
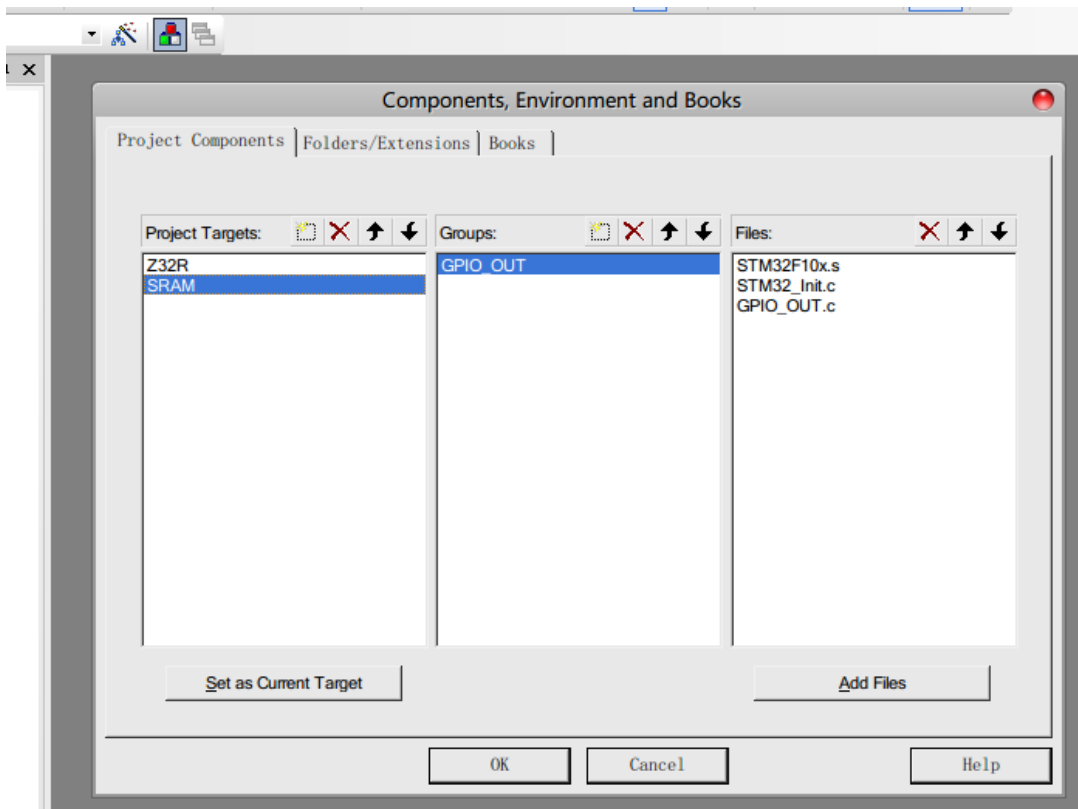


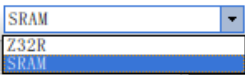
## 图解 Keil MDK 3.5/UV4beta Jlink V7 在 STM32 中 SRAM 调试成功方法成功

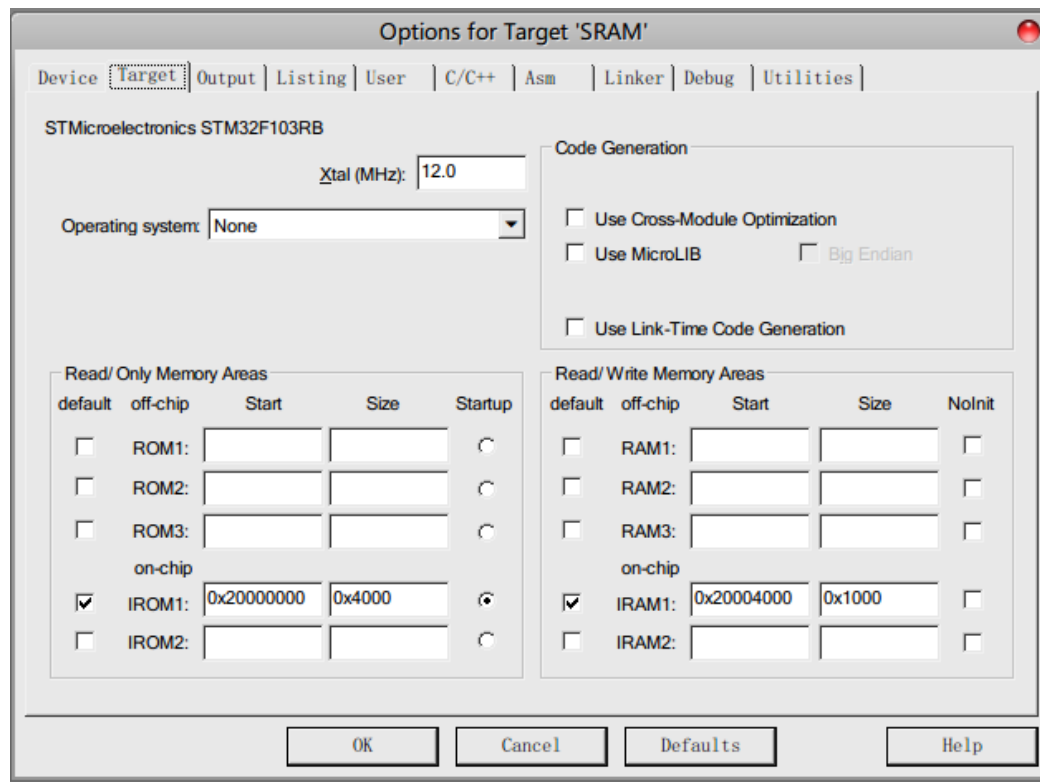
目的是避免 FLASH 擦写次数的限制，和避免新手学习很快让 MCU 结束生命。以本人 STM32F103RBT6 为例，128K flash，20K SRAM,借鉴前人 ULINK 下调试的方法。

把内存分为 16K 用做 FLASH，4K 用做内存使用，这里以 GPIO 点亮 LED 为例子，项目为 GPIO\_OUT。

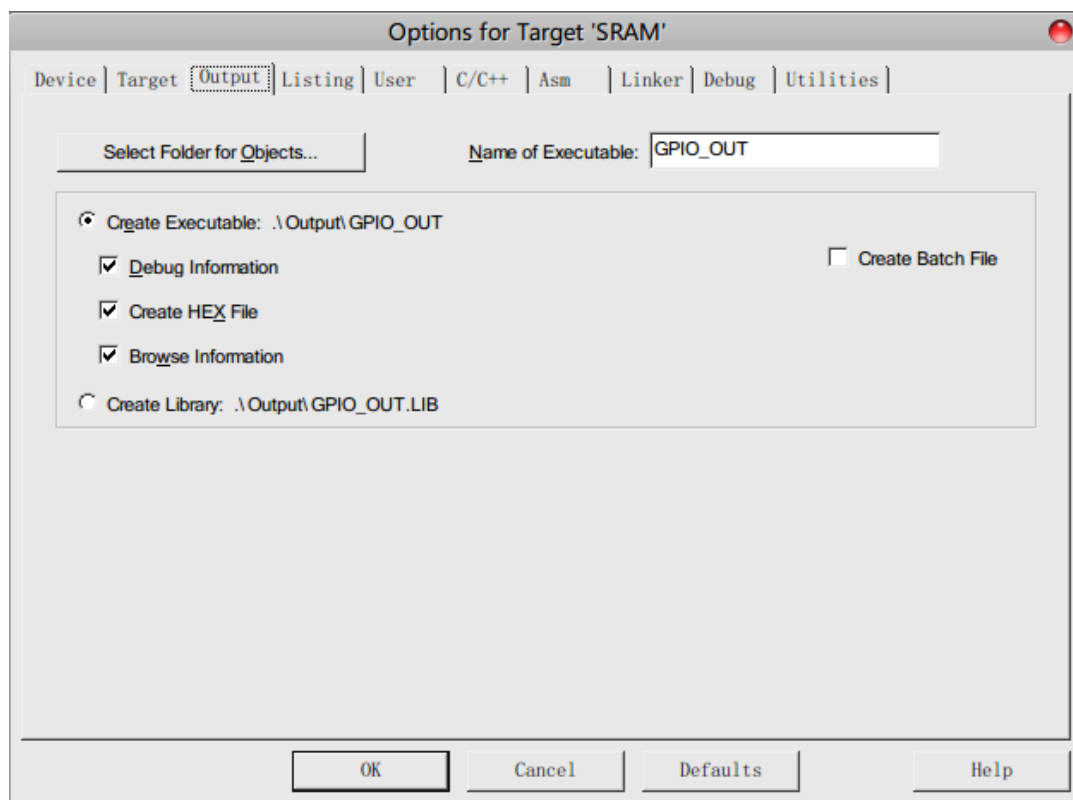
- 1、copy Keil 安装目录下，例如 C:\Keil\ARM\Boards\Keil\MCBSTM32\STLIB\_Blinky 中的 RAM.ini 到项目目录，然后双击 GPIO\_OUT.uvproj，启动 Keil。
- 2、单击  启动下图对话框，在 Project Targets 一项单击 New(Insert)按钮，增加一个目标，命名为 SRAM，然后 OK。



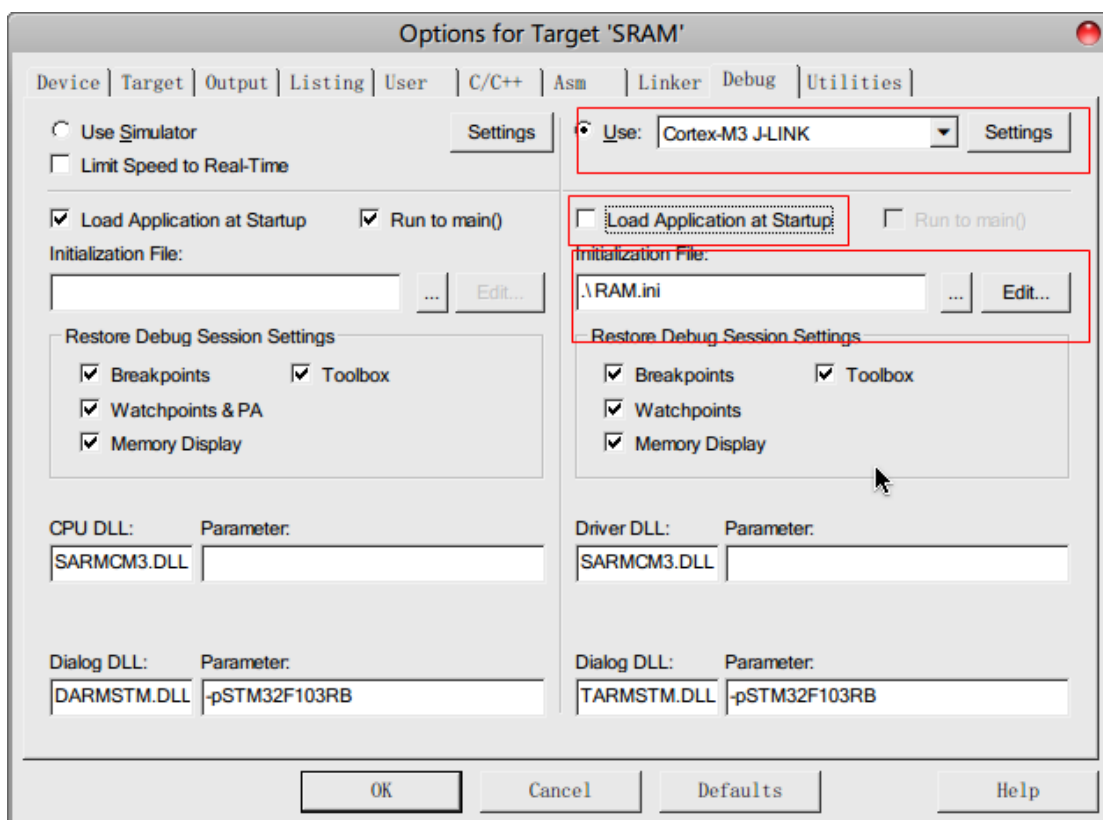
- 3、然后再 Select Target  中选择 SRAM。
- 4、左侧项目栏中选择 SRAM，单击右键选择 Options for Target “SRAM”，打开选项对话框。



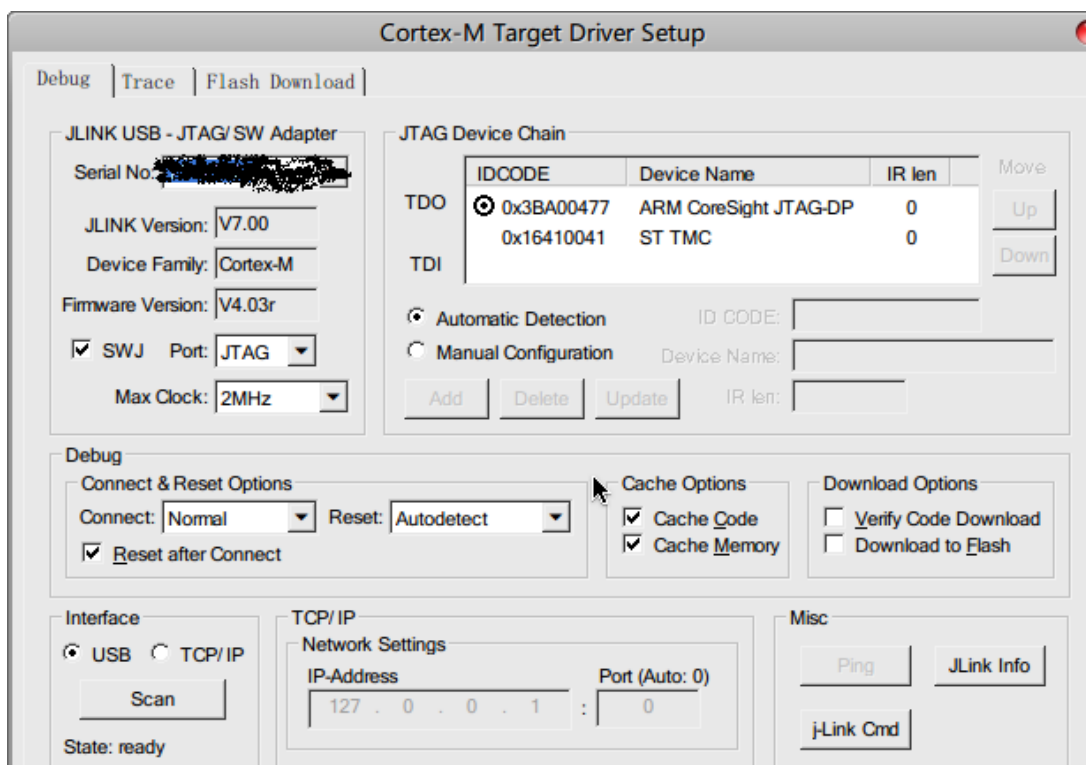
- 5、如上图所示，在 target 选项中修改 IROM1 的 Start 值为：0x20000000，Size：0x4000  
IRAM1 的 Start 值为：0x20004000，Size：0x1000；  
即内存分为 16K 用做 FLASH，4K 用做内存使用。
- 6、 OutPut 选项卡中也选中 Create HEX File，如下图所示。



- 7、在 Debug 选项中仿真器选择 Cortex-M3 J-LINK,Initialization File 中点击 Edit 按钮选择刚才 copy 过来的 RAM.ini 文件，取消 Load Application at Startup 复选框，设置如下：

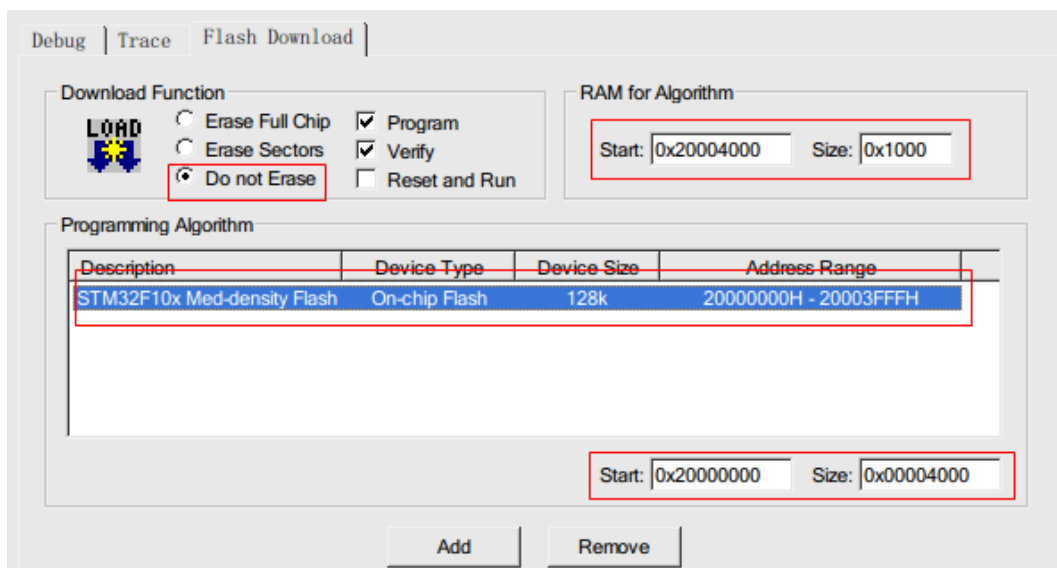


并点击仿真器一栏，Settings 按钮，进一步设置,如下图，注意 Debug 选项中 Interface 中选择 USB，

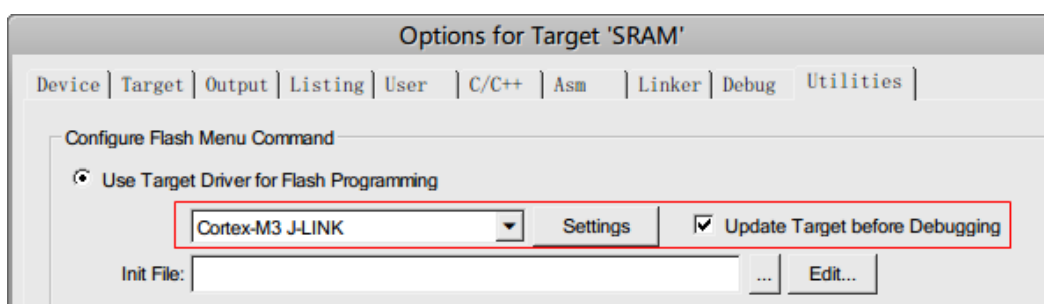


如下图所示,在 Flash Download 中配置，选中 Do not Erase，在 RAM for Algorithm 中 Start

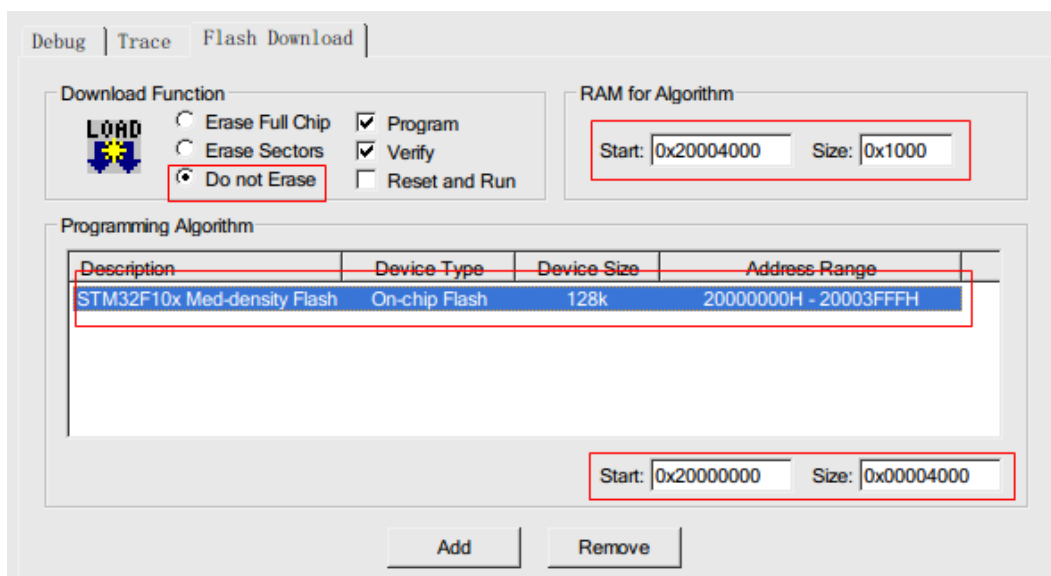
设置为 0x20004000,Size 设置为 0x1000,设置 STM32F10x Med-density Flash 的地址 Start: 0x20000000, size: 0x00004000, 点击 OK。注意：如果在 Programming Algorithm 中没有 STM32F10x Med-density Flash, 就单击 Add 按钮添加, 然后修改即可。



- 8、如下图所示,在选项设置对话框中 Utilities 中, Use Target Driver for Flash Programming 中选择 Cortex-M3 J-LINK, 并点击 setting 进一步设置,



可以看到设置同第 7 步中一样, 不再更改。



- 9、到此为止设置完成, 开始调试。

- 10、 开始调试，重建项目（重新编译）。
- 11、 然后启动 **Start/Stop Debug Session**（或 **Crtl+F5**），开始调试，进入调试模式后，单击 **Run** 按钮或 **F5** 运行。
- 12、 注意：再次过程中不要点击 **Flash** 菜单下面的下载。
- 13、 最后记住几个数字：

<b>0x20000000</b>	为内部 <b>SRAM</b> 的起始地址
<b>0x4000</b>	为 <b>16K RAM</b> 大小，模拟 <b>FLASH</b>
<b>0x20004000</b>	设定的 <b>4K</b> 大小 <b>RAM</b> 的起始地址，模拟 <b>RAM</b>
<b>0x1000</b>	设定的 <b>4K</b> 大小 <b>RAM</b>