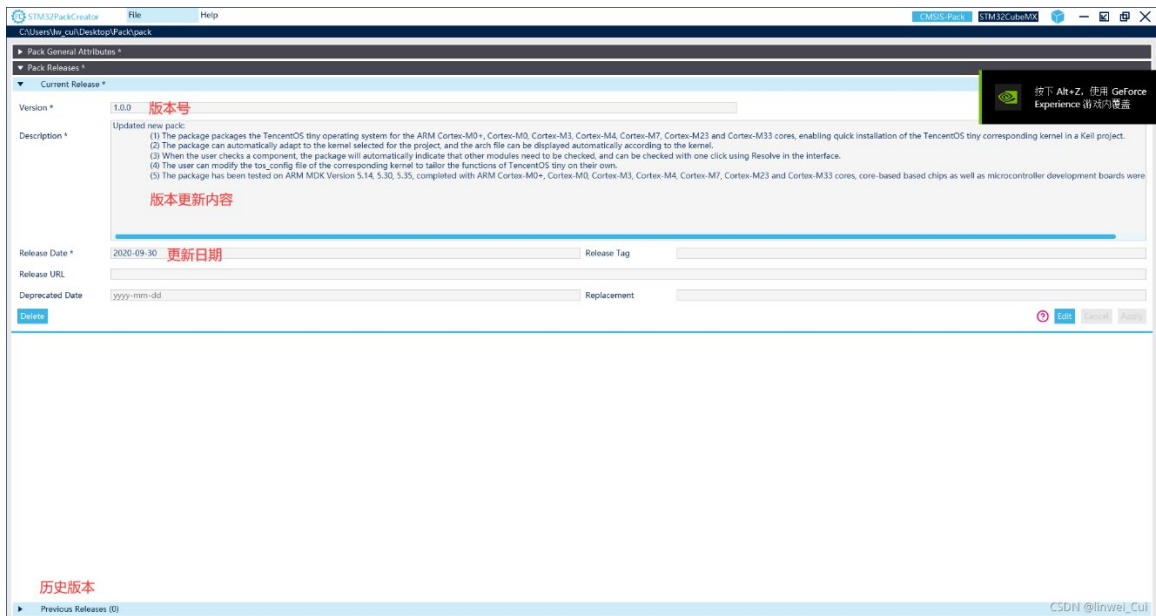
Support Contact 1797878653@qq.com **支持人员联系方式**

URL (pack location) [https://github.com/OpenAtomFoundation/TencentOS-tiny/tree/master/tools/mdk\\_pack/](https://github.com/OpenAtomFoundation/TencentOS-tiny/tree/master/tools/mdk_pack/) **网络地址**

PACKAGE IS CUBE RULES COMPLIANT

CSDN @linwei\_Cui

(a) 基础配置



(b) 软件包版本控制

图 1.9 软件包配置

(5) 制作软件包的关键在于 Pack Details 的配置，这里实际上就是手工编写 PDSC 文件的图形化界面显示，我们按照 ARM PDSC 文件的编写规范设计 condition、设计组件结构、添加组件即可。

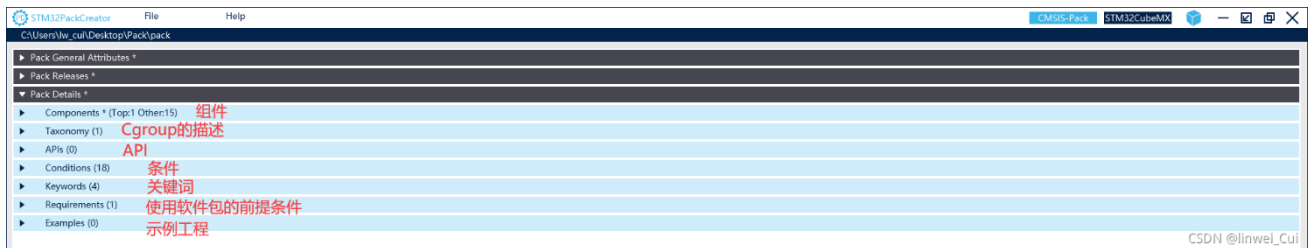


图 1.10 软件包组件配置

(6) 添加附件，在 Additional Files 中，我们需要添加上软件包的说明文件(README.md)、利用软件包生成的示例工程(Projects)和 STM32CubeMX 文件夹，此时我们点击 File-Save and Generate Pack，可以在之前设定的路径下看到软件包。然后我们再次将这个软件包也加入到 Additional Files 中，如图 1.11 所示，我们再次点击 File-Save and Generate Pack，就可以生成一个完整的 STM32CubeMX 软件包了。



图 1.11 STM32CubeMX 文件夹

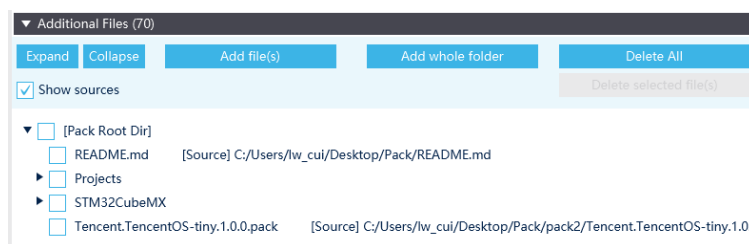


图 1.12 Additional Files 界面



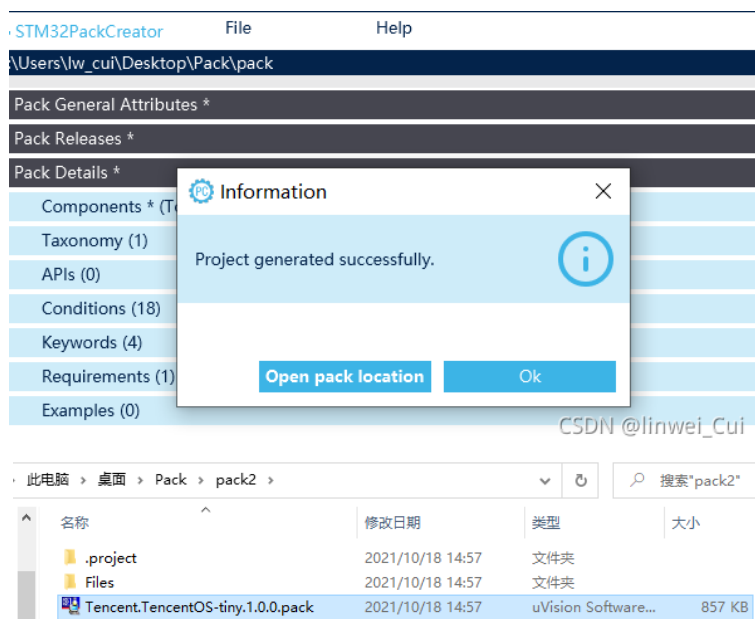


图 1.13 生成软件包

(7) 另外，STM32PackCreator 提供了参数预先配置的功能，使得我们在 STM32 Cube MX 生成工程时，可以对软件包的参数进行提前配置。首先在图 1.13(a)中添加我们需要用户提前配置的一系列参数，然后在图 1.13(b)中制作一个模板文件，此处我设置为头文件的形式，然后将之前设置的一系列参数添加过来并保存，这样用户在使用软件包生成工程的时候就可以预先配置参数，并在生成工程时自动生成名为 `tos_config` 的文件。

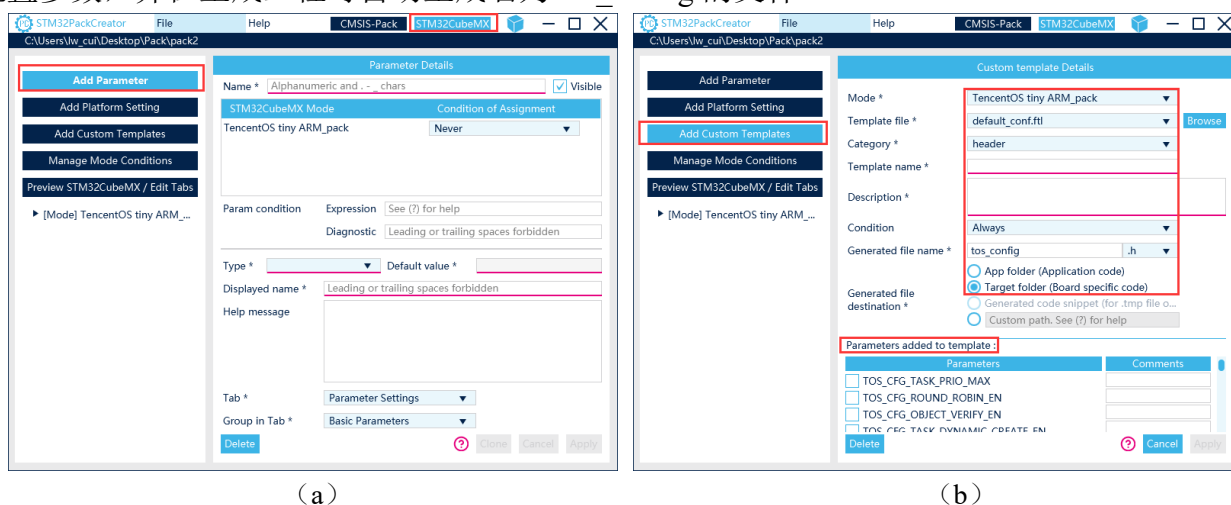


图 1.14 软件包参数配置

## 2、TencentOS-tiny 软件包

### 2.1 软件包内容

结合 TencentOS tiny 的算法架构，本文设计的软件包包括如表 2-1 所示的内容：

表 2-1 软件包内容

内容		功能
arch		包括 TencentOS-tiny\arch\arm 下内核为 Cortex-M0+、Cortex-M0、Cortex-M3、Cortex-M4、Cortex-M7、Cortex-M23、Cortex-M33 的 arch 文件
kernel		包括 TencentOS-tiny\kernel 下的 core、hal 路径中的文件、tos_config 文件
cmsis_os		对应 TencentOS-tiny\osal\cmsis_os 的文件
example	helloworld_main	用于测试软件包的 main 文件
	mcu_it.c	移植软件包时需要按照该文件对中断函数进行修改
	mcu_platform.h	用户可在此文件添加对应单片机的头文件

软件包具有以下功能：

（1）软件包针对 ARMCortex-M0+、Cortex-M0、Cortex-M3、Cortex-M4、Cortex-M7、Cortex-M23 和 Cortex-M33 内核进行了 TencentOS tiny 软件的封装，用户在安装软件包后能够快速将 TencentOS tiny 相应内核的 Keil 工程中；

（2）软件包能够自动适应用户所选的内核，arch 文件能够根据内核自动显示，从而方便用户使用；

（3）用户在勾选一个组件时，软件包会自动提示还需要勾选其他模块，并可利用界面中的 Resolve 一键勾选，防止遗漏；

（4）用户可在 STM32CubeMX 上修改 tos\_config 文件中的参数，对 TencentOS tiny 的功能进行裁剪。

### 2.2 软件包安装

接下来介绍 Tencent.TencentOS-tiny 软件包的安装，首先打开 STM32 Cube MX 软件，如图 2.1(a)所示，点击 INSTALL/REMOVE，然后将软件包拖进来，点击 I agree to all the terms of the preceding License Agreement，再点击 next 进行安装，安装完成界面如图 2.1(b)所示。

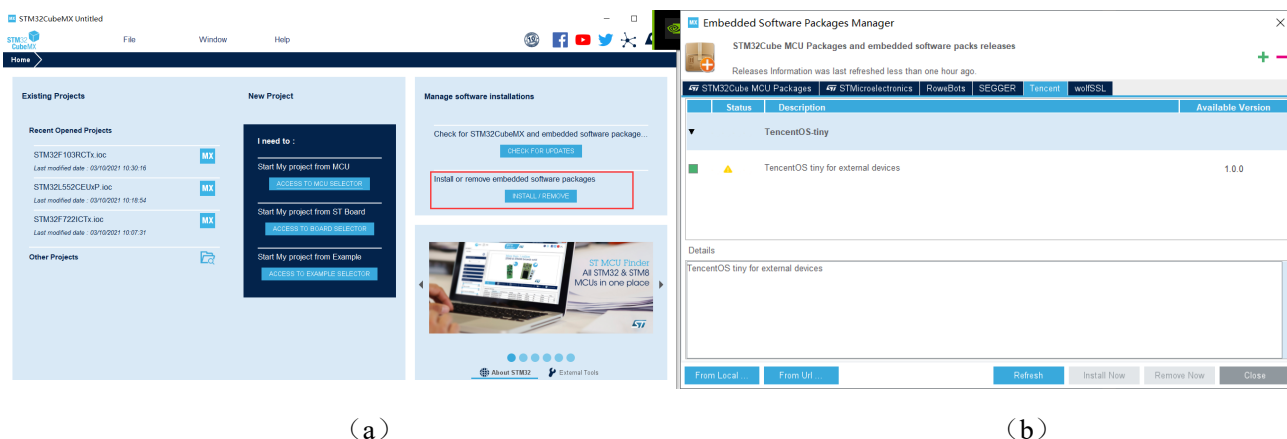
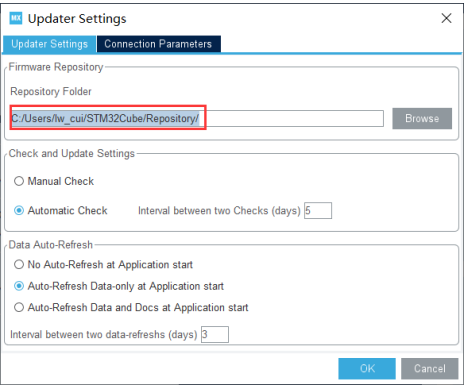


图 2.1 软件包安装

安装好的软件包可以在安装路径中查看内容，路径在 Help-Updater Settings 中查看。



(a)安装路径

用户 > lw\_cui > STM32Cube > Repository > Packs > Tencent > TencentOS-tiny > 1.0.0

搜索"1.0.0"

名称	修改日期	类型	大小
Middlewares	2021/10/18 14:57	文件夹	
Projects	2021/10/18 14:57	文件夹	
STM32CubeMX	2021/10/18 14:57	文件夹	
LICENSE.txt	2021/10/18 14:57	文本文档	18 KB
OpenAtomFoundation_TencentOS-tiny_腾讯物联网终端操作系统.html	2021/10/18 14:57	Microsoft Edge ...	247 KB
pdscpackage.xml	2021/10/18 14:57	XML 文档	2 KB
README.md	2021/10/18 14:57	MD 文件	3 KB
Tencent.TencentOS-tiny.1.0.0.pack	2021/10/18 14:57	uVision Software...	482 KB
Tencent.TencentOS-tiny.pdsc	2021/10/18 14:57	PDSC 文件	48 KB

(b) 软件包安装查看  
图 2.2 软件包安装

### 3、软件包测试

#### 3.1 STM32 MCU 移植测试（生成 STM32 CubeIDE 工程）

移植步骤如下：

（1）点击 ACCESS TO MCU SELECTOR:

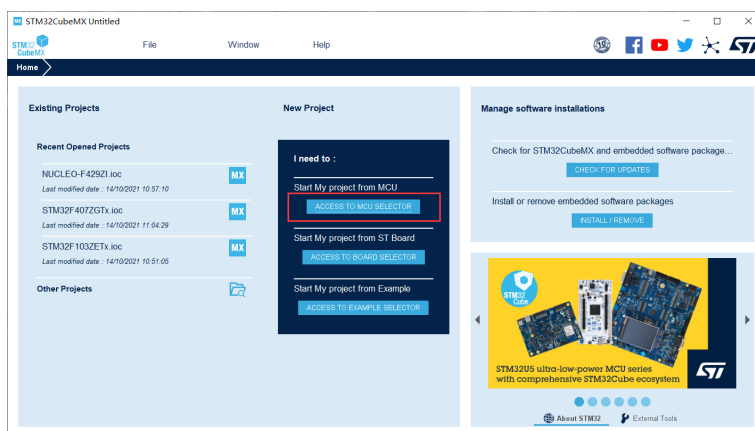


图 3.1 打开 MCU 选择的界面

（2）选择 STM32F407ZGTx，点击 Start Project:

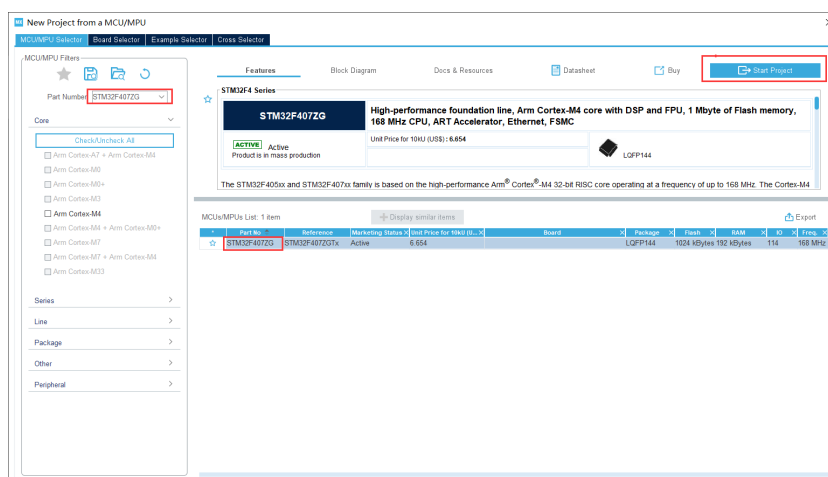


图 3.2 选择 MCU

（3）点击红框中选项

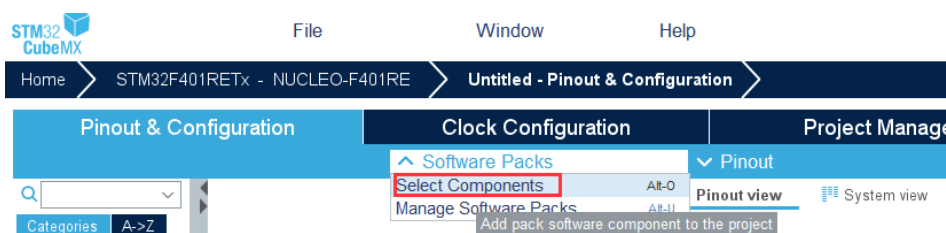
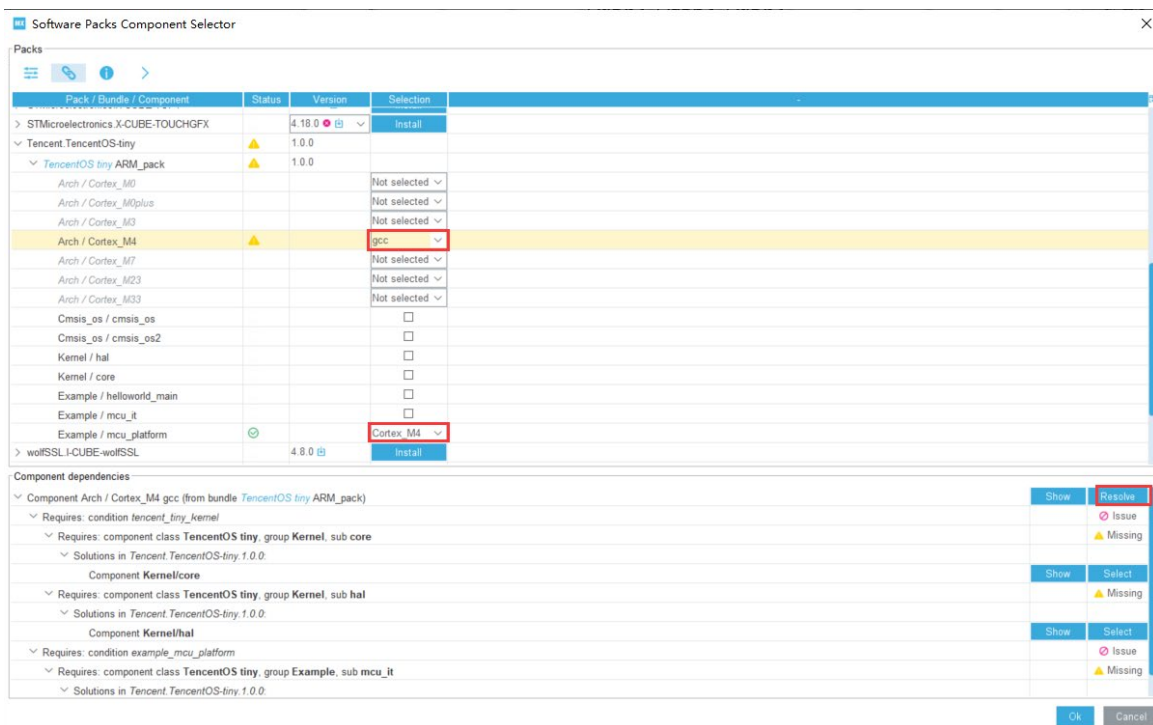


图 3.3 选择软件包

（4）此时生成使用 STM32 CubeIDE 编译的工程，首先选择 gcc 版本的 arch（如果是 MDK-ARM 版本的话就点击 armcc 版本的 arch），然后点击黄色感叹号，再点击 Resolve。之后手动添加 helloworld\_main 和 cmsis\_os，并选择对应 MCU 版本的 mcu\_platform。最后点击 ok 完成配置。



▼ Tencent.TencentOS-tiny	✅	1.0.0	
▼ TencentOS tiny ARM_pack	✅	1.0.0	
Arch / Cortex_M0			Not selected
Arch / Cortex_M0plus			Not selected
Arch / Cortex_M3			Not selected
Arch / Cortex_M4	✅		gcc
Arch / Cortex_M7			Not selected
Arch / Cortex_M23			Not selected
Arch / Cortex_M33			Not selected
Cmsis_os / cmsis_os	✅		<input checked="" type="checkbox"/>
Cmsis_os / cmsis_os2			<input type="checkbox"/>
Kernel / hal	✅		<input checked="" type="checkbox"/>
Kernel / core	✅		<input checked="" type="checkbox"/>
Example / helloworld_main	✅		<input checked="" type="checkbox"/>
Example / mcu_it	✅		<input checked="" type="checkbox"/>
Example / mcu_platform	✅		Cortex_M4

图 3.4 软件包配置

(5) 点击 Software Packs，打勾，并在下方进行软件包的参数配置。

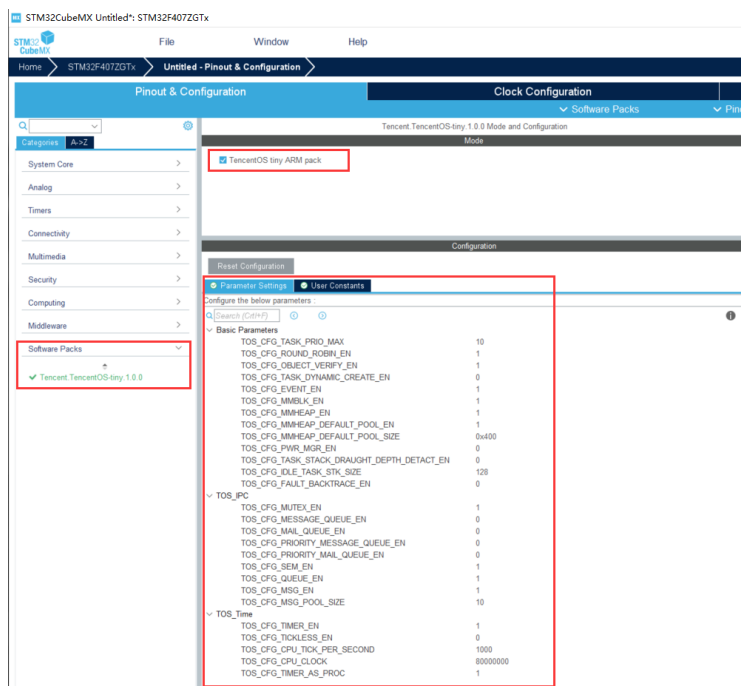


图 3.5 软件包参数配置

(6) 注释掉中断函数 PendSV()、void SysTick\_Handler(void)等，这些在软件包中都进行了配置。

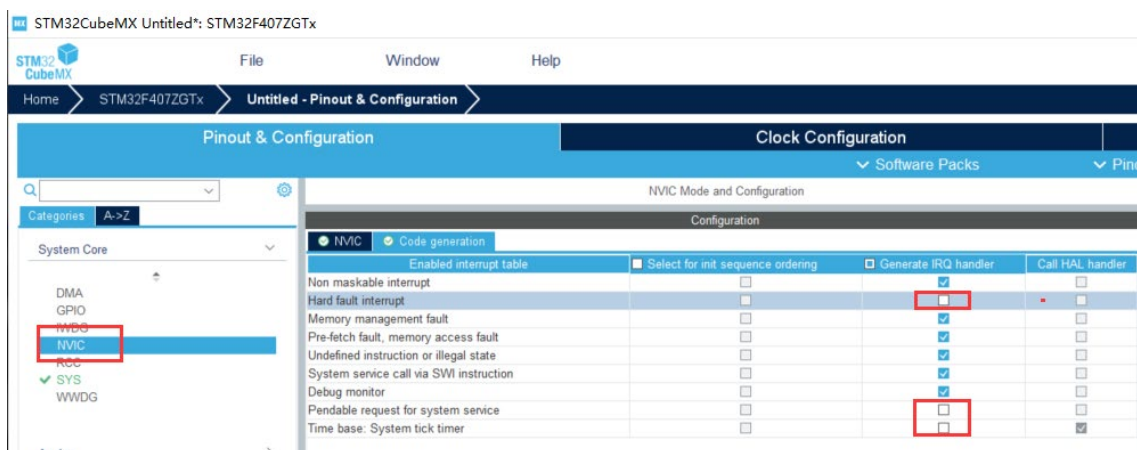


图 3.6 中断函数处理

(7) 进行生成工程配置，按如下界面进行配置，最后点击 GENERATE CODE，并点击 Open Project。

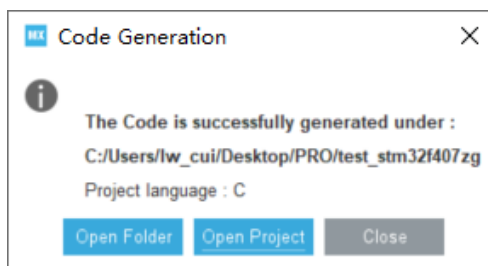
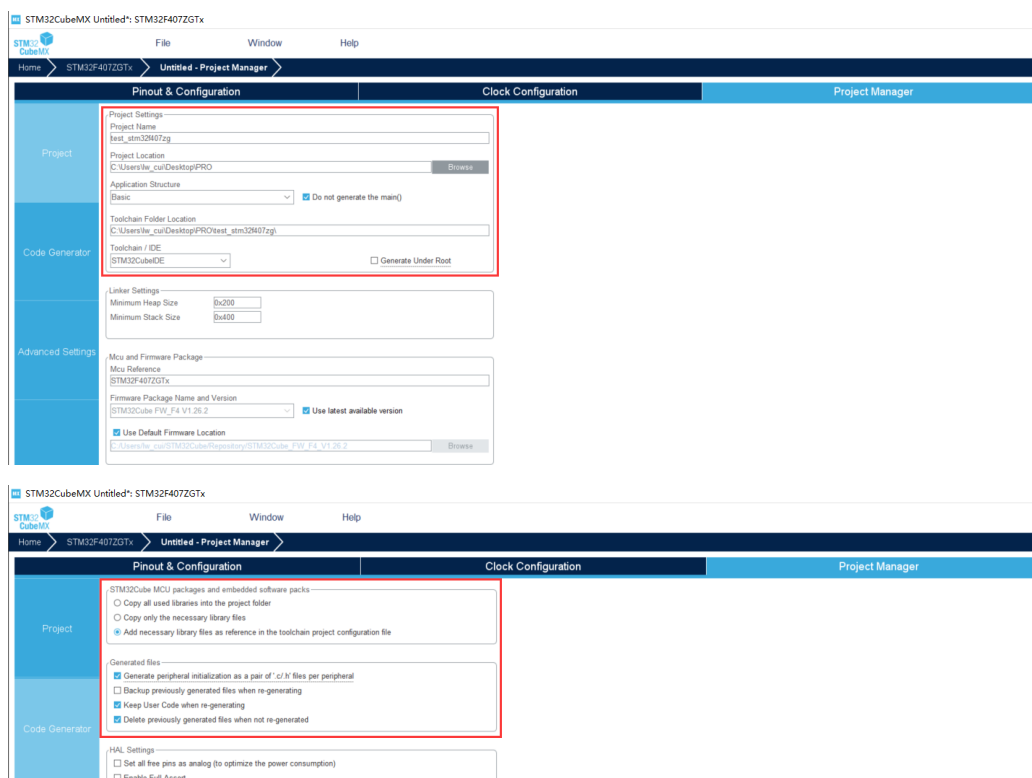


图 3.7 工程生成配置

(8) 在 tos\_config 头文件中加入头文件：#include "mcu\_platform\_M4.h"，点击 Build All，如下界面所示，编译通过，完成软件包的移植。

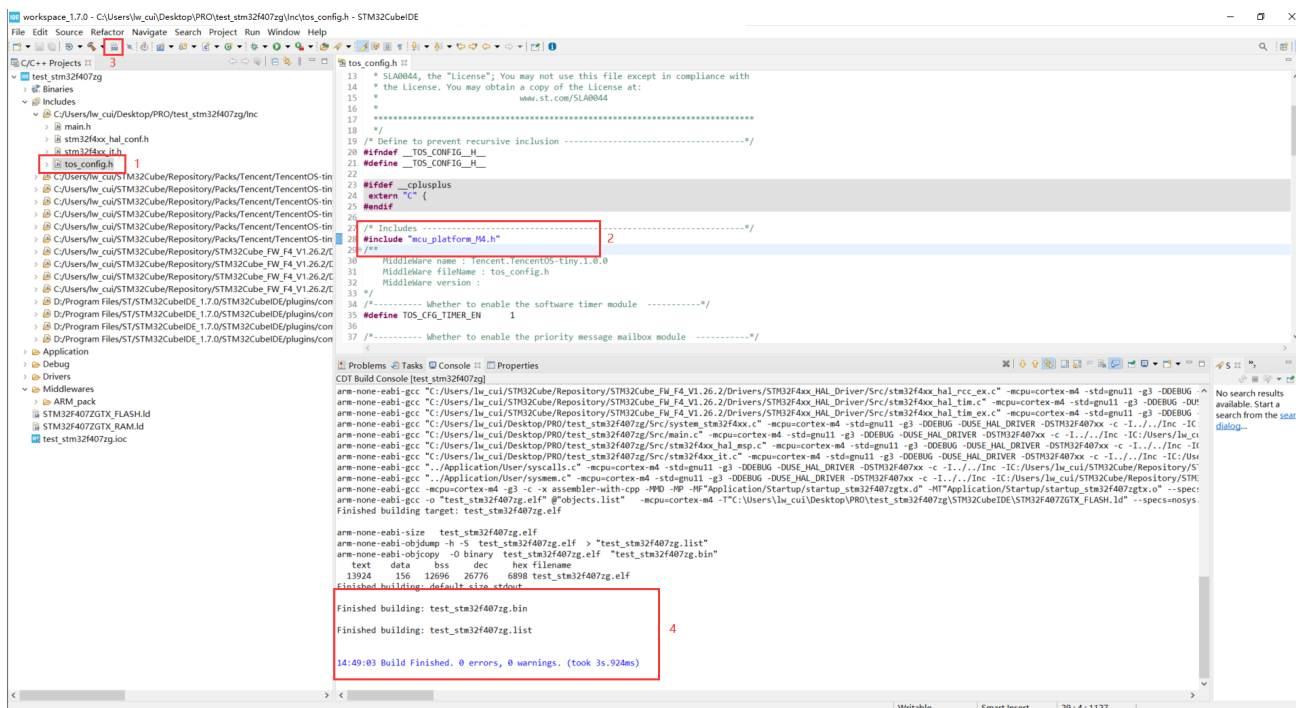


图 3.8 工程配置与编译

## 3.2 STM32 Board 移植测试（生成 MDK-ARM v5 工程）

（1）点击 ACCESS TO BOARD SELECTOR, 选择 Board 为 NUCLEO-F401RE, 点击 Start Project:

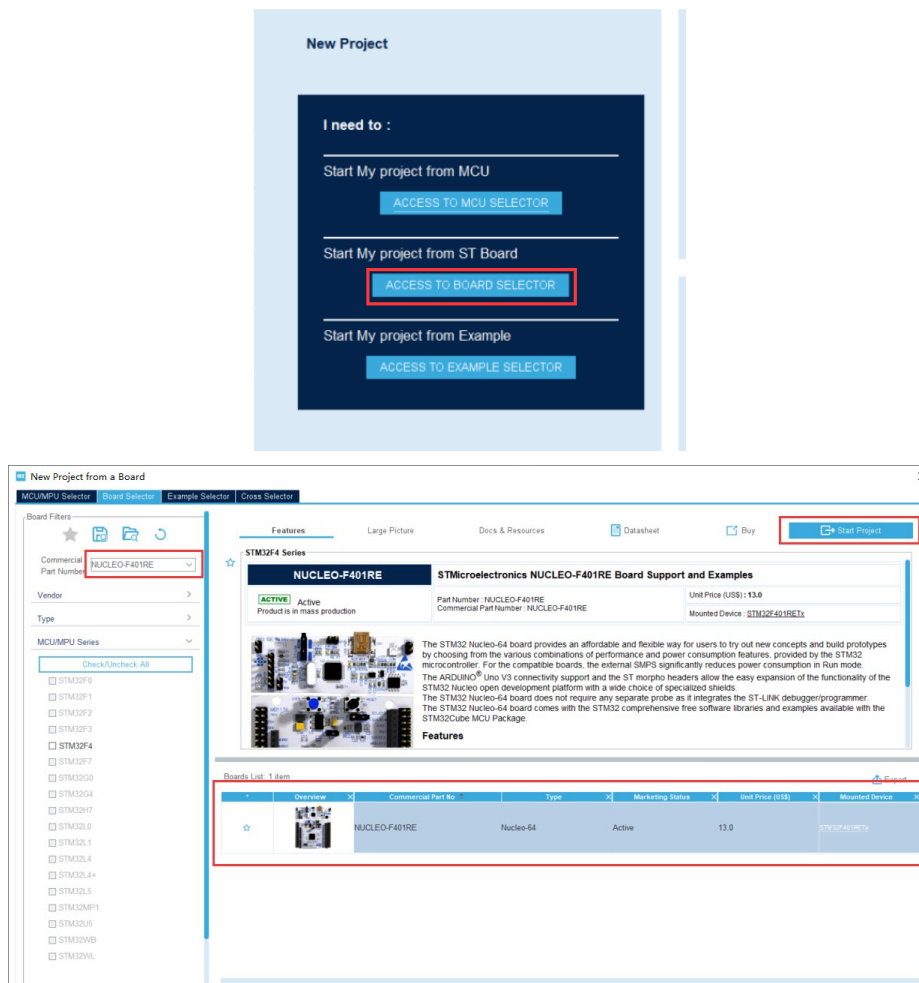


图 3.9 Board 选择



(2) 点击红框中选项

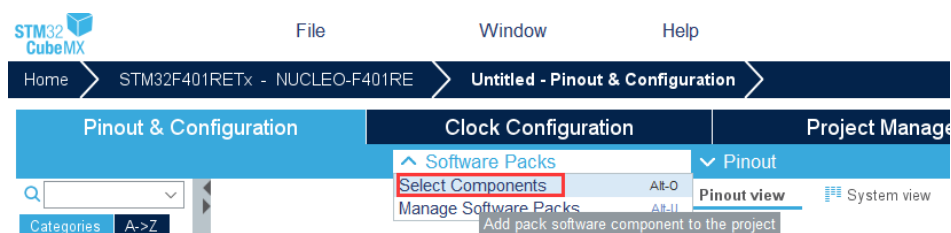


图 3.10 选择软件包

(3) 此时生成使用 MDK-ARM 编译的工程，首先选择 armcc 版本的 arch（如果是 STM32 CubeIDE 版本的话就点击 gcc 版本的 arch），然后点击黄色感叹号，再点击 Resolve。之后手动添加 helloworld\_main 和 cmsis\_os，并选择对应 MCU 版本的 mcu\_platform。最后点击 ok 完成配置。

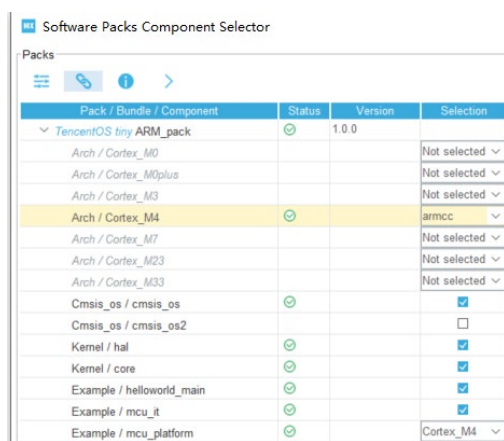
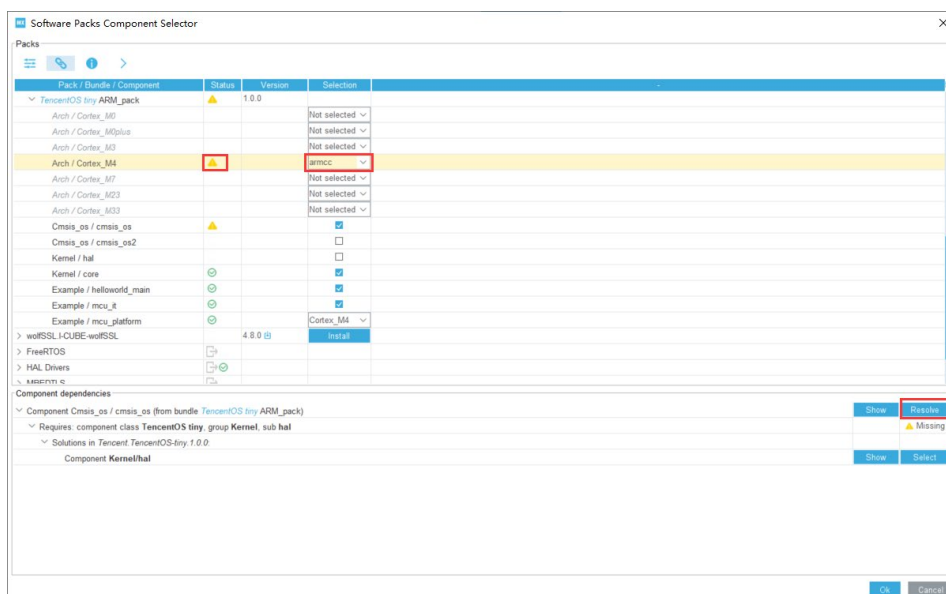


图 3.11 软件包配置

(4) 点击 Software Packs，打勾，并在下方进行软件包的参数配置。



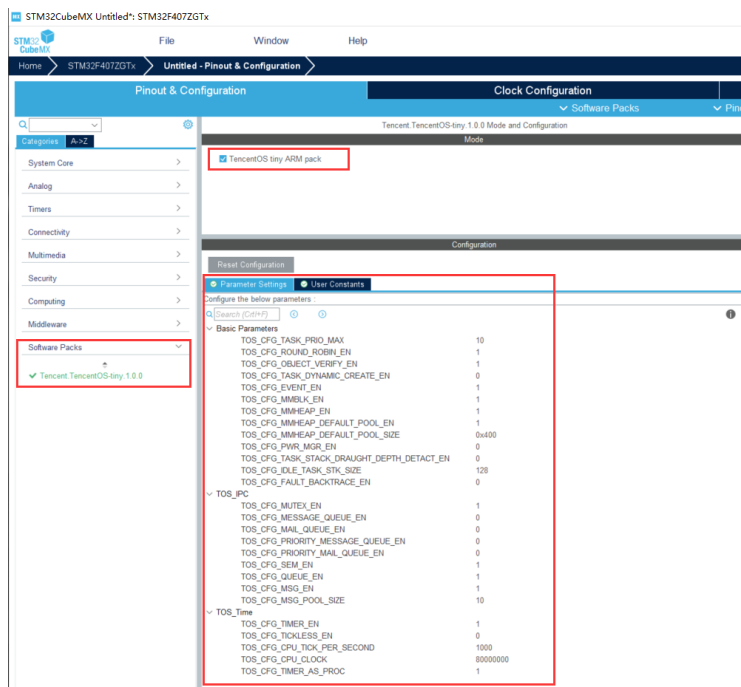


图 3.12 软件包参数配置

(5) 注释掉中断函数 PendSV()、void SysTick\_Handler(void)等，这些在软件包中都进行了配置。

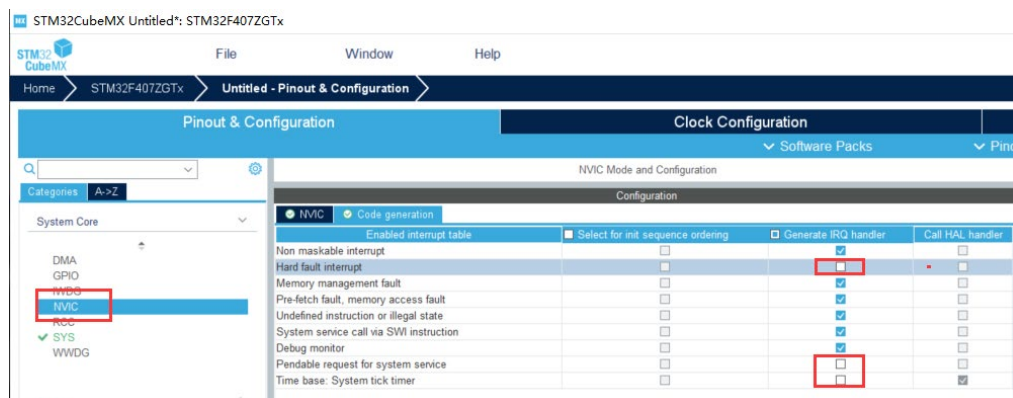
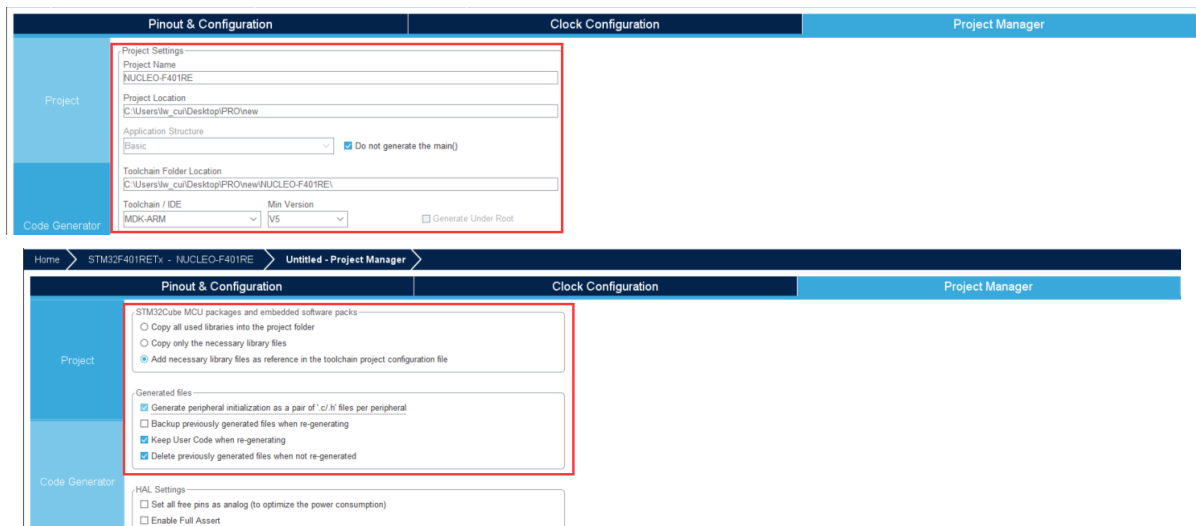


图 3.13 中断函数处理

(6) 进行生成工程配置，按如下界面进行配置，最后点击 GENERATE CODE，并点击 Open Project。



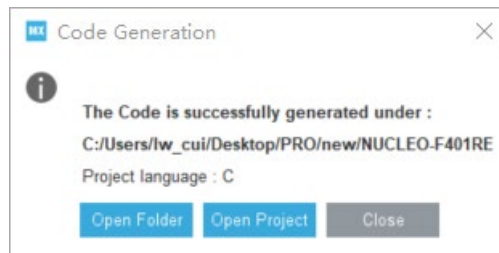


图 3.14 工程生成配置

(8) 打开 main\_example.c, 跳转到 tos\_config 头文件, 在其中加入头文件: `#include "mcu_platform_M4.h"`, 点击 Build All, 如下界面所示, 编译通过, 完成软件包的移植。

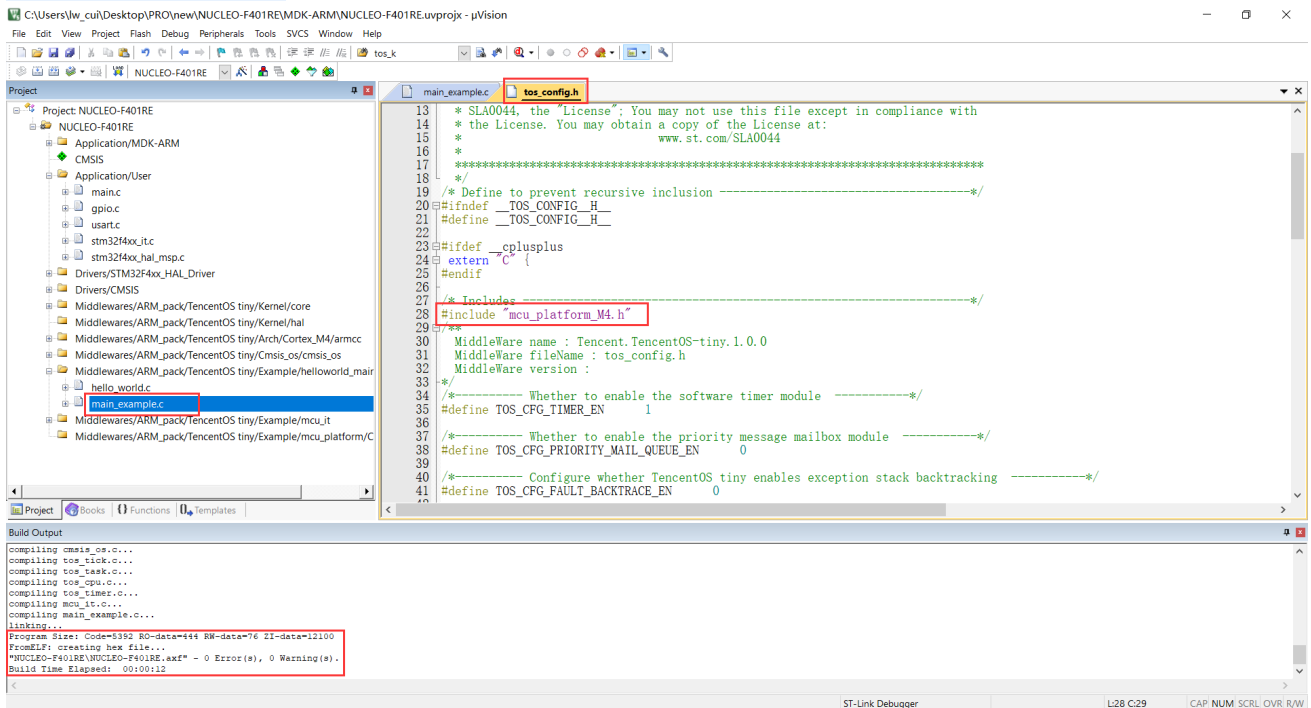


图 3.15 移植编译

### 3.3 单片机测试

对单片机开发板进行测试, 以正点原子探索者 STM32F407ZGT6 为例介绍 TencentOS-tiny 软件包的移植。

(1) 点击 ACCESS TO MCU SELECTOR:

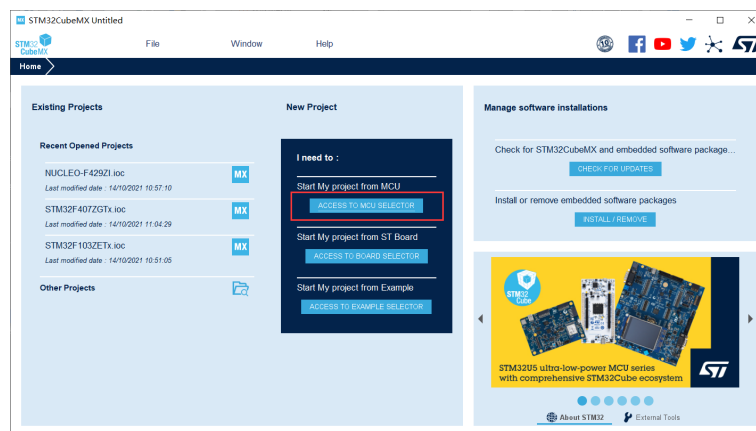


图 3.16 打开 MCU 选择的界面

(2) 选择 STM32F407ZGTx, 点击 Start Project:

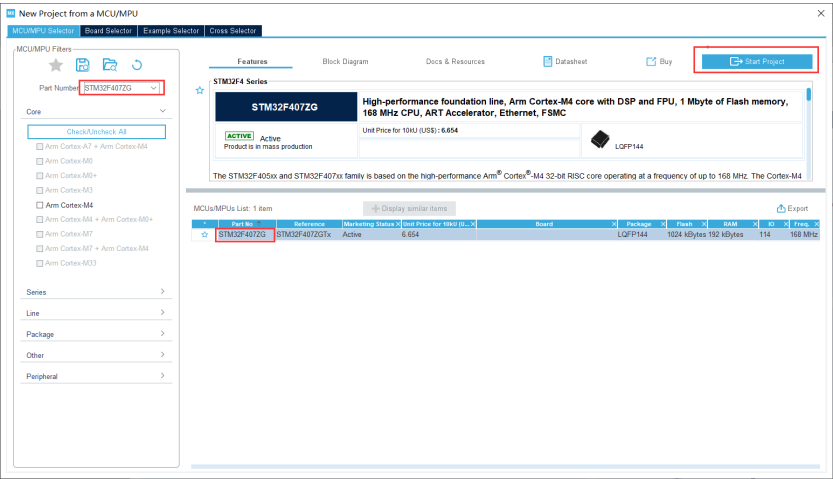


图 3.17 选择 MCU

(3) 点击红框中选项

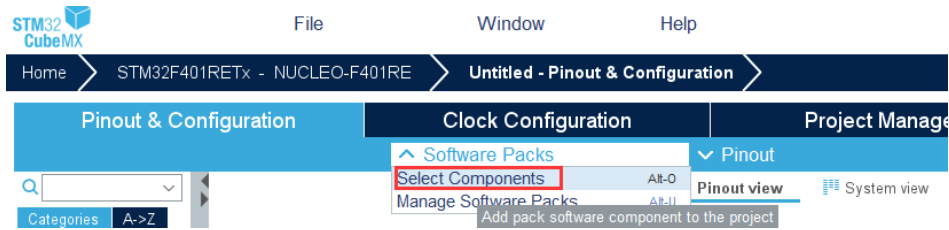


图 3.18 选择软件包

(4) 此时生成使用 MDK-ARM 编译的工程，首先选择 armcc 版本的 arch（如果是 STM32 CubeIDE 版本的话就点击 gcc 版本的 arch），然后点击黄色感叹号，再点击 Resolve。之后手动添加 cmsis\_os，并选择对应 MCU 版本的 mcu\_platform。最后点击 ok 完成配置。

Pack / Bundle / Component	Status	Version	Selection
▼ Tencent.TencentOS-tiny	✓	1.0.0	
▼ TencentOS tiny ARM_pack	✓	1.0.0	
Arch / Cortex_M0			Not selected ▼
Arch / Cortex_M0plus			Not selected ▼
Arch / Cortex_M3			Not selected ▼
Arch / Cortex_M4	✓		armcc ▼
Arch / Cortex_M7			Not selected ▼
Arch / Cortex_M23			Not selected ▼
Arch / Cortex_M33			Not selected ▼
Cmsis_os / cmsis_os	✓		☑
Cmsis_os / cmsis_os2			☐
Kernel / hal	✓		☑
Kernel / core	✓		☑
Example / helloworld_main			☐
Example / mcu_it	✓		☑
Example / mcu_platform	✓		Cortex_M4 ▼

图 3.19 软件包配置

(5) 点击 Software Packs，打勾，并在下方进行软件包的参数配置。

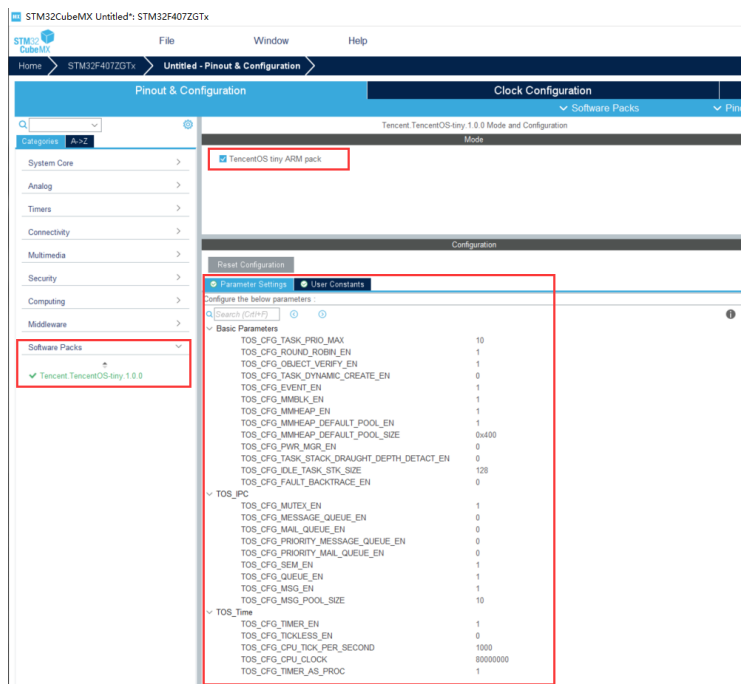
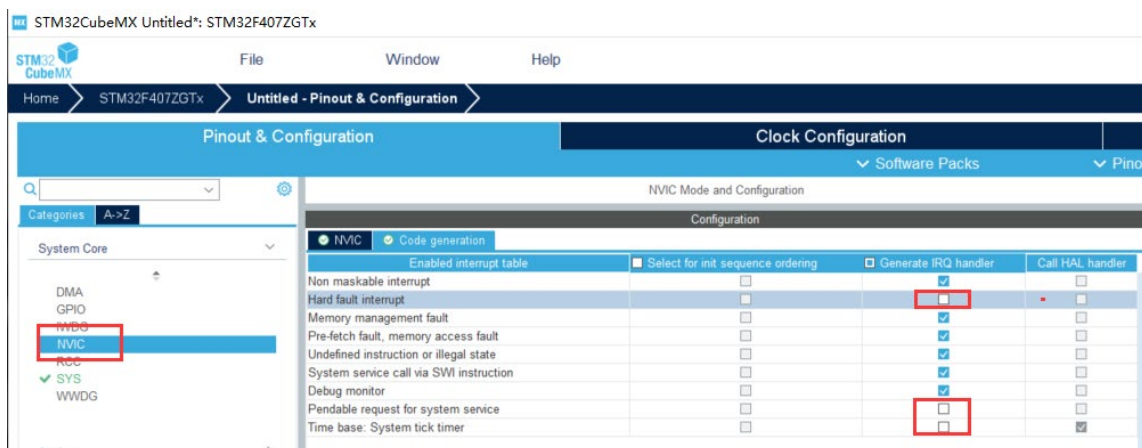
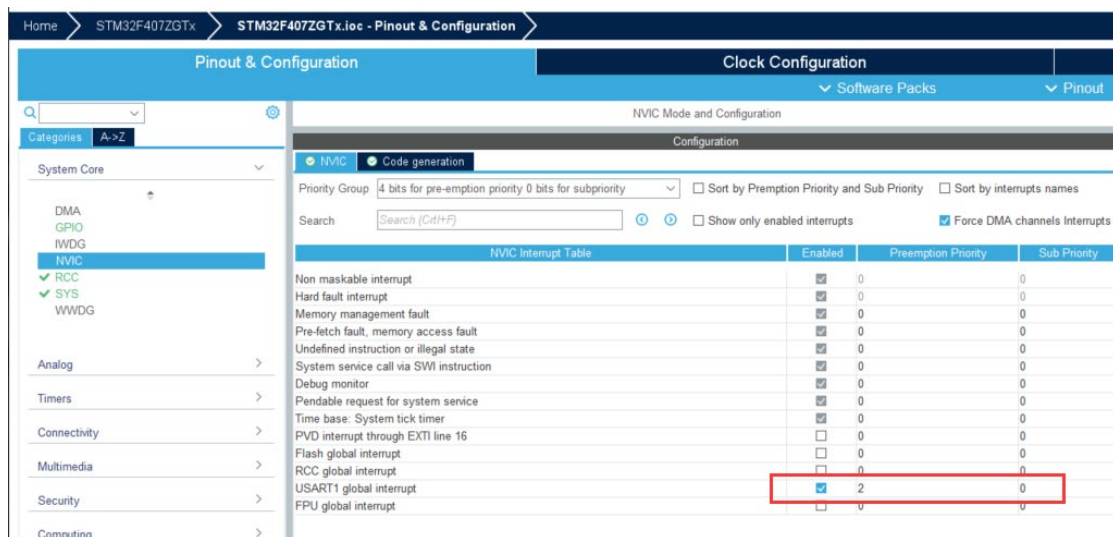


图 3.20 软件包参数配置

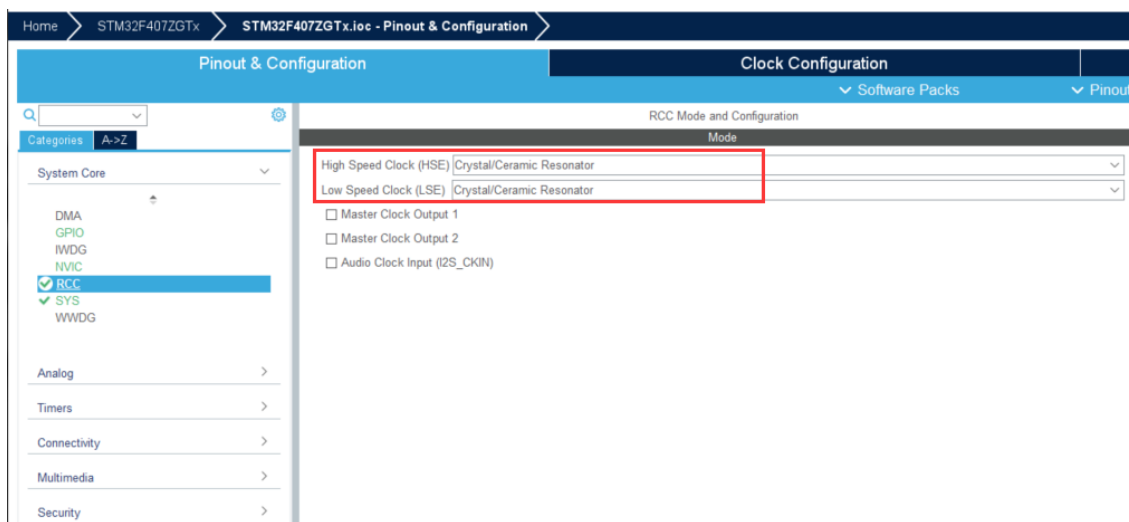
(6) 注释掉中断函数 PendSV()、void SysTick\_Handler(void)等，这些在软件包中都进行了配置，然后我们还需要对串口、下载等模块进行修改，如下：



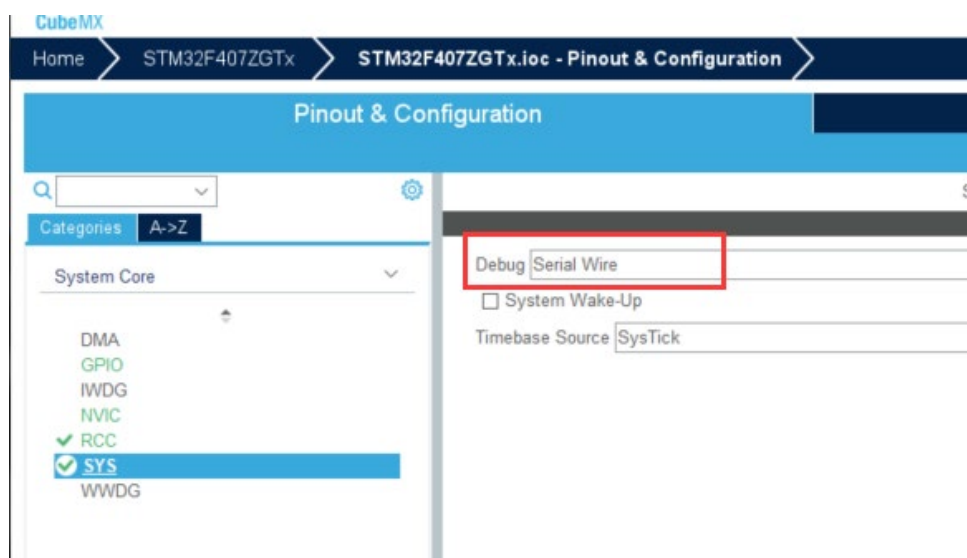
(a) 中断函数处理



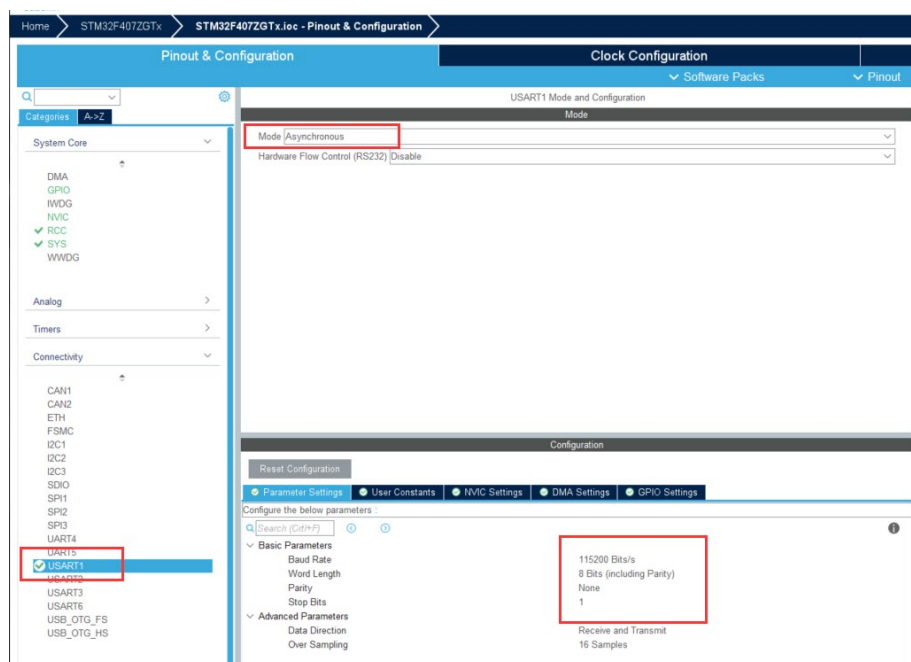
(b) 中断优先级配置



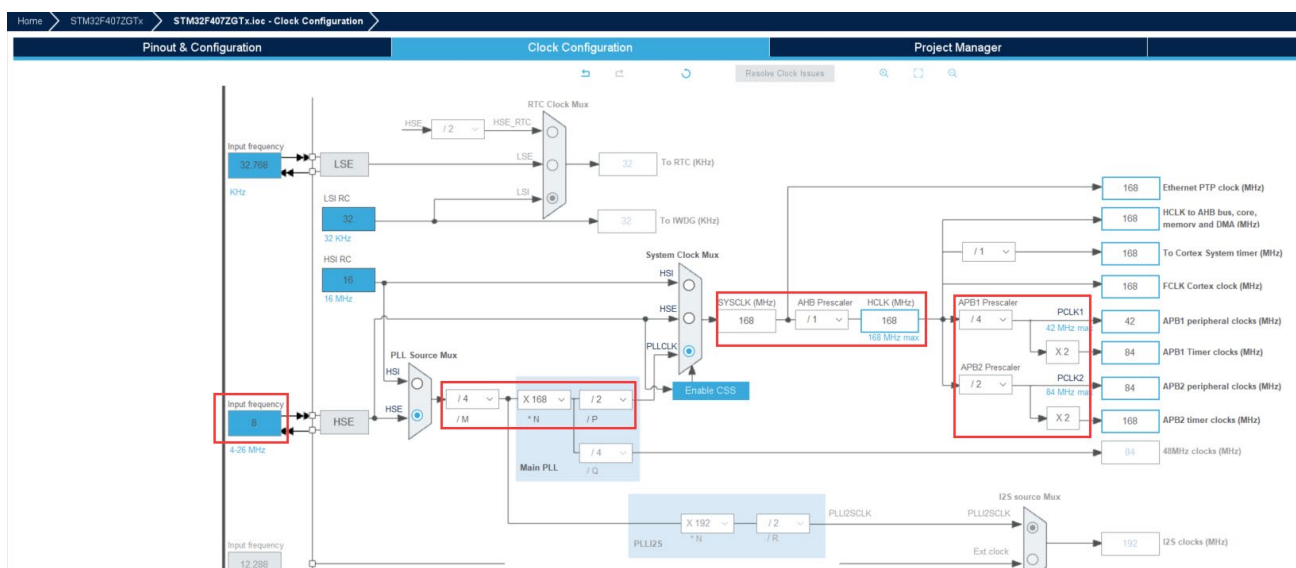
(c) RCC 配置



(d) SYS 配置



(e) 串口 1 配置



(f) 时钟树配置

图 3.21 模块配置

(7) 进行生成工程配置，按如下界面进行配置，最后点击右上角 GENERATE CODE，生成工程后点击 Open Project。

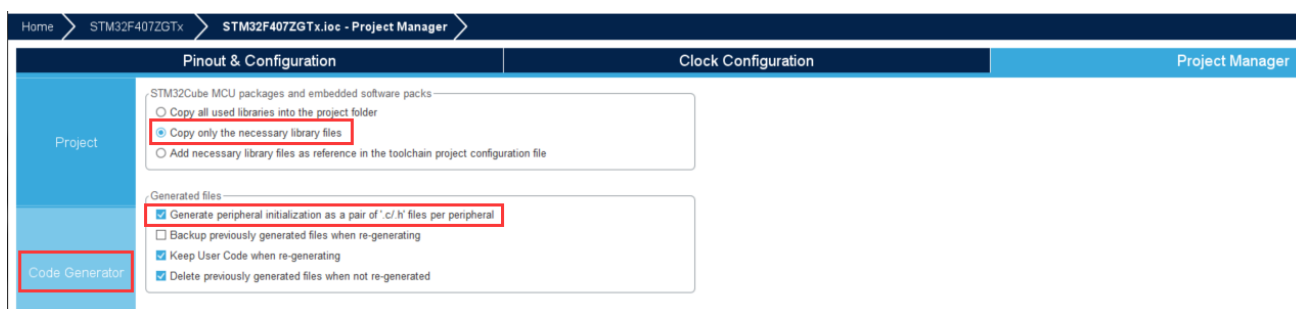
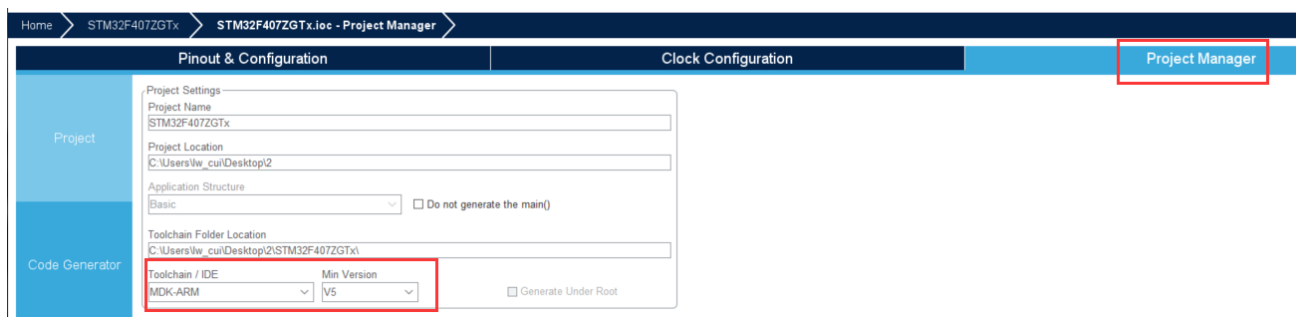


图 3.22 工程配置

(8) 打开 `tos_config` 头文件，在其中加入头文件：`#include "mcu_platform_M4.h"`，然后使用如下的 main 文件：

```
/* USER CODE BEGIN Includes */
#include "cmsis_os.h"
/* USER CODE END Includes */

/* USER CODE BEGIN 0 */
//task1
#define TASK1_STK_SIZE 512
void task1(void *pdata);
osThreadDef(task1, osPriorityNormal, 1, TASK1_STK_SIZE);

//task2
#define TASK2_STK_SIZE 512
void task2(void *pdata);
osThreadDef(task2, osPriorityNormal, 1, TASK2_STK_SIZE);

void task1(void *pdata)
{
    int count = 1;
    char buffer[64] = {0};
    while(1) {
        snprintf(buffer, sizeof(buffer), "task 1 %04d\r\n", count++);
    }
}
```

```

        HAL_UART_Transmit(&huart1,      (uint8_t*)buffer,      /* USER CODE BEGIN Init */
        strlen(buffer), 0xFFFF);
        osDelay(2000);
        /* USER CODE END Init */
    }
}
/* Configure the system clock */
void task2(void *pdata)
{
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_USART1_UART_Init();
    /* USER CODE BEGIN 2 */
    osKernelInitialize();
    /* USER CODE END 2 */
    /* TOS Tiny
kernel initialize
osThreadCreate(osThread(task1), NULL); // Create task1
osThreadCreate(osThread(task2), NULL); // Create task2
osKernelStart();
//Start TOS Tiny
*/
    /* USER CODE END 2 */

    /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    while (1)
    {
        /* USER CODE END WHILE */

        /* USER CODE BEGIN 3 */
    }
    /* USER CODE END 3 */
}

int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /*
    MCU
    Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and
    the Systick. */
    HAL_Init();
}

```

### (9) 编译，然后使用 ST-LINK 下载到正点原子探索者单片机中

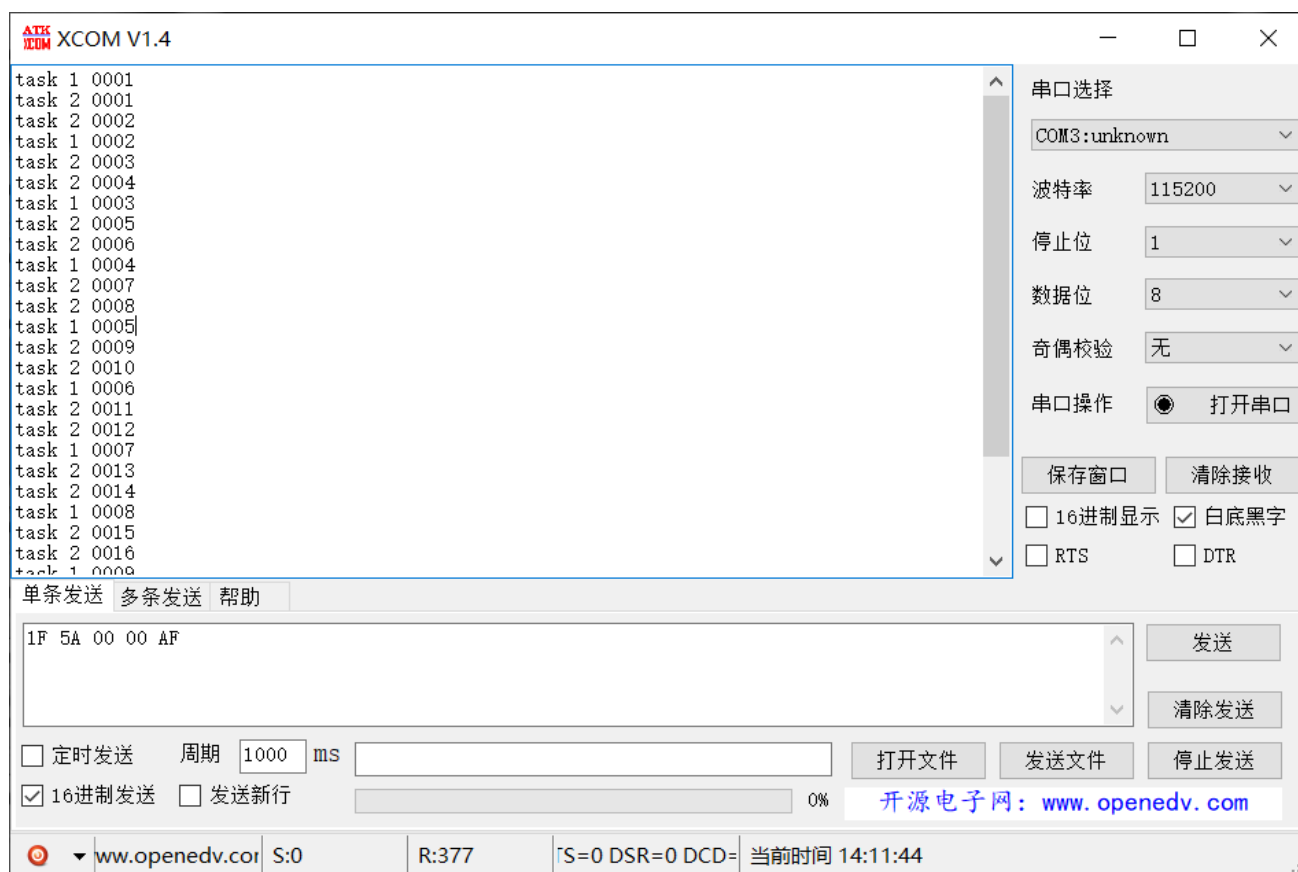
```

Build Output
Build target 'STM32F407ZGTx'
compiling port_s.S...
compiling main.C...
linking...
Program Size: Code=10360 RO-data=536 RW-data=88 ZI-data=6928
FromELF: creating hex file...
STM32F407ZGTx\STM32F407ZGTx.hex: Warning: Q9931W: Your license for feature Keil will expire in 4 days
Finished: 0 information, 1 warning and 0 error messages.
"STM32F407ZGTx\STM32F407ZGTx.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:04
Load "STM32F407ZGTx\STM32F407ZGTx.axf"
Erase Done.
Programming Done.
Verify OK.
Flash Load finished at 14:10:22

```

### 3.23 编译、下载程序

#### (10) 使用串口助手查看，两个任务的延时设置不一样，运行效果也不一样：





#### 4、开发参考

- 1、腾讯物联网操作系统网址 <https://github.com/OpenAtomFoundation/TencentOS-tiny>
- 2、ST 官方教程 [How to develop a STM32Cube Expansion Package - stm32mcu \(stmicroelectronics.cn\)](#)
- 3、制作 STM32Cube 软件包-STM32 PackCreator 的使用及软件包的制作 - 极术社区 - 连接开发者与智能计算生态 ([aijishu.com](http://aijishu.com))
- 4、(1) STM32 CubeMX 软件包的制作---STM32PackCreator 的使用 [linwei\\_Cui 的博客-CSDN 博客](#)
- 5、(2) STM32 CubeMX 软件包的制作---简易软件包的制作 [linwei\\_Cui 的博客-CSDN 博客](#)