# 目录

[目录 1](#_Toc522552210)

[第1章 Cortex-M4基础 2](#_Toc522552211)

[1.1 Cortex-M4概述 2](#_Toc522552212)

[1.1.1 ARM单片机简介 2](#_Toc522552213)

[1.1.2 ARM单片机发展史 2](#_Toc522552214)

[1.1.3 ST芯片简介 2](#_Toc522552215)

[1.1.4 STM32F407ZGT6资源介绍 3](#_Toc522552216)

[1.1.5 XYD-Cortex-M4-开发板介绍 4](#_Toc522552217)

[1.2 STM32F407ZGT6最小系统 4](#_Toc522552218)

[1.2.1 最小系统构成 4](#_Toc522552219)

[1.2.2 电源电路 5](#_Toc522552220)

[1.2.3 振荡电路 5](#_Toc522552221)

[1.2.4 复位电路 5](#_Toc522552222)

[1.2.5 自举模式 5](#_Toc522552223)

[1.3 STM32F407ZGT6芯片架构 5](#_Toc522552224)

[1.4 STM32F407ZGT6芯片开发环境 7](#_Toc522552225)

[1.4.1 开发软件 7](#_Toc522552226)

[1.4.2 开发软件注册器 10](#_Toc522552227)

[1.4.3 开发芯片支持包 11](#_Toc522552228)

[1.4.4 开发芯片烧写工具驱动 12](#_Toc522552229)

[1.4.5 串口驱动 12](#_Toc522552230)

[1.5 STM32F407ZGT6项目工程创建 13](#_Toc522552231)

[1.5.1 新建项目文件夹 13](#_Toc522552232)

[1.5.2 创建启动文件夹 14](#_Toc522552233)

[1.5.3 创建编写代码文件夹 15](#_Toc522552234)

[1.5.4 创建项目工程 15](#_Toc522552235)

[1.5.5 配置项目工程 15](#_Toc522552236)

[1.5.6 编写主函数并且编译工程 17](#_Toc522552237)

# Cortex-M4基础

## Cortex-M4概述

### ARM单片机简介

由ARM公司构建的内核，ARM公司不生产芯片。由于芯片的内核是ARM公司设计的，所以市场上都是在说ARM单片机，很少会讲ST单片机或NXP单片机等等。

ARM公司就是卖专利。

生产ARM单片机的公司：ST(意法半导体)、NXP(恩智浦)、瑞萨、TI(德州仪器)、飞思卡尔、三星、高通、联发科。

### ARM单片机发展史

08年以前，市场上主要芯片：ARM9/11等等。

08年以后，市场上发生了变化，ARM公司也从以前ARM变成了三种系列的芯片：Cortex-A、R、M系列。

Cortex-A系列：主要用于消费电子。搭载Linux操作系统。

Cortex-R系列：主要用于军工方面。市场上流通很少。

Cortex-M系列：主要用于工控、电源控制方面。可以搭载实时操作系统；也可以使用裸机。

### ST芯片简介

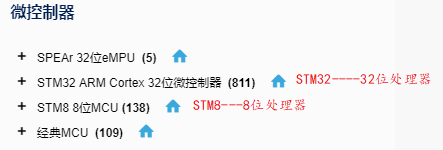
查找ST相关的芯片通过官网查找。

ST官网：<https://www.st.com/content/st_com/zh.html>

查找ST公司的芯片：搜索产品目录 🡪 微控制器

网站：<https://www.st.com/content/st_com/zh/product-selector-welcome2.html>

微控制芯片简介：



STM32系列芯片：



不同产品选用不同的芯片，考虑芯片的时候从以下几个方面考虑：flash、RAM、各种外设、芯片的价格。

### STM32F407ZGT6资源介绍

了解一款芯片资源，简介以及后面所需要开发相关的资料都在官网寻找(国外的芯片都可以这样操作)。

查找相应的芯片介绍以及相关手册资料：

<https://www.st.com/content/st_com/zh/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32-high-performance-mcus/stm32f4-series/stm32f407-417/stm32f407zg.html>

资源介绍：

* Core: ARM®32-bit Cortex®-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator™) allowing 0-wait state execution from Flash memory, frequency up to 168 MHz, memory protection unit, 210 DMIPS/1.25 DMIPS/MHz (Dhrystone 2.1), and DSP instructions
* Memories
  + Up to 1 Mbyte of Flash memory
  + Up to 192+4 Kbytes of SRAM including 64-Kbyte of CCM (core coupled memory) data RAM
  + Flexible static memory controller supporting Compact Flash, SRAM, PSRAM, NOR and NAND memories
* LCD parallel interface, 8080/6800 modes
* Clock, reset and supply management
  + 1.8 V to 3.6 V application supply and I/Os
  + POR, PDR, PVD and BOR
  + 4-to-26 MHz crystal oscillator
  + Internal 16 MHz factory-trimmed RC (1% accuracy)
  + 32 kHz oscillator for RTC with calibration
  + Internal 32 kHz RC with calibration
  + Sleep, Stop and Standby modes
  + VBATsupply for RTC, 20×32 bit backup registers + optional 4 KB backup SRAM
* 3×12-bit, 2.4 MSPS A/D converters: up to 24 channels and 7.2 MSPS in triple interleaved mode
* 2×12-bit D/A converters
* General-purpose DMA: 16-stream DMA controller with FIFOs and burst support
* Up to 17 timers: up to twelve 16-bit and two 32-bit timers up to 168 MHz, each with up to 4 IC/OC/PWM or pulse counter and quadrature (incremental) encoder input
* Debug mode
  + Serial wire debug (SWD) & JTAG interfaces
  + Cortex-M4 Embedded Trace Macrocell™
* Up to 140 I/O ports with interrupt capability
  + Up to 136 fast I/Os up to 84 MHz
  + Up to 138 5 V-tolerant I/Os
* Up to 15 communication interfaces
  + Up to 3 × I2C interfaces (SMBus/PMBus)
  + Up to 4 USARTs/2 UARTs (10.5 Mbit/s, ISO 7816 interface, LIN, IrDA, modem control)
  + Up to 3 SPIs (42 Mbits/s), 2 with muxed full-duplex I2S to achieve audio class accuracy via internal audio PLL or external clock
  + 2 × CAN interfaces (2.0B Active)
  + SDIO interface
* Advanced connectivity
  + USB 2.0 full-speed device/host/OTG controller with on-chip PHY
  + USB 2.0 high-speed/full-speed device/host/OTG controller with dedicated DMA, on-chip full-speed PHY and ULPI
  + 10/100 Ethernet MAC with dedicated DMA: supports IEEE 1588v2 hardware, MII/RMII
* 8- to 14-bit parallel camera interface up to 54 Mbytes/s
* True random number generator
* CRC calculation unit
* 96-bit unique ID
* RTC: subsecond accuracy, hardware calendar

### XYD-Cortex-M4-开发板介绍

《XYDSTM32F4开发板简介.pdf》

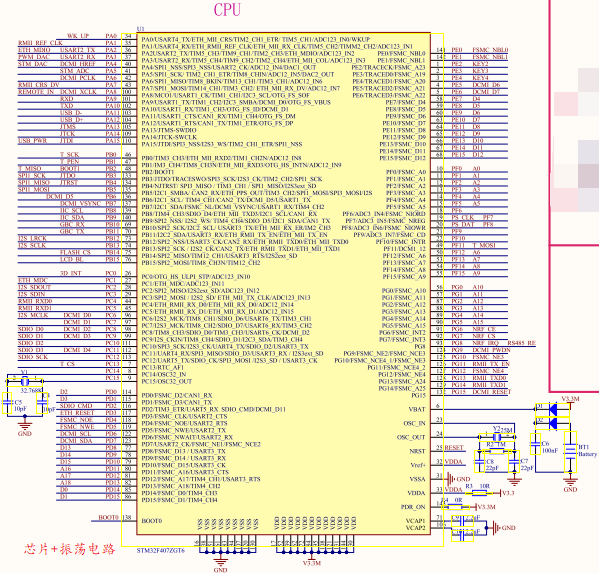
## STM32F407ZGT6最小系统

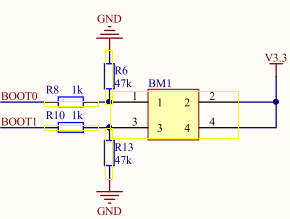
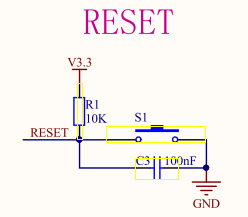
最小系统的作用：芯片能够正常运行的最小单元。

### 最小系统构成

构成：电源电路、振荡电路、复位电路。

离不开的器件或电路：芯片、在ST的芯片中有一个BOOT设置(手册中称为：自举模式)。





### 电源电路

STM32F407ZGT6芯片的供电电源。

STM32F407ZGT6最大承受电压为3.6V。标准电压为3.3V。

从市电需要经过降压处理并且需要将电源进行转换(转成直流电源)。市场上变压器+转换器合起来成为适配器。电源经过适配器以后还需要转换成可以用的电源，需要使用特殊的电源芯片(ASM-1117型号：3.3V)。

补充：

CMOS电平：MOS管能够正常工作的一个电平。STM32F407ZGT6的构成就是MOS管。

高电平：3.3V

低电平：0V

TTL电平：晶体管(三极管)能够正常工作的一个电平。STC89C52RC的构成就是晶体管。

高电平：5V

低电平：0V

### 振荡电路

目的：为芯片正常工作提供心跳节拍。

例如：电脑CPU有一个关注点：主频。

RC振荡器、晶振、声表面波振荡电路。

RC振荡器：频偏比较大；设计非常简单。

晶振：频偏比较小；设计比较复杂；稳定性比较高；容易坏(容易摔坏)。

声表面波振荡电路：频偏最小，设计最复杂；稳定性最高；不容易坏；价格最贵。

### 复位电路

目的：让芯片从头开始正常工作。

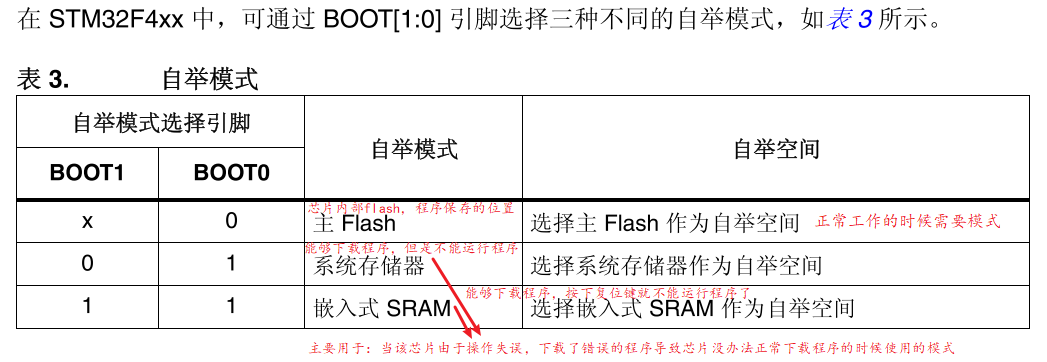
STM32中复位方式有：硬件复位(上电复位)、软件复位、看门狗复位。

硬件复位详解：

高电平复位：高电平能够让CPU复位，称为高电平复位。例如：STC89C52RC

低电平复位：低电平能够让CPU复位，称为低电平复位。例如：STM32F407ZGT6

### 自举模式



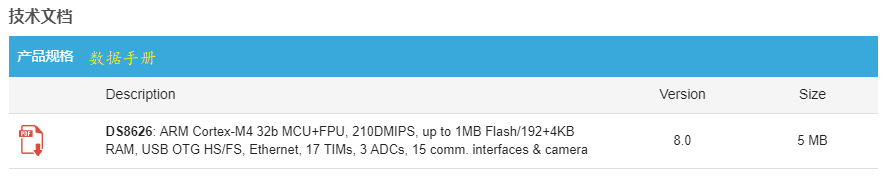
## STM32F407ZGT6芯片架构

目的：通过芯片架构，了解芯片的控制原理以及相关硬件的连接方式。

了解芯片的架构必须查看芯片的数据手册：

官网：[https://www.st.com/content/st\_com/zh/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32-high-performance-mcus/stm32f4-series/stm32f407-417/stm32f407zg.html#design-scroll](https://www.st.com/content/st_com/zh/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32-high-performance-mcus/stm32f4-series/stm32f407-417/stm32f407zg.html" \l "design-scroll)

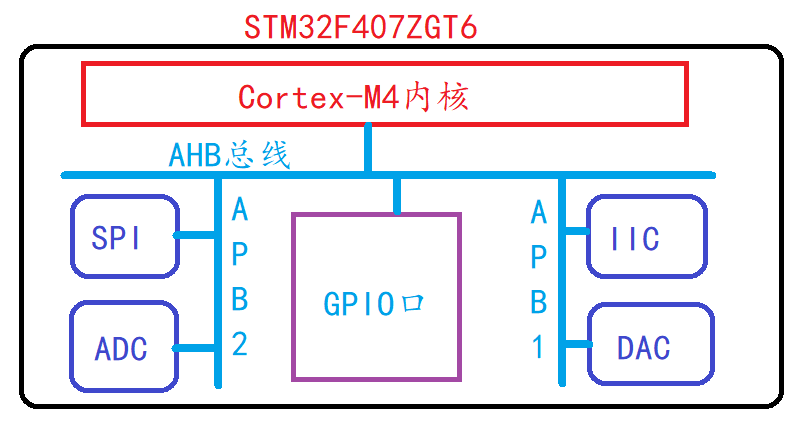








芯片架构图：



## STM32F407ZGT6芯片开发环境

### 开发软件

STM32F407ZGT6芯片开发软件：IAR、MDK-ARM、VSCode等等。

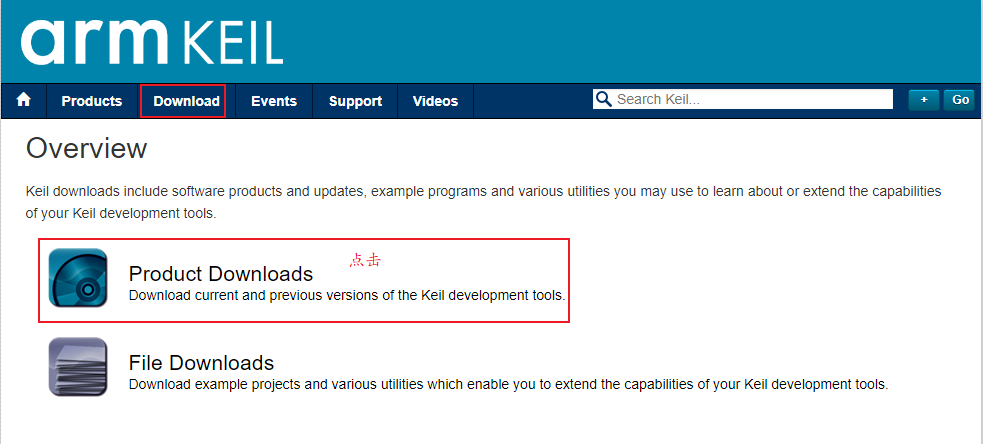
ST官网推荐的是MDK-ARM平台。

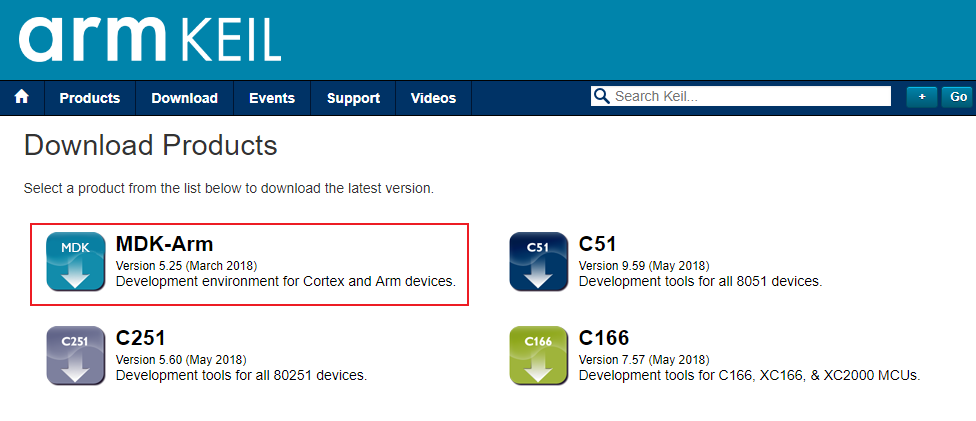
使用MDK-ARM平台(俗称：Keil软件)。

软件获取：从官网去获取。

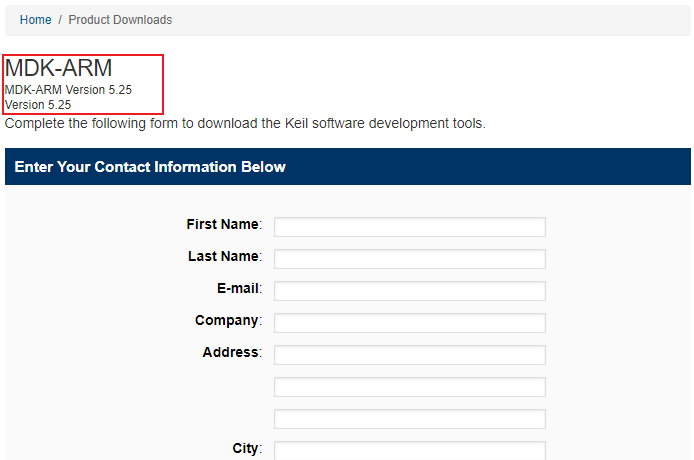
keil官网：<http://www.keil.com/>

找到DownLoad。



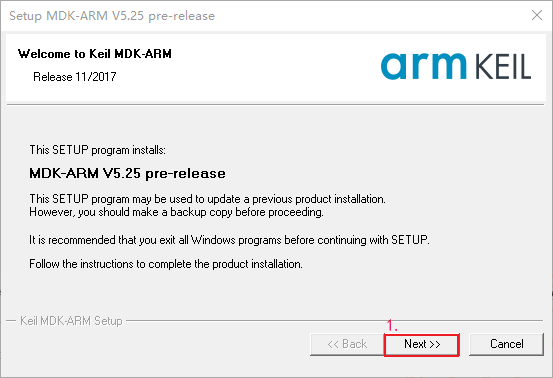


注册以后即可下载相应的软件，注册和下载都是免费的。

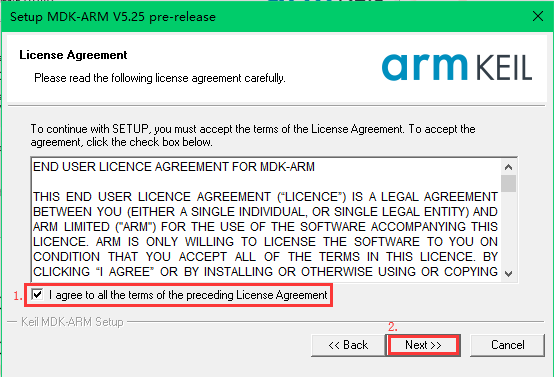


找到软件工具中的软件文件夹，进入软件文件夹，里面有MDK525pre.EXE文件。

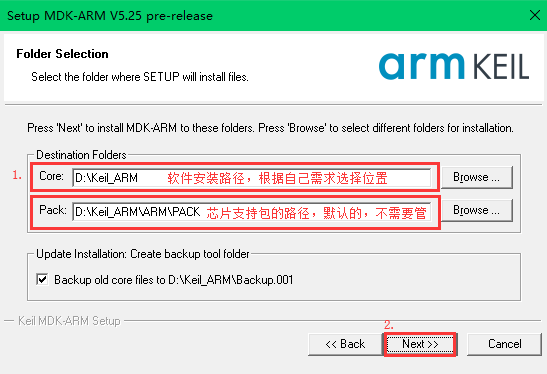
1. 双击运行该软件
2. 选择Next



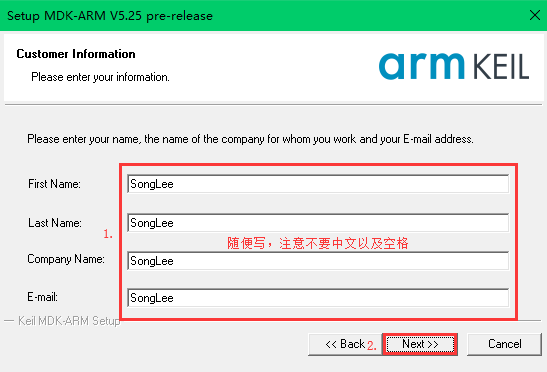
1. 同意许可协议



1. 选择自己的安装路径



1. 填写用户名等信息



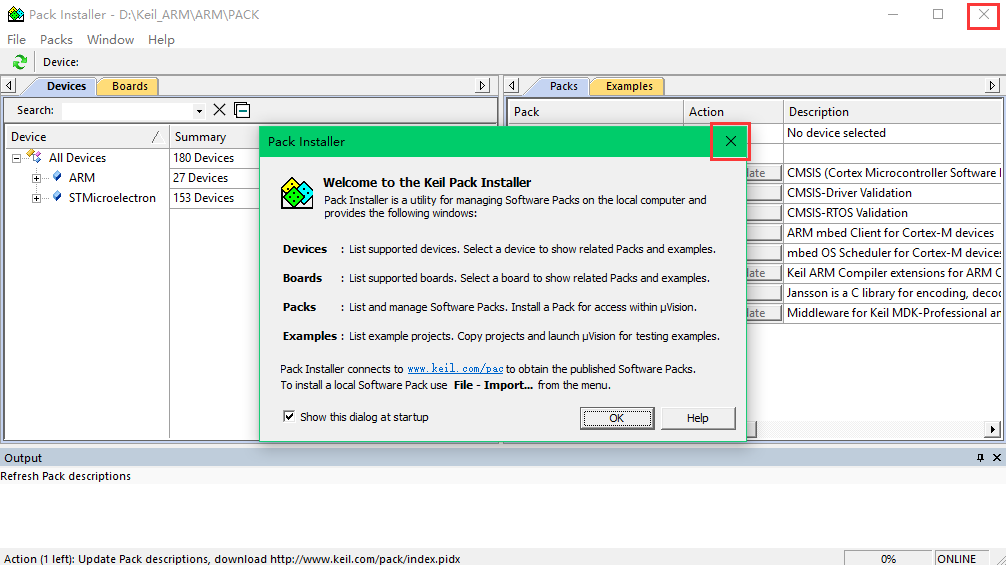
1. 软件正在安装



1. 安装完成



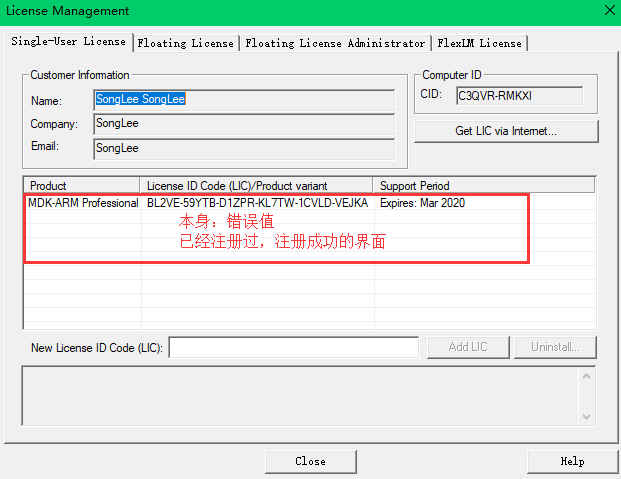
1. 出现Pack安装界面直接关闭



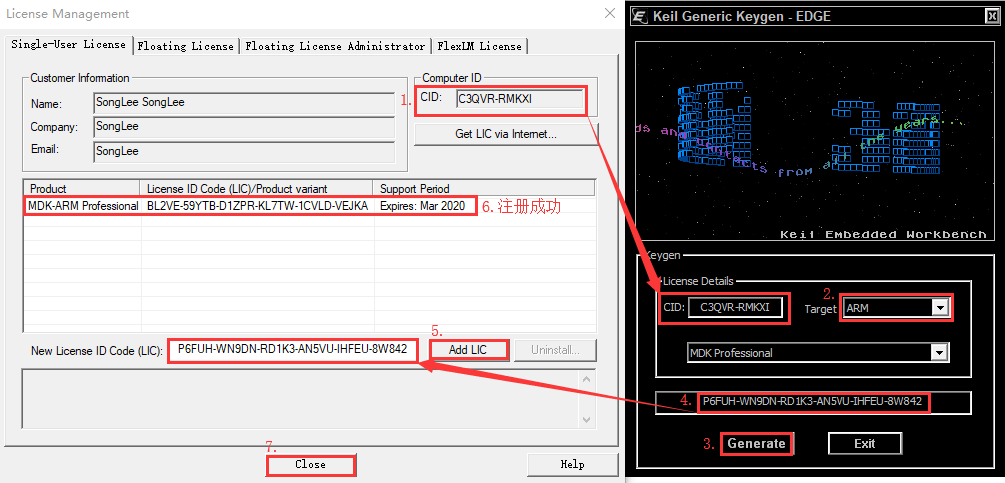
### 开发软件注册器

注册：破解，学习使用可以；商用不行。在工作中使用正版。编译程序代码需要注册后才能编译。

1. 双击运行注册器
2. 点击桌面的软件图标，右键以管理员权限运行。
3. 在软件中任务栏上有File按钮；点击File按钮；选择里面的License。



1. 注册。

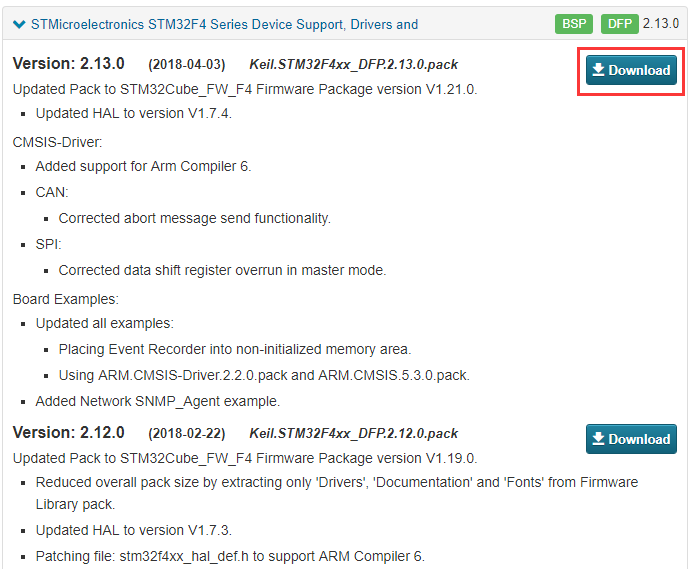


### 开发芯片支持包

目的：不同的芯片进行编译及下载的时候需要芯片支持包的支持。

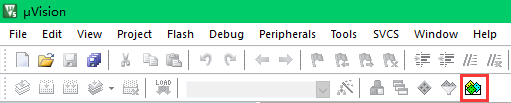
获取芯片支持包。

官网查找：keil的官网中有：<http://www.keil.com/dd2/pack/>

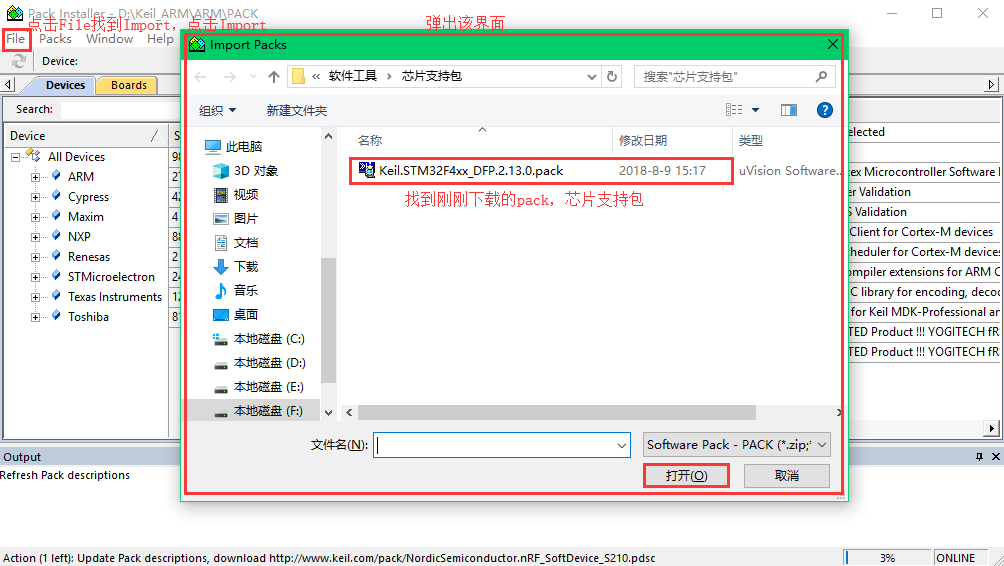


芯片支持包安装：

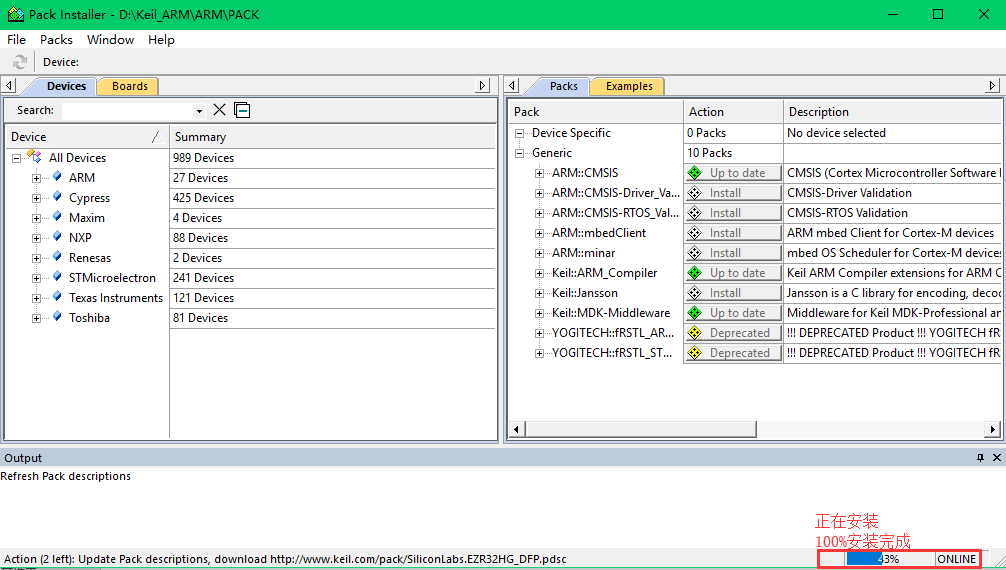
1. 点击图标



1. 选择芯片支持包，点击打开

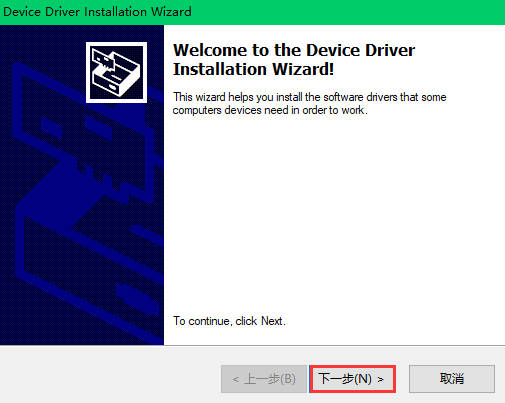


1. 安装中以及完成

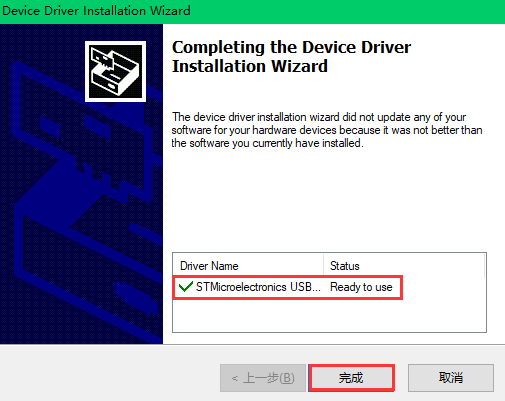


### 开发芯片烧写工具驱动

1. PC连接好烧录器，如果烧录器上的LED灯存在红色的闪烁状态表示没有安装驱动。
2. 打开文件夹：ST-LINKV2 USB driver1.0 for Windows7 and Windows8, 32 and 64 bits
3. 双击运行软件：dpinst\_amd64.exe(64位操作系统使用)
4. 下一步

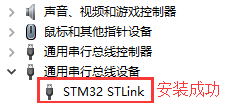


1. 完成



1. 查看是否安装成功

打开设备管理器。电脑🡪属性🡪设备管理器。



### 串口驱动

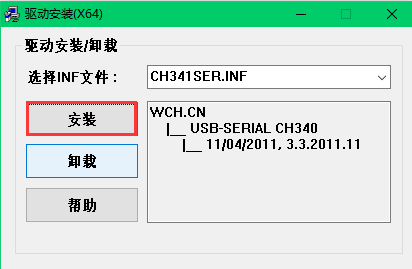
目的：产品如果需要使用串口，在做串口测试的时候需要安装好串口驱动，才能与PC进行串口测试。

不同的串口驱动芯片，驱动程序不一致。

目前使用的是CH340作为串口驱动芯片。安装CH340芯片驱动程序。

1. 进入文件夹：CH340驱动(USB串口驱动)\_XP\_WIN7共用
2. 运行软件：SETUP.EXE
3. 安装软件

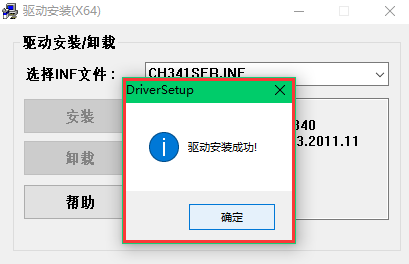
安装之前将开发板与电脑进行连接。



1. 正在安装

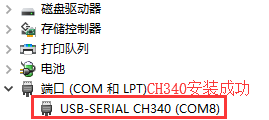


1. 安装成功



1. 查看是否安装成功

打开设备管理器。



## STM32F407ZGT6项目工程创建

编写程序，编译程序文件比较多，一个文件存不了，并且很多芯片都需要启动文件支持，需要创建好项目工程。

STM32中创建项目工程有三类：HAL库的工程(由软件直接可以生成代码)、Lib库工程(日常工作用的最常见的工程)、寄存器版本工程(用的不多，但是学习时用的工程，只有这个才能了解更底层的工作原理)。

HAL库不能使用Lib库的程序。

Lib库不能使用HAL库的程序。

HAL库以及Lib库都可以使用寄存器版本程序。

项目工程是寄存器版本的。

### 新建项目文件夹

1. 选择自己放程序代码的文件夹
2. 在此文件夹下创建好工程文件夹：project\_demo。

### 创建启动文件夹

1. 进入project\_demo文件夹
2. 新建cmsis文件夹
3. 拷贝启动文件到cmsis文件夹下。
   1. 获取启动文件

官网(ST官网)：

<https://www.st.com/zh/microcontrollers/stm32f4-series.html?querycriteria=productId=SS1577>

* 1. 进入官网后找到旁边的工具与软件



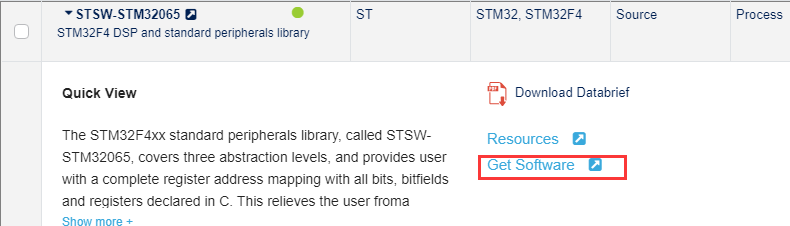
* 1. 找到Lib库程序包，库程序包中有相应的启动文件



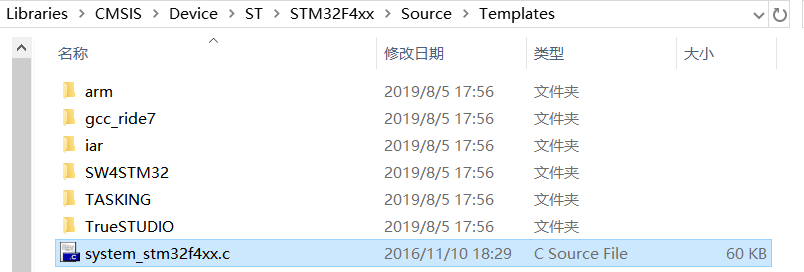
* 1. 下载相应文件

官网链接：

<https://www.st.com/content/st_com/zh/extended-query.html?querycriteria=productId=LN1939$$associatedTo=SS1577>



* 1. 在参考资料中已经进行包含：en.stm32f4\_dsp\_stdperiph\_lib
  2. 进入文件夹：en.stm32f4\_dsp\_stdperiph\_lib\STM32F4xx\_DSP\_StdPeriph\_Lib\_V1.8.0\Libraries\CMSIS
  3. 进入文件夹：CMSIS\Device\ST\STM32F4xx\Include；拷贝里面的所有文件到cmsis文件夹中。//
  4. 进入文件夹：CMSIS\Device\ST\STM32F4xx\Source\Templates；拷贝system\_stm32f4xx.c到cmsis文件夹中。//



* 1. 进入文件夹：CMSIS\Device\ST\STM32F4xx\Source\Templates\arm；拷贝startup\_stm32f40\_41xxx.s到cmsis文件夹中。//
  2. 进入文件夹：CMSIS\Include；拷贝里面所有文件到cmsis文件夹下。//

拷贝完成。

### 创建编写代码文件夹

以后自己写的程序代码放在该文件夹下。创建一个文件夹user。

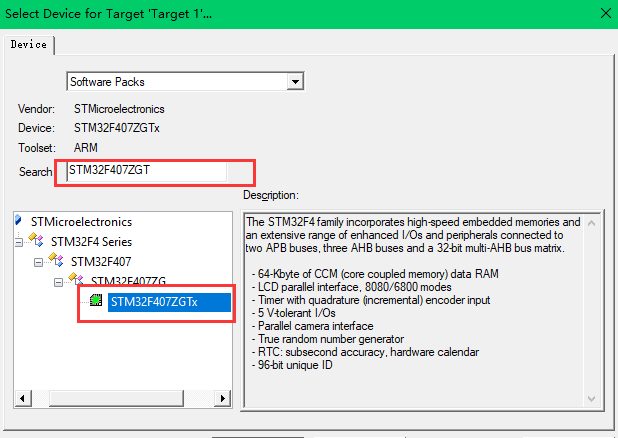
在user文件夹下创建inc和src文件夹。

inc：存放自己写的头文件

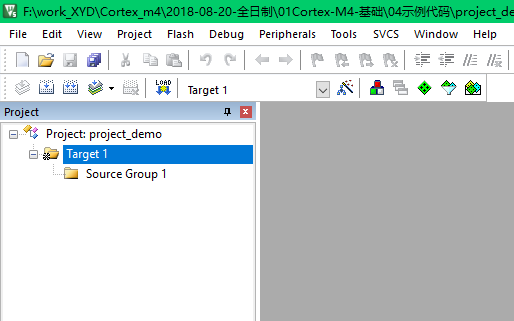
src：存放自己写的源文件

### 创建项目工程

1. 打开软件
2. 新建工程：选择软件菜单栏中的Project。Project中有New Project。
3. 在弹出界面中，用来保存工程的路径设置。保存到project\_demo文件夹下
4. 保存以后，在弹出的框中选择目前使用的芯片型号：STM32F407ZGT6

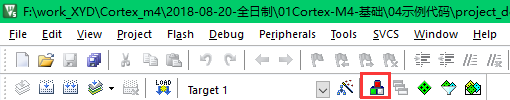


1. 点击OK，将弹出的界面关闭。

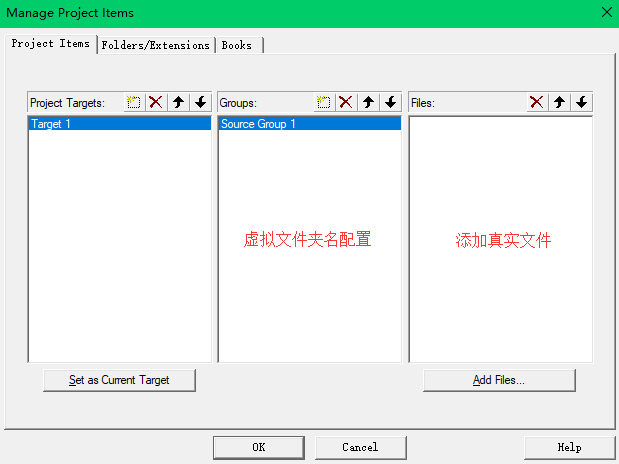


### 配置项目工程

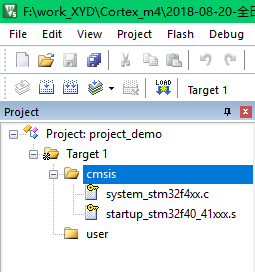
1. 配置虚拟文件夹：分明别类的存放文件，与真实文件夹下的文件一致。



配置虚拟文件夹图标。

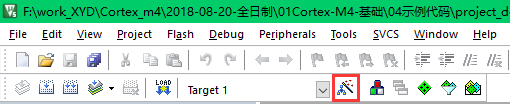


虚拟文件夹以及文件配置OK。

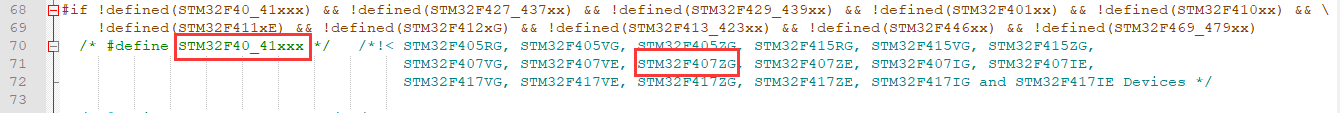


1. 配置工程编译

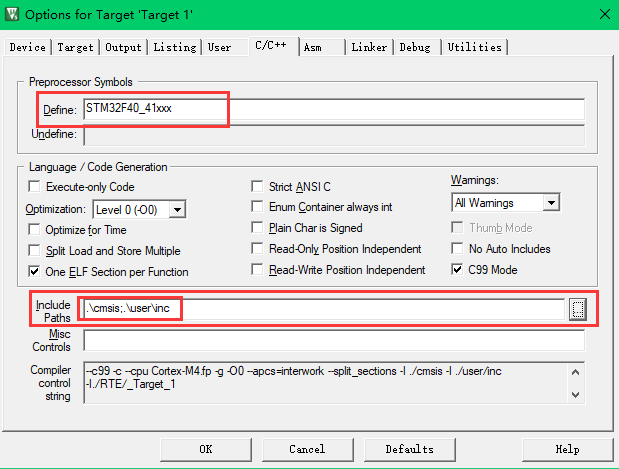
点击图标进行配置。



C/C++中配置宏在stm32f4xx.h中查找

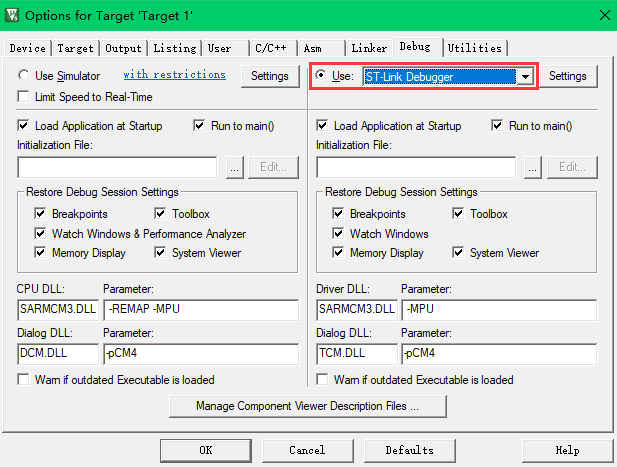


配置好头文件路径以及宏定义：

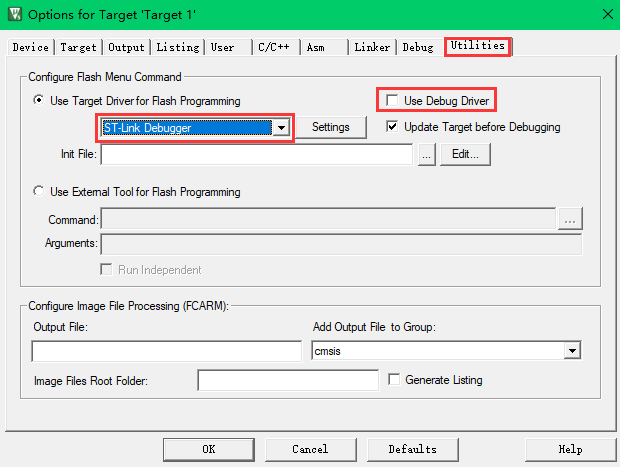


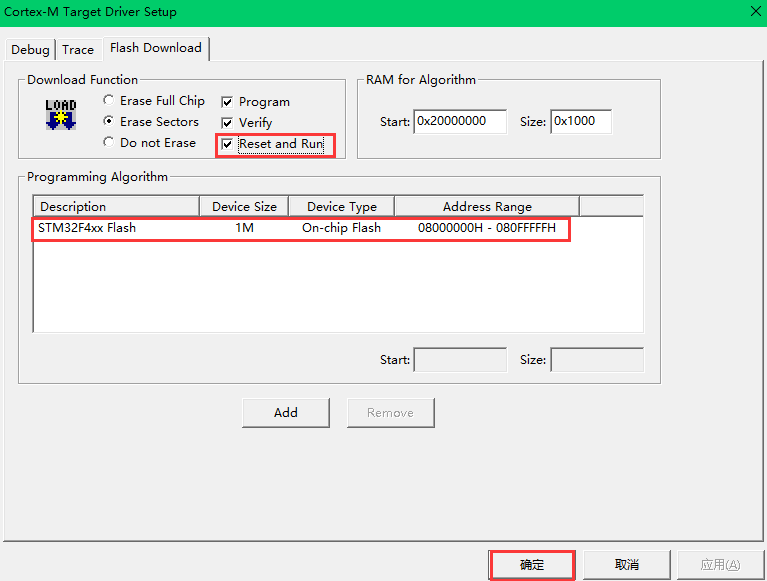
配置调试与下载部分：

调试：选择ST-Link



下载：选择ST-Link以及点击setings配置好复位and运行；flash文件添加好(默认添加OK的)。



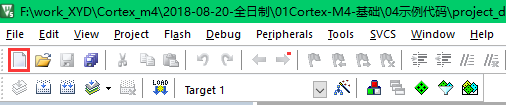


工程配置到此已经完成。

### 编写主函数并且编译工程

1. 新建源文件，由于现在存放的是main函数，文件名叫做main.c。

点击图标：

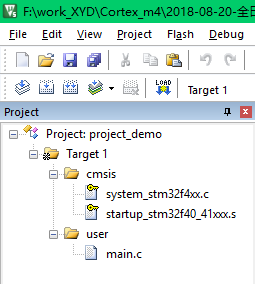


保存文件：使用ctrl+s。

保存文件位置：user下的src中。

1. 由于user已经有了一个main.c，需要在虚拟文件中添加进来。

方法：通过双击user这个虚拟文件夹。进入user的src文件夹下。



1. 编写好主函数。

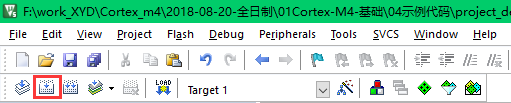
包含头文件路径：#include “stm32f4xx.h”

编写主函数：

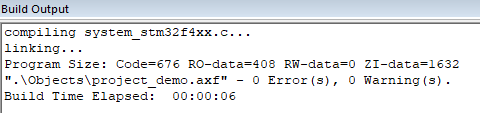
|  |
| --- |
| int main(void)  {  while(1)  {  ;  }  } |

1. 编译工程

点击编译图标：

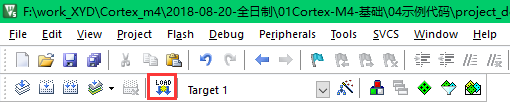


编译结果：



1. 下载程序到芯片

点击图标：



下载完成：

