

【工具使用】Keil5软件使用-基础使用篇

原创 知识噬元兽 已于 2023-07-07 20:01:22 修改 93523 收藏 1361

分类专栏: 工具使用 # Keil 文章标签: 单片机 嵌入式硬件

InsCode 文章已被社区收录

工具使用 同时被 2 个专栏收录

80 订阅 15 篇文章

写文章 写代码 发动态 提

他们都在参与话题



如何看待unity新的收费模式?

创作者商单

 创作稿酬 200 供稿得现金奖励,

一、概述

本文面向未接触过 **Keil** 的新手，如果是职场老手可跳过此篇。为了快速上手，本文会跳过很多细节及解释，如需要了解原理，请移步进阶篇。

二、软件介绍

Keil提供了包括 **C编译器**、宏汇编、链接器、库管理和一个功能强大的仿真调试器等在内的完整开发方案，通过一个集成开发环境（**uVision**）将这些部分组合在一起。

目前软件对中文的支持不友好，不建议安装网上的一些汉化包之类的。另外建立的工程文件路径也尽量不要存在中文，否则可能会出现一些异常。

演示版本：5.24a

三、名词释义

IDE：Integrated Development Environment，是开发集成环境的意思，一般包括代码编辑器、编译器、调试器和图形用户界面等工具。因为做嵌入式开发的，首先需要有可以编写代码的编辑器，然后需要有把代码转换成机器可执行代码的编译器，有时候可能需要在对应硬件芯片上调试，那就需要有个调试器，为了操作方便，一般会把这些都图形化。这么下来的一整套工具集成在一个应用程序里，就是所谓的IDE，这篇文章介绍的Keil uVision就是IDE的一种。

Keil：其实Keil是个公司名，其旗下开发的软件都以Keil开头命名。目前有Keil MDK-ARM、Keil C51、KeilC166和KeilC251等版本。

uVision：uVision是由keil公司开发的集成开发环境（IDE），可以进行代码编辑，文件管理，程序的编译调试等。

MDK、C51：如果想要开发ARM内核的芯片，需要安装MDK-ARM，如果想要开发51内核的芯片，那就需要安装c51。也就是说，MDK、c51只是Keil针对不同芯片内核开发的不同开发集成环境。

CMSIS：ARM Cortex™ 微控制器软件接口标准(CMSIS: Cortex Microcontroller Software Interface Standard) 是 Cortex-M 处理器系列的与供应商无关的硬件抽象层。

烧录：烧软件只是一种叫法，来源于最早刻录光盘，因为是用激光讲数据烧在光盘上，所以后来人们把这种不可逆的记录方式称之为烧录。

bin文件：二进制文件，其用途依系统或应用而定。一种文件格式binary的缩写。一个后缀名为".bin"的文件，只是表明它是binary格式。一般来讲是机器代码，汇编语言编译后的结果。

axf文件：axf文件是ARM芯片使用的文件格式，它除了包含bin代码外，还包括了输出给调试器的调试信息，例如每行C语言所对应的源文件行号等。

Hex文件：Intel HEX文件是由一行行符合Intel HEX文件格式的文本所构成的ASCII文本文件。在Intel HEX文件中，每一行包含一个HEX记录。这些记录由对应机器语言码和/或常量数据的十六进制编码数字组成。Intel HEX文件通常用于传输将被存于ROM或者EPROM中的程序和数据。

四、下载链接

直接[官网](#)下载，用注册机生成Lisence即可，个人用没什么问题，企业用建议购买正版，不然会被查到的。（会有邮件警告到公司老总那去，别问我为什么知道）

五、软件使用


1、软件安装

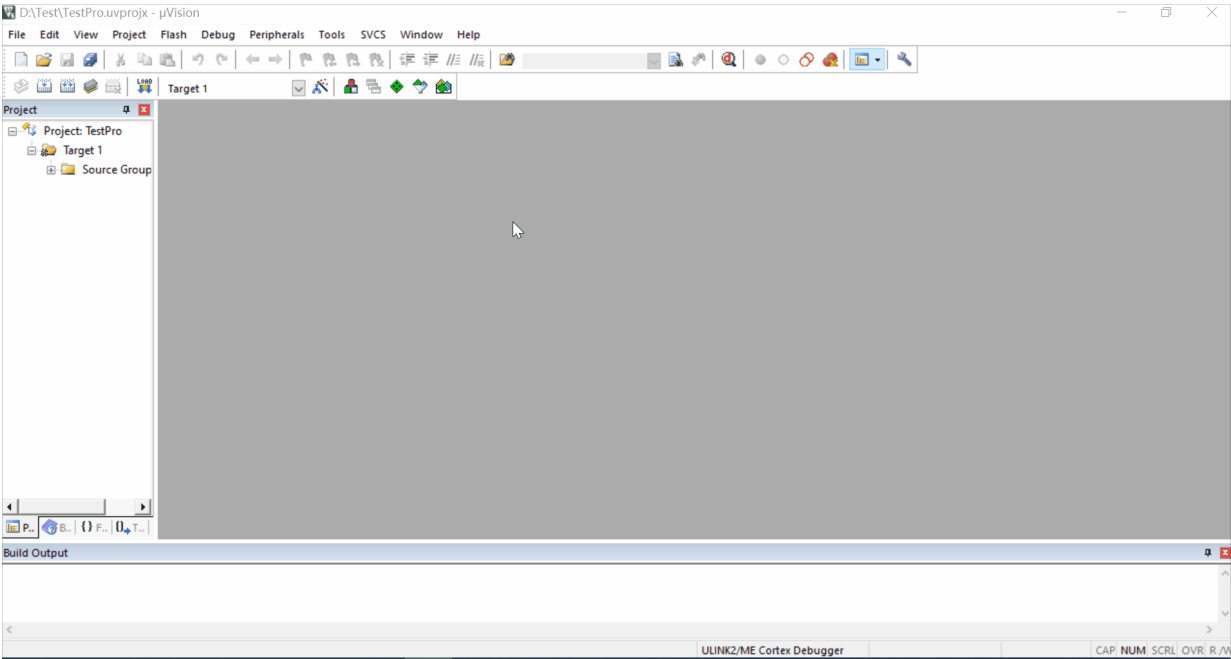
这个没什么特殊的，一路next就行。如果是自己用的，一般不建议装在C盘，因为后续芯片包都会默认安装在此目录下，文件会变得很大。

2、下载/安装芯片包

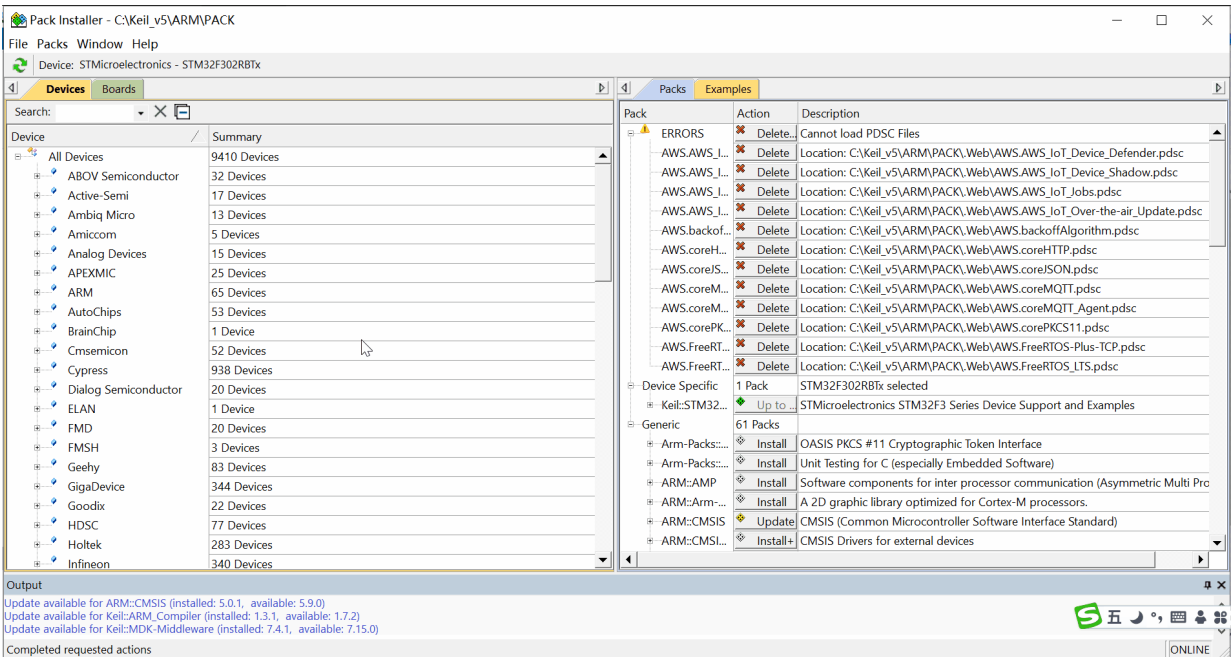
跟旧版本的Keil不一样，Keil4以前是软件里集成芯片包，也就是只要下载安装完Keil的软件，就有对应的一些芯片可供选择使用。但由于现在软件支持的芯片越来越多，软件的体量也就越来越大，如果把所有芯片包都跟软件打包一起下载，那软件估计得有几百个G。所以为了简洁，Keil把集成环境跟芯片包、例程等东西分开单独下载，具体要用到什么芯片再自己去下载安装对应的芯片包。

• 在Keil里下载

点击"Pack Installer"，也就是这个图标，第一次打开可能会很慢，因为要从官网自动更新所有芯片包的信息。这时候很坑的一个点是，你不更新，可能又没法找到你想要的芯片，所以在信息更新完前，只能乖乖地等它更新完。然而这种国外链接，又时不时给你断一下。



在更新完后，在左边的窗口中找到想要安装的芯片型号，可以选择整个系列（或具体芯片，结果是一样的），这里以STM32G071为例。然后在右边窗口中，找到"Device Specific"里面"Keil:STM32G0xx_DFP"这一项，点击按钮"Install"下载安装。（同样的，网络卡的时候下载很慢，还经常断开链接）



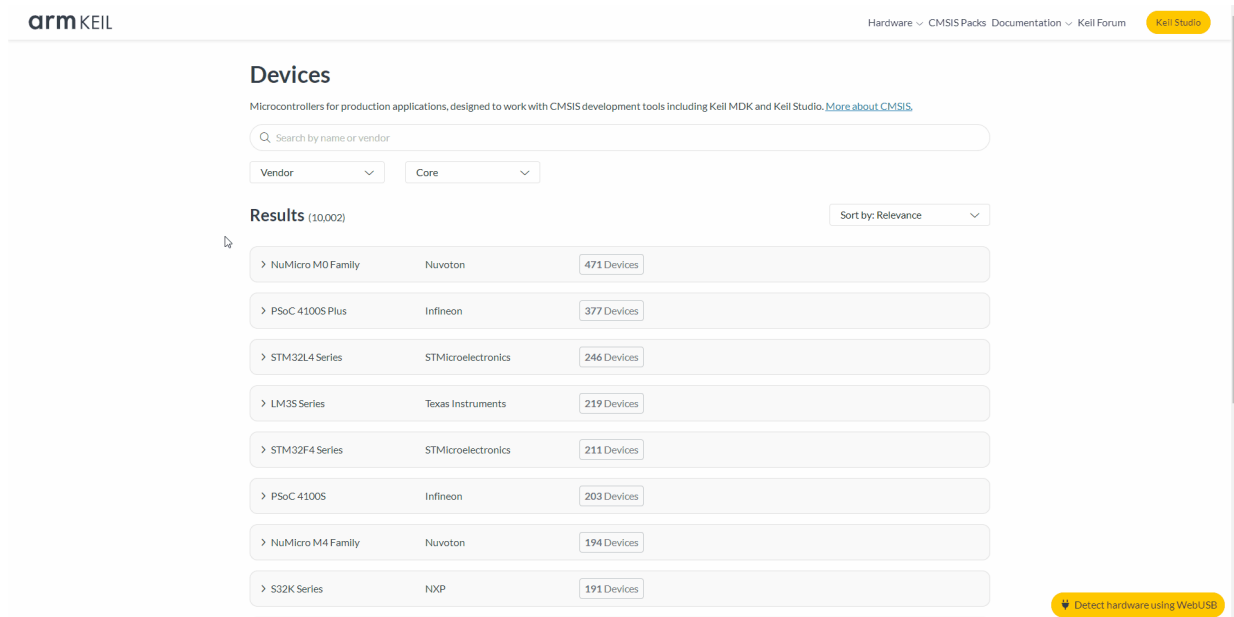
等到下载安装完成后（右下角有进度条），"Install"键会变成"up to date"，这时候说明安装成功。

• 在官方网站下载

前面那种方法在很多时候都是行不通的，不是网络不好就是网络不好。所以可以通过另外一个方式更新芯片包，那就是先从官网提前下载好芯片包，再点击安装。

官网链接：<https://www.keil.com/dd2/Pack/>

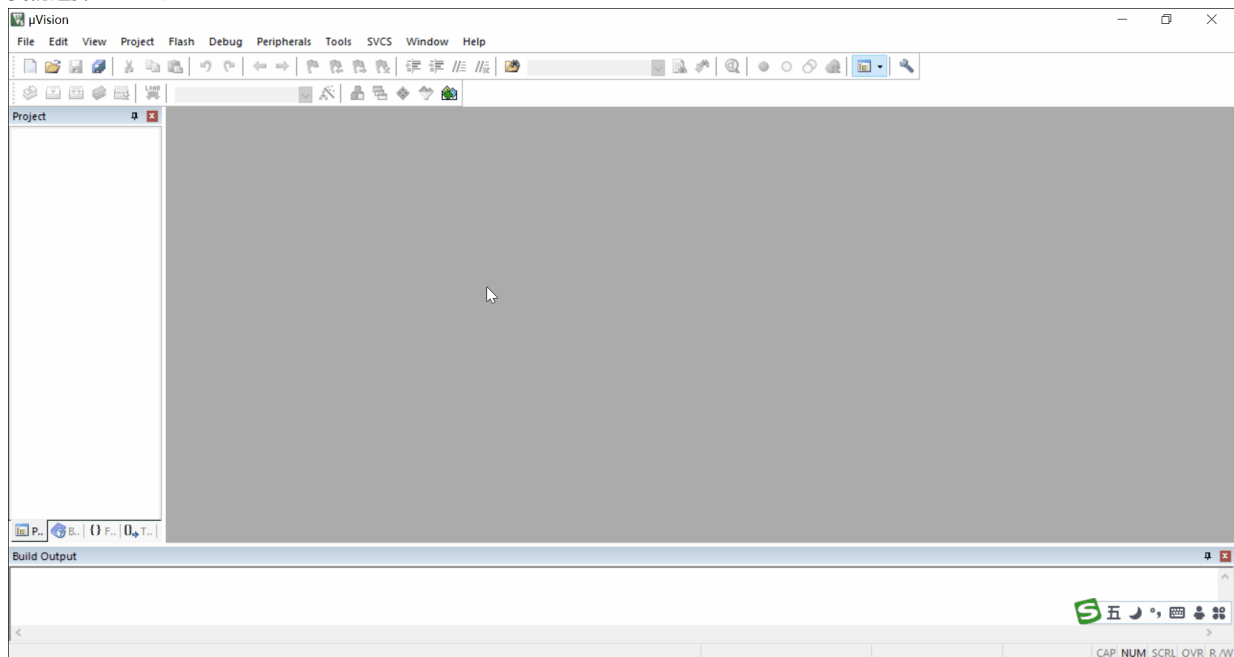
同样，先找到对应的芯片（这里展开时可能不会显示所有芯片，找不到对应芯片时，点击下面"View all in family"展示所有芯片），选择"xxxxx_DFP"，点"Download Recommended Pack"即可下载。



这种方式有个好处就是即使在官网无法下载时，也可以假借他人之手去下载。为了防止小伙伴们在官网下载也出现网络中断的情况，这里提供STM32常用系列的芯片包供下载使用（版本可能不是最新的，可以凑和着用）。

3、新建工程

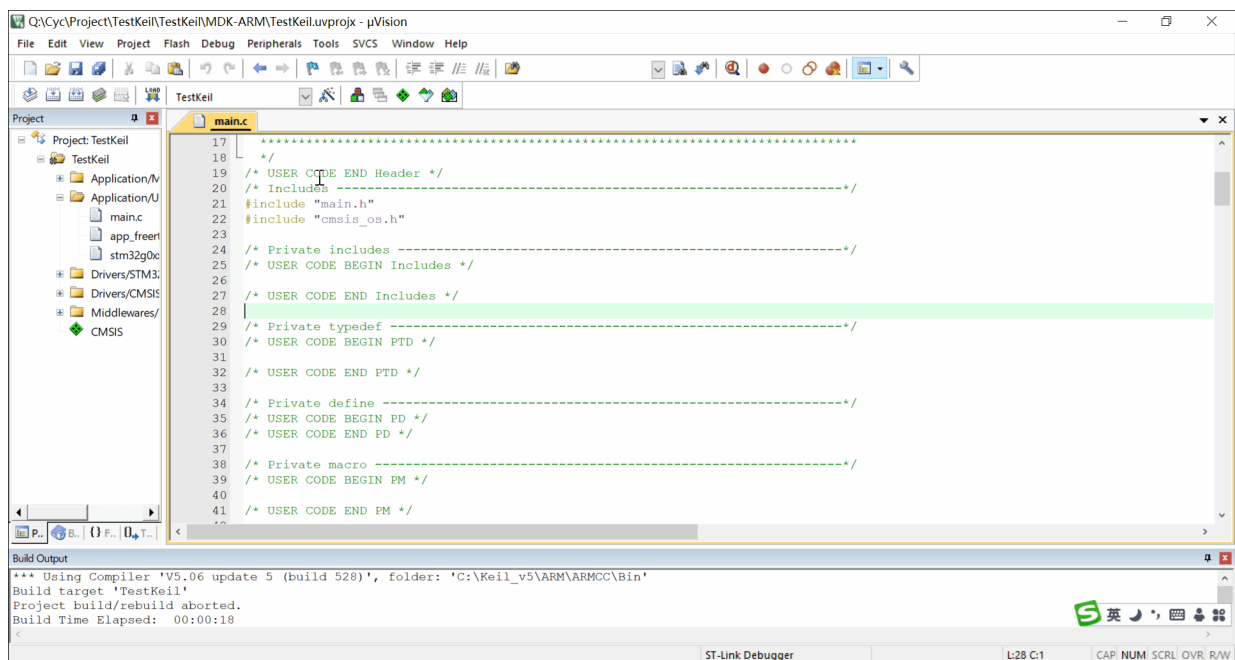
点击"Project"-">"New Project"新建工程，在弹出的窗口中选择工程要保存的路径，设置好工程名称，点击保存。由于是新建工程，软件会默认弹出芯片选择框，让你选择当前使用的单片机型号，这里我们选择STM32G071RBTx，然后是选择接口标准，这里我们选择CMSIS。



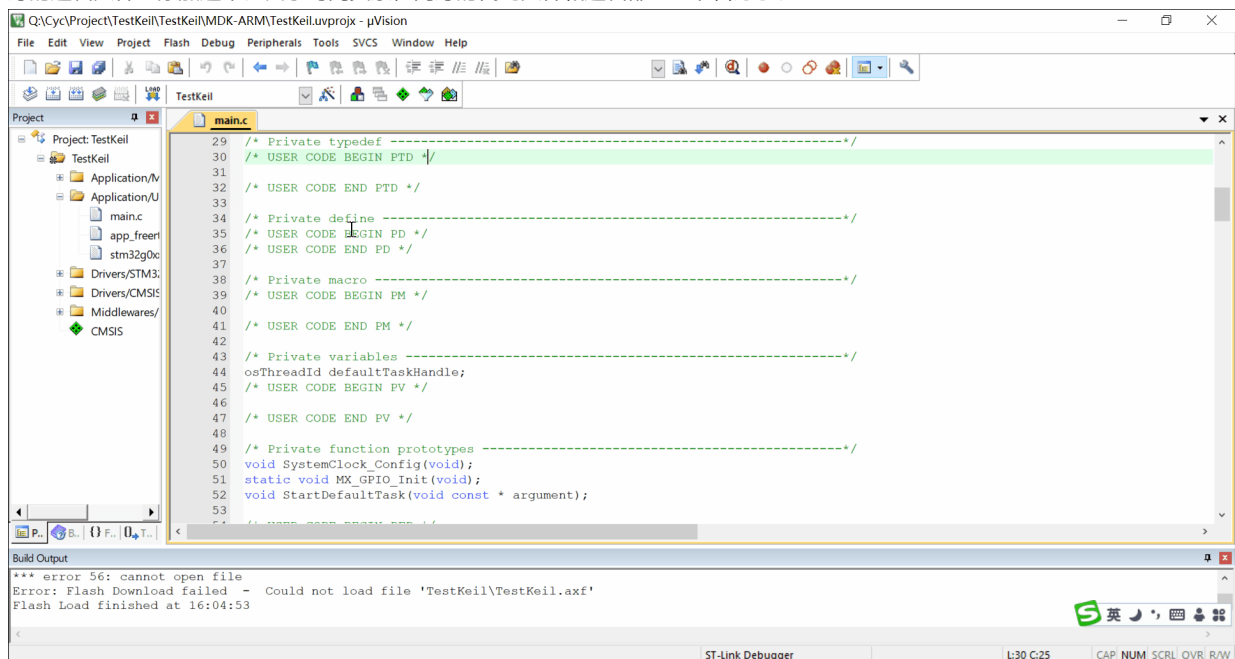
做完以上几步，一个简单的工程就新建完成了。由于时间关系，这里我们使用了超能力——STM32CubeMX配置并生成了一个工程。这里面实现了调试口的初始化及一个GPIO口的初始化，也就是下来我们要用到的LED灯所使用的PA5口。

4、新建/添加文件

有了工程还不够，就像建房子一样，这时候只是把钢筋水泥弄好，做出个房子的外型，离真正可以居住的温馨小屋还差亿点点距离。所以这时候就得加点核心进来，也就是我们的源代码。



因为这里有演示使用的是STM32CubeMX生成的工程，所以里面已经添加好一些初始化的代码文件，我们现在要做的就是将闪烁的逻辑文件也添加进来。由于时间关系，闪烁的代码文件和逻辑都已经准备好了。

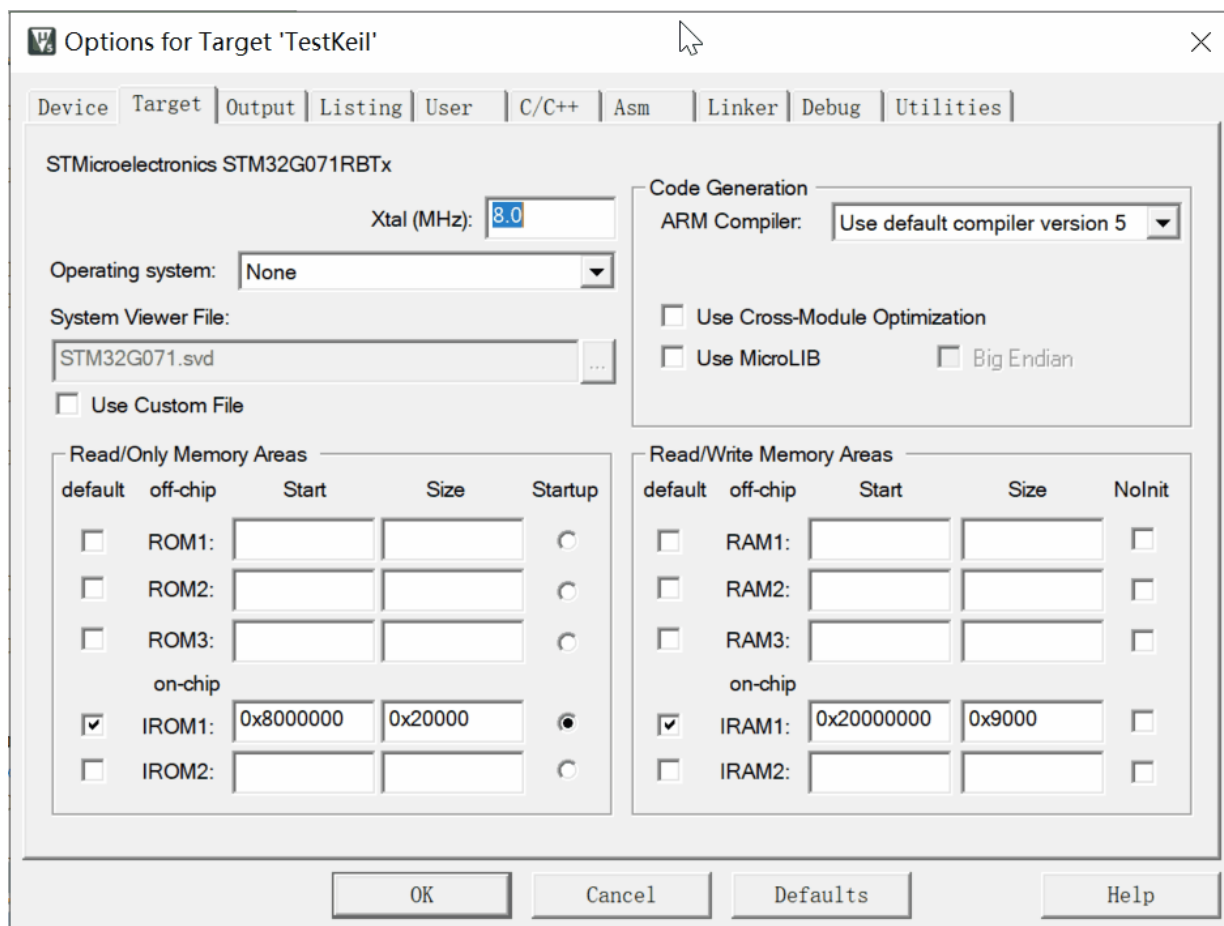


5、配置工程

为了可以正常地跟开发板及烧录器进行匹配，这里我们还需要对工程进行一些配置。

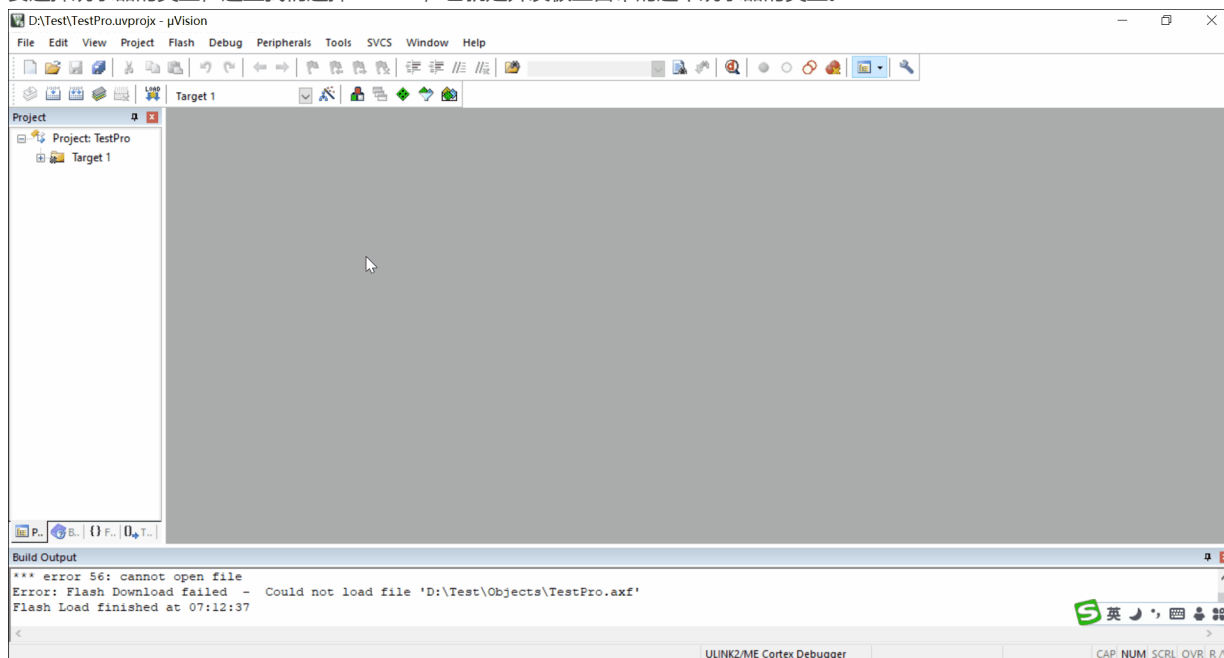
• 包含头文件路径

前面只是添加了必要的核心代码文件，这时候这些文件还是属于各自管各自的。虽然代码里可能有在文件A里引用了B文件（include进行包含），但实际Keil这个软件并不知道A文件B文件各自放在什么地方，要去哪里找。所以需要我们要告诉它每个文件的路径。由于.c文件是直接添加到工程里的，此时文件路径就已经包含进来，但是对于.h文件，软件还不清楚路径在哪，所以我们需要手动添加.h所在的路径。



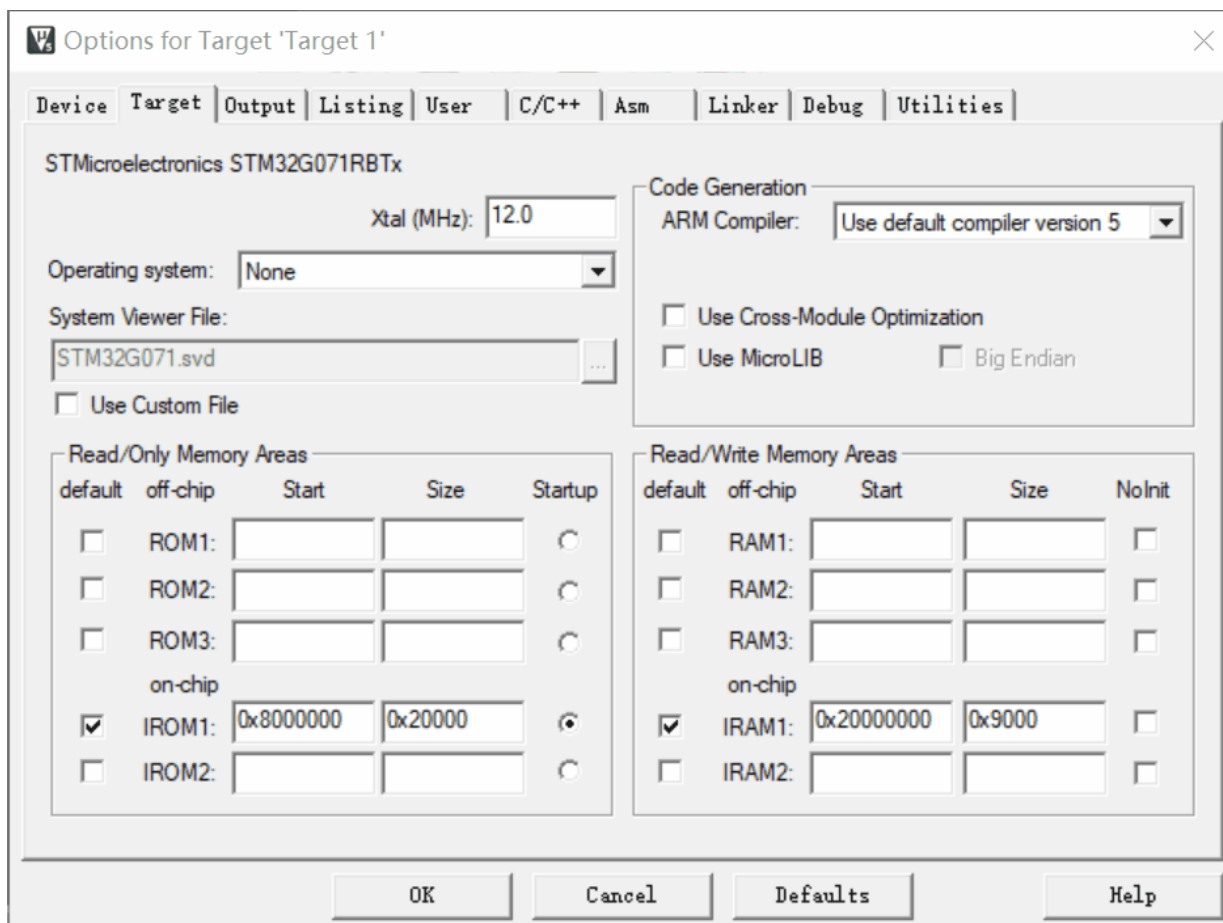
• 选择烧录器

这里可以选择在线调试或离线仿真两种方式，如果有开发板，建议选在线调试直接在板子上进行实验。选择了在线调试后，需要选择烧录器的类型，这里我们选择ST-Link，也就是开发板上自带的这个烧录器的类型。



• 生成Hex

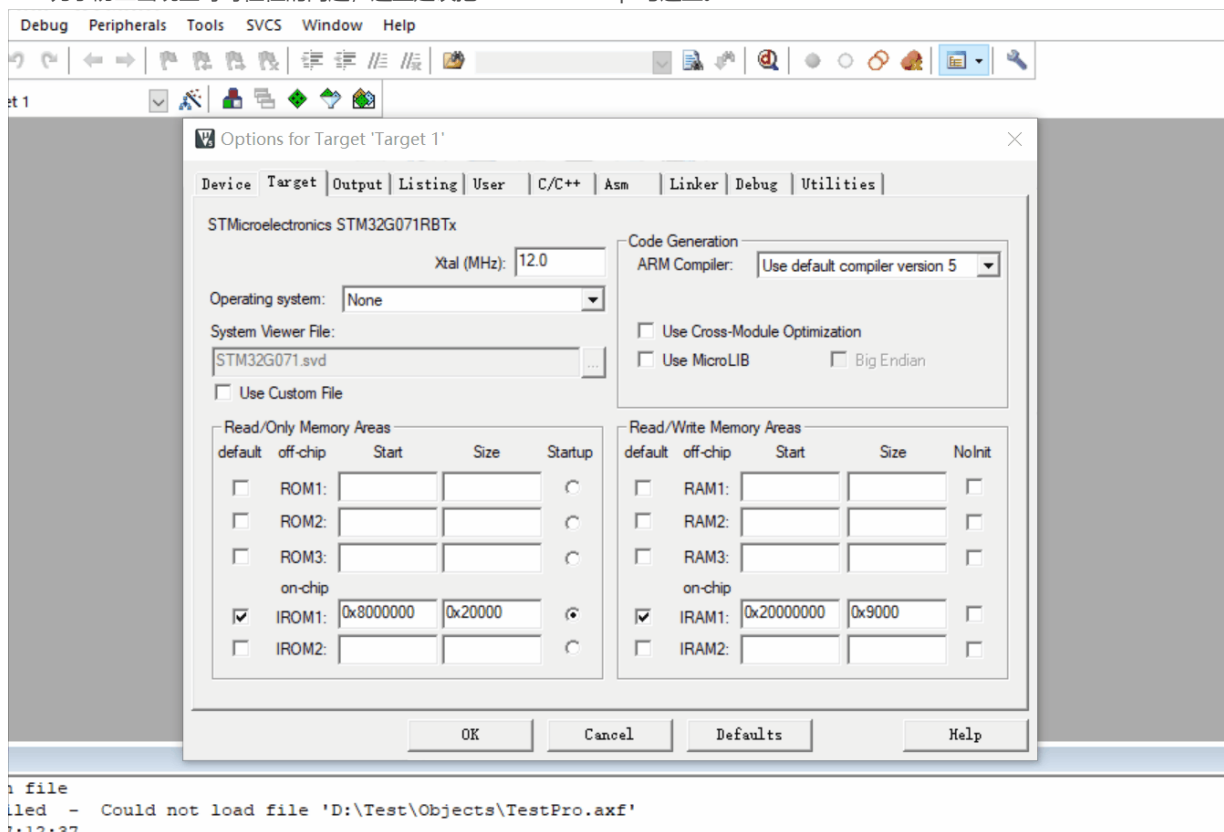
有些场合下，只能通过Hex文件进行烧录，这时候就需要在工程里勾选"Create HEX File"，这样在编译之后工程会自动生成Hex的可执行文件。



• 选择烧录后复位

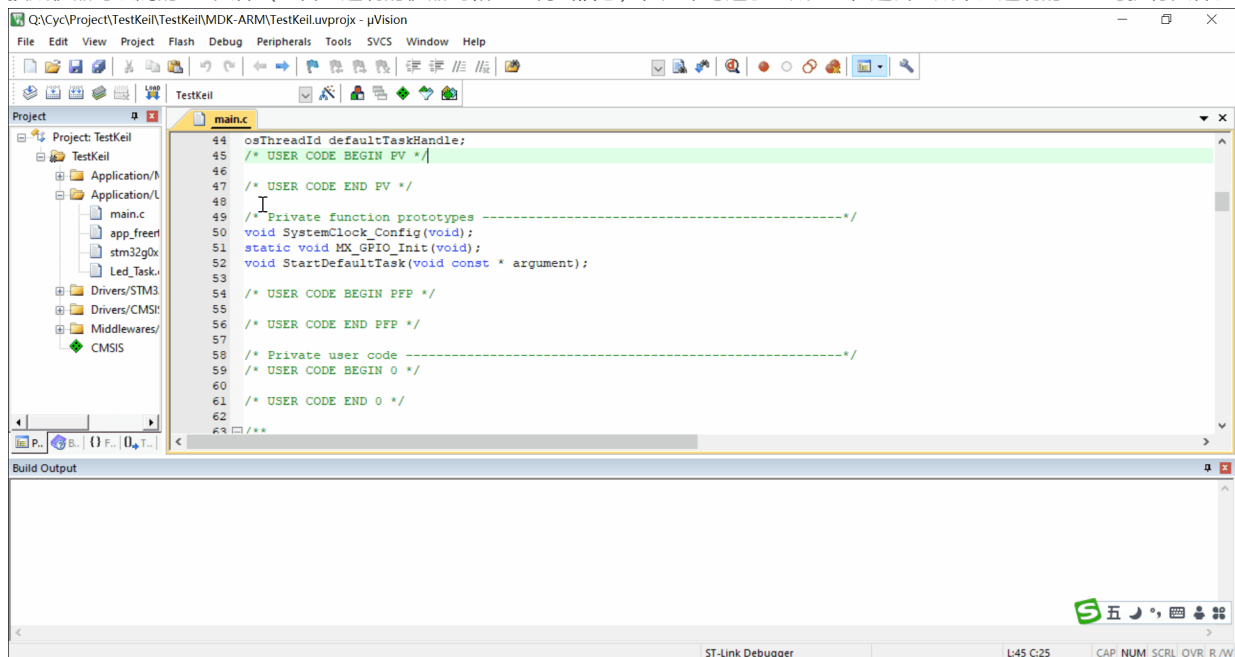
选择"Reset and Run"可以在软件烧录后立即运行，不然会发现一个现象，就是烧完后程序不会跑，需要手动复位后才能跑起来。

为了防止出现些奇奇怪怪的问题，这里建议把"Erase Full Chip"勾选上。



6. 编译

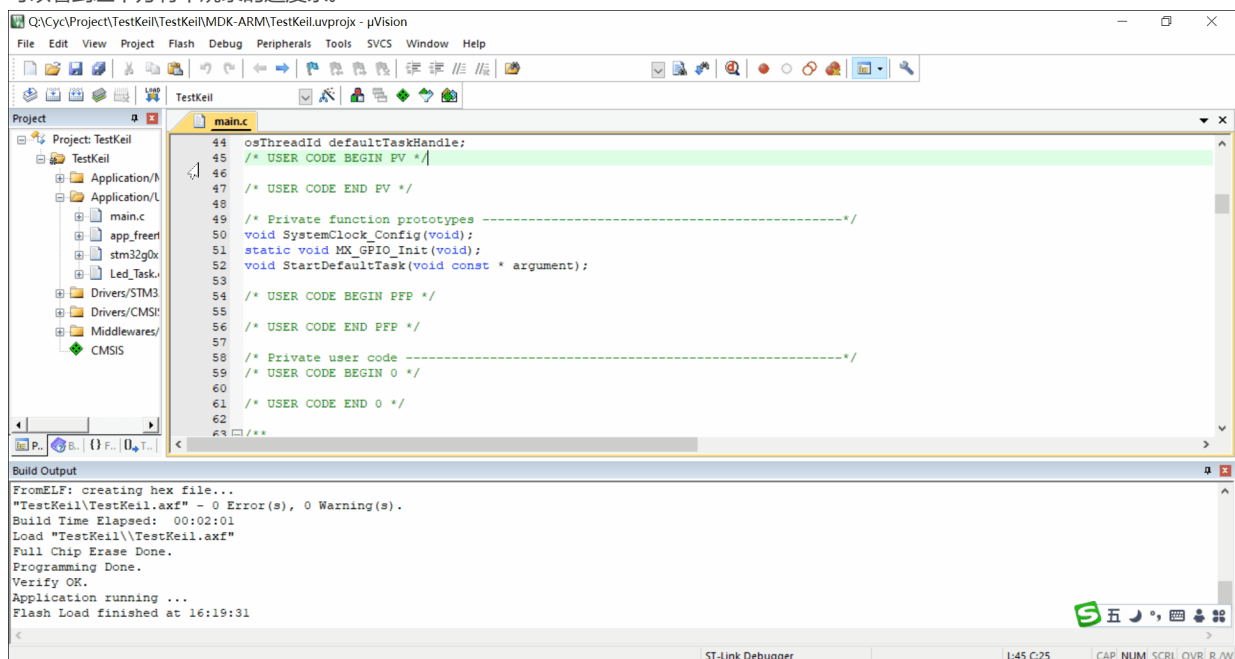
添加完文件，配置好工程后，我们就可以点编译，这时候软件会自动调用编译器、链接器等一步步把源代码（.c.h文件），转换成机器可识别的.axf文件（包含二进制的机器码和一些调试信息），如果勾选了生成Hex，还会生成十六进制的.Hex可执行文件。



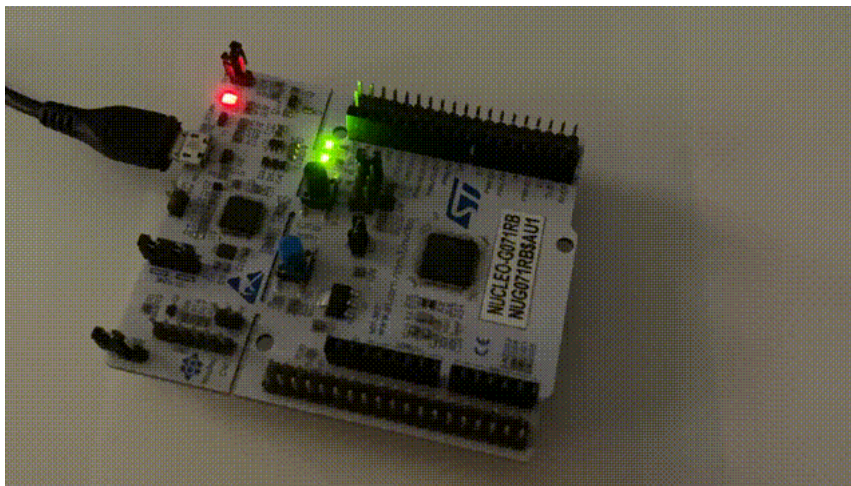
注：如果此时出现了Error信息，则无法编译通过，需要修改源代码直到编译结果0-Error，才可以正常生成可执行文件。如果出现Warning信息，可视情况进行处理，Warning信息不影响编译结果，但建议多关注下这些Warning信息，很多Bug就隐藏在里面。

7、烧录

经过前面的步骤，我们已经得到了机器中识别的执行文件.axf。并且工程也已经配置了对应的烧录器及芯片，此时先连接烧录器到开发板上，点击"Project"-">"Optinos for Target 'xxx'"-">"Debug"-">"Setting"查看当前是否已正常连接。如果已正常连接，点击，可以看到左下方有个烧录的进度条。



烧录完成后，我们看下板子，灯开始正常闪，说明程序已经正常跑起来了。



8、调试

• 在线调试

在有开发板的情况下，可以把软件烧进单片机中，并使用烧录器对运行的程序进行实时监控。可查看当前单片机内核、外设、ROM、RAM的变化情况，并且Keil提供了一套比较友好的人机交互界面，可以帮助快速定位查找问题。

• 离线仿真

如果是没有开发板的情况，想要大致跑下自己写的逻辑有没有问题，可以使用离线仿真这种方式。但如果逻辑中涉及跟单片机外设或开发板外围电路的一些配合，则不适合使用这种方式调试。

六、注意事项

1. 工程路径下不可以有中文，否则可能编译或调试时会有影响。
2. 工程文件路径不可过长，这个软件对路径的识别是有长度限制的，当文件的总路径长度超过一定限制后，编译时可能会报无法查找到相关文件。

七、相关链接

[Keil5软件使用-进阶工程配置篇](#)、[Keil5软件使用-进阶调试篇](#)

显示推荐内容



知识噬元兽

关注

👍 191



★ 1361



💬 6



专栏目录