

编译 BLHeli_S 的正确方式

源码简介

我就随便讲讲我所知道的 BLHeli 历史吧。然后有源码上的问题可以问，我尽量回答。我 BLheli 是欧洲一位神人 S 君写的最早是在华科尔的 10A ESC 上实现的。其主控是 Atmel8 位的芯片。代码实现参考了德国另外一位高手的无刷 C 代码。但是因为 BLHeli 用 Atmel 芯片且用汇编写的，程序是完全独立的。

那个时候 Atmega8 芯片是锁住的，刷程序很麻烦，去除保护位需要用到一个高电压编程，是个并口，要接 10 多条线才能把那个保护位去掉，而且需要比较高级的编程板子。S 君都是直接换个新的上去。曾经我还做过一个转接板，但是也相当麻烦。那时候基于 Atmel 的 Blheli 基本上没有什么人玩，要焊芯片下来需要热风枪。事情发生转变是在 S 君把 Blheli 移植到 silabs 的芯片上去，一般是 C8051FX 系列，比如 C8051F330。当时的电调是 xp7a 和 xp3a 这些。推动这个的另外一个爆发点事地平线的 mcpX 小直升机，当时无福翼的小飞机第一次做这么小，非常火。但是马力欠缺，很多人 DIY 改无刷。silabs 的电调不要焊下来可以直接刷 Blheli，一下子被大家追捧起来。

而本文所说的 Blheli_S 是 S 君又把 Blheli 程序移植到 silabs 公司的另一款 MCU 上，EFF8XX 系列（一款 8 位的 MCU，行业熟称小蜜蜂，因其 LOGO 是个小蜜蜂）。并且驱动电机采用硬件双路互补 PWM，这种同步整流电调，加减速非常快，市场反应很好。

总结一下：

	MCU	版本定义方式	PWM 产生方式	油门信号支持
Blheli	Silabs 公司的 C8051 系列和 Atmel8	根据各电调厂商名字定义	定时器 IO 模拟	普通油门 OneShot125 OneShot42 Multishot 自动识别
Blheli_S	Silabs 公司的 EFM8BB10X(主频 24M) EFM8BB21X(主频 48M)	根据引脚定义命名，具体后续会讲	硬件互补 PWM 可插入死区控制	普通油门 OneShot125 OneShot42 Multishot 自动识别

BLHeli_S 相关下载地址

源码下载地址: <https://github.com/bitdump/BLHeli>
上位机下载地址: <https://www.mediafire.com/folder/dx6kfaasyo24l/BLHeliSuite>

源码结构分析:

目录结构如所示, 可以看出只有 Atmel/BLHeli_S SiLabs/SiLabs 三个文件夹有用, 举例说明: 当你想用 EFM8 系列的 MCU 作为主控, 那么你就可以直接拷贝 BLHeli_S SiLabs 文件夹加入到你的工程, 其他的都可以删除, 同理也是一样。本教程以 BLHeli_S SiLabs 文件夹为例说明, 因为这个是最新源码, 并且长久更新。

名称	修改日期	类型	大小
Atmel			
BLHeli_S SiLabs			
SiLabs			
Vendor specific versions			
.gitattributes			
.gitignore			
COPYING			
README			

该目录下存放 BLHeli 支持 Atmel 的源码

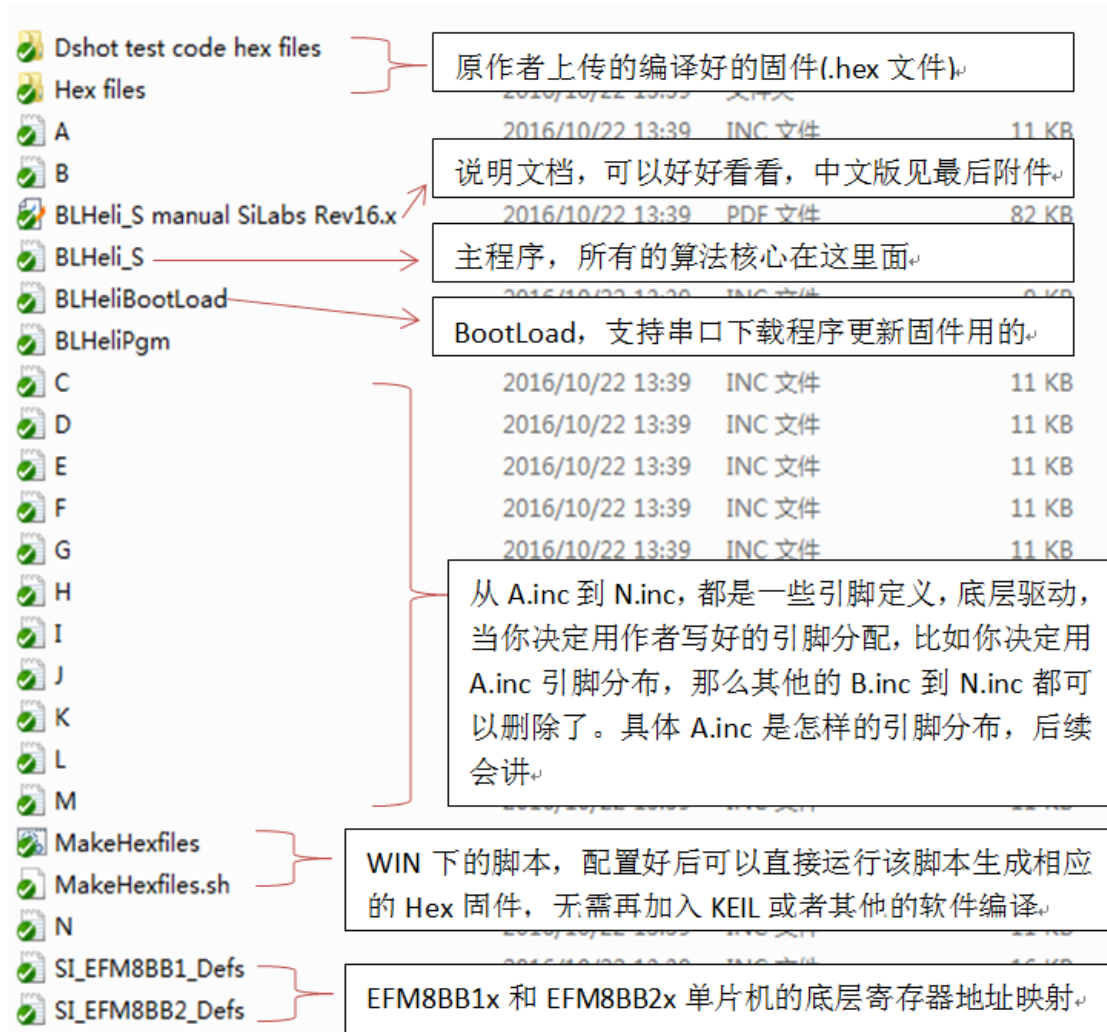
该目录下存放 BLHeli 支持 EFM8 的源码, 也就是 BLHeli_S

该目录下存放 BLHeli 支持 C8051 的源码

这些不用管, 这是一些 github 的文件

介绍文档

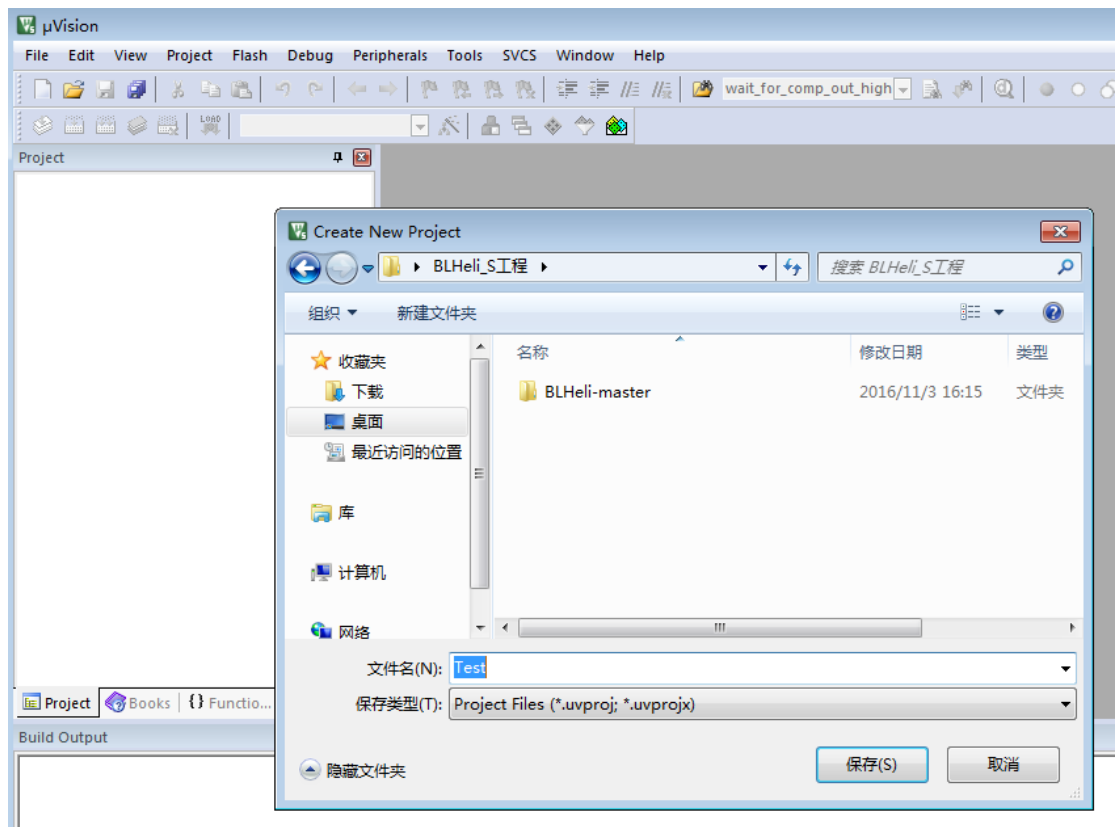
打开 BLHeli_S SiLabs 文件夹, 说明如所示。



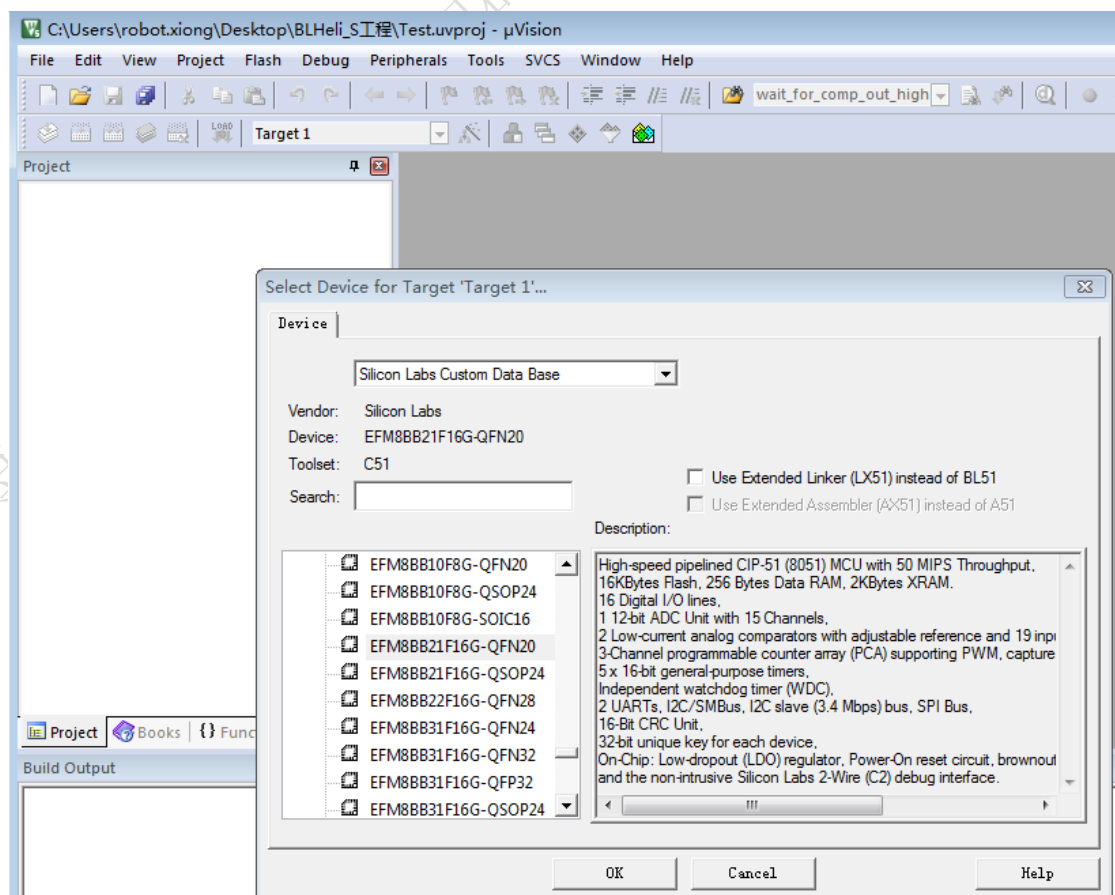
加入 Keil 编译

首先建个文件夹，改个名字(随意改，我的是 BLHeli_S 工程)，然后拷贝刚下载的源码工程到 BLHeli_S 工程文件夹下，如所示。

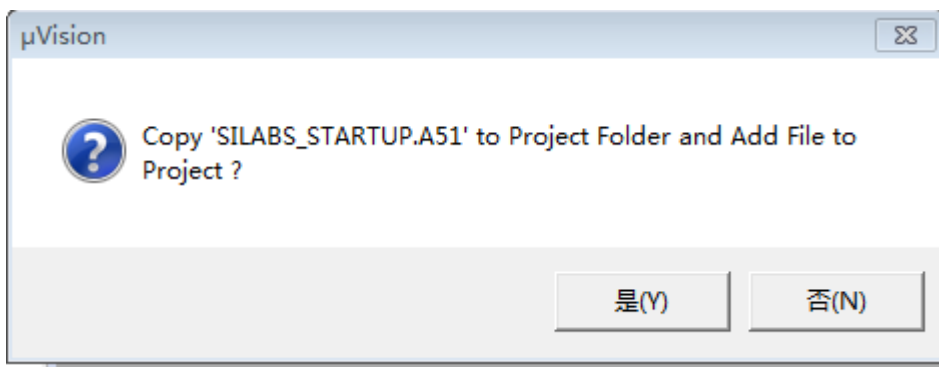
1.Keil 新建工程，弹出对话框，如所示；



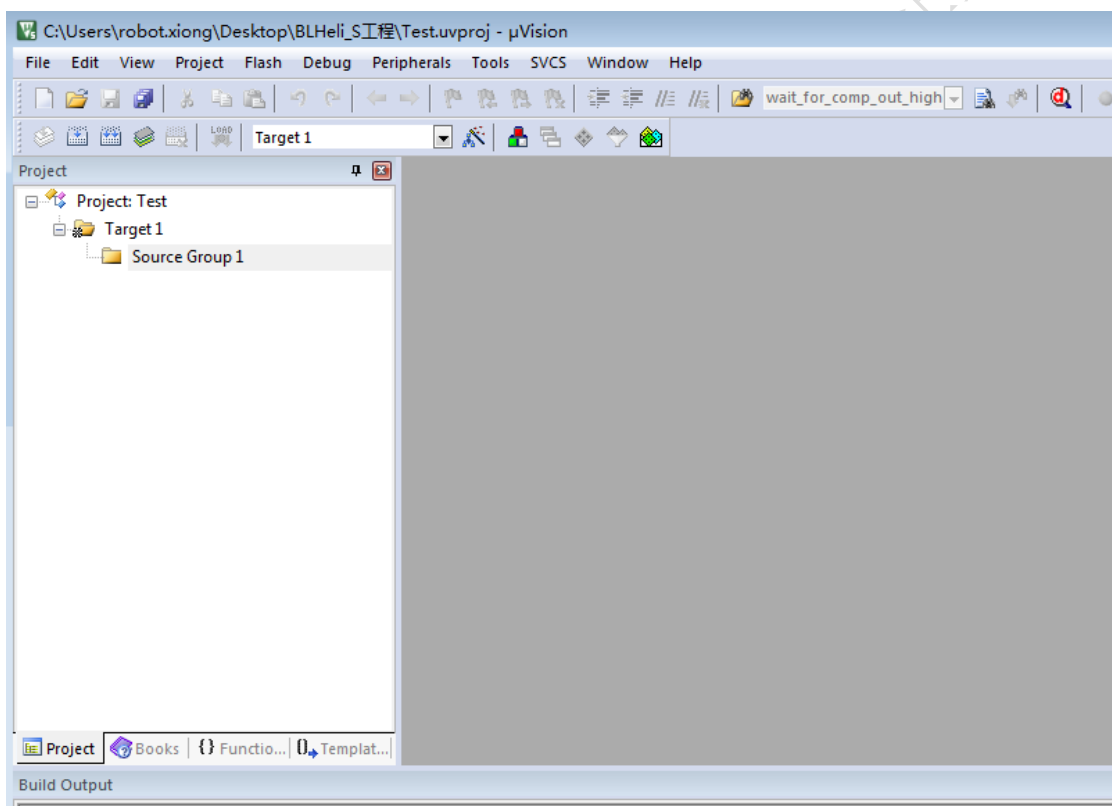
2.选择芯片为 EFM8BB21F16G-QFN20，点击 OK，如所示；



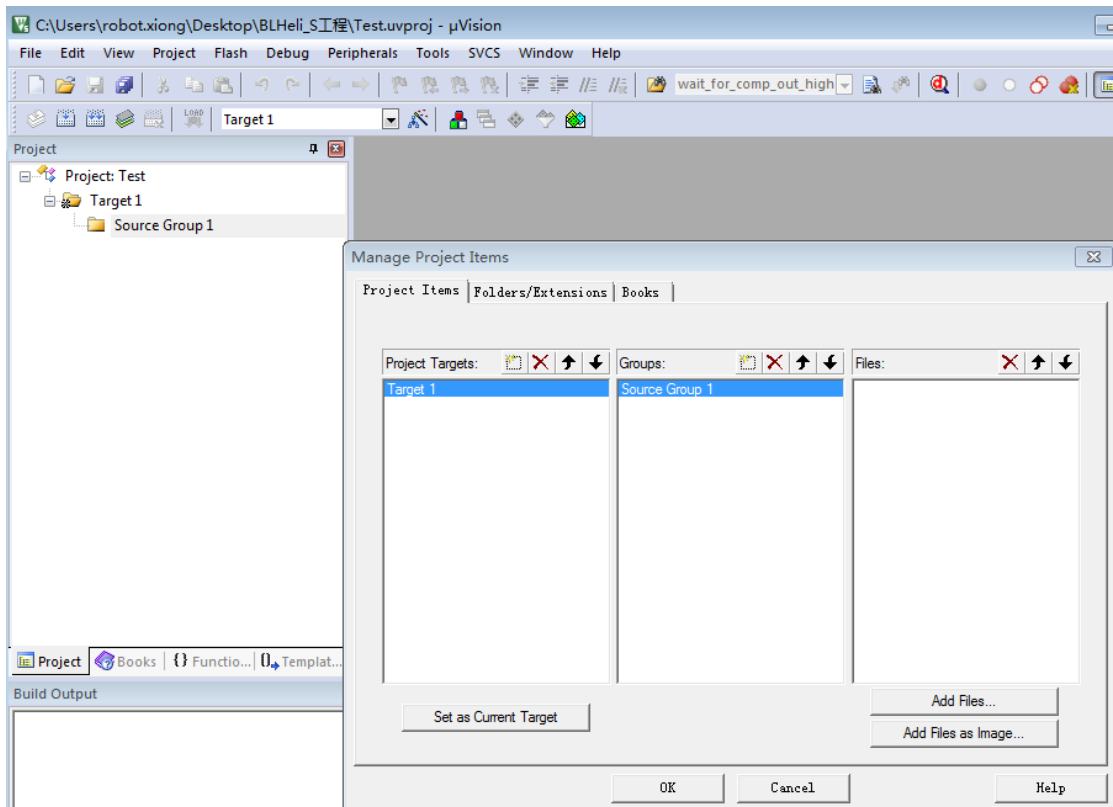
3.弹出对话框如图所示，然后点击否；



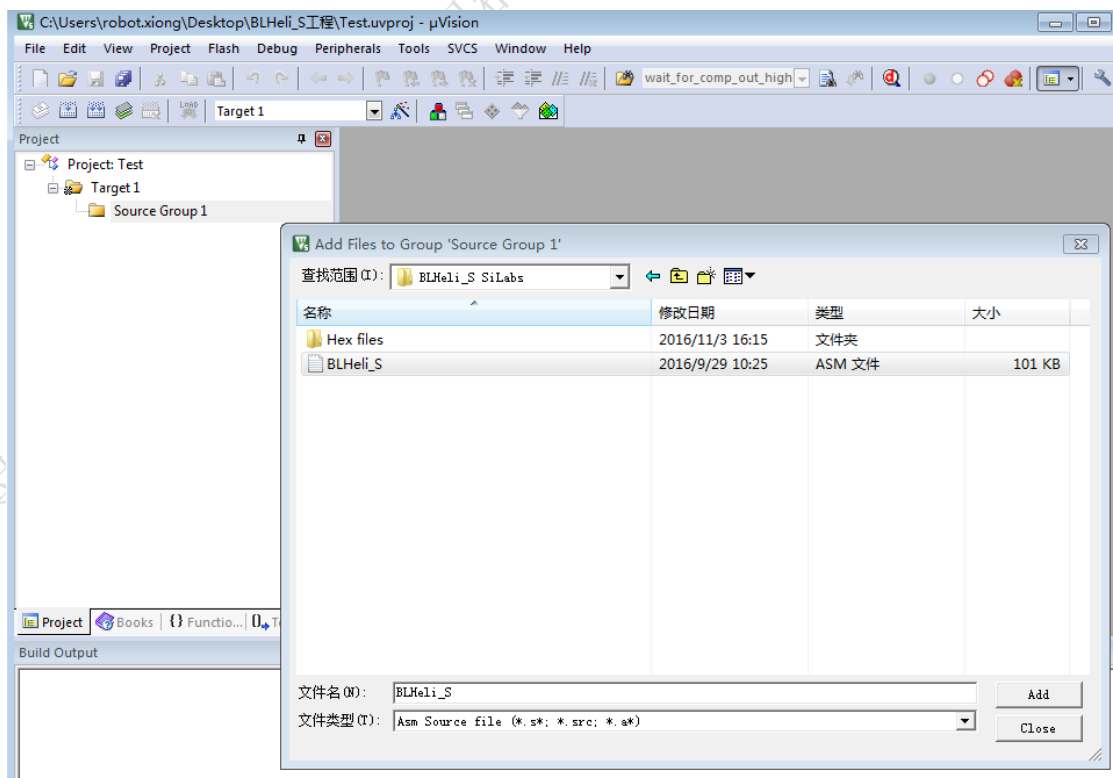
4.最后可以看到 KEIL 空工程如图所示。



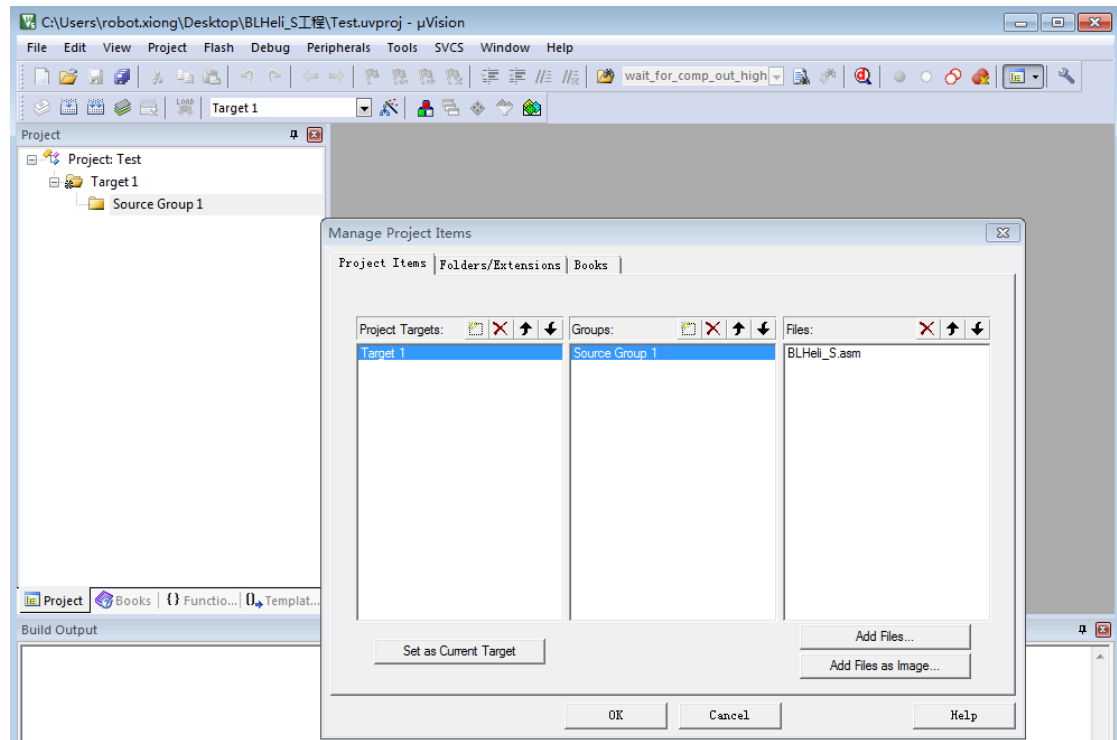
5.向工程中添加 BLHeli_S 的源码文件，点击 Add Files ，找到刚才的源码目录



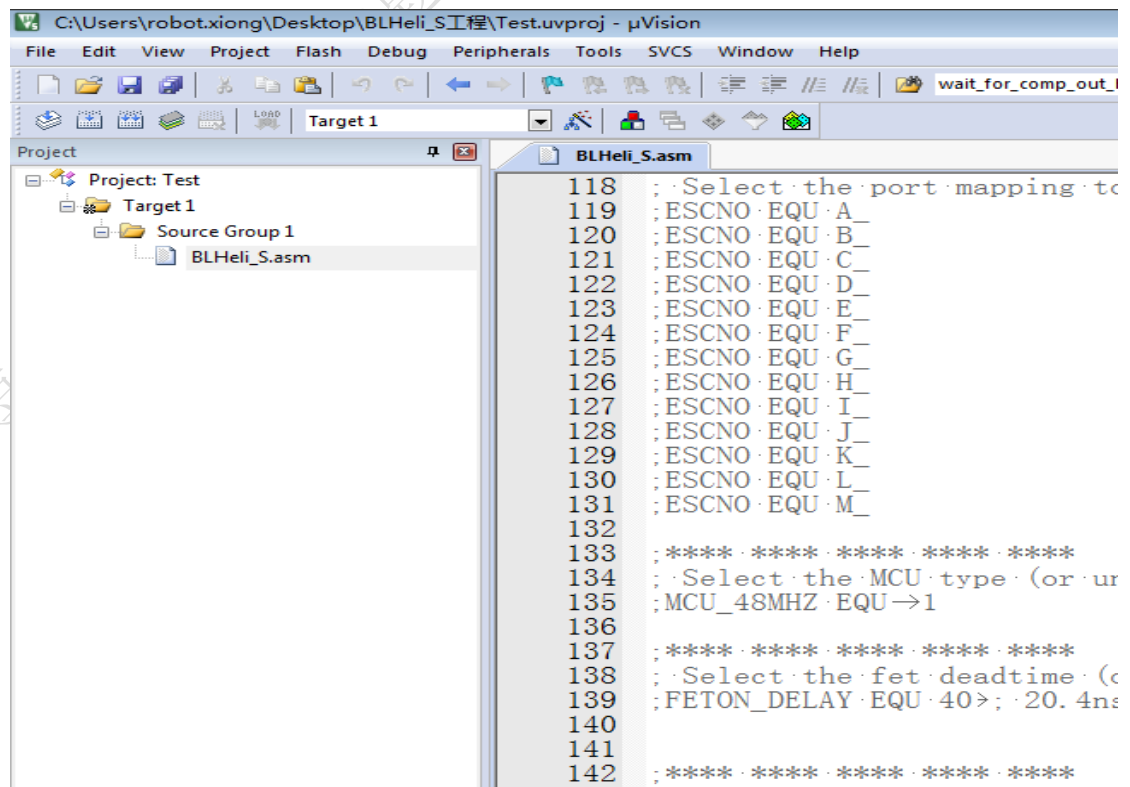
6. 点击 OK 后弹出如图所示的对话框，然后把文件类型选择如图所示，点击 BLHeli_S 点击一下 Add，然后关掉该对话框



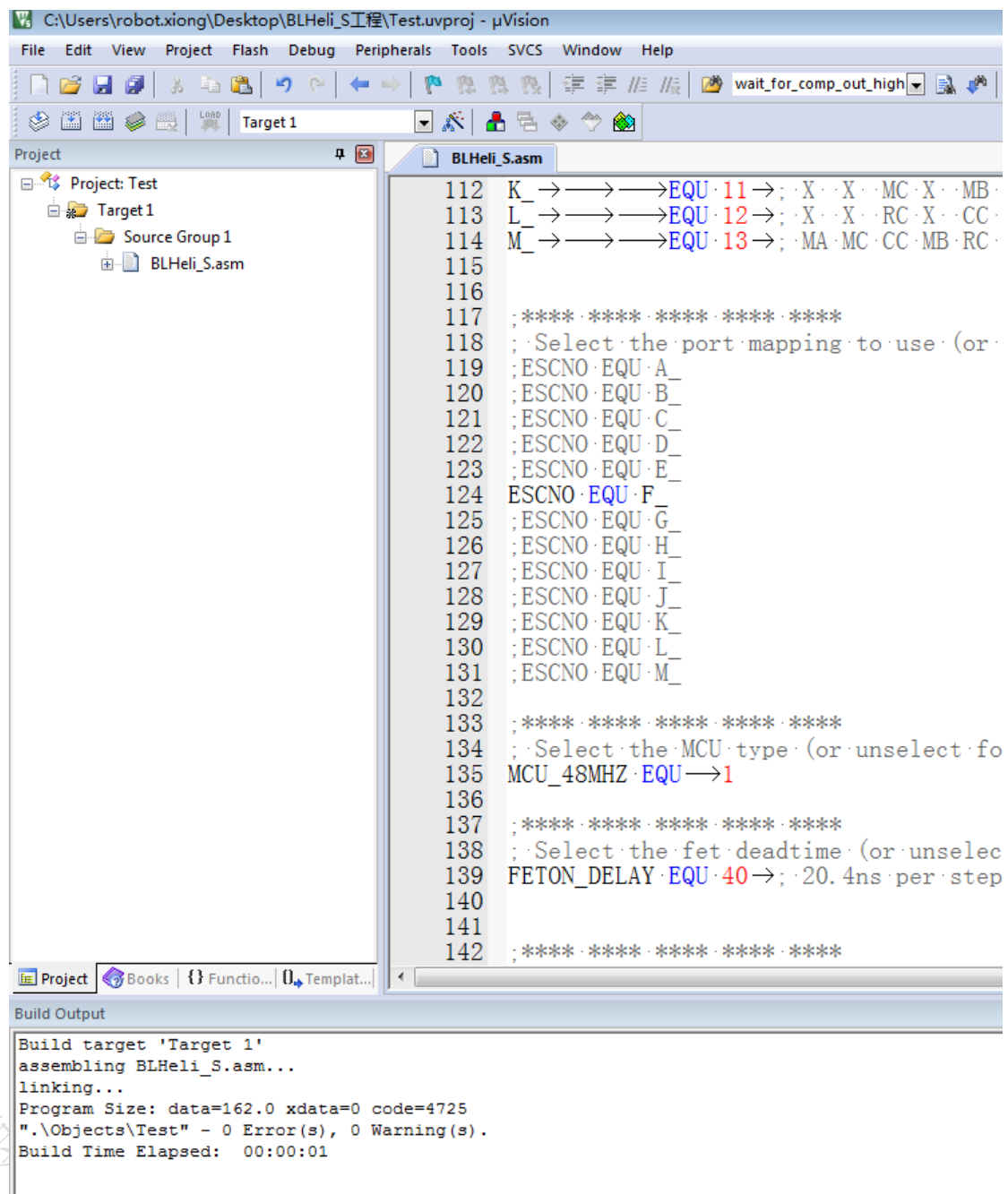
7. 点击 OK,



8. 可以看到如所示的 KEIL 工程。



9.然后打开刚加入的 BLHeli.asm 文件，找到 124 行，135 行，139 行，把前面的分号去掉，即去掉注释，然后点击编译，如所示。编译完成，就可以下载编译后的固件，以后大家应该知道怎么下载固件了吧，不知道的话群里面问。



引脚定义解释说明

打开 BLHeli_S.asm 文件可以看到如下图所示的定义

A	EQU 1	; X X RC X MC MB MA CC X X Cc Cp Bc Bp Ac Ap
B	EQU 2	; X X RC X MC MB MA CC X X Ap Ac Bp Bc Cp Cc
C	EQU 3	; Ac Ap MC MB MA CC X RC X X X Cc Cp Bc Bp
D	EQU 4	; X X RC X CC MA MC MB X X Cc Cp Bc Bp Ac Ap Com fets inverted
E	EQU 5	; L1 L0 RC X MC MB MA CC X L2 Cc Cp Bc Bp Ac Ap A with LEDs
F	EQU 6	; X X RC X MA MB MC CC X X Cc Cp Bc Bp Ac Ap
G	EQU 7	; X X RC X CC MA MC MB X X Cc Cp Bc Bp Ac Ap Like D, but noninverted com fets
H	EQU 8	; RC X X X MA MB CC MC X Ap Bp Cp X Ac Bc Cc
I	EQU 9	; X X RC X MC MB MA CC X X Ac Bc Cc Ap Bp Cp
J	EQU 10	; L2 L1 L0 RC CC MB MC MA X X Cc Bc Ac Cp Bp Ap LEDs
K	EQU 11	; X X MC X MB CC MA RC X X Ap Bp Cp Cc Bc Ac Com fets inverted
L	EQU 12	; X X RC X CC MA MB MC X X Ac Bc Cc Ap Bp Cp
M	EQU 13	; MA MC CC MB RC L0 X X X Cc Bc Ac Cp Bp Ap X Inverted LED

该图就是 BLHeli_S 目前支持的所有引脚映射。以 F_为例说明：

F_ EQU 6 ;X X RC X MA MB MC CC X X Cc Cp Bc Bp Ac Ap

第一部分：X X RC X MA MB MC CC：该部分是 P0 口定义，从底到高依次如下：

CC：P0.0 比较器公共端

MC：P0.1 C 相加入比较器输入端

MB：P0.2 B 相加入比较器输入端

MA：P0.3 A 相加入比较器输入端

X：P0.4 X 代表没有用，空着，以下都是如此

RC：P0.5 油门输入端

X：P0.6 X 代表没有用，空着，以下都是如此

X：P0.7 X 代表没有用，空着，以下都是如此

第二部分：X X Cc Cp Bc Bp Ac Ap：该部分是 P1 口定义，从底到高依次如下：

Ap：P1.0 A 相 PWM 输入端

Ac：P1.1 A 相 IO 口输入端

Bp：P1.2 A 相 PWM 输入端

Bc：P1.3 A 相 IO 口输入端

Cp：P1.4 A 相 PWM 输入端

Cc：P1.5 A 相 IO 口输入端

X：P1.6 X 代表没有用，空着，以下都是如此

X：P1.7 X 代表没有用，空着，以下都是如此

最后附上 BLHeli_S 的 KEIL 工程，见连接：链接：<http://pan.baidu.com/s/1pLIX6H1> 密码：问群主，一定给