使用 ADS 建立 S64 的应用

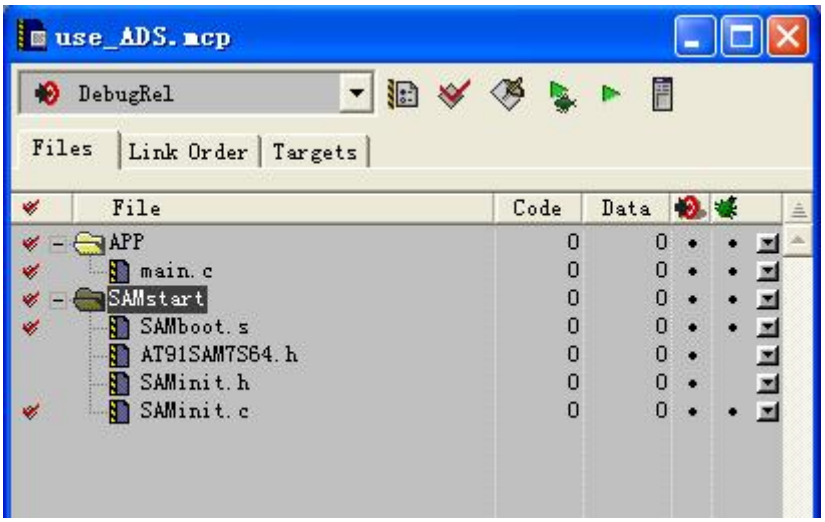
ADS 是 ARM 官方推出的 IDE，AT91SAM7S64（以下简称为 S64）为 ATMEL 推出的基于 ARM7TDMI 的 MCU，这篇文章旨在引导初学者使用 ADS 建立 S64 的应用，掌握相关设置，编译，调试与下载。

本文并不关注程序本身，重点在于开发的过程，以下言归正传。

一，建立工程

打开 ADS 的 IDE,选择 NEW，新建一个 project，类型为 ARM Executable Image，比如叫做 use_ADS，默认情况下 ADS 会为新工程建立一个以工程名命名的文件夹。

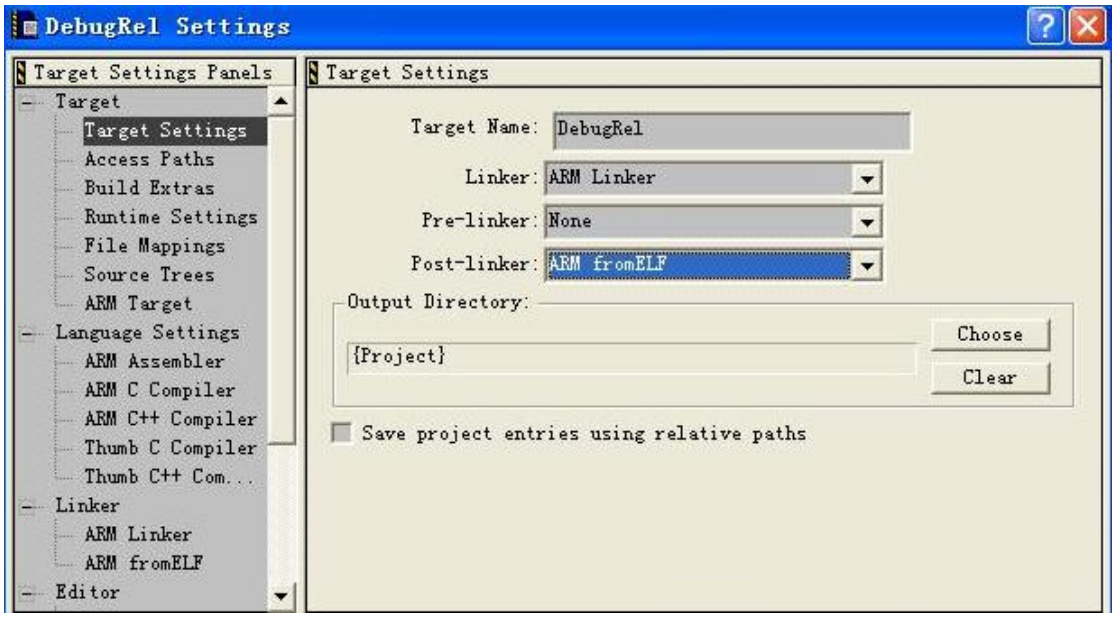
工程建立后，工程管理窗口出现在右侧，此时可以为工程建立文件组，并添加文件，如下图：



然后应对工程进行相关设置，具体步骤如下：

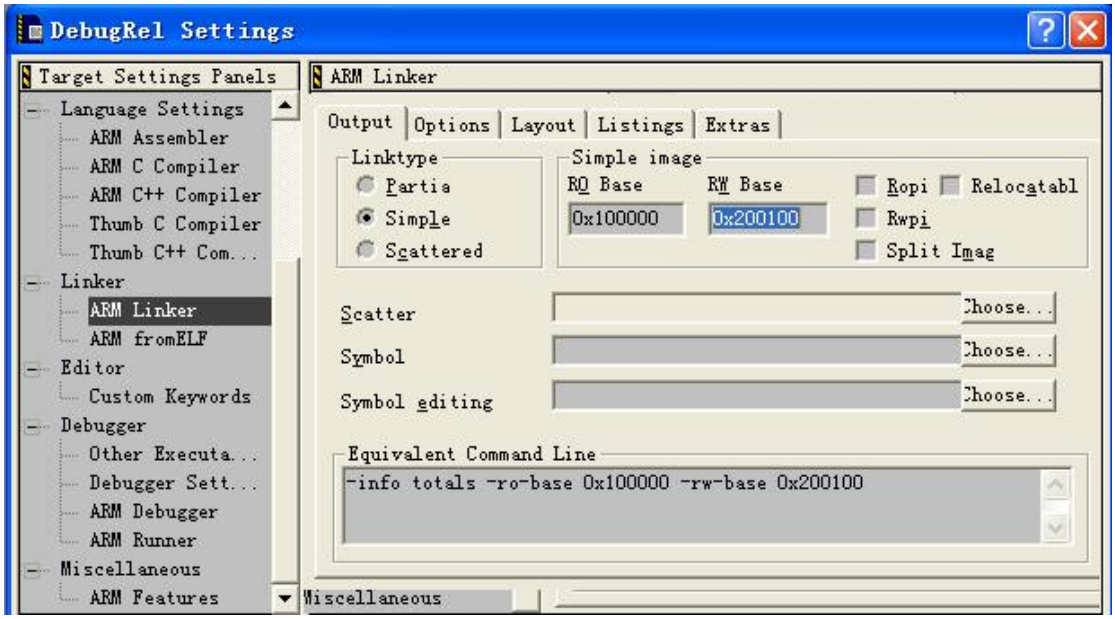
点击工程 settings 图标(上图第一排左起第一个图标,第三个为 make,第四个为 debug)出现工程设置的选项卡

1） target settings

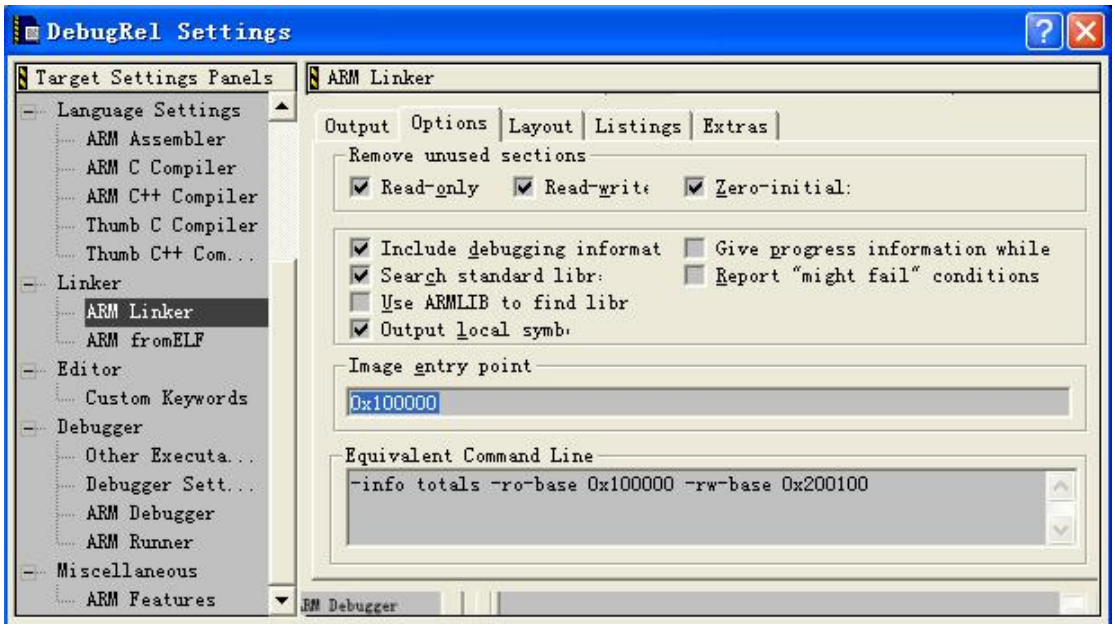


将 Post-linker 修改为 ARM fromELF 以得到相应的输出

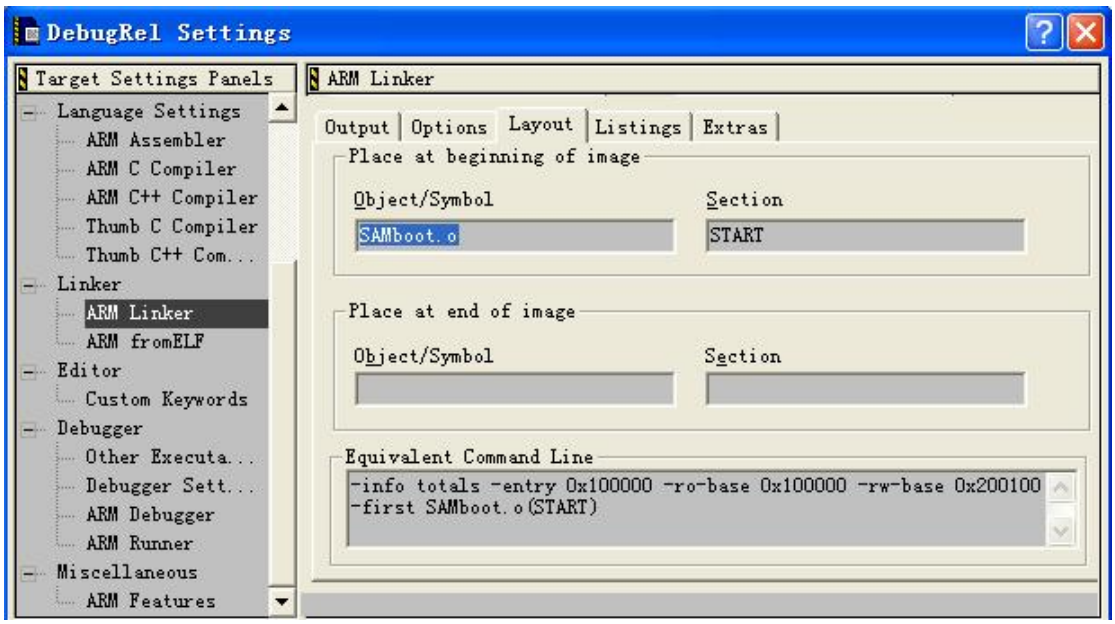
2）linker



在 output 选项卡中将 RO base 指向 flash 起始处，也就是程序开始的地方；RW base 指向 0x200100，之所以不从 RAM 其实地址开始，是为向量保留一些空间，这样在使用 remap 之前将向量从 flash 复制到 ram 中，加速异常处理速度。

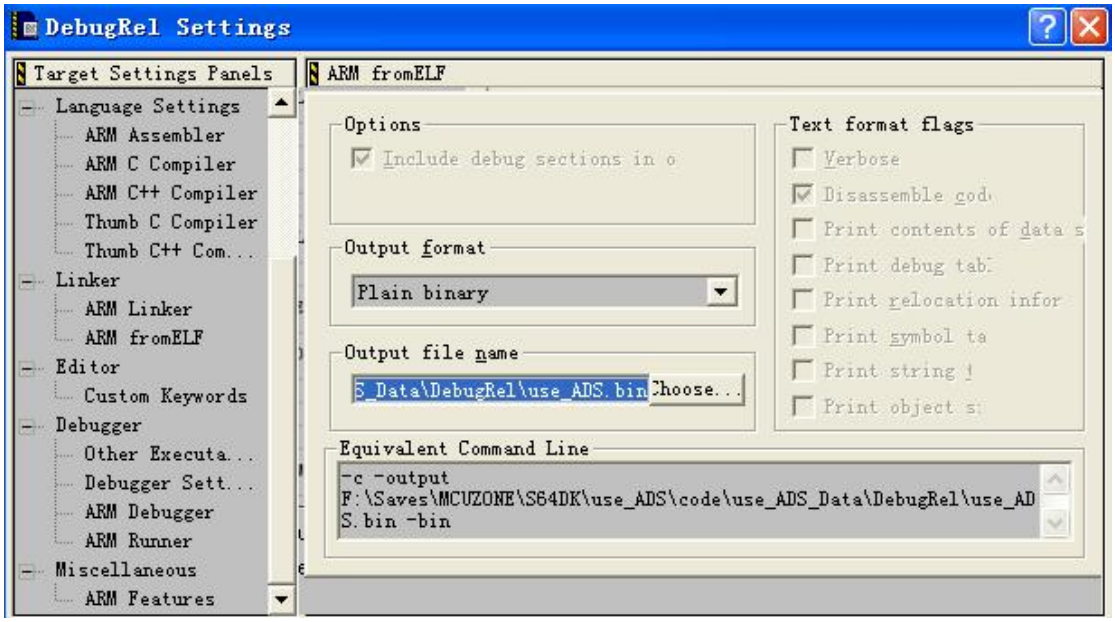


在 options 设置 Image entry point 为程序起始地址。



Layout 选项卡中配置各个段的 layout（“摆放位置”），重要的是将启动代码段放置到 Image（映像）的起始处。

3） ARM formELF

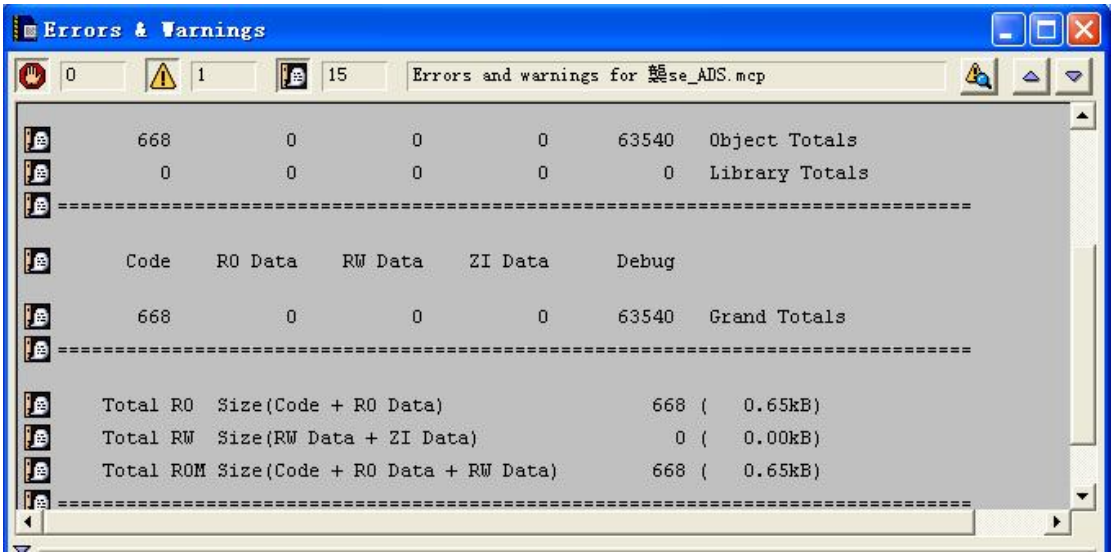


在这里设置编译最终的输出，输出格式默认为 Plain binary,输出文件要选择一个输出文件名，需要注意的是，虽然格式已经指定为 bin，但是文件还是应该加上后缀名，比如 use_ADS.bin，而不是 use_ADS。

以上设置完成后，可以选择 apply,接受改变。

二，编译

设置完成后，即可点击 make 图标，进行编译与连接，完成后从弹出的 errors & warnings 窗口可以检查编译结果。



如果有错误或者警告可以通过点击信息提示进行错误的定位。
如果没有错误将报告生成代码的大小，同时生成选择过的.bin 文件，如上图。

三，程序下载运行

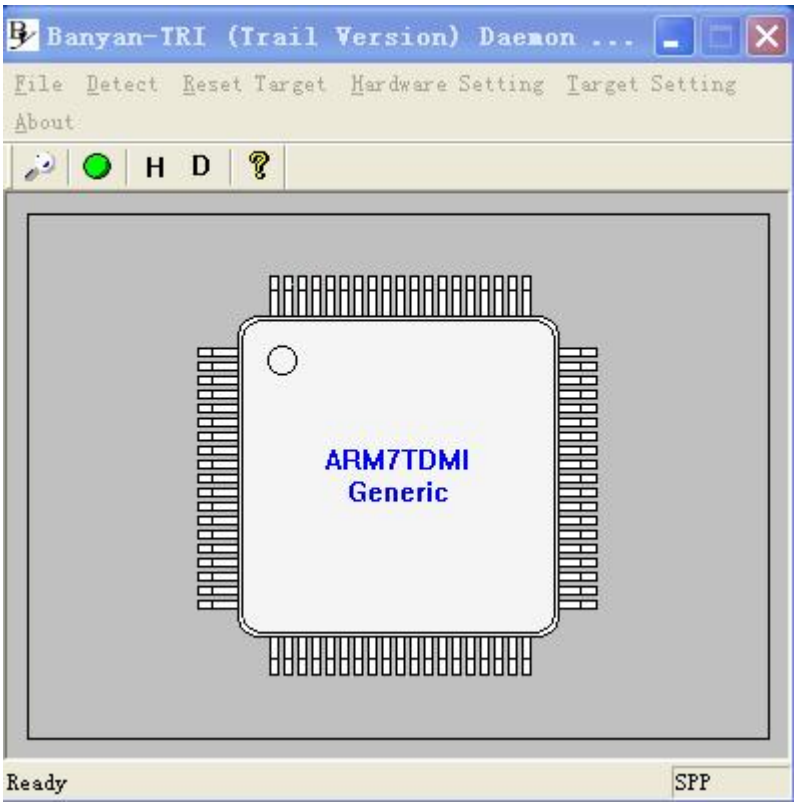
生成.bin 文件后，就可以利用 SAM-BA 进行程序的下载。

具体程序的下载请参考相关文章，或者请版主链接，谢谢！

下载完成后，按动 reset 按钮，将看到 3 个 LED 顺序点亮，说明运行正常。

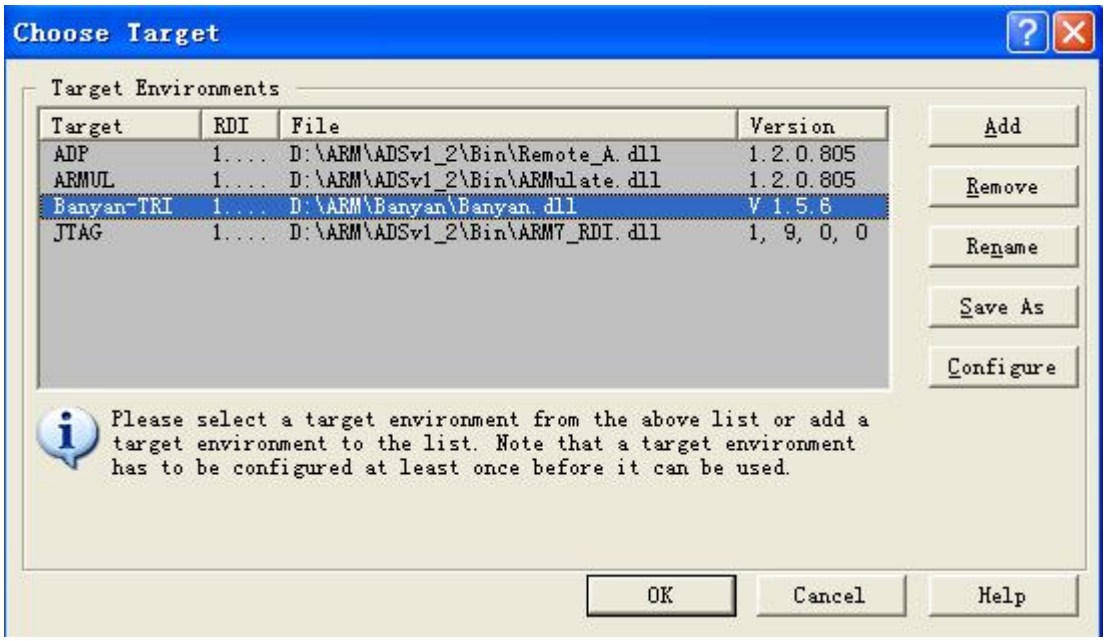
四，调试程序（DEBUG）

此时接上 wiggler 仿真头，打开 banyan，若出现

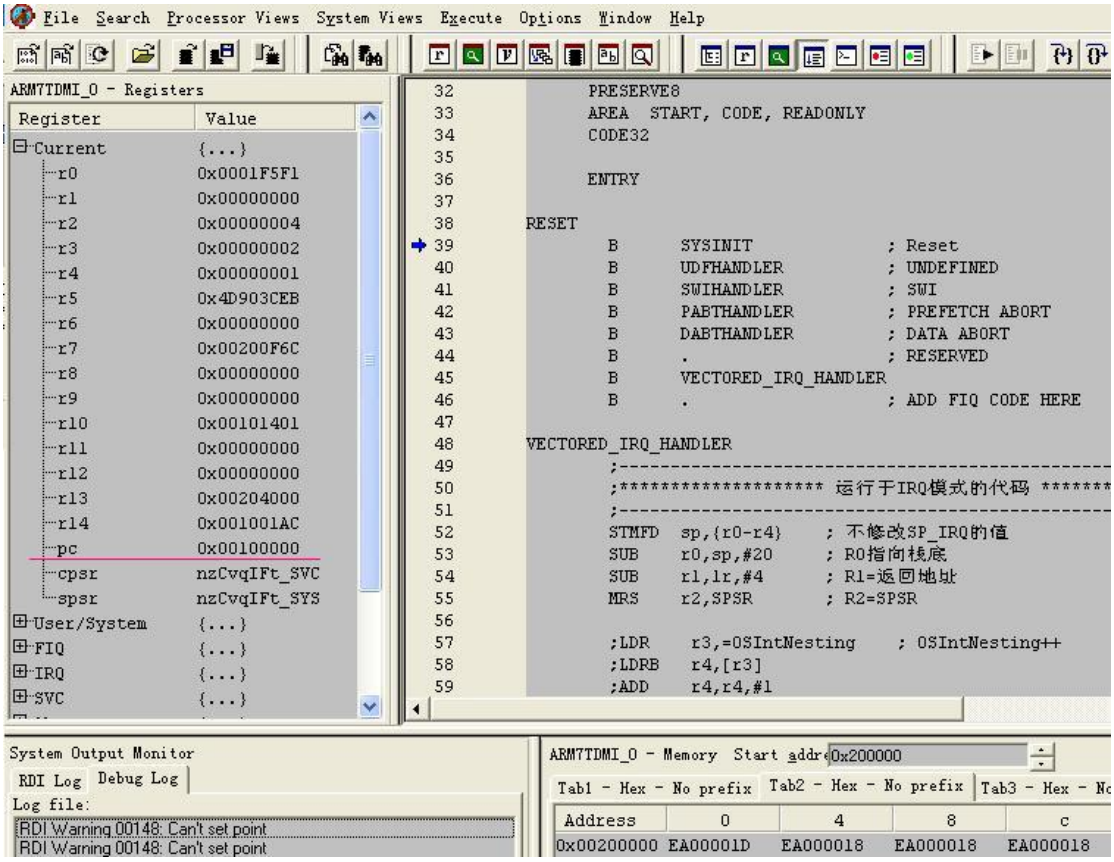


说明连接正确，否则请检测连接。

正确连接后在 ADS 中点击 debug 图标，将会启动 AXD，并加载工程映像，并试图使用上次默认的连接对系统进行连接，如果无法连接，请确定是否选择以 banyan 为调试代理



正确连接后，AXD 将加载映像相关的各种符号，然后 PC 会指向程序开始的地方



注意此时 PC 的值就是 R0 的地址，而不是 0；

此时按 F8 进行单步运行，将出现 RDI Warning 00148:Can’t set point,且程序不会单步运行，需要点击 Execute→delete all breakpoints,再次点击 F8，程序已经可以单步运行了；或者在程序合适位置设置断点（双击所在行行号），然后让程序运行到指定位置。由于这个 banyan 是免费的评估版本，因此断点有很多的限制。

五, 在 RAM 中运行代码

以上编译的程序,运行时代码位于 FLASH,要让代码运行于 RAM,需要增加相应的代码复制程序(示例中已包含).

修改 RO = 0x200000,RW = 0x202000,Image Entry Point = 0x200000,重新编译,生成新的代码,下载到 S64 中运行,可以看到 LED 闪烁速度有明显的提高,有兴趣,有设备的话,可以用示波器比较两者的周期差异.

六, 其它

可以看到, 使用工具并不是很难, 也希望更多的加入学习, 交流的行列, 相互学习, 共同提高.