**使用JLink ARM-OB STM32 简化ARM Cortex处理器开发**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

iBOard 电子学堂官方论坛 [http://www.oshcn.com](http://www.oshcn.com/)

官方博客：[http://XiaomaGee.cnblogs.com](http://xiaomagee.cnblogs.com/)

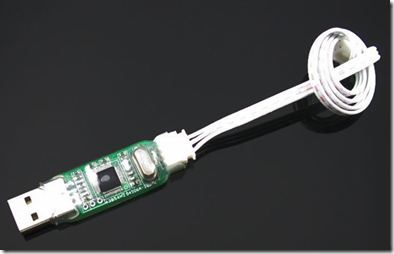
官方微博：<http://weibo.com/iBoard>

《iBoard 电子学堂 群【A】》：204255896（500人超级群）   
《iBoard 电子学堂 群【B】》：165201798（500人超级群）   
《iBoard 电子学堂 群【C】》：215053598（200人高级群）   
《iBoard 电子学堂 群【D】》：215054675（200人高级群）   
《iBoard 电子学堂 群【E】》： 215055211（200人高级群）   
《iBoard 电子学堂 群【F】》： 78538605  （200人高级群）

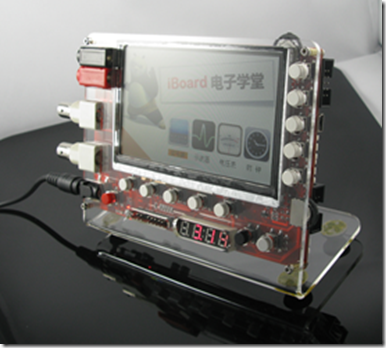
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

专为 《iBoard 电子学堂》打造的便携式调试器

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210141206680.png)

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210141246761.png)

                     JLink ARM-OB STM32

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/20120221014126896.png)

《iBoard 电子学堂》

**一、 Jlink ARM-OB STM32 简介**

Jlink ARM-OB STM32 是 SEGGER 公司为开发板定制的板载Jlink调试方案。除了不能测试目标板电压外，此Jlink 与正式版功能几乎一致

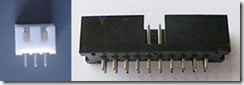
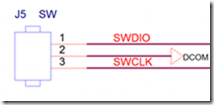
（速度限制到 **4MHz**）。

受 Ourdev论坛 minux 网友一篇文章启发，参考minux 网友贡献的资料，笔者制作了此文所提的 Jlink 简化版，经过将近半年的测试，其不但工作稳定，而且携带方便，故此与大家分享。其特点为：

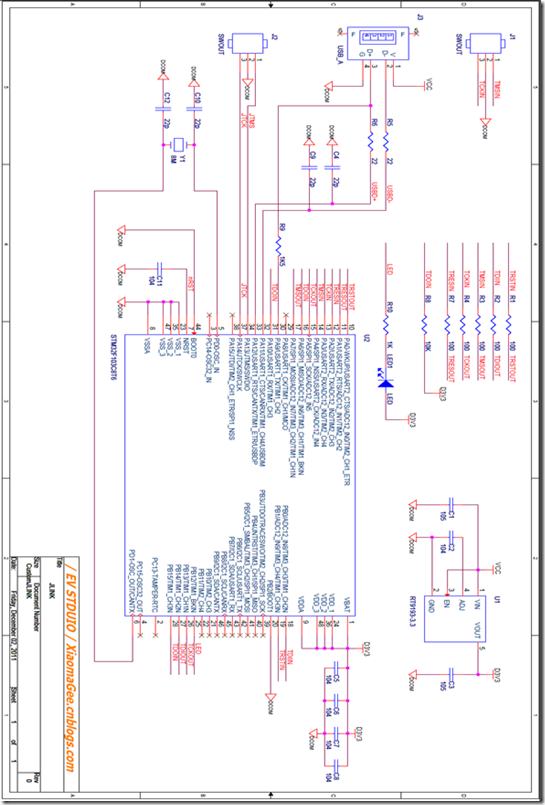
1. 支持IAR EWARM 和 KEIL MDK；
2. 仅支持 SW 两线调试；
3. 工作稳定，无丢失固件现象；
4. 仅U盘大小，携带方便；
5. 直接与电脑USB 接口连接，无需USB 线缆；
6. 接口简化，仅用三根线，完成高速调试、下载；
7. 仅需要三脚 XH2.54 连接器，大大节约PCB面积；
8. 与JTAG 相比，仅需两根线（TMS / TCK），可节约若干条 IO 口。

**二、 硬件电路及接口**

传统20脚JTAG底座个头大，占用PCB面积多，连接线复杂。此Jlink经过精简，使用三脚 XH2.54 连接器，完成 SWD 接口的物理连接，经过半年多测试，即使连接线达 40cm的情况下，此Jlink 依然能可靠下载和调试。图一为接口形式对比，图二为接口定义，图三为硬件电路图。

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210141316374.png) [](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210141317836.png)

    图一 接口形式与传统JTAG对比                 图二接口定义

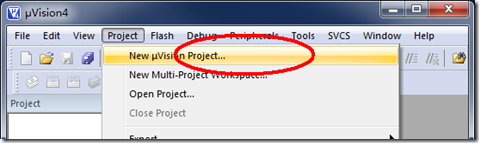
[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210141332527.png)

图二 原理图

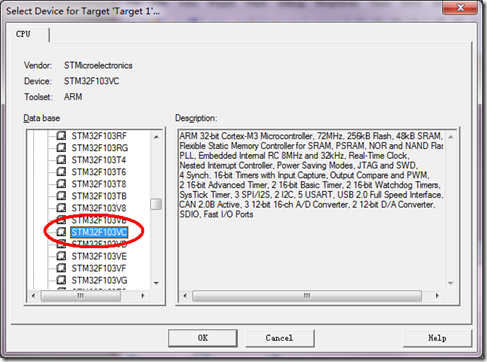
**三、 在Keil MDK 中使用Jlink**

本节介绍在Keil MDK 4.14中使用Jlink 下载、调试流程，目标CPU 为 STM32F103VC，。首先，我们安装 Keil MDK 开发环境和 Jlink软件。建立工程步骤如下：

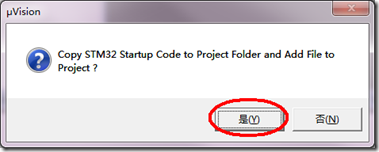
① 打开KEIL MDK 开发环境，并点击如图：

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210141363937.png)

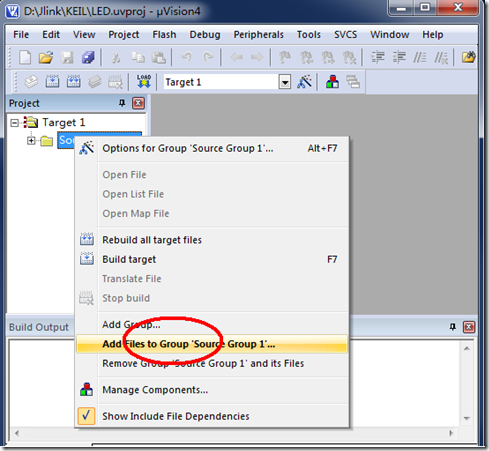
② 在硬盘合适的位置，建立一个工程并保存，本文命名为 LED，然后跳出CPU 选择，我们选择STMicroelectronics 公司的STM32F103VC，如图所示：

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210141375083.png)

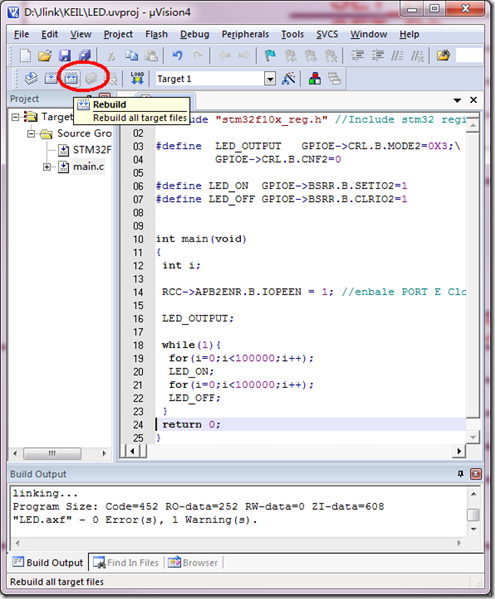
③ 弹出对话框，加入 Startup 文件，选择“是（Y）”，如图所示：

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210141398138.png)

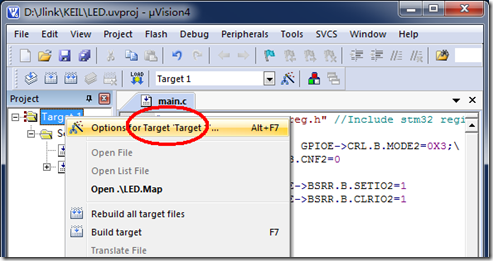
④ 把附件内的 main.c 和 stm32f10x\_reg.h文件拷贝到工程文件夹内，并把main.c加入到工程中，如图所示：

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210141413876.png)

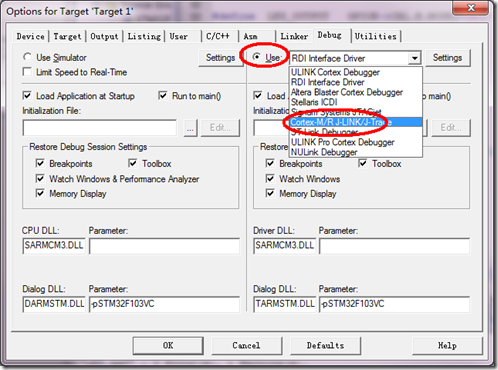
⑤ 添加完成后，我们可以点击“[clip_image005[4]](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210141439614.jpg)”来编译整个工程。完成后如图所示：

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210141433268.png)

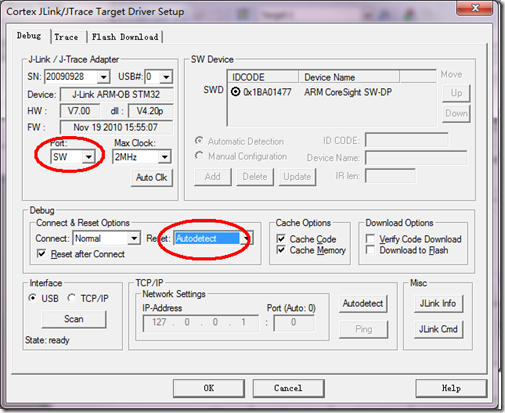
⑥ 正确无误后，我们来设置工程属性，以便可以通过Jlink 下载、调试。首先进入工程属性设置，我们可以通过如图所示方法，进入工程属性页面，也可以通过 Project 菜单下的Options for Target …… 菜单进入属性设置。

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210141463042.png)

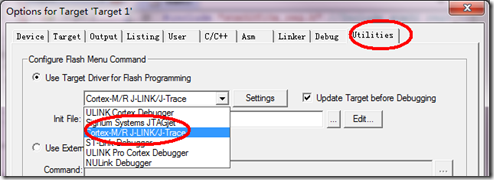
⑦ 进入属性设置页面后，我们选择 Debug Tab选项，然后选择 Use Cortex-M/R J-LINK/J-Trace，然后点击Settings选项。

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210141487700.png)

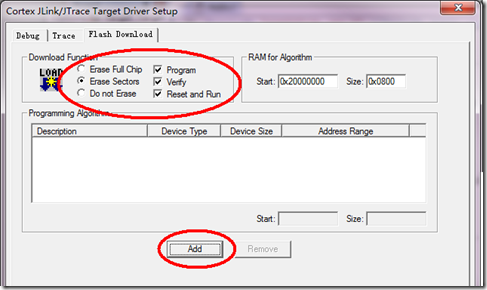
⑧ 调试端口我们选择 SW模式，复位模式选择 Atuodetect模式，如果此时Jlink连接了上电后的CPU，则能在SW Device栏检测到CPU ，如图所示。设置完毕，按OK退出此设置页面。

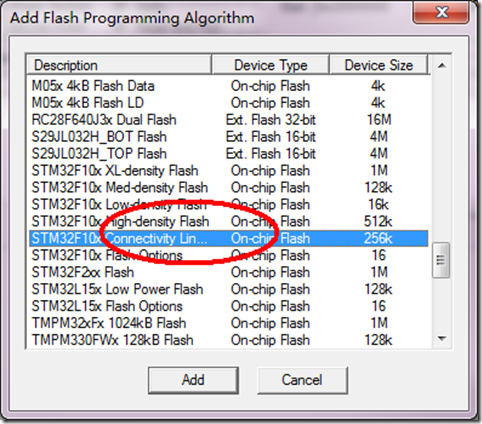
[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210141512948.png)

⑨ 然后进入 Utilities 页面，并选择 Cortex-M/R J-LINK/J-Trace，然后点击Settings，如图所示：

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210141538228.png)

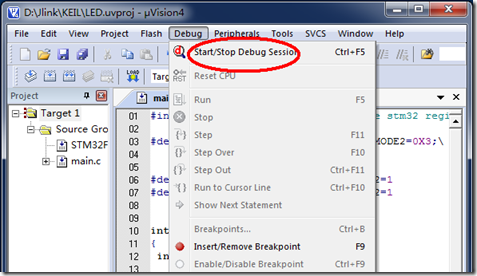
⑩ 然后勾选上如图设置，并按 Add 按钮添加变成对象，如图所示：

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210141557031.png)

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210141583674.png)

点击 Add 后，并按OK 退出设置菜单，回到工程界面。

⑪ 如果仅仅需要下载程序，则只需要点击 [clip_image013](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210142008398.png) 按钮，如果需要调试，则可以点击Debug菜单下 Start/Stop Debug session 来进入、退出Debug 模式。如图所示。

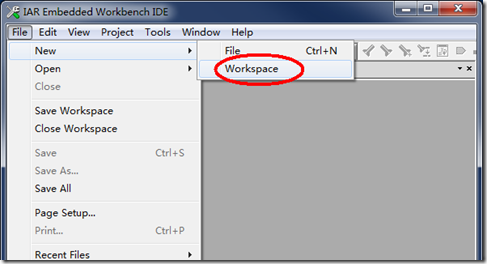
[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210142004528.png)

至此，我们完成了用 Keil MDK 建立一个新工程，并通过 Jlink下载、调试。

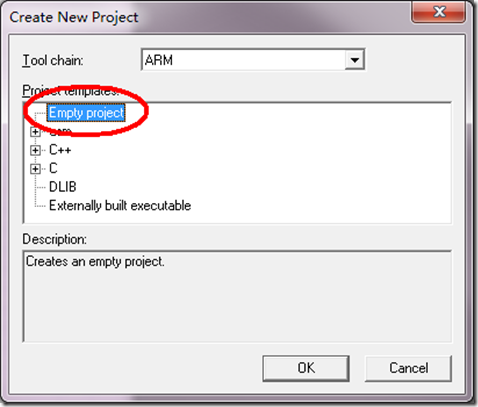
**四、 在IAR EWARM 中使用Jlink**

本节使用 IAR EWARM 5.41，并配合Jlink 完成下载、调试工作，目标CPU 为 STM32F103VC。在IAR EWARM 中使用Jlink 与Keil MDK中大同小异，具体方法如下。

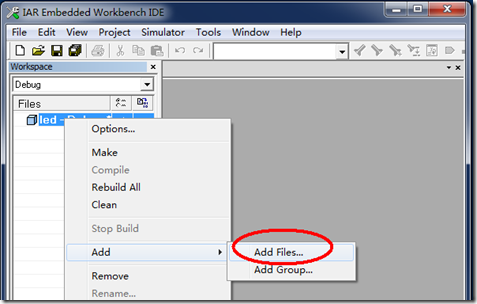
① 首先安装IAR EWARM 5.41 开发环境，并安装安装目录下的Jlink驱动包，完成后，打开IAR EWARM 图标，并建立一个新的工作区，如图所示。

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210142026612.png)

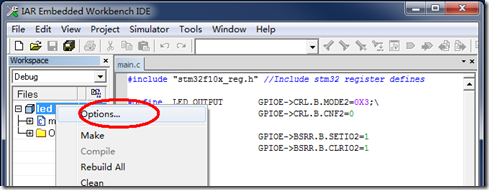
② 然后点击Project 菜单下的Create New Project…，我们选择一个空工程（Empty project），如图所示，点击 OK 确认操作，并存盘为 led.eww 工程。

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210142037267.png)

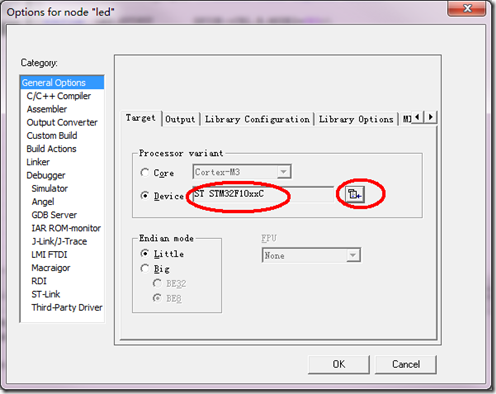
③ 把附件内的 main.c 和 stm32f10x\_reg.h文件拷贝到工程文件夹内，并把main.c加入到工程中，如图所示：

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210142036777.png)

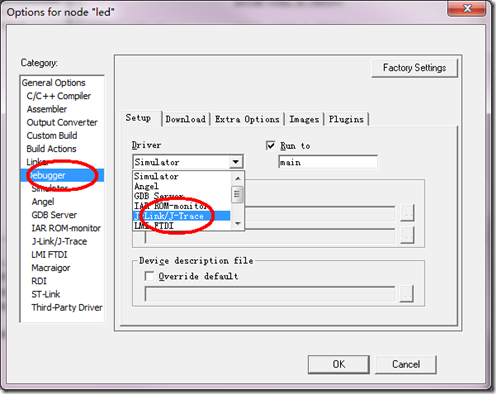
④ 加入后，我们可以通过点击“[clip_image004[9]](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210142057498.png)”来编译工程，成功后，我们来设置工程属性，以便使用Jlink。如图所示：

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/20120221014205355.png)

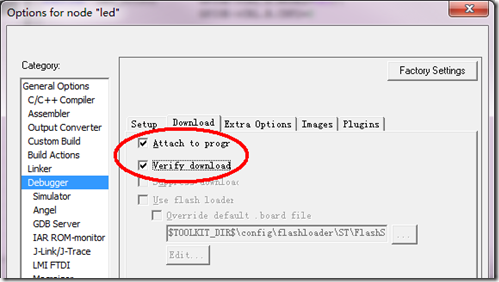
⑤ 进入后，我们通过General Options，来选择CPU,如图所示：

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210142074632.png)

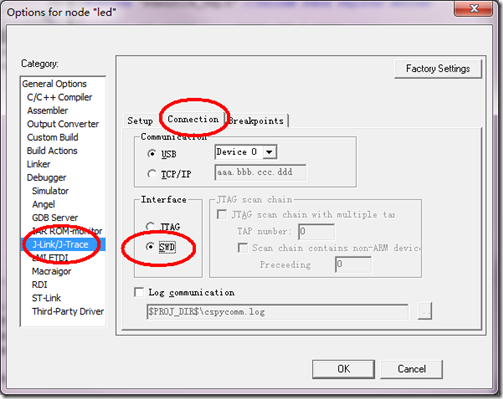
然后选择Debugger选项，并在Setup tab里选择J-Link/J-Trace，

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210142088700.png)

⑥ 进入Download tab，并按下图设置选项。

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210142107993.png)

⑦ Download 设置完成后，我们进入J-link/J-Trace 选项，并选择SWD调试接口，如图所示。

[](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/20120221014211143.png)

然后按OK 结束设置。此时我们可以通过[clip_image010[5]](http://images.cnblogs.com/cnblogs_com/xiaomagee/201202/201202210142504521.png)按钮来下载和调试工程。

至此，一个简单的利用Jlink来下载、调试Cortex-m3内核的工程建立完毕。