

# Hanker TI M4 开发板开发应用教程

作者: 青风

QQ: 157736409

淘宝店: <http://qfv5.taobao.com>

邮箱: wanqin\_002@126.com

硬件平台: Hanker 开发板 v2.2

### 第三章 Hanker TM4C123 快速入门

#### 3.1 keil MDK 集成开发环境

这节主要讨论一下如何使用 keil MDK 进行 TM4C 的开发与使用。使用版本建议大家选择 MDK 4.5 到最新的 MDK4.7 这个范围内的版本。

后面的开发都是基于 TM4C 库函数的开发，那么我们首先需要预备下载一下 TM4C 的库，给一个 TI 的官方地址下载：

[http://www.ti.com/lscds/ti/microcontroller/tiva\\_arm\\_cortex/c\\_series/tm4c\\_arm\\_cortex-m4/tools\\_software.page#tivaware](http://www.ti.com/lscds/ti/microcontroller/tiva_arm_cortex/c_series/tm4c_arm_cortex-m4/tools_software.page#tivaware)

这个网页下一个 TIVAware，下载后名字为：SW-TM4C-1.0.exe 的库安装文件就 ok 了，这个是准备工作。然后我们点击安装，安装好后出现如下文件夹：

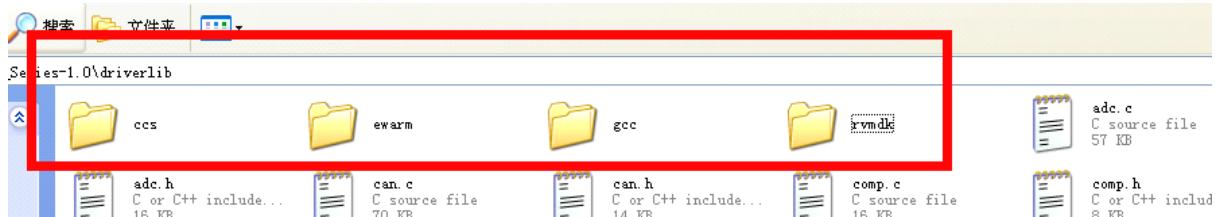


TI 的一下库函数比较复杂，因为 TI 考虑的比较多，所以大家看见如此多的文件夹会感到比较头疼，那么我们就稍微介绍下，剥丝抽茧，找到我们需要用的：

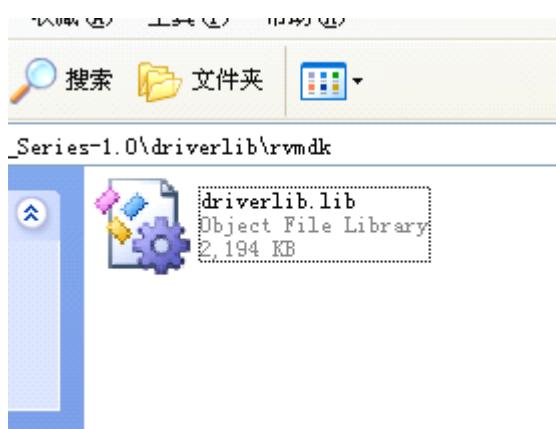
**Boot\_loader** 文件夹：这个很清楚，通过它的命名就知道里面应该是关于 Boot 的相关代码，关于 BOOT 我们后面再讲。

**Docs** 文件夹: 命名为文件夹, 肯定是里面放的都是相关文档了, 关于 TI 大部分 TM4C 的文档都在这里面了, 包括一些开发板的使用说明呀, 当然没有中文文档哦, 要看中文文档还得看我写的了。

**Driverlib** 文件夹: 外设驱动文件夹, 这里面放的就是我们要使用的外设的基本驱动库函数了, 这个也是我们学习的主要内容了, 这个后面我们会常常使用的。打开这个文件夹如下出现 4 个子文件:



我们点开其中的一个 **rvmdk**:



这个命名为 **dmk** 的子文件夹里放的就是一个封装好的 **.lib** 文件, 这个文件我们就可以在 **MDK keil** 中直接调用的了, 它封装了所有外设的驱动, 当然你也可以把外设的子驱动一一调用。其他的几个子文件夹正好对应了不同的编译器, 这里提供 4 中可以使用的编译器库函数封装。

**Grlib** 文件夹: 这个和上面的 **driverlib** 一样, 也是库函数, 但是给的是 TI 自己的图形库函数, 图形库函数的介绍在我们的产品光盘里有几篇详细的介绍与使用文章, 这个如果用的好的话, 可以在 **TFT** 上开发出华丽的人机用户界面, 这个是后话了。

**Sensorlib** 文件夹: 这个是 TI 新增加的关于传感器的库函数。

**IQmath** 文件夹: 很早 TI 就加入的数学库, 在 CORTEX M4 加入了 **dsp** 浮点运算单元后应该用途更加的大了。关于他的使用我们提供一个官方演示实例, 这个需要有合适的应用场合。

**Inc** 文件夹: 一些公共功能的头文件命名, 这个会在之后的应用中加以介绍。

**Example** 文件夹: ti 给的几个简单的实例。

**third\_party** 文件夹: 第三方的文件系统, 比如我们 sd 卡使用的 FATFS 文件系统, 或者一些操作系统。

那么整个 TI 给出的 TM4C 文件系统库函数文件包就介绍完毕了, 下面我们就开始来建立一个工程项目。

### 3.2 TM4C 工程项目的创立:

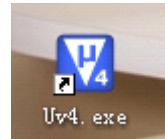
首先, 我们在刚才的文件安装包内建立自己的工程文件夹, 这个随便怎么命名了, 我先建一个名字为: qf-hanker 如下图所示



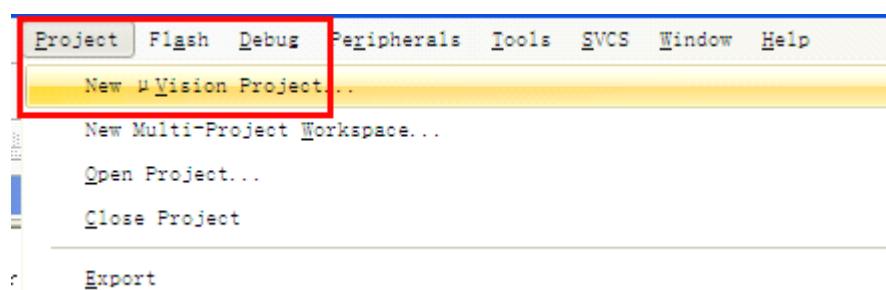
然后打开文件夹 qf-hanker 建立一个演示文件夹, 并且在里面建立一个 main.c 的文档:



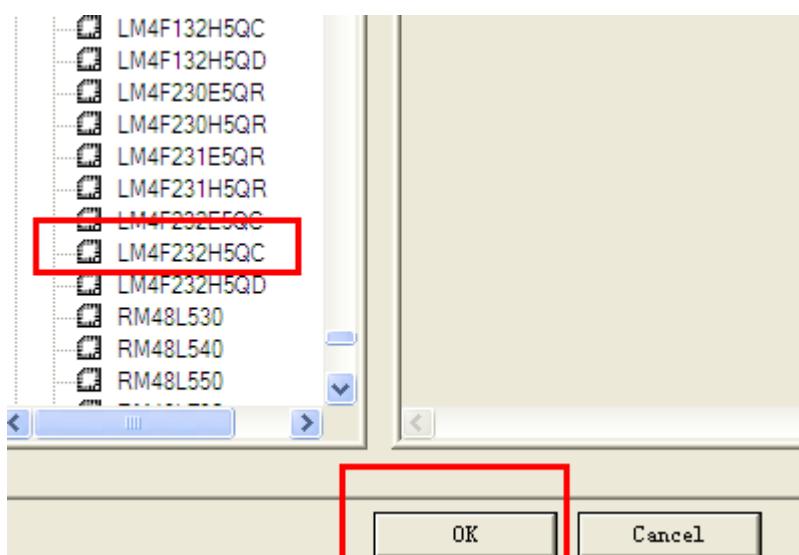
下面准备开发环境, 打开安装好的 MDK 软件:



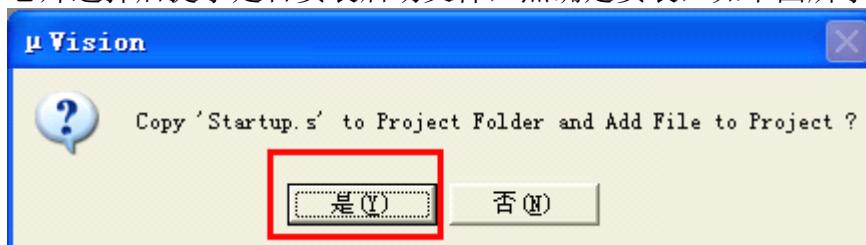
点击 project 上, 选择 new uvision project 如下图所示, 新建一个工程项目, 项目放置到刚才我们兴建的样例文件夹:



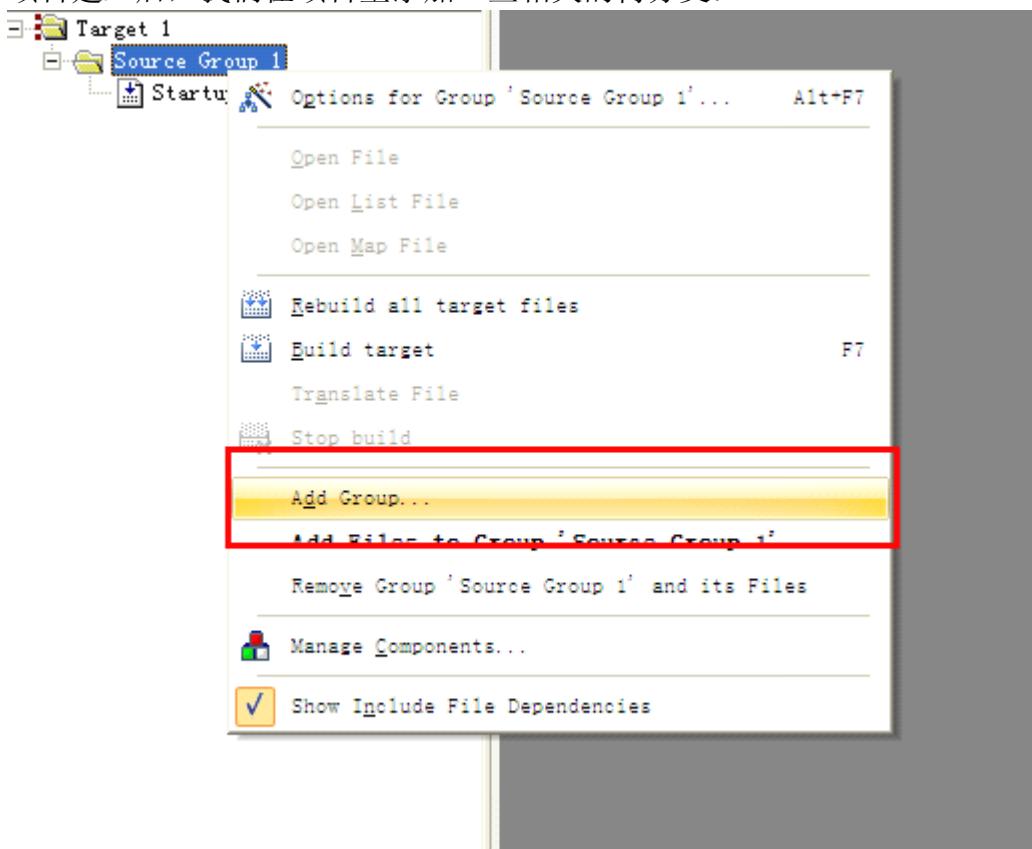
把工程命名为 example, 确定后选择芯片类型, 由于 MDK 最新的 4.7 版本命名还没有改过来, 大家还是用原来的 lm4f232h5qc:



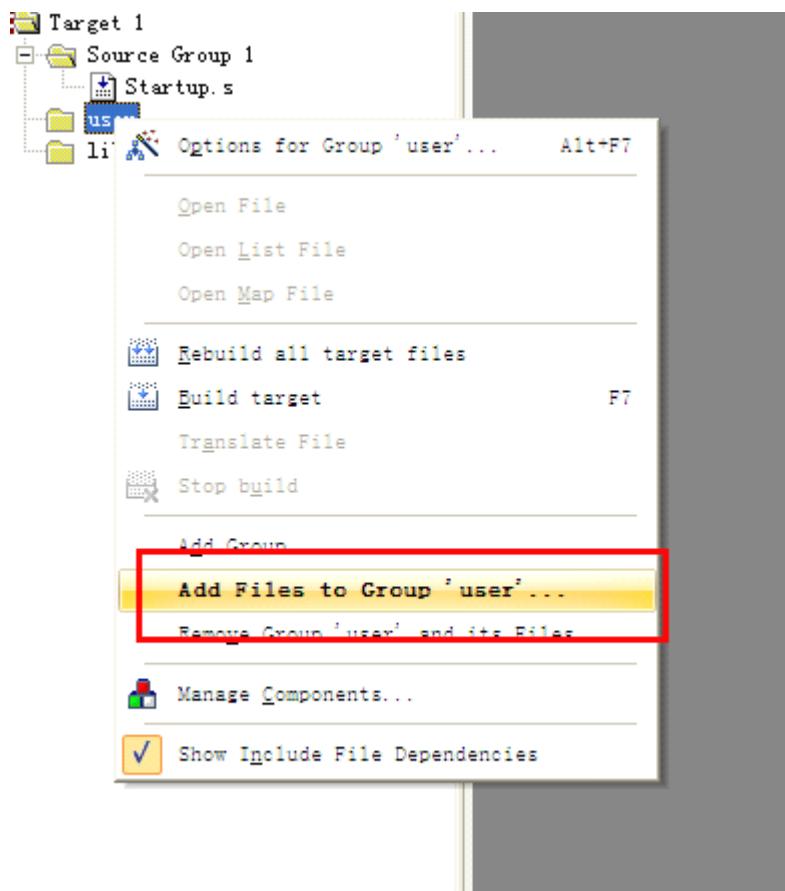
芯片选择后提示是否安装启动文件，点确定安装，如下图所示：



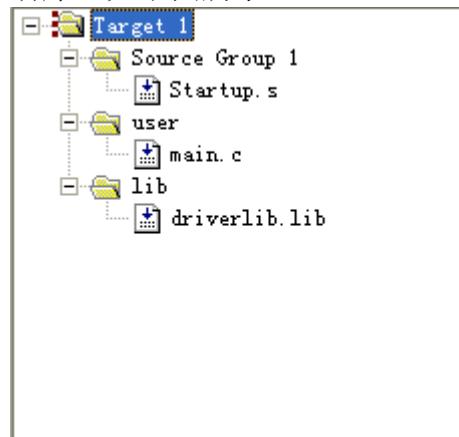
项目建立后，我们在项目里添加一些相关的树分支：



建立一个 user 组，一个工程组：



然后向组里面添加我们的文件了，在 user 里添加刚才的主函数文件，在 lib 里添加驱动库，如下图所示：



Ok，工程建立好了，下面我们来写一个模板，打开 main 函数文件，添加如下代码：

```
//-----  
  
// 文件名    : 演示实例  
// 作者      : 青风  
// ARM 内核  : Cortex-M4  
// 使用芯片  : LM4F232H5QD/tm4c123gh6pz  
// 开发环境  : KEIL  
// 版本记录  : 2013-6-10 14:30
```

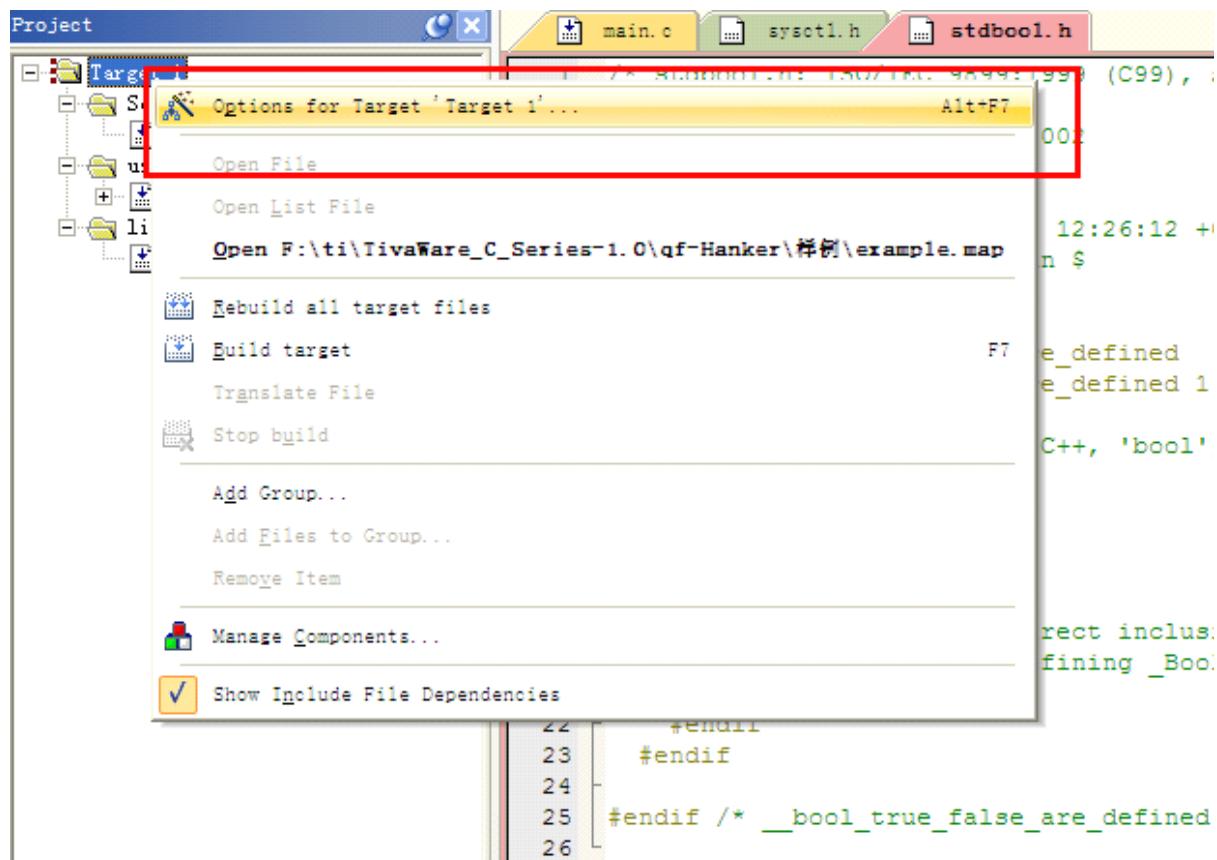
//函数功能说明:

//-----

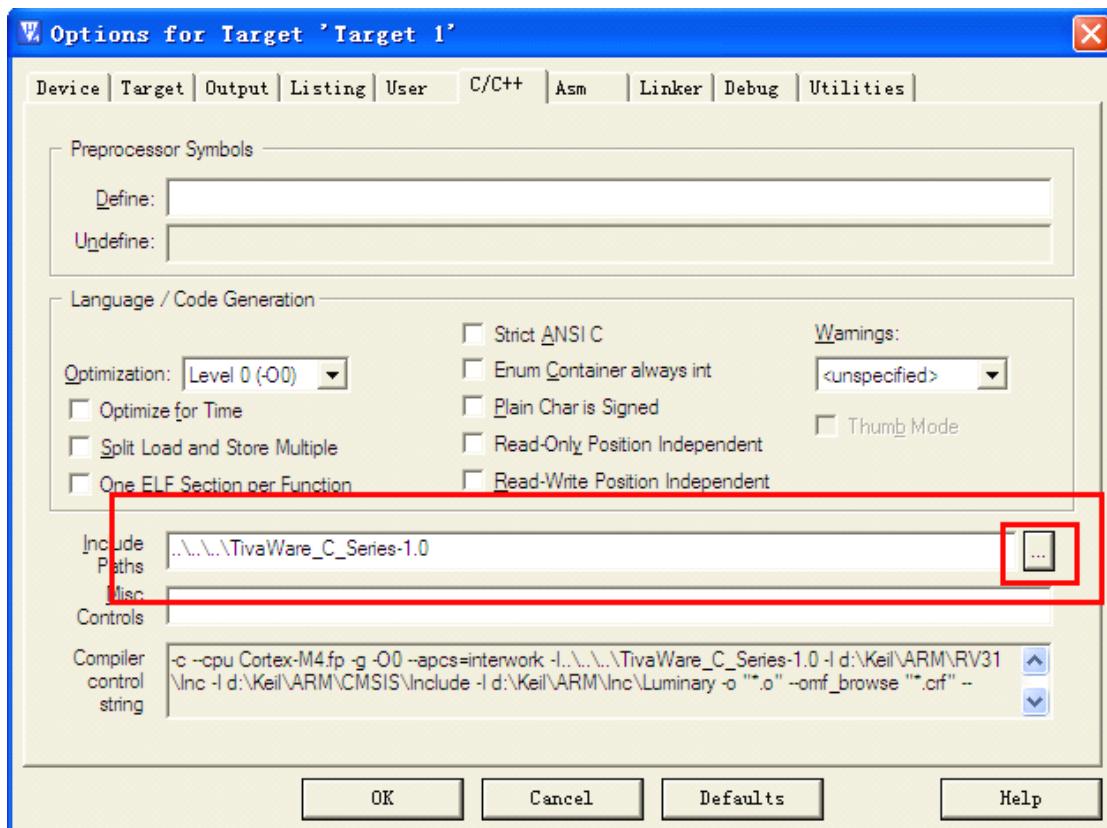
```
#include "inc/tm4c123gh6pz.h"
#include <stdint.h>
#include <stdbool.h>
#include "inc/hw_types.h"      //一些位带操作的宏定义
#include "inc/hw_memmap.h"     //外设基地址的定义
#include "driverlib/sysctl.h"   //系统时钟
#include "driverlib/gpio.h"     //gpio 定义
```

```
int main(void)
{
    /*写入任何你想写入的代码*/
}
```

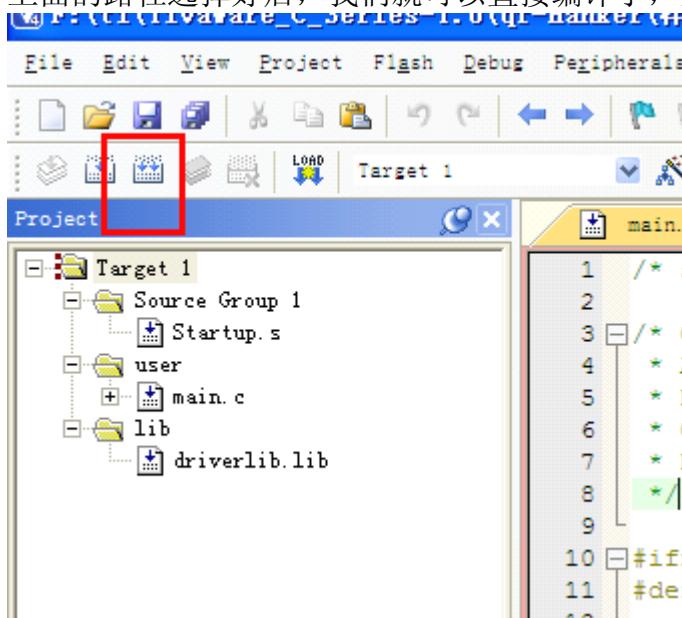
这个例子什么都没写，当然头文件大家根据外设来选择，后面在讲历程的时候要详细说明。模板里添加了几个基本头文件，如上面的例子所示，大家可以按照上面的写法来写。当然这些头文件需要给路径，我们点击工程项目点击右键，如下图所示：



在 C/C++ 内选择我们链接的文件的路径：



上面的路径选择好后，我们就可以直接编译了，点击整体编译：



编译成功后：

```

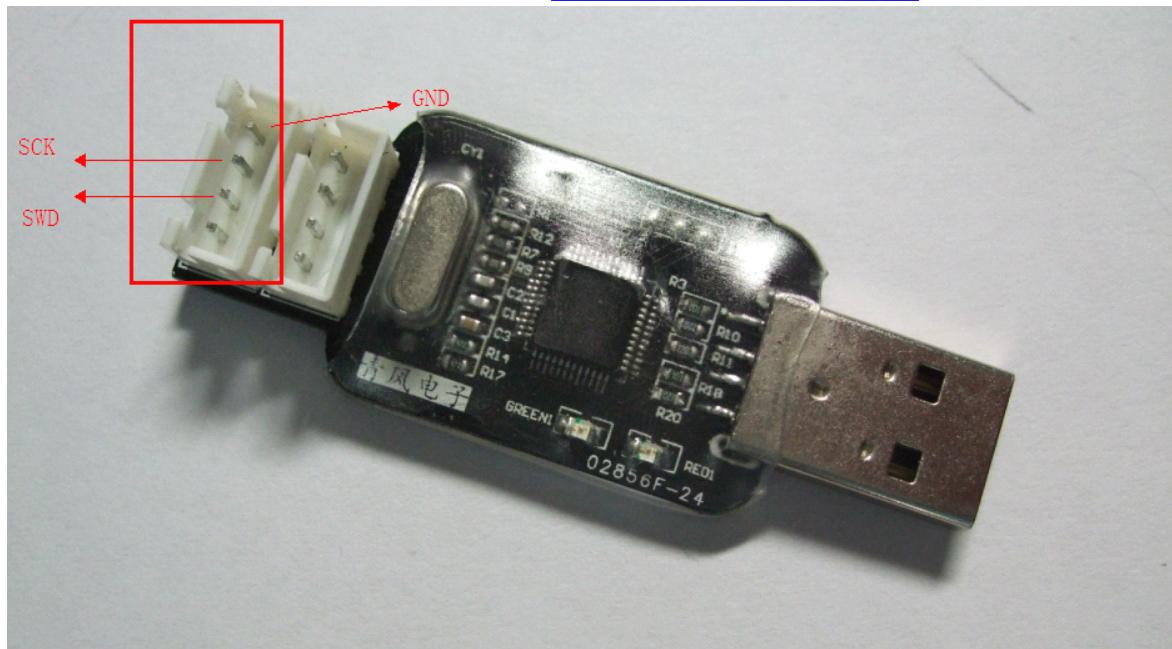
Rebuild target 'Target 1'
assembling Startup.s...
compiling main.c...
linking...
Program Size: Code=928 RO-data=16 RW-data=0 ZI-data=352
"example.axf" - 0 Error(s), 0 Warning(s).

```

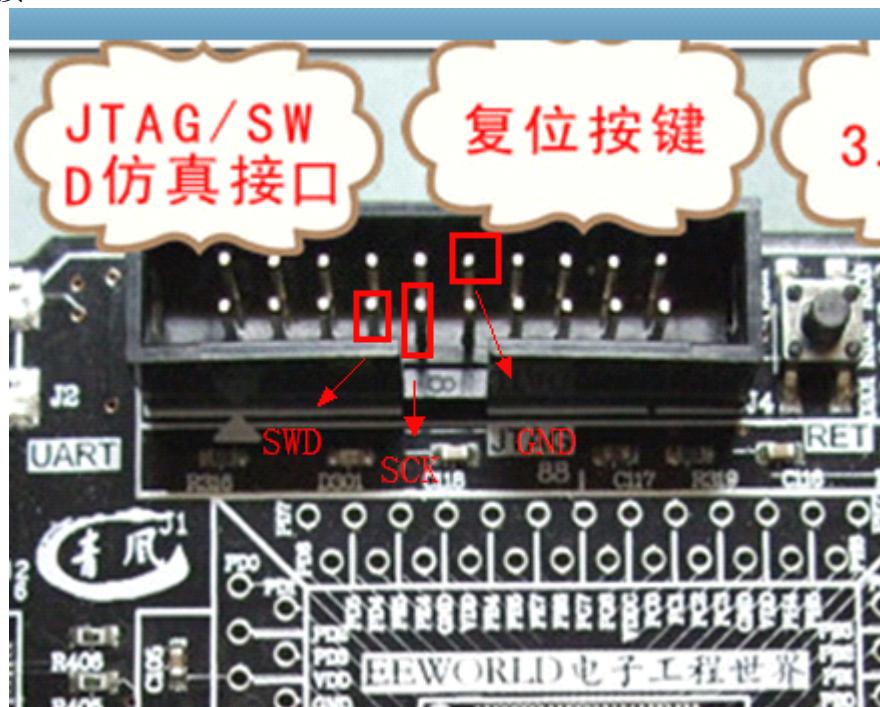
### 3.3 工程项目的下载与仿真:

这里面只谈使用仿真器进行仿真与下载，关于使用串口下载会在以后专门写一篇文章讲解下：

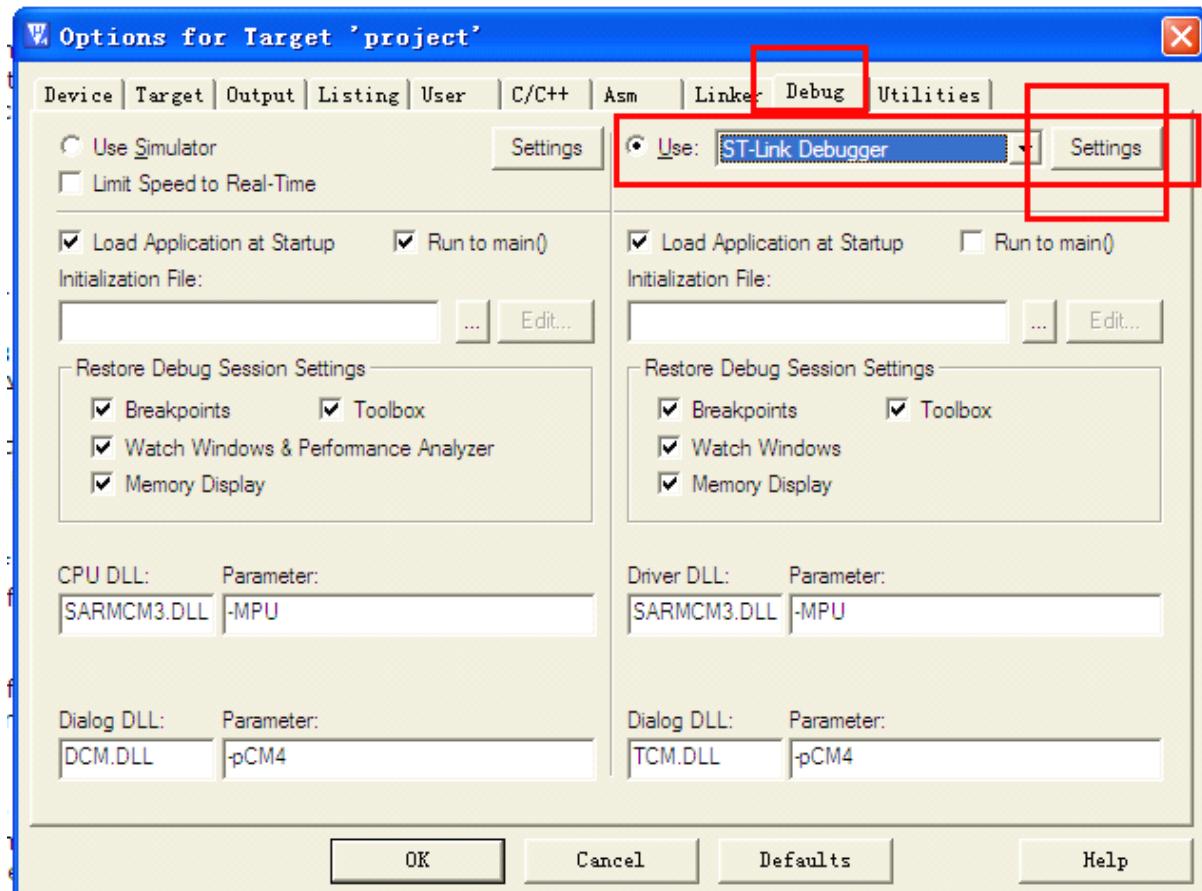
硬件关于仿真器：仿真器可以选择 JLINK, ULINK. 这里我选择廉价的开源仿真器 STLINK V2，价格非常低，支持的器件非常的多，不仅仅支持 arm，而且支持 stm8 器件，所以这里比较推荐使用，购买地址：<http://qfv5.taobao.com/> 如下图所示：



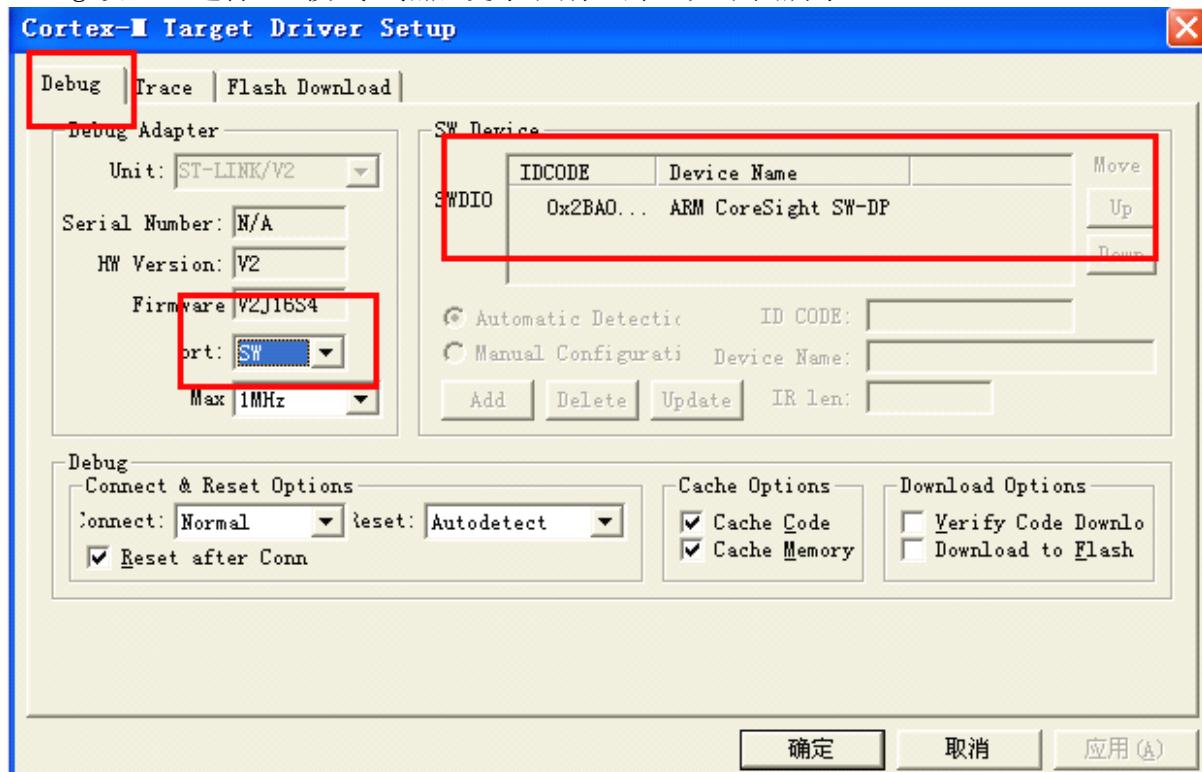
和 hanker M4 开发板只需要连接 3 根线，分别为 GND, SCK, SWD. 对比上下两附图进行连接：



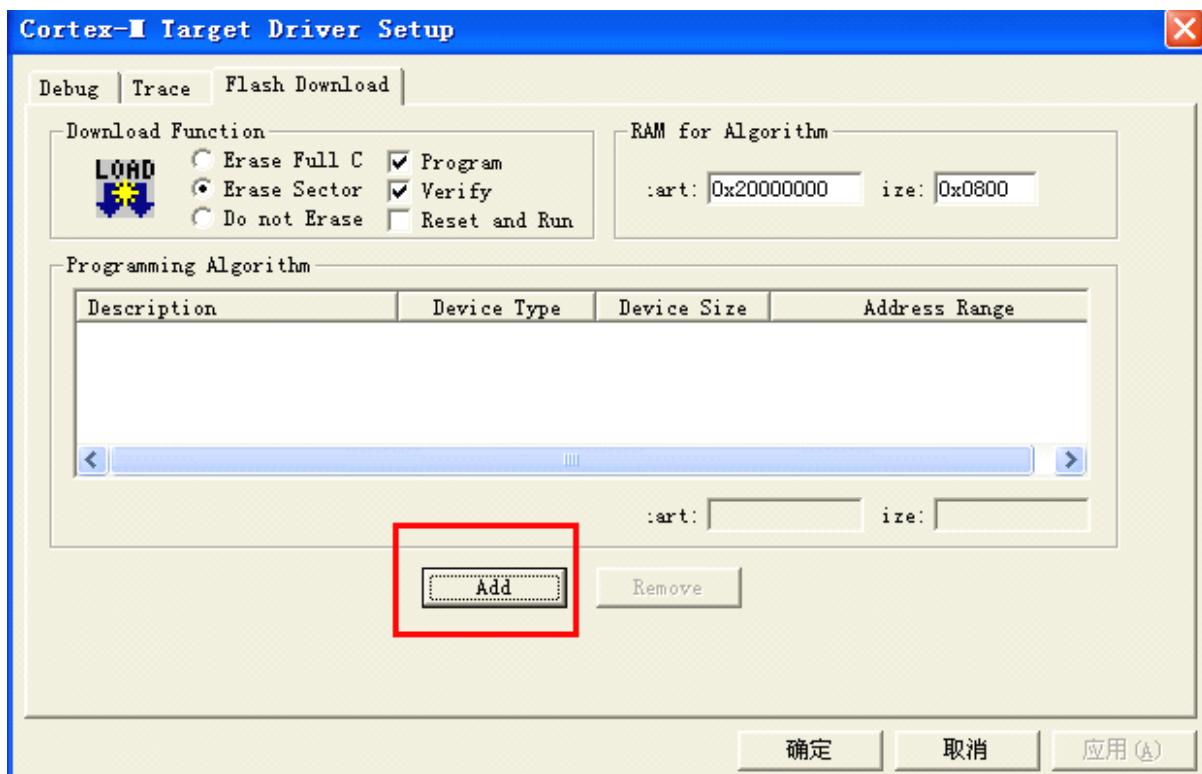
打开项目设置，设置仿真和下载项目，首先是仿真设置，Bebug 选择 ST-LINK，并且点击 settings，如下图所示：



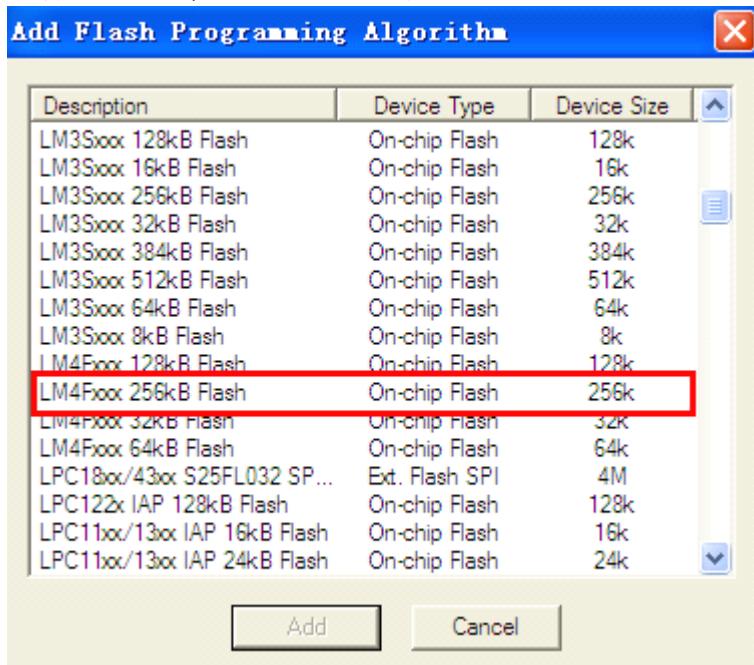
Debug 设置，选择 SW 模式，然后提示识别芯片，如下图所示：



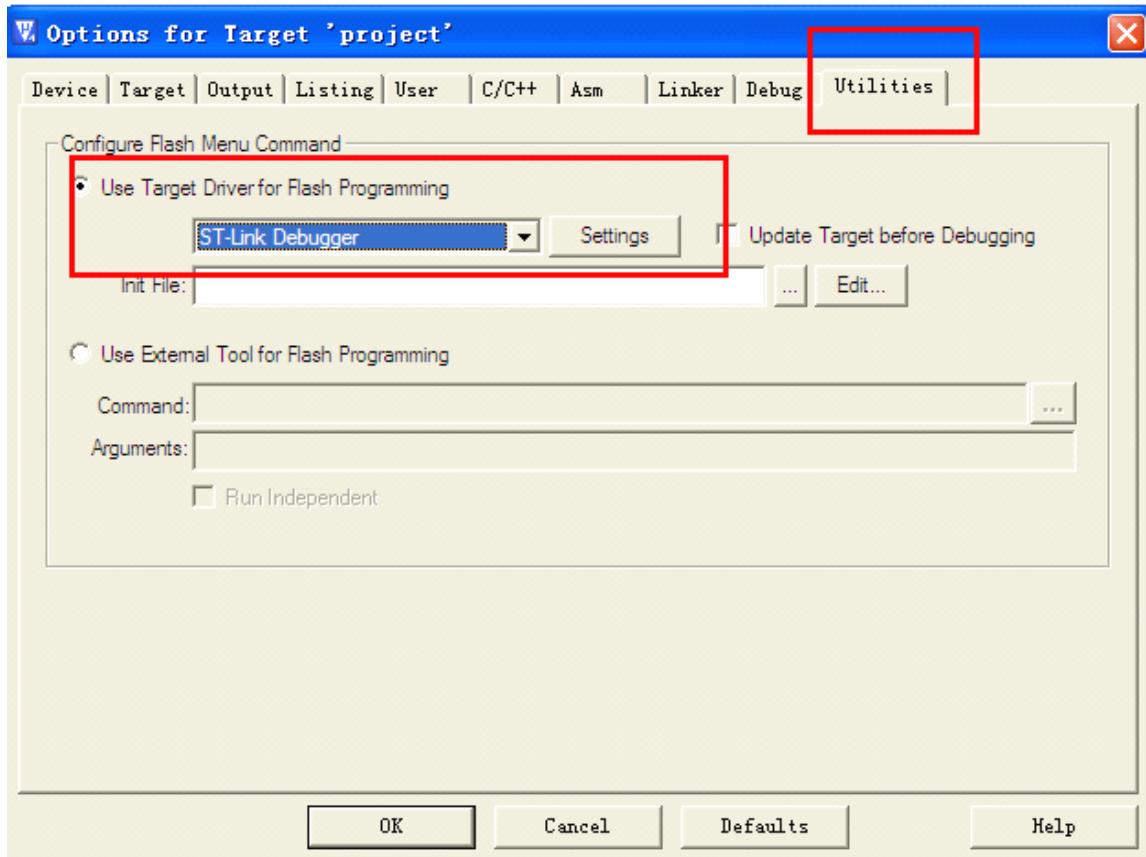
在 FLASH DOWNLOAD, 选择 FLASH 大小：



选择 LM4Fxxx, 256k 的 FLASH 大小:



最后设置 Utilites 选项，设置 stlink 选项:



上面所有的设置设置好了，大家就可以进行仿真与下载了。

这里面注意一点，不管使用 JLINK 还是 stlink v2，部分寄存器内部都无法观看，如果需要观察寄存器内部值的朋友，需要购买 TI 专门为 M4 设计的专业 ICDI 仿真器，购买地址：<http://item.taobao.com/item.htm?spm=a1z10.3.w17-18016875483.10.tJ4gaY&id=24237988393&>

如下图所示：

