

## 目录

第 1 章	Cortex-M4 基础 .....	2
1.1	Cortex-M4 介绍 .....	2
1.1.1	何谓单片机.....	2
1.1.2	单片机与电脑的区别.....	2
1.1.3	M4 单片机结构框架 .....	2
1.2	芯片资料下载.....	5
1.3	开发环境搭建.....	7
1.3.1	安装 CH340 的驱动 .....	7
1.3.2	ST_Link 驱动.....	7
1.3.3	安装 keil 软件.....	7
1.3.4	安装芯片型号支持包.....	9
1.4	Cortex-M4 单片机最小系统 .....	10
1.4.1	电源电路.....	10
1.4.2	震荡电路.....	11
1.4.3	复位电路.....	11
1.4.4	自举模式电路.....	12
1.5	新建工程.....	12
1.6	添加源文件.....	16

## 第1章 Cortex-M4 基础

### 1.1 Cortex-M4 介绍

#### 1.1.1 何谓单片机

单片机，是典型的嵌入式微控制器（Microcontroller Unit），常用英文字母的缩写“MCU”表示单片机。实际上，单片机是一种集成电路芯片，是采用超大规模集成电路技术把具有数据处理能力的中央处理器 FPU、随机存储器 RAM、只读存储器 ROM、多种 I/O 口和中断系统、定时器/计数器等功能（可能还包括显示驱动电路、脉宽调制电路 PWM、模拟多路转换器、A/D 转换器等电路）集成到一块硅片上构成的一个小而完善的计算机系统。

RAM: 数据存储器 存储一些中间变量 读写速度快 数据掉电丢失 内存条 运行内存  
ROM: 程序存储器 数据掉电不丢失 FLASH: 闪存 数据掉电不丢失 硬盘 本机内存  
I/O: 输入输出口 IO 一定是芯片的引脚 电源管脚不是 IO  
中断: 打断(不正常: 异常事件)  
定时器: 51 单片机功能比较少 但是 STM32 定时器功能非常多  
定时: 计一次数的时间  $1s * \text{计多少次}$  10 10S  
基本定时器: 只有定时器功能  
通用定时器: 包含了基本定时器的所有功能, PWM(占空比)和捕获(红外)  
高级定时器: 包含了通用定时器的所有功能, 编码  
ADC: 模数转换器 模拟信号(连续) 数字信号(离散)  
DAC: 数模转换器

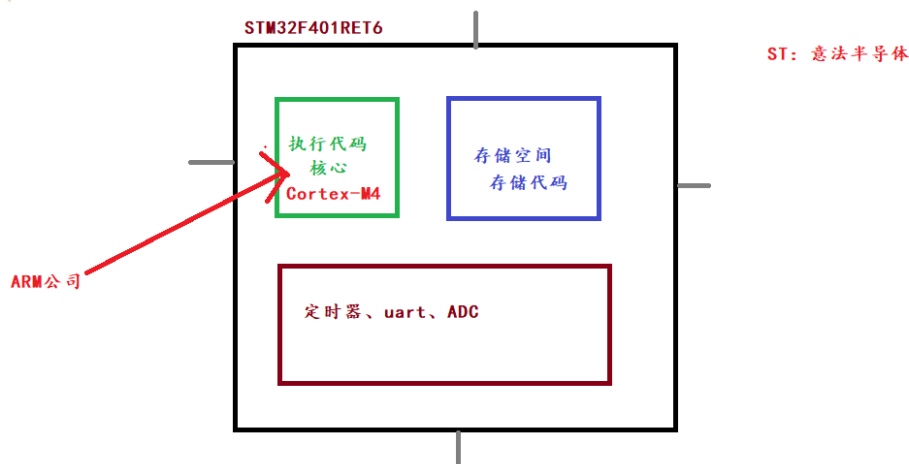
单片机就是一个微型计算机

#### 1.1.2 单片机与电脑的区别

写程序代码，都要站在单片机角度去写代码 单片机很蠢

	电脑	单片机
芯片	CPU	MCU
芯片核心（内核）	GPU	FPU
输入设备	鼠标、键盘、麦克风、摄像头	按键、触摸屏(电容/电阻)
输出设备	屏幕、扬声器	LED、LCD、OLED(消费电子)
存储器	内存条、硬盘	RAM、FLASH

#### 1.1.3 M4 单片机结构框架



STM32F4xx中文... 参考手册: 对芯片使用说明

27.1MB

工程师...

23次

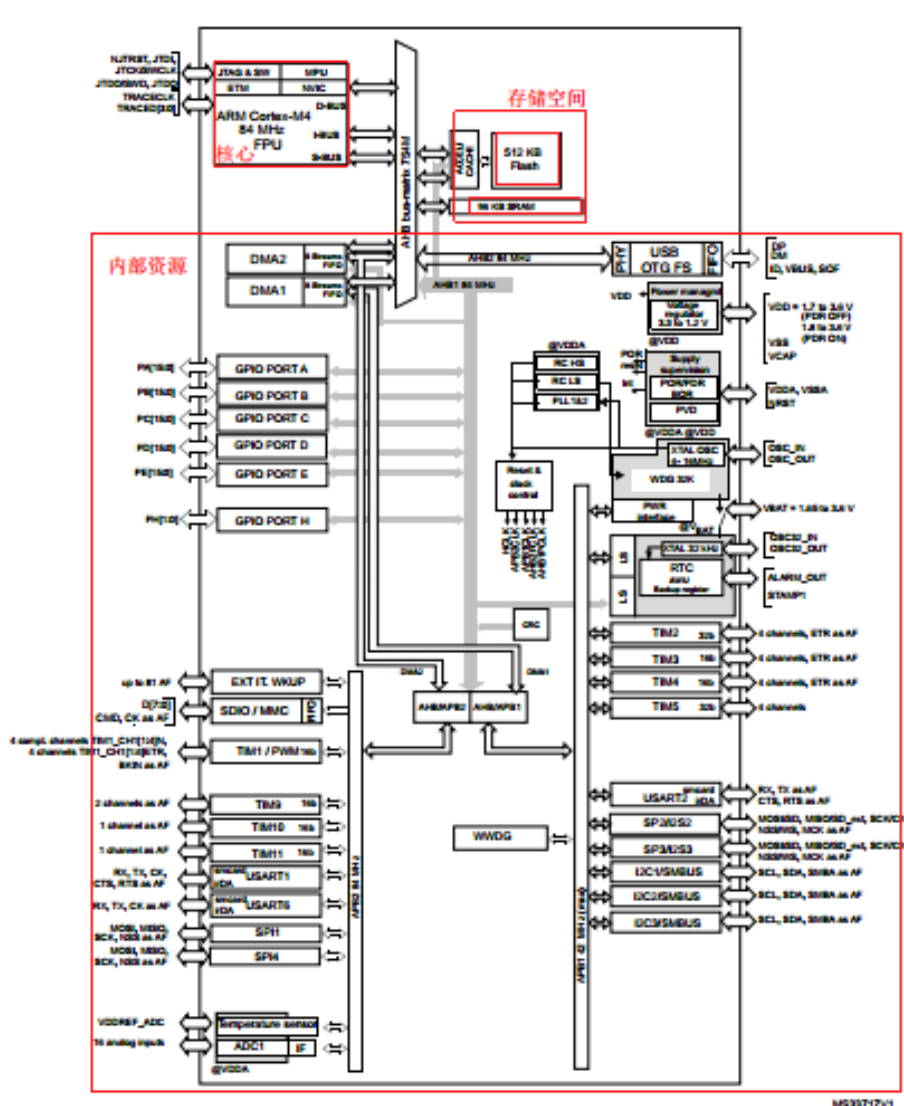
stm32f401re.pdf 数据手册: 对芯片参数有详细描述

2.01MB

工程师...

25次

当前使用的芯片型号: STM32F401RET6



去了解: 下午: 2 点--4 点

1. 什么外设? 什么是片上外设、片内外设、内部资源、片外外设?

核心外就叫外设

片上外设、片内外设、内部资源: 在核心外, 在芯片内

片外外设: 芯片外的

2. 回顾一下 C 语言按位运算符 &amp; | ~ 先将数据转换成二进制

&amp;: 全 1 为 1 有 0 为 0

|: 全 0 为 0 有 1 为 1

~: 1 变 0 0 变 1

3. 回顾一下 C 语言数据类型(有四大类)

(1) 基本数据类型: 整型 字符型 浮点型(单精度 float 和双精度 double)

(2) 构造数据类型: 数组类型 结构体类型 共用体 枚举 链表

(3) 指针类型: 数据类型 \* int \*p; // 指针变量

指针变量与整型变量有什么区别?

`int *p;` 本质: 存储地址

`int p;` 本质: 存储数据

(4) 空类型: `void`

4. 回顾: `char`、`short`(有符号和无符号): 取值范围

`char`: 1 字节

定义变量: 有修饰符(有符号还是无符号)

`signed char a;` //1 字节

取值范围: -128 ~ 127

`unsigned char a;` //1 字节

取值范围: 0 ~ 255

`char i;` //取值范围-128 - 127

`for(i = 0; i < 200; i++)` // i 加到 127 之后再加 1 → -128

{

循环几次出来?

}

非常严重问题: 不会警告、不会有错误。

使用这个变量之前, 要明确变量的取值范围。

`short`: 2 字节

`signed short a;` 2 字节

取值范围: -32768 - 32767

`unsigned short a;` 2 字节

取值范围: 0- 65535

`int`: 跟平台有关 51 单片机(8bit)

32 位以下的: `int` → 2 字节

32 位及以上: `int` → 4 字节

`long`: 固定 4 字节

`float`: 4 字节

`double`: 8 字节

惩罚: 表演节目(5 分钟才艺表演)

STM32F401RET6

ST: 意法半导体生产出来

ARM 公司: 研发核心

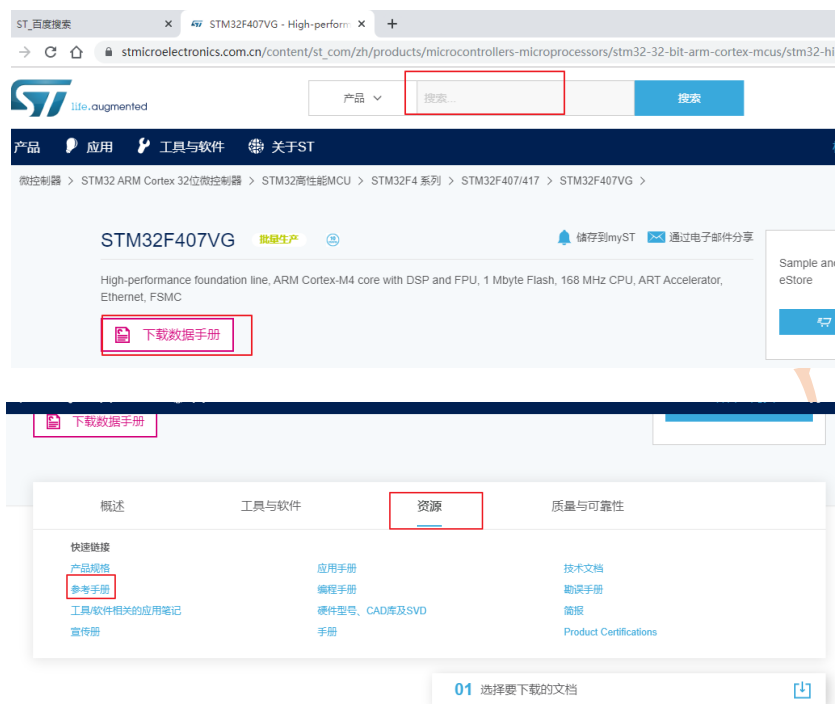
A: 高端消费电子

R: 军工

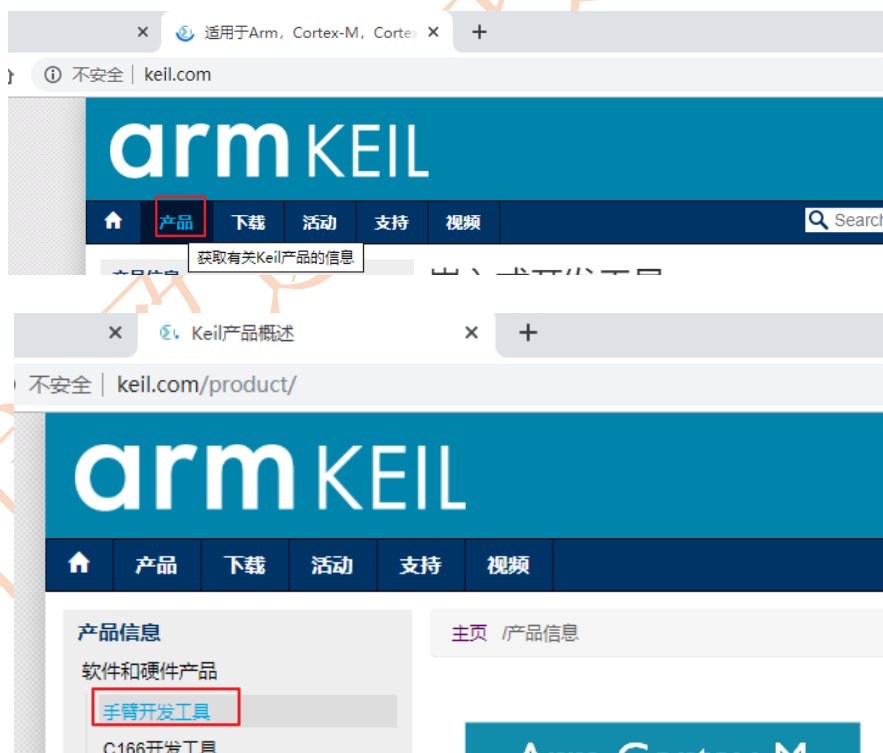
M: 中低端消费电子 M0 M3 M4 M7

## 1.2 芯片资料下载

### 1. 打开 ST 官网



### 2. 下载 keil 软件 打开官网





3. 芯片型号支持包  
打开 keil 官网



STMicroelectronics STM32F4系列设备支持, 驱动程序和示例

BSP DFP 2.14.0

版本: 2.14.0 (2019-07-24) Keil.STM32F4xx\_DFP.2.14.0.pack

下载

警告: --C99不再通过设备说明来实施。在“目标选项”对话框的“C / C ++”选项卡中启用“C99模式”。

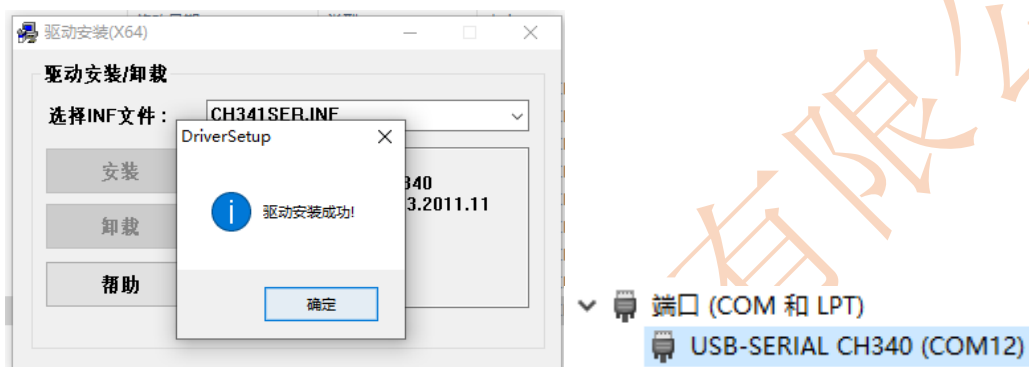
使用HAL驱动程序V1.7.6更新了Pack, 以包括STM32Cube\_FW\_F4固件包V1.24.1版本的子集。

- 已更正RTE\_Device.h UART5引脚配置。
- MX\_Device.h.ftl:
  - 更新了对USART虚拟模式的解析。
  - 更新了宏的生成: 添加了对“(和)”符号的处理。

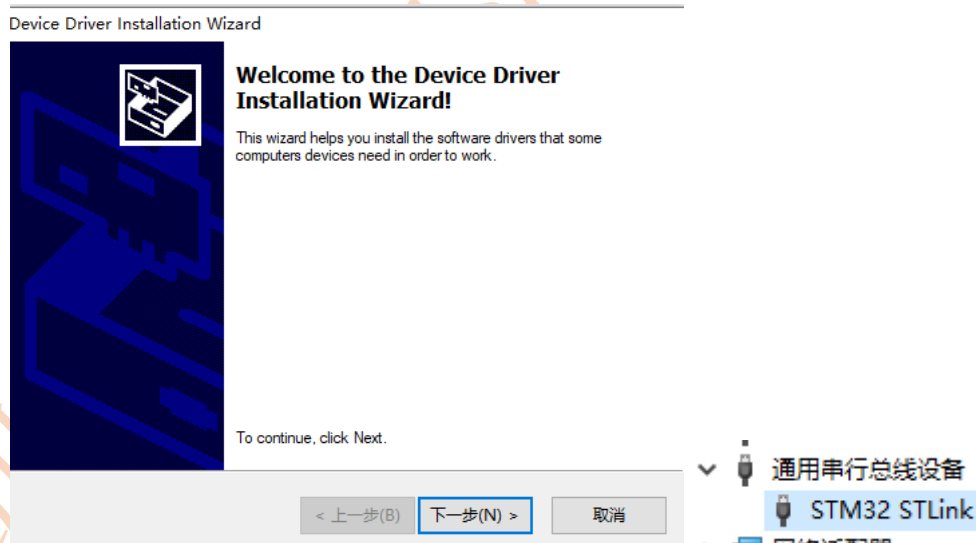
纠正了通过“播放”按钮启动STM32CubeMX来为现有项目覆盖新的STM32CubeMX项目文件而不是加载现有

## 1.3 开发环境搭建

### 1.3.1 安装 CH340 的驱动 (有板子的同学装)

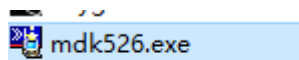


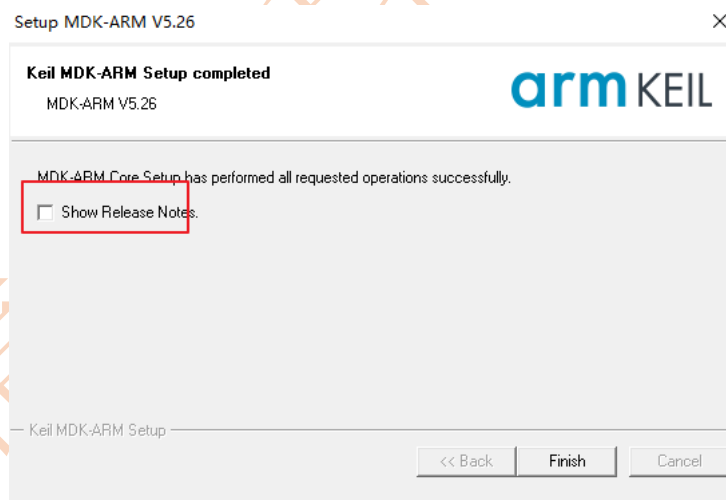
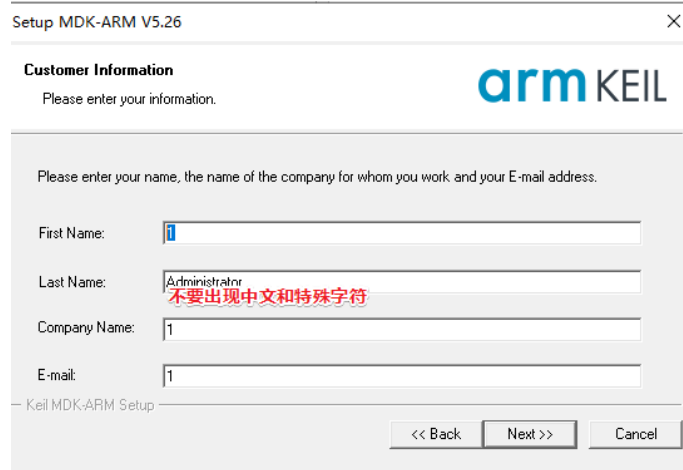
### 1.3.2 ST\_Link 驱动 (有板子的同学装)



### 1.3.3 安装 keil 软件(都要装)

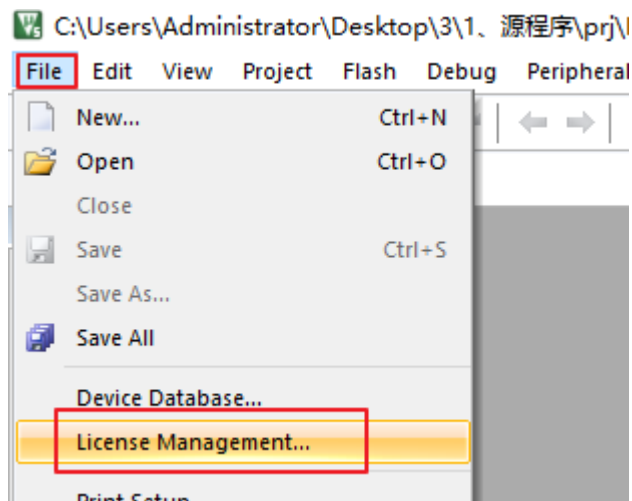
安装路径最好不要有中文名





破解软件  
打开 keil 软件



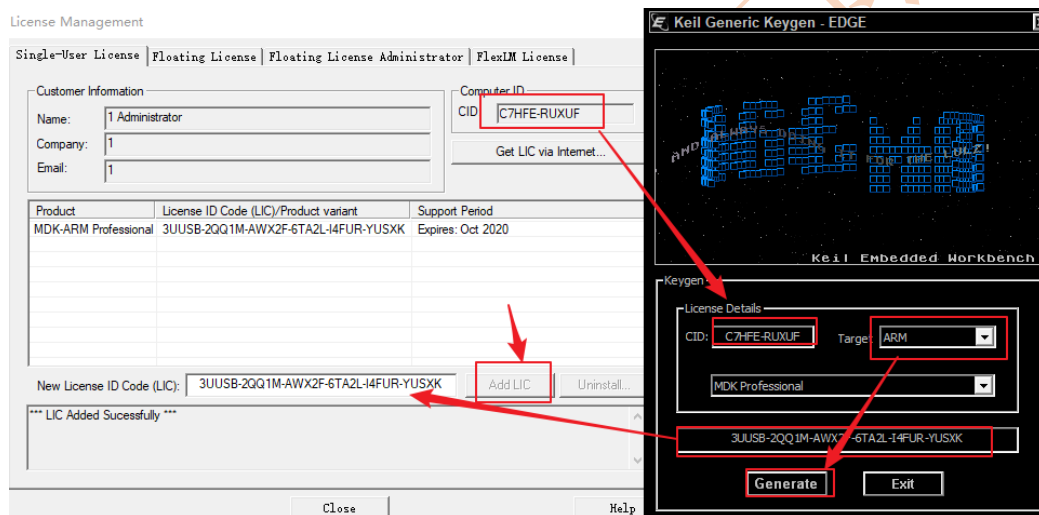


keygen\_new2032... 7小时前

破解软件

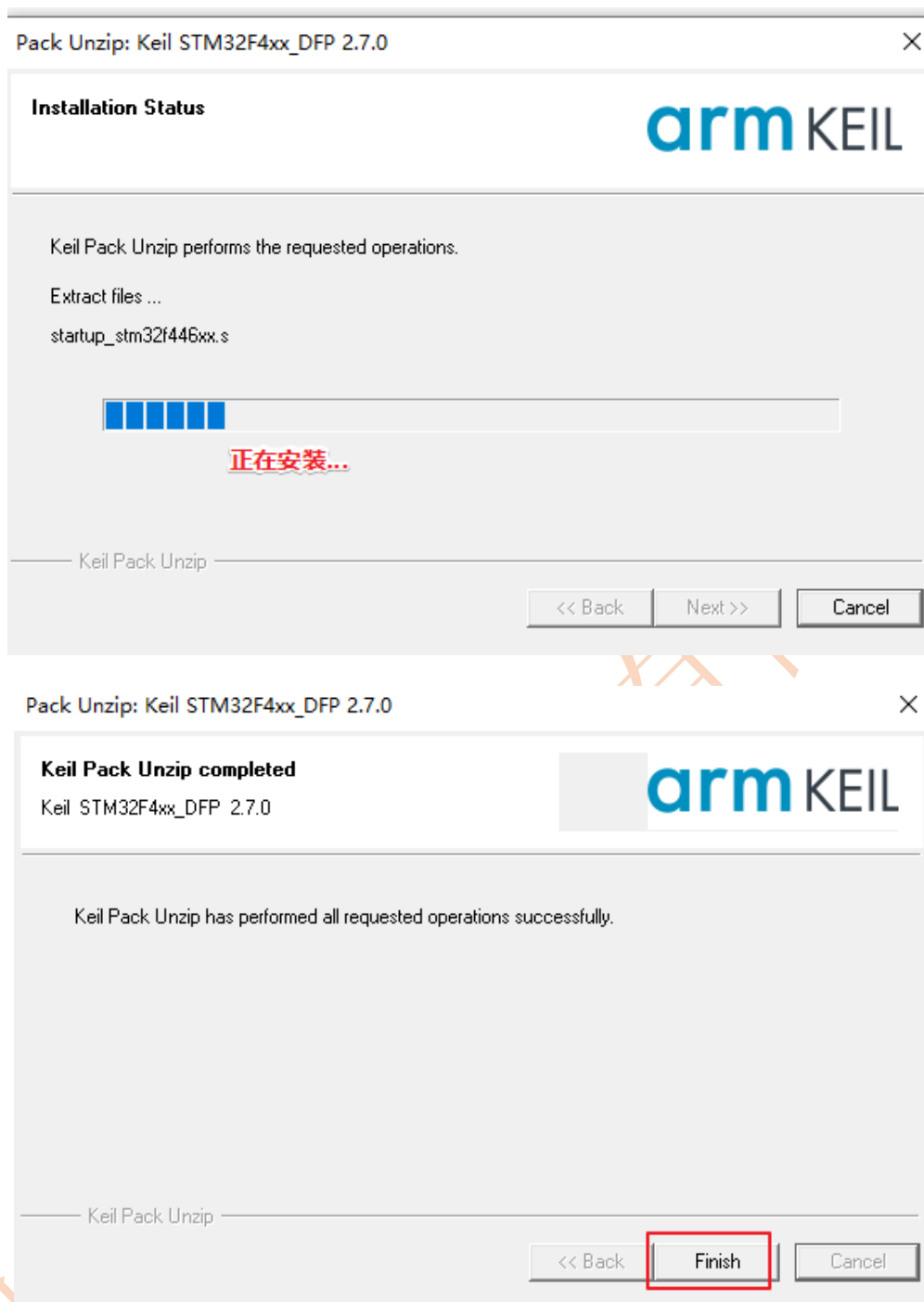
破解时：所有杀毒软件都关闭

温馨提示：打开这个软件时，音量调小



### 1.3.4 安装芯片型号支持包(没装：新建工程之后，就会出现没有芯片型号选择)

Keil.STM32F4xx\_DFP.2.7.0.pack



## 1.4 Cortex-M4 单片机最小系统

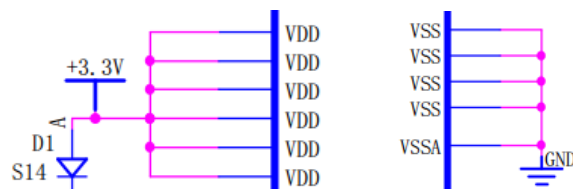
单片机最小系统：单片机能正常工作的最基本条件也是必要条件  
不是说给你单片机，接上电就可以用。

51 单片机：由电源电路、振荡电路、复位电路、单片机、EA 引脚这些东西组成。

STM32 单片机：由电源电路、振荡电路、复位电路、单片机、启动方式这些东西组成。

### 1.4.1 电源电路

所有机器工作都需要电



stm32 单片机: 3.3V 供电      CMOS 电平

51 单片机: 5V 供电      TTL 电平

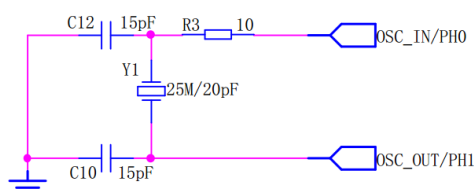
单片机只能识别数字量: 0 或者 1

高电平: '1'

低电平: '0'

### 1.4.2 振荡电路 心跳

因为单片机都需要连续的脉冲信号才能正常工作。



晶振: 提供连续的脉冲信号 频偏较小 比较便宜

内部 RC 振荡电路: 不够稳定, 频偏较大

表面声波振荡器: 频偏很小 但是贵

### 1.4.3 复位电路

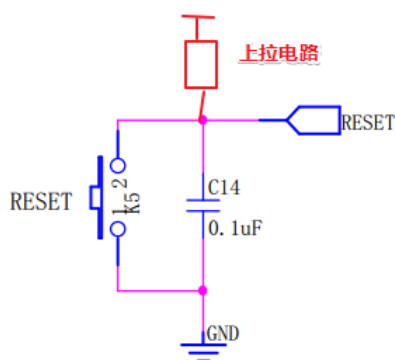
任何的单片机都有一个复位引脚

找到这个复位引脚之后, 往这个引脚输入一个电平(是高是低取决于单片机的种类)

复位: 程序重头开始运行

STM32 单片机: 低电平复位

51 单片机: 高电平复位



复位方式

电容: 饱和: 隔直通交      充电: 导通

按键复位

上电复位

软件复位

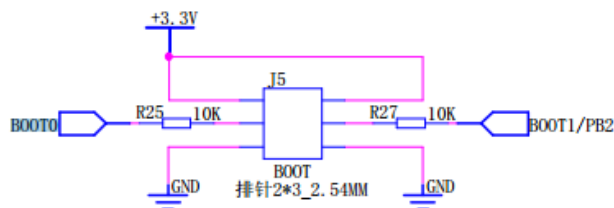
看门狗复位: 检测程序会不会跑飞

### 1.4.4 自举模式电路

只是 ST 芯片特有的

表 3. 自举模式

自举模式选择引脚		自举模式	自举空间
BOOT1	BOOT0		
x	0	主 Flash	选择主 Flash 作为自举空间
0	1	系统存储器	选择系统存储器作为自举空间
1	1	嵌入式 SRAM	选择嵌入式 SRAM 作为自举空间



1. 将代码下载到主 FLASH, 代码从主 FLASH 启动

2. 将代码下载到系统存储器(FLASH), 代码启动不了

3. 将代码下载到 SRAM, 代码从 SRAM 启动(超频)

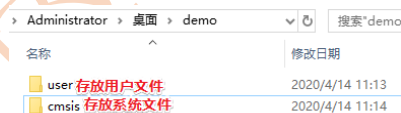
### 1.5 新建工程

关于寄存器新建工程模板---只需要新建一次(后面使用都不需要), 直接在这个工程里面添加模块

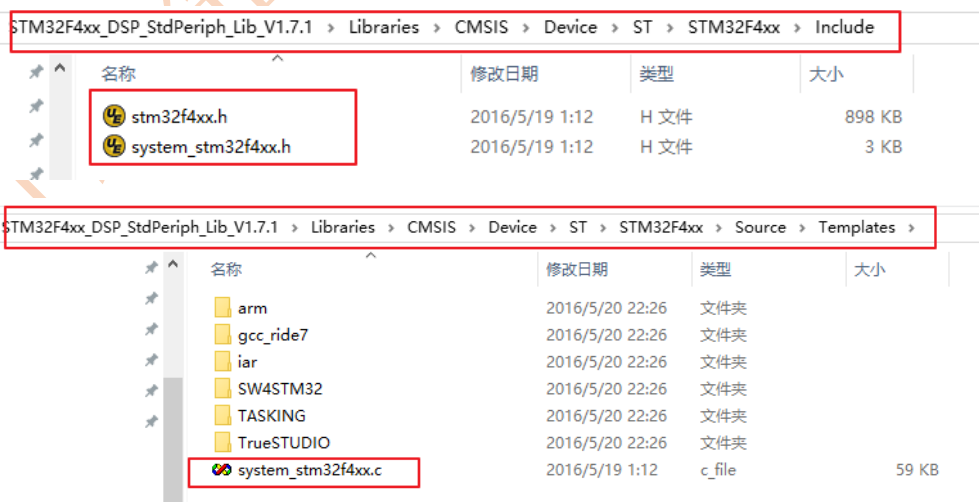
#### 多文件操作

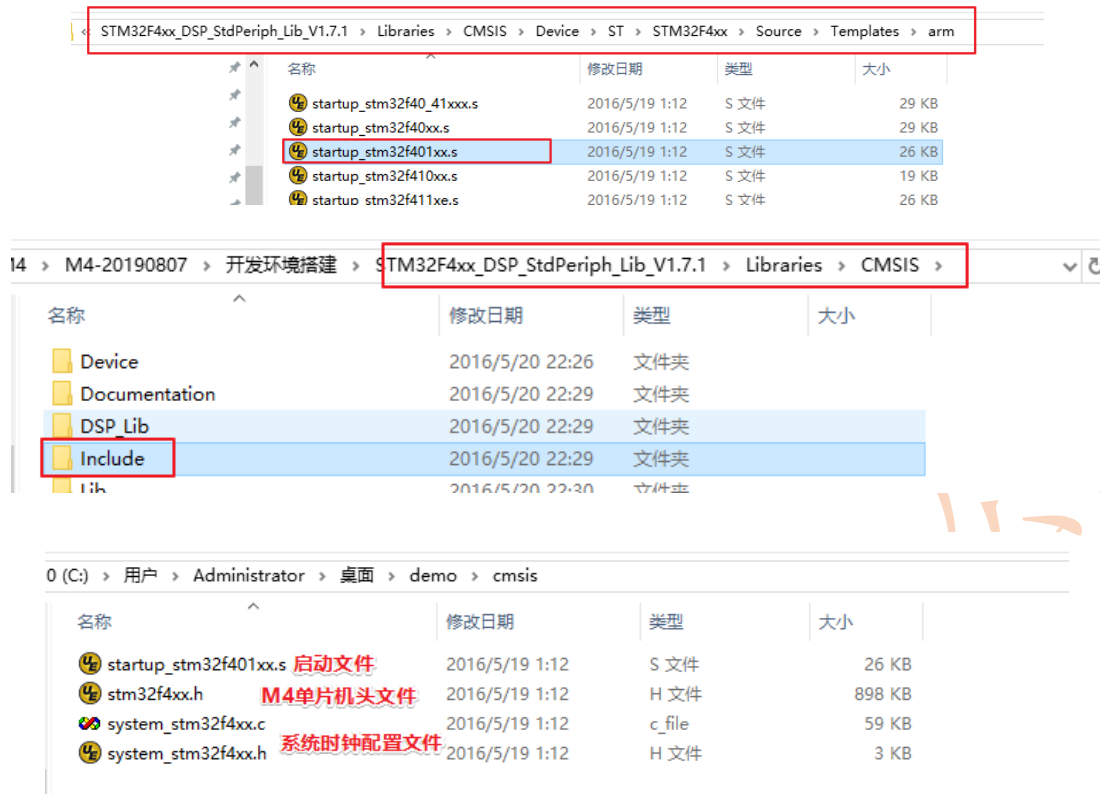
1. 新建一个文件”demo” //可以是中文

2. 在 demo 文件夹下新建 user 文件夹和 cmsis 文件夹

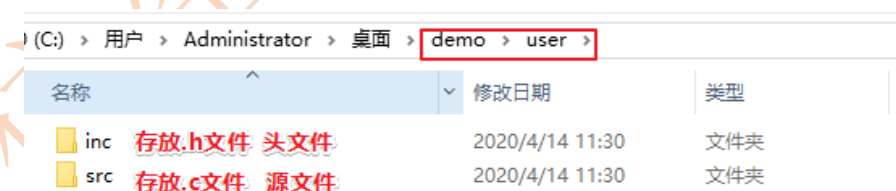


3. 将芯片所需的系统文件添加到 cmsis 文件夹下

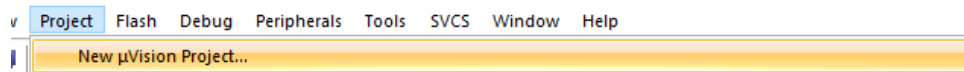


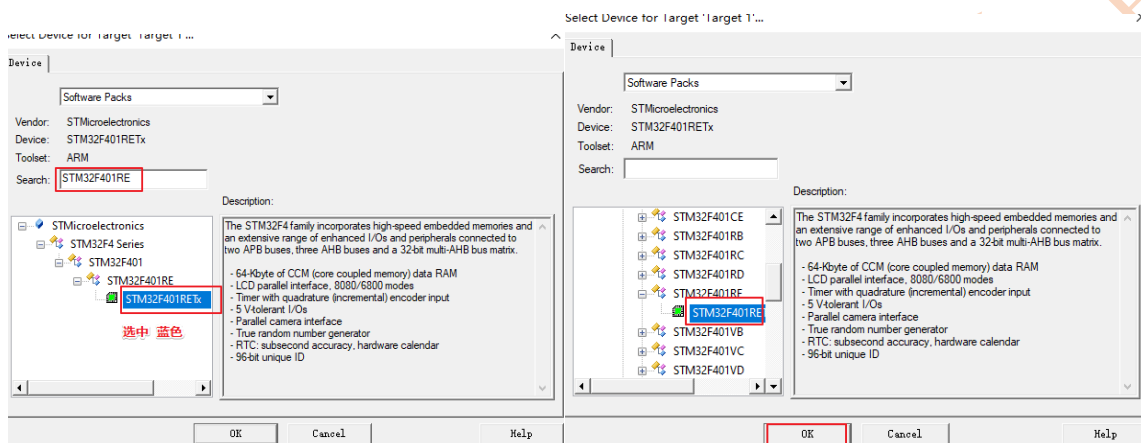
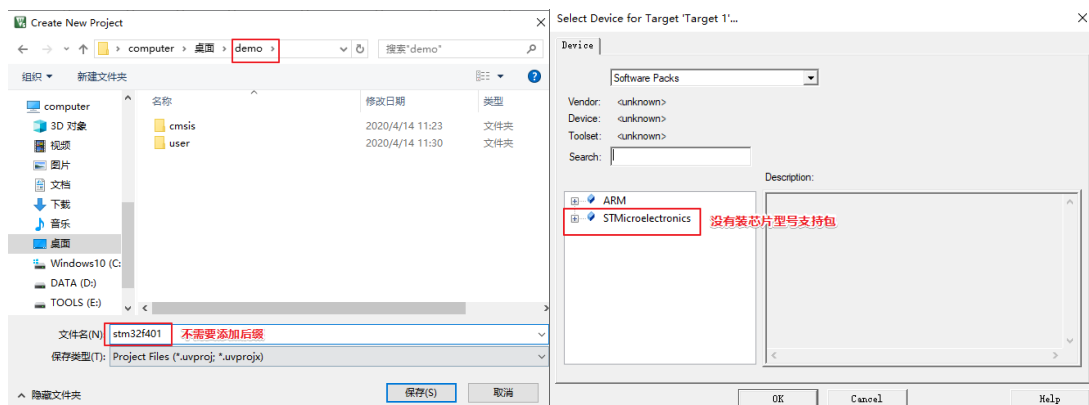


#### 4. 在 user 文件夹下新建 src 文件夹和 inc 文件夹

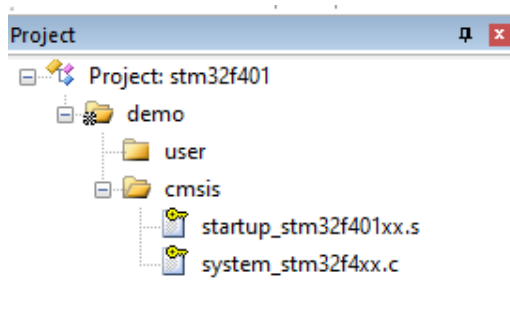
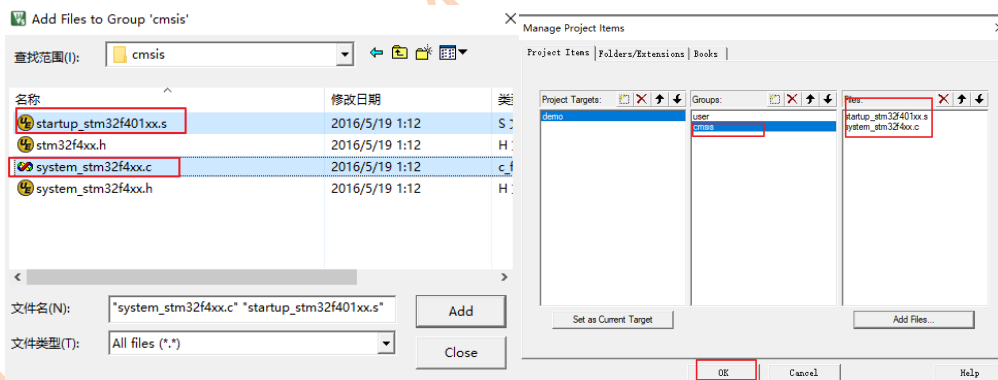


#### 5. 打开 keil 软件，新建工程，选择芯片型号



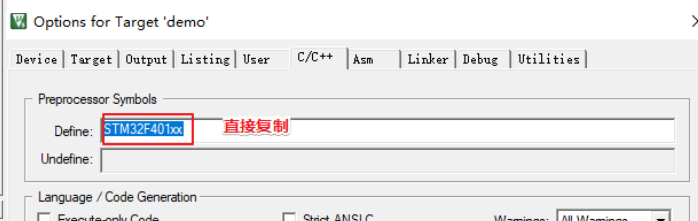
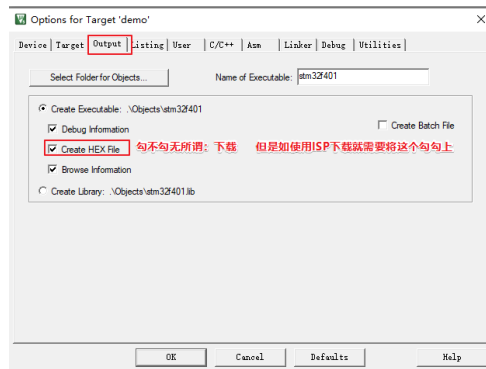


## 6. 设置虚拟工程树

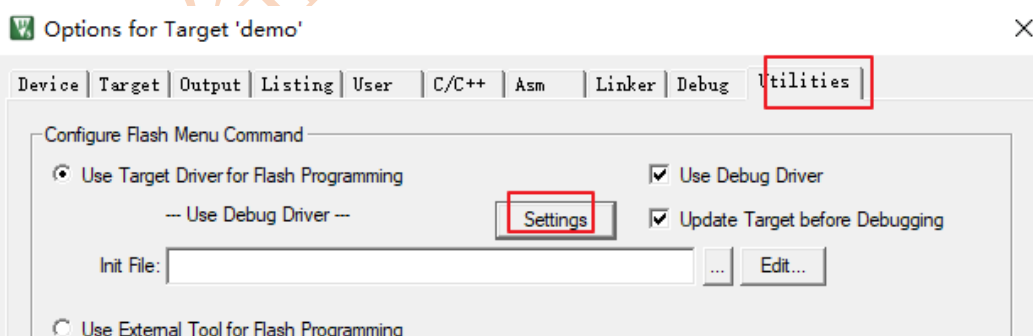
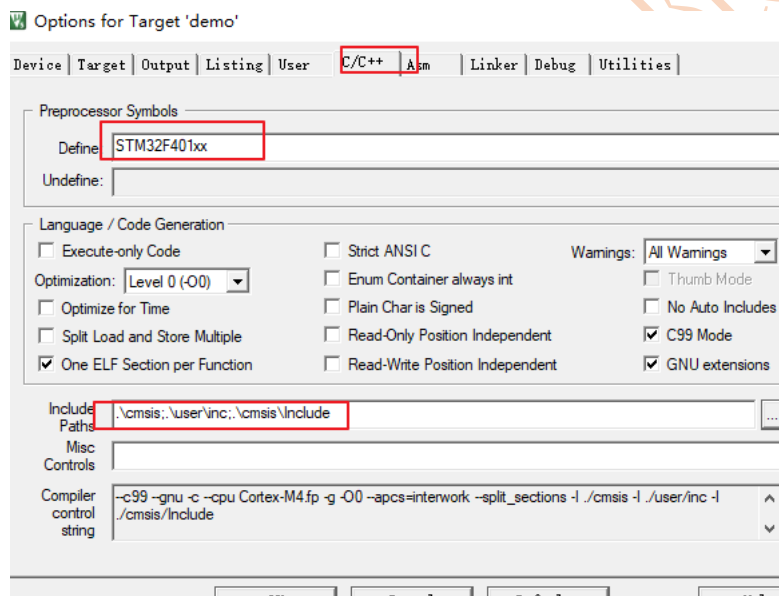
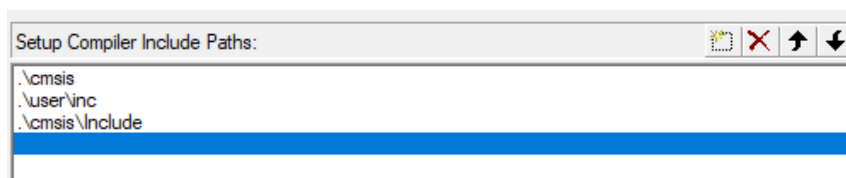


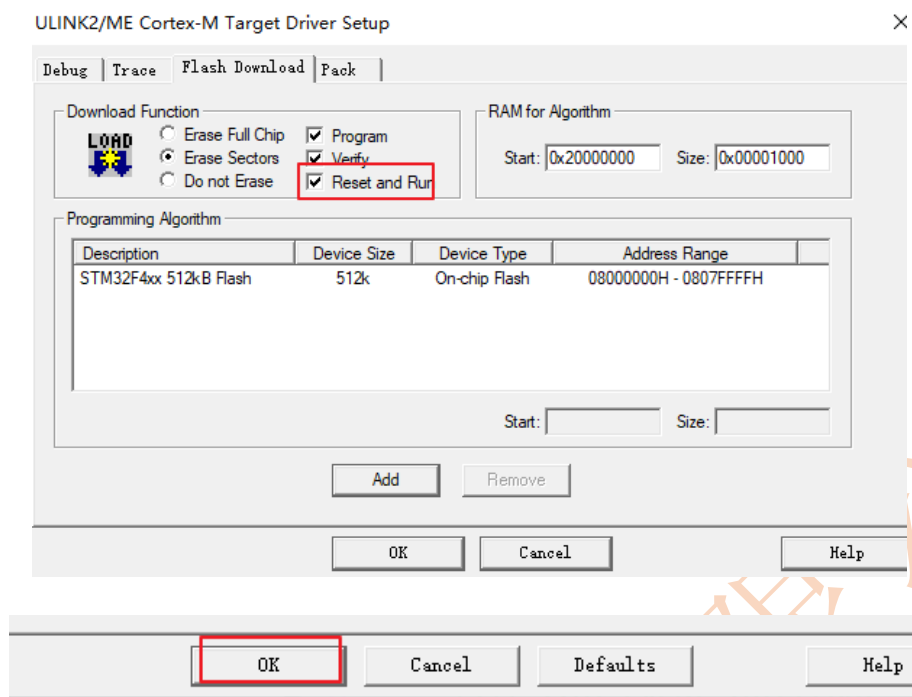
## 7. 设置工程环境





Folder Setup





## 1.6 添加源文件

