ARM微控制器开发入门

----使用GNU开源工具链

大纲：

ARM微控制器项目代码组成
Windows平台上使用CooCox的CoIDE
Windows/Ubuntu/Mac上使用Eclipse以及GNUARM插件

随着物联网的兴起，ARM公司的微控制器以其卓越的性能和极低的功耗设计再一次得到了越来越多的关注。创客（maker）是当前物联网浪潮的另一大特色，他们大都具备深厚的特定领域知识，但并不是专业的嵌入式开发工程师，他们更多的是想借助ARM微控制器的良好特性来实现自己的创新想法。为了让更多人能够更加容易的在ARM微控制器上实现自己的梦想，ARM公司围绕微控制器编程投入了大量的资源，构建了多个蓬勃发展的开发平台和社区。全球的创客和开发工程师基于这些平台和社区，碰撞想法，分享经验，共享代码，完成了一个又一个精彩的项目。了解这些平台和社区，善用已有的代码模块，无疑可以加速你的微控制器编程之旅。本文将会介绍如何使用由ARM公司精心打造的免费GNU开源工具链来开启嵌入式编程之旅，希望这些介绍能够帮你更好的站在巨人的肩膀上来完成自己的梦想。

本文中的微控制器指的是基于ARM公司Cortex-M系列的微控制器如Cortex-M0和Cortex-M7.这些微控制器各有特色，在ARM官方网站有详细的介绍。一个完整的微控制器项目通常包含如下的代码：

1. 硬件启动代码，初始化启动向量表，代码段等
2. 硬件初始化代码，初始化具体硬件，如果微控制器主频，GPIO初始化等
3. 项目核心代码
4. 内存布局脚本，指出如何加载最终项目文件到板子

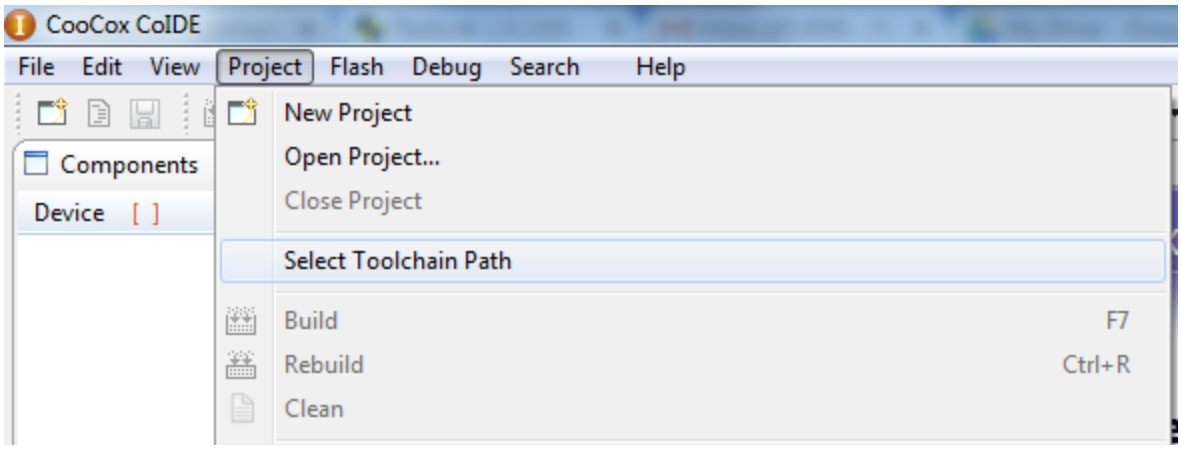
不同的开发工具对这些部分有不同的命名和组织形式。一般来讲，开发板厂商应该提供除了第三部分外的其他部分。得益于ARM微控制的广泛应用，对于常见的开发板，很多微控制器开发套件已经包含了这些代码模板，使得开发人员可以迅速上手。同样得益于C语言的可移植性，我们可以在不同的开发套件中重用这些代码。需要指出来的是，不同的开发套件会使用不同的方法或者格式来描述内存布局。

网站https://launchpad.net/gcc-arm-embedded是ARM公司开源工具链的主网站，版本发布，用户支持，都是在这个网站进行。 ARM公司提供的GNU开源工具链是一个命令行模式的工具链，可以运行在Windows，Linux和Mac平台，对于Windows平台，用户可以选择安装包，也可以

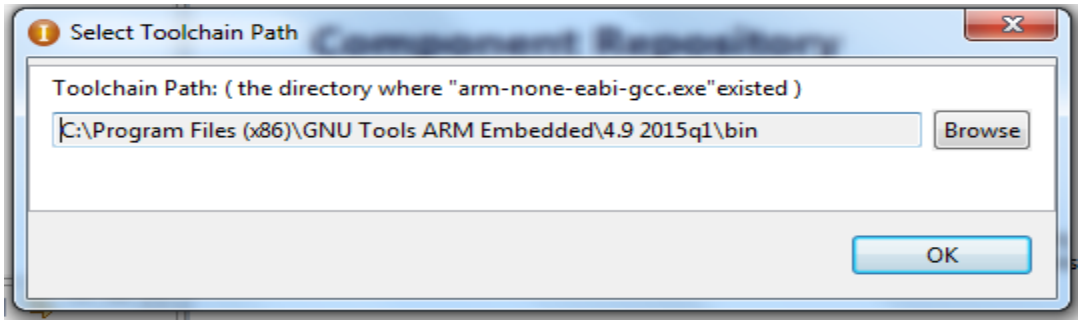
选择免安装的zip压缩包，解压缩后就可以立即使用。每年第四个季度会推出一个主要版本更新，大版本号会更新，主要包含新的功能。每个季度都会推出一个当前版本的季度更新版本，季度更新版本主要包含上一个季度版本的错误修正。对于习惯命令行模式的用户，下载并解压缩相应工具链，就可以直接使用，不需要注册，也没有收费。对于习惯集成开发环境的用户，我们推荐同样是开源的CoIDE和Eclipse。CoIDE可以在其官网http://www.coocox.org 下载，由于最新的CoIDE在本文写作时还处于beta阶段，所以这里推荐版本v1.7.8。除了没有集成工具链外，CoIDE提供了所有常用的微控制器开发功能。

下面我们介绍如何使用CoIDE和ARM开源工具链建立我们的第一个微控制器项目，开发板我们选择的是意法半导体的STM32F4 [http://www.st.com/stm32f4-discovery/]，这是一款基于Cortex-M4微控制器的开发板。

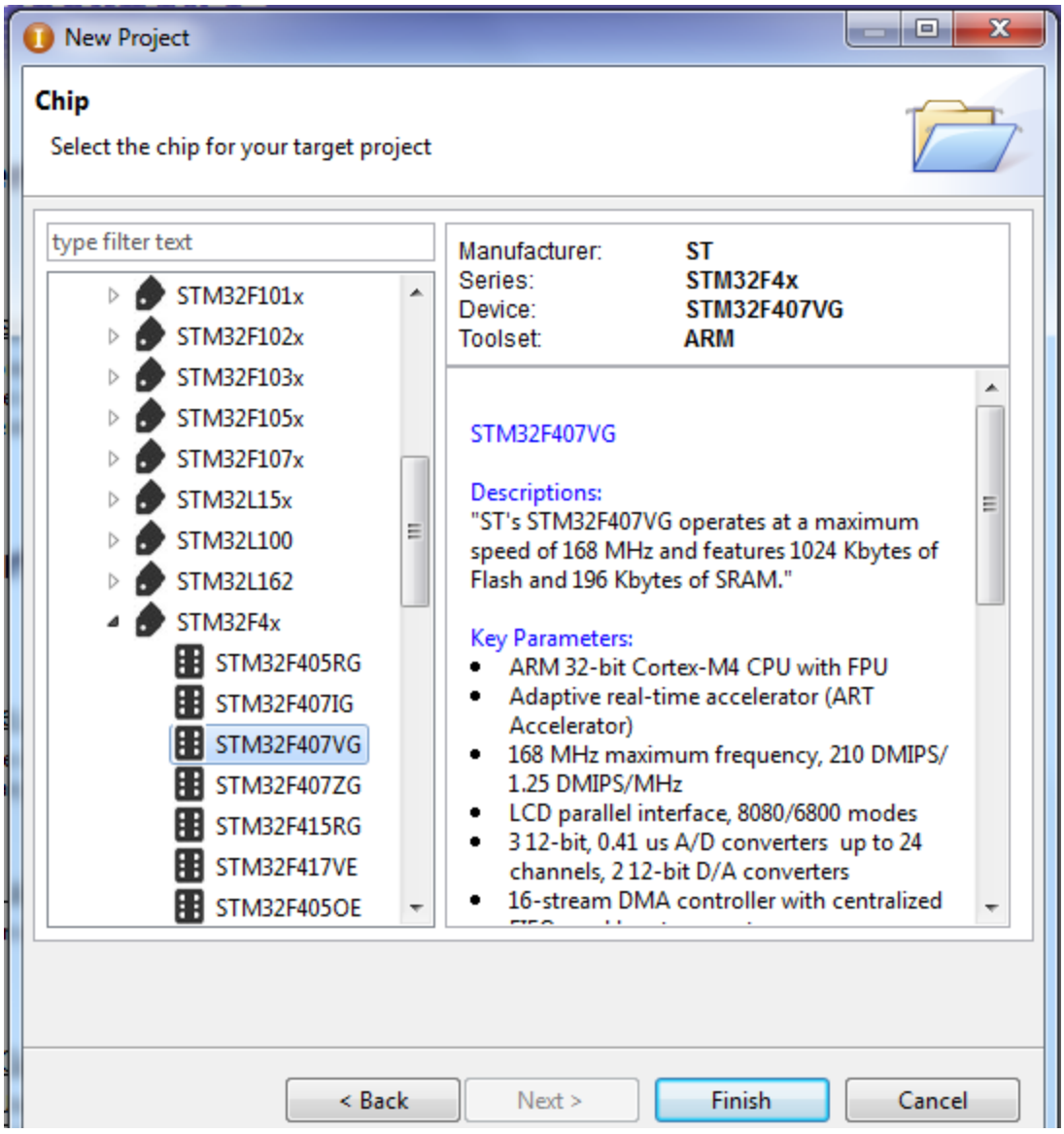
1. 下载并安装ARM开源工具链
2. 下载并安装CoIDE
3. 启动CoIDE，点击Project菜单，选择Select Toolchain Path来选择要使用哪一个安装好的工具链



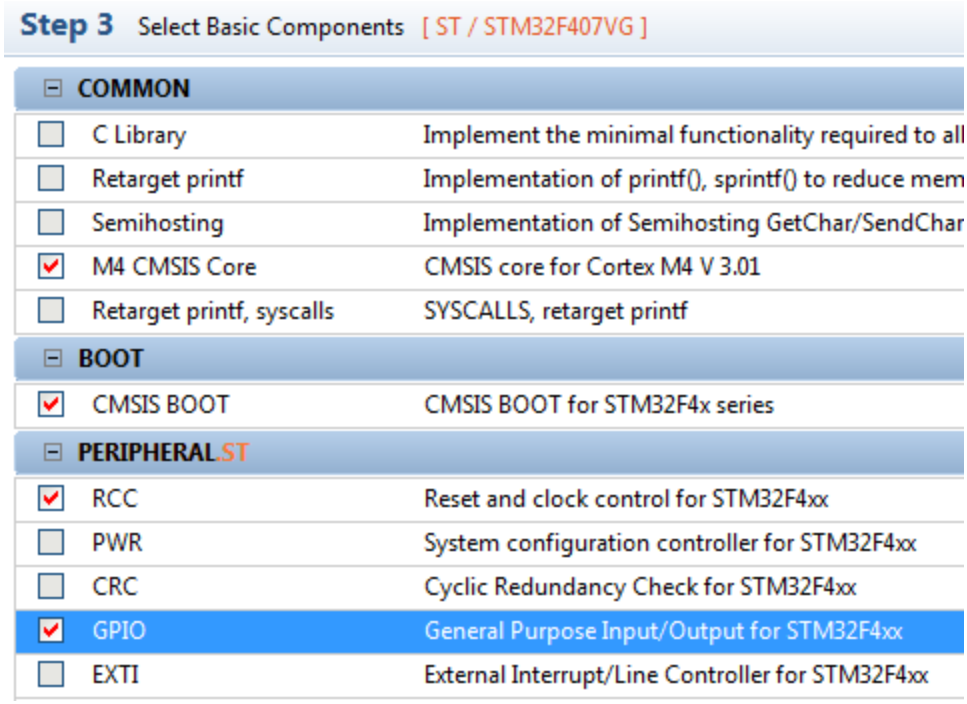
指定好的路径如下图所示：



4. 在CoIDE主界面我们点击“Create a New Project”，并做如下选择，来为我们的板子创建工程

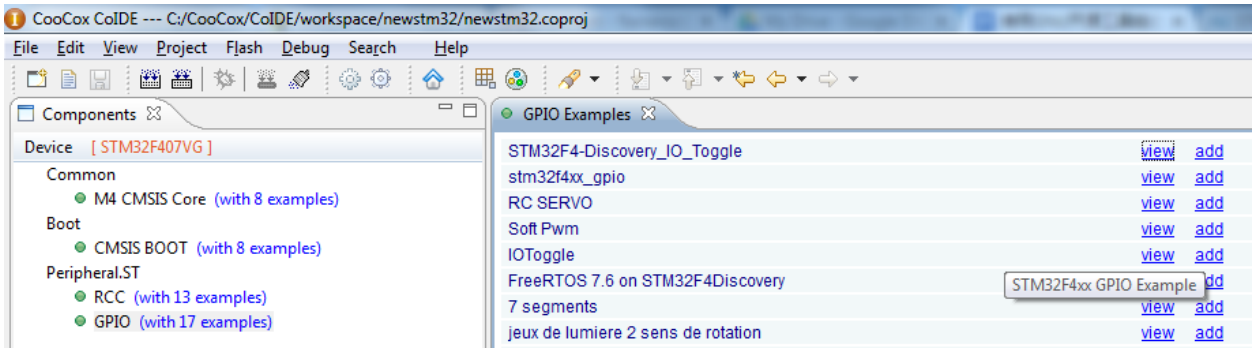


5. 点击Finish按钮完成工程创建第一步。在接下来的Step 3中，我们可以勾选需要的模块的对应的代码，本例中我们想控制板子上的LED灯，所以我们选择如下：

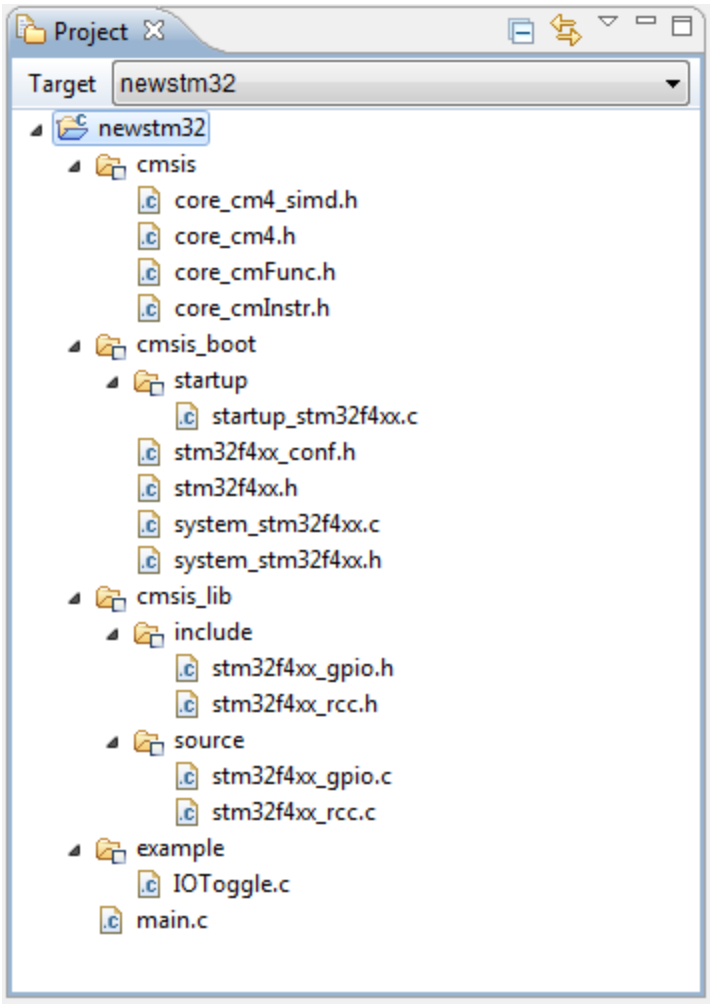


板载LED灯属于外设中的GPIO类设备。

6. 集成开发环境的左上角会显示当前开发板的已经存在的例子程序，单击GPIO后面的例子链接，在右面有更详细的例子列表，在这里我们使用由CooCox提供的官方示例IOToggle，单击对应的add链接，把该例子程序加入到我们的当前工程中。

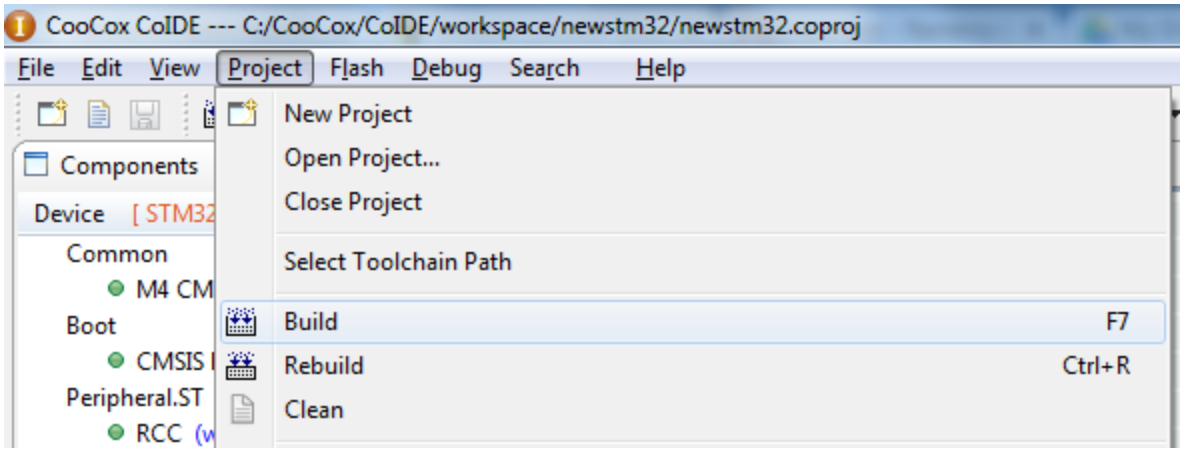


7. 这样点几下鼠标，不费吹灰之力，我们就得到了一个可以工作的微控制器项目：

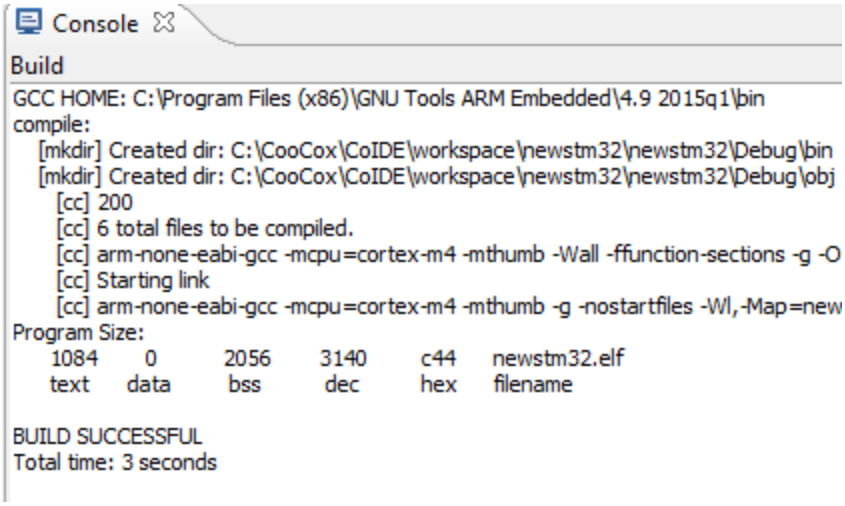


接下来我们需要使用开源工具链编译这个项目，并加载生成的工程文件到板子。

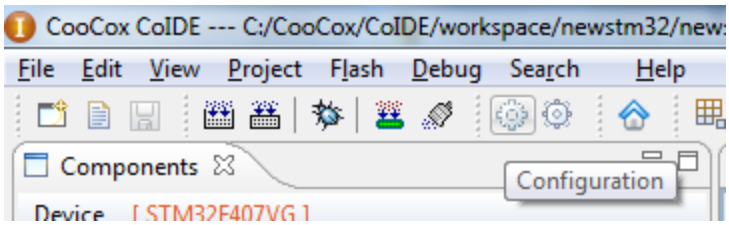
点击Project并选择Build



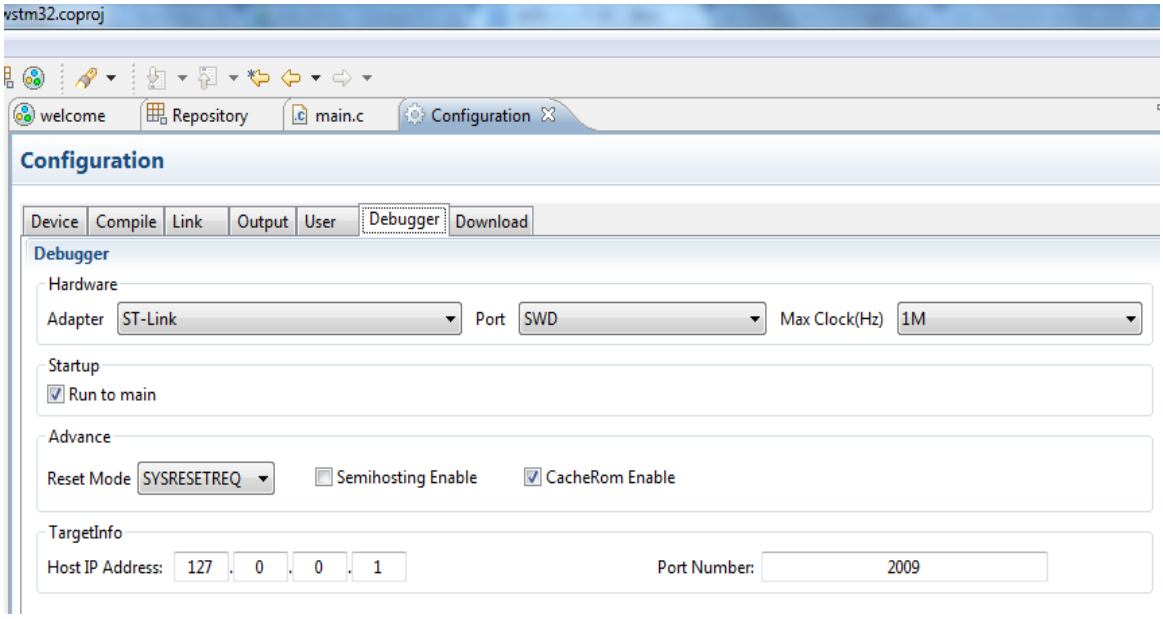
一切顺利的话，应该会看到



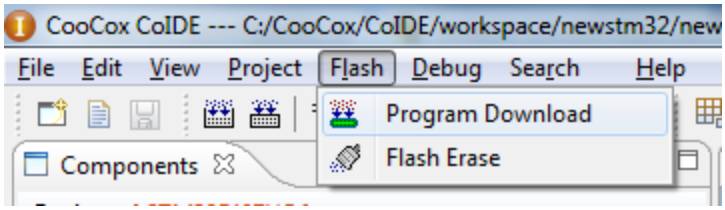
8. 接着我们使用USB线把板子和电脑连接起来。点击如下按钮来配置CoIDE和我们的板子的通信方式：



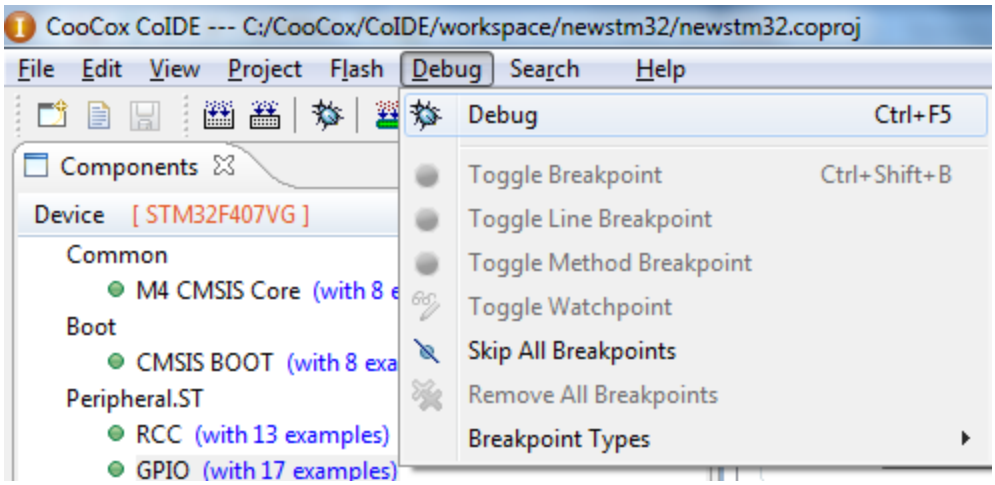
在弹出的配置界面中，选择Debugger标签，并作如下配置：



可以直接使用下面的菜单把编译链接生成的项目文件加载到板子中



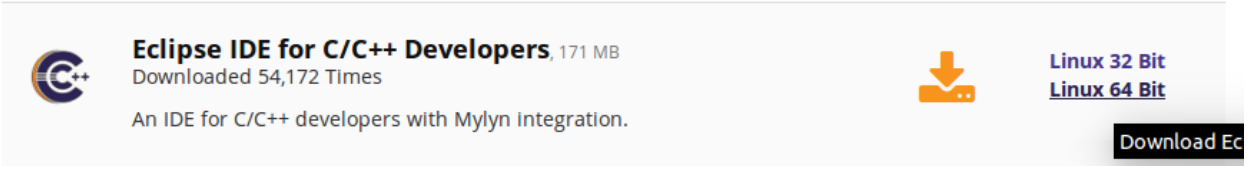
也可以使用下面的菜单命令进行单步的调试



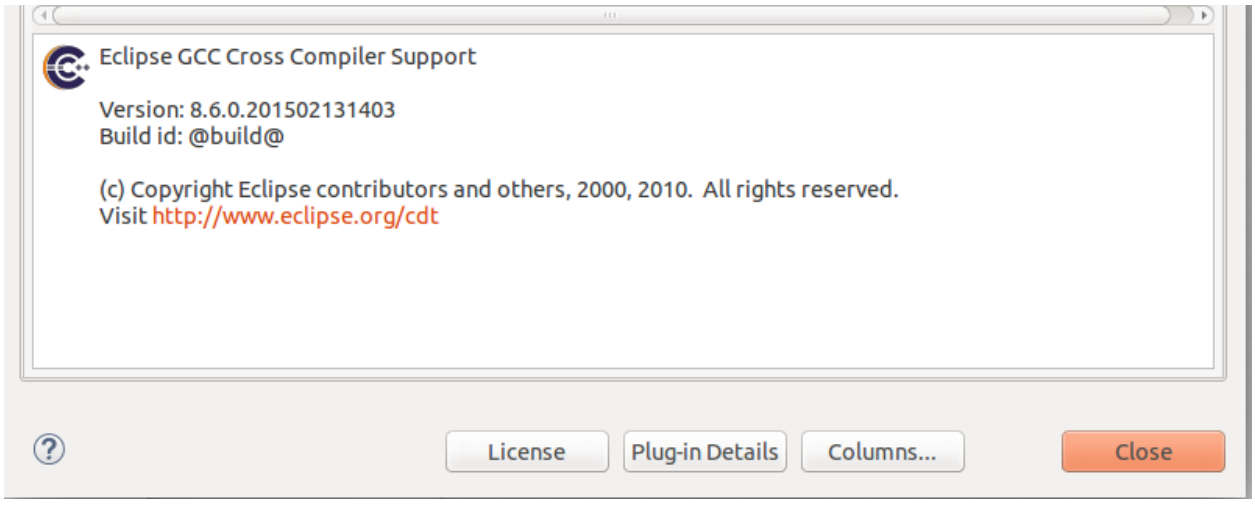
9. 上述的例子工程可以让我们通过板子上的黑色按钮来和LED灯交换，当按钮按下后，LED灯会闪烁一下。如果不做任何操作，板子上的四个灯会依次点亮然后自动熄灭。
10. 了解我们的工程是由哪些文件组成，对于后续开发至关重要，所以让我们再复习一下我们的项目文件树：
    a. 在cmsis目录下的文件是用来配置微控制器的，以及一些封装好的C语言函数，方便我们直接调用他们来使用微控制器提供的硬件特性
    b. 在cmsis_boot目录下提供了板子启动初始化文件
    c. 在cmsis_lib目录下使用C语言对板载资源进行封装，方便我们使用这些板载资源
    d. 在example目录下提供了我们工程的核心实现

Eclipse是另外一个优秀的集成开发环境，并且可以运行在多个主流的操作系统上如Windows，Linux和MacOS。下面我们介绍如何在Ubuntu 14.04 64位操作系统上把Eclipse打造为微控制器集成开发环境：

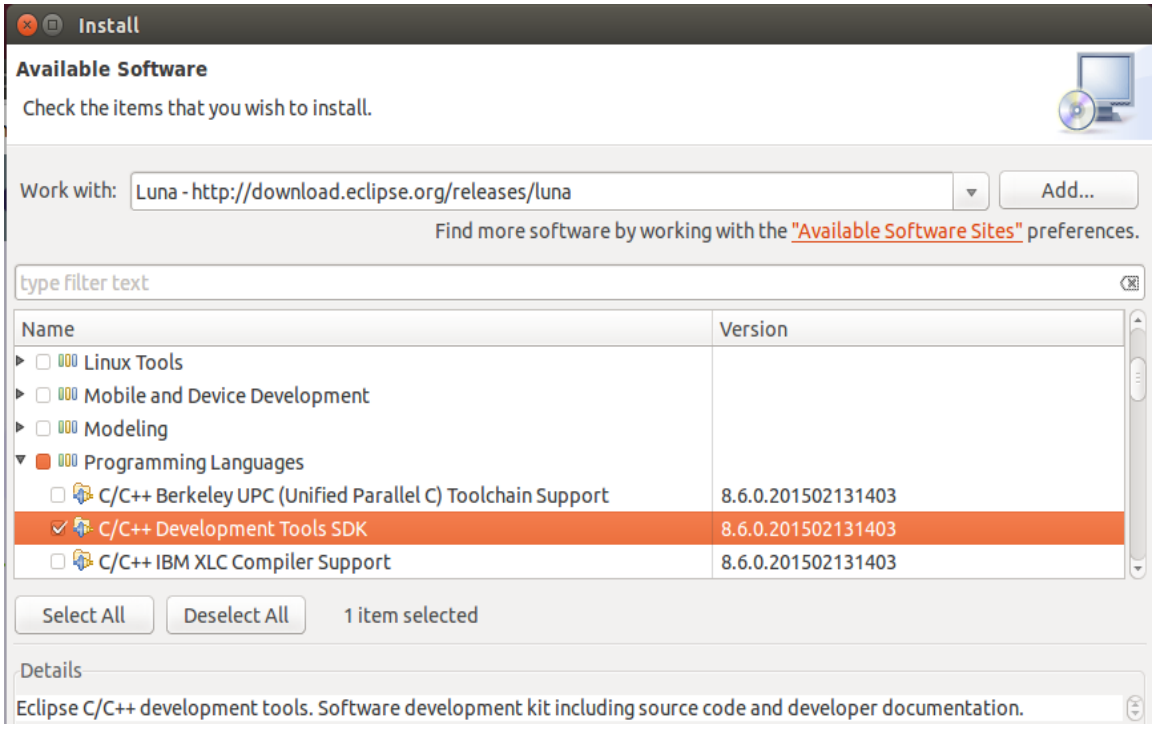
1. 从Eclipse官网下载如下版本的Eclipse，这里选择的是64位版本：



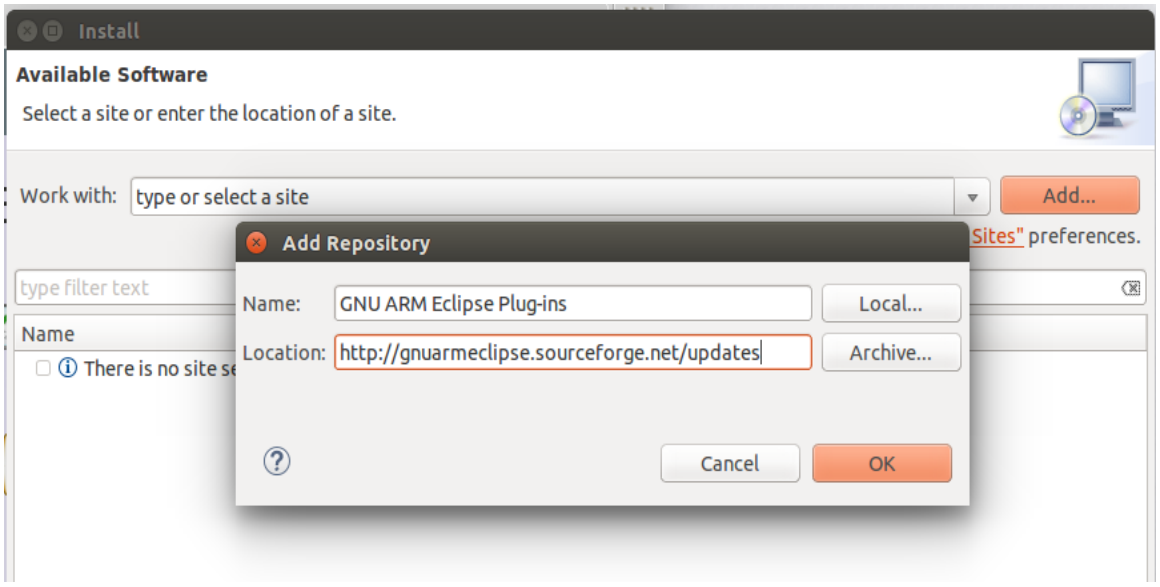
解压缩并启动Eclipse，点击Help菜单中的About Eclipse，可以看到版本信息如下，请确保你的版本和该版本一致或者更新：



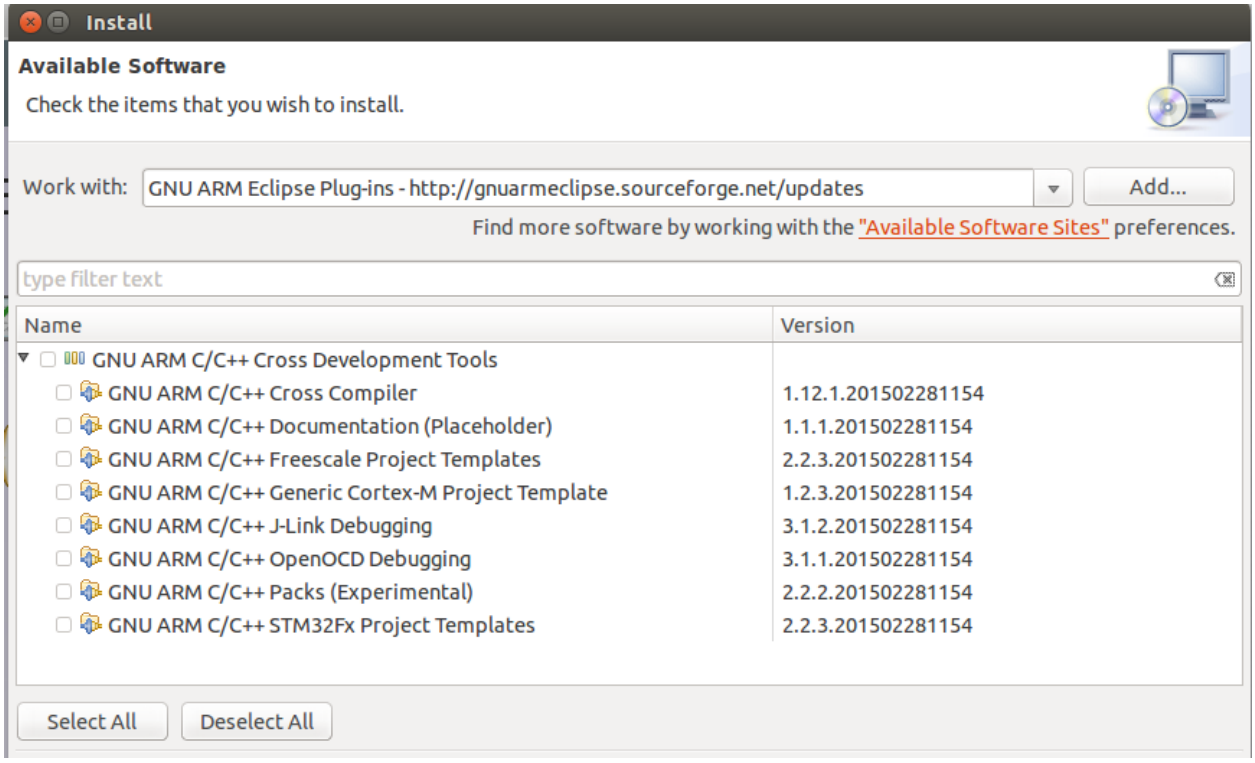
2. 现在我们拿到了一个符合要求的Eclipse，接着安装GNU ARM Eclipse Plug-ins。点击Help菜单中的Install New Software，并做如下选择：



点击Next按钮，在后续步骤中选择同意安装，继续执行直至完成安装，有可能需要重启Eclipse来完成最终安装。再次点击Help菜单中的Install New Software，并点击弹出界面中的Add按钮，做如下添加：

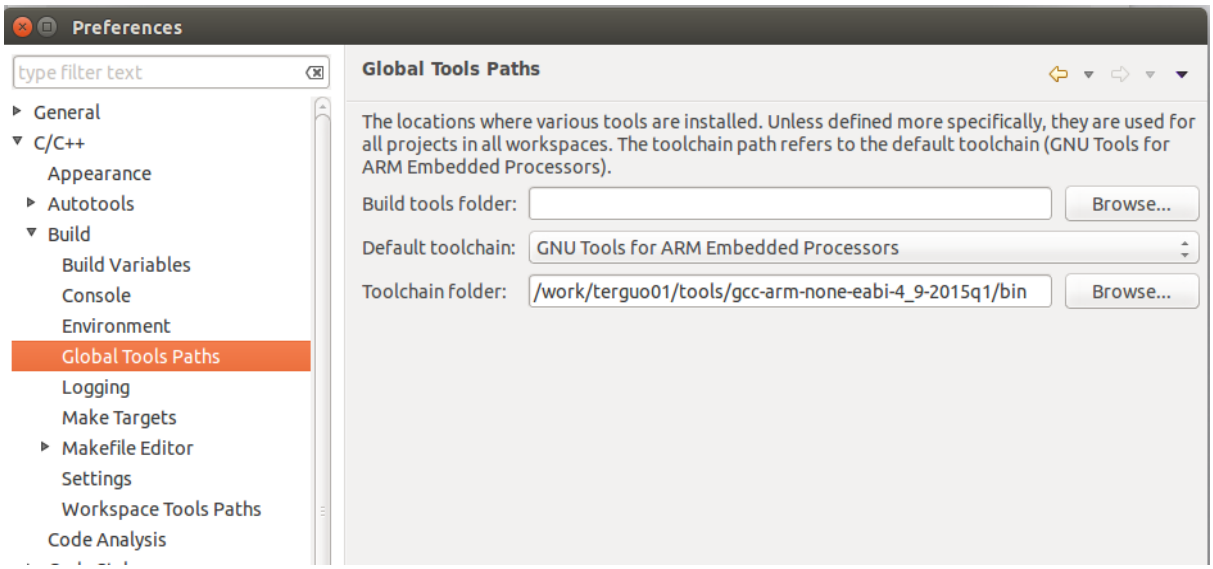


点击OK按钮，并等待一段时间，Eclipse会列出关于该插件的详细信息，如下图所示：

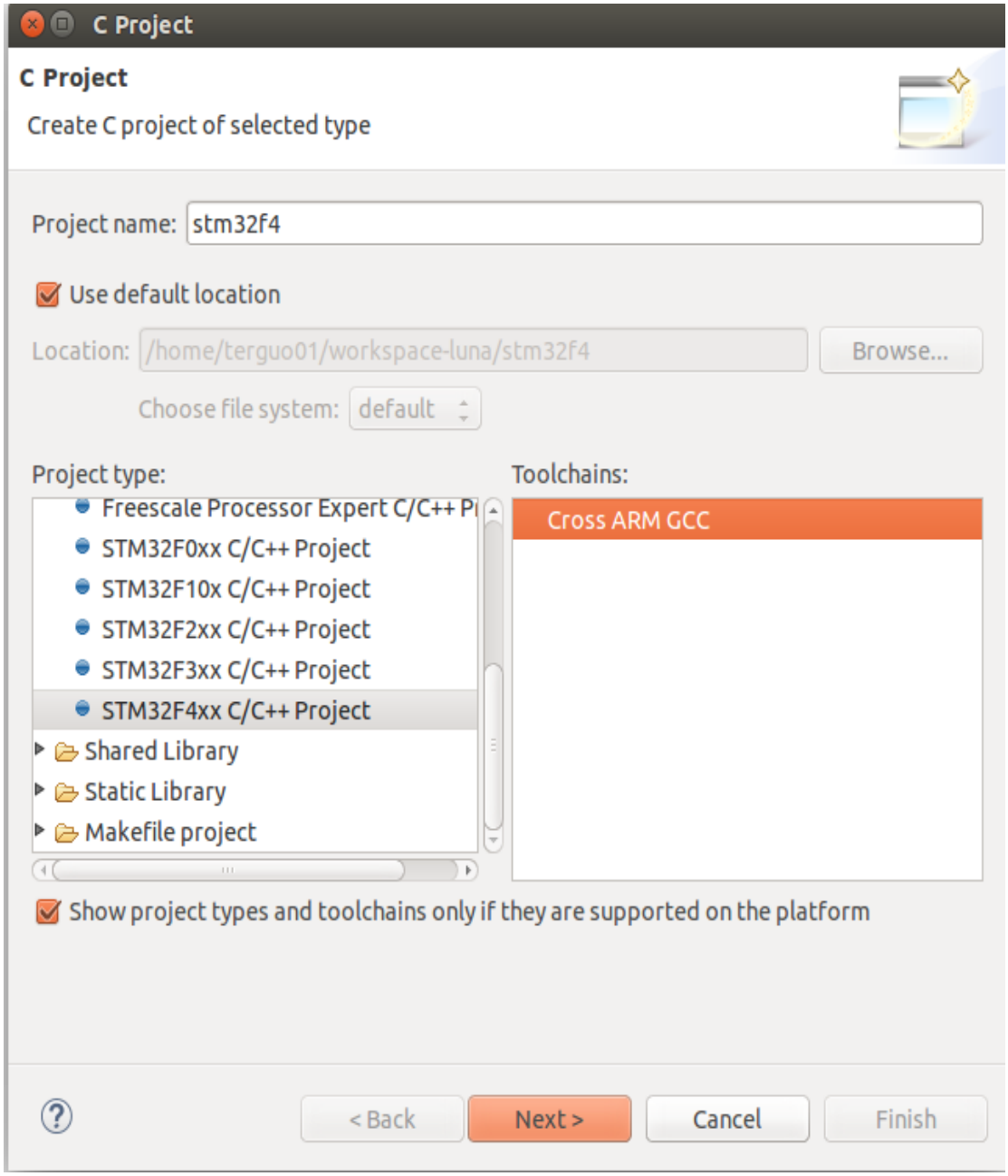


全选，并按照屏幕提示来完成安装，可以忽略警告信息。

3. 从ARM开源工具链官网下载名字中包含linux.tar.bz2的工具链。因为工具链是32位的程序，而我们的宿主机是64位，所以需要执行下述命令来安装32位程序的依赖包：
   sudo apt-get install lib32z1 lib32ncurses5 lib32bz2-1.0
4. 在Eclipse全局设置中，我们做如下配置，设置好开源工具链的路径

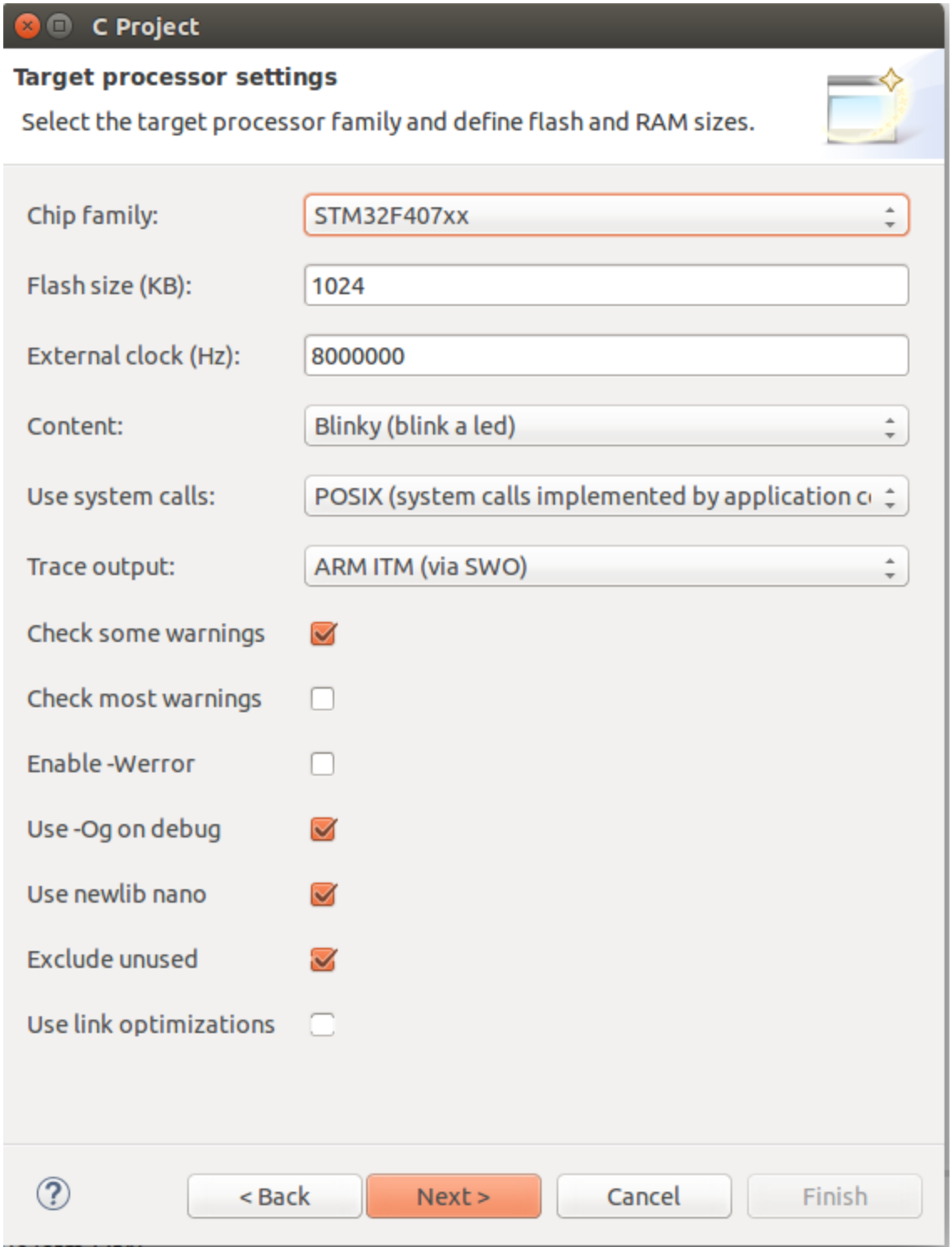


5. 回到Eclipse主界面后，通过File -> New -> C Project来创建一个新的微控制器工程，在新工程配置界面，我们做如下选择：

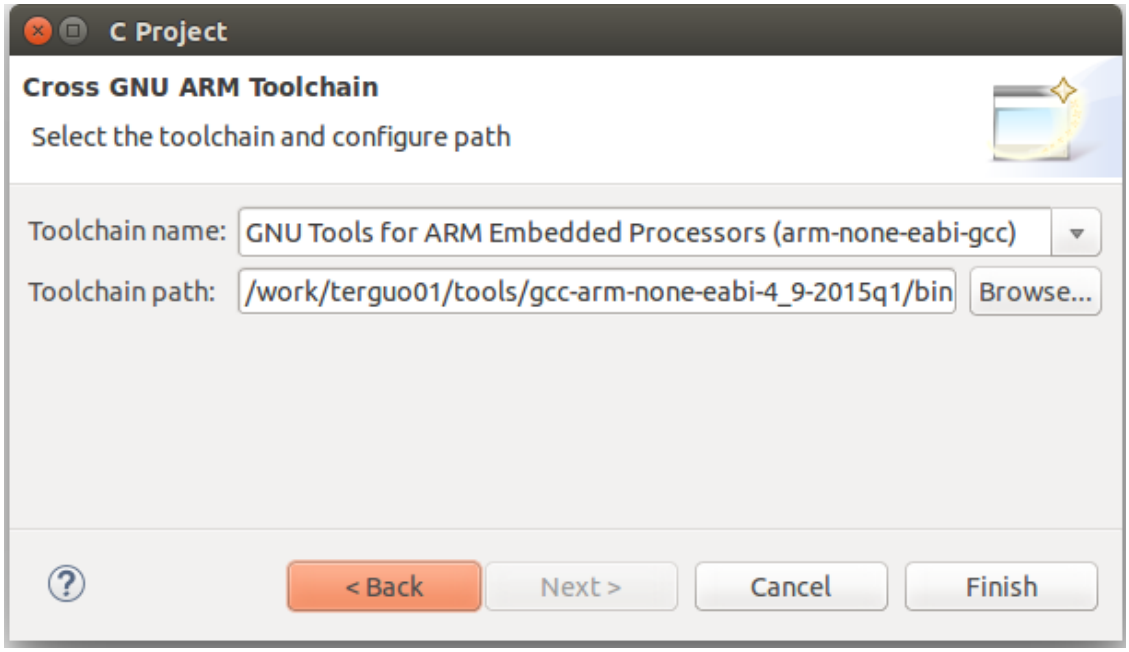


然后点击Next按钮

6. 继续配置目标微控制器参数如下图所示：

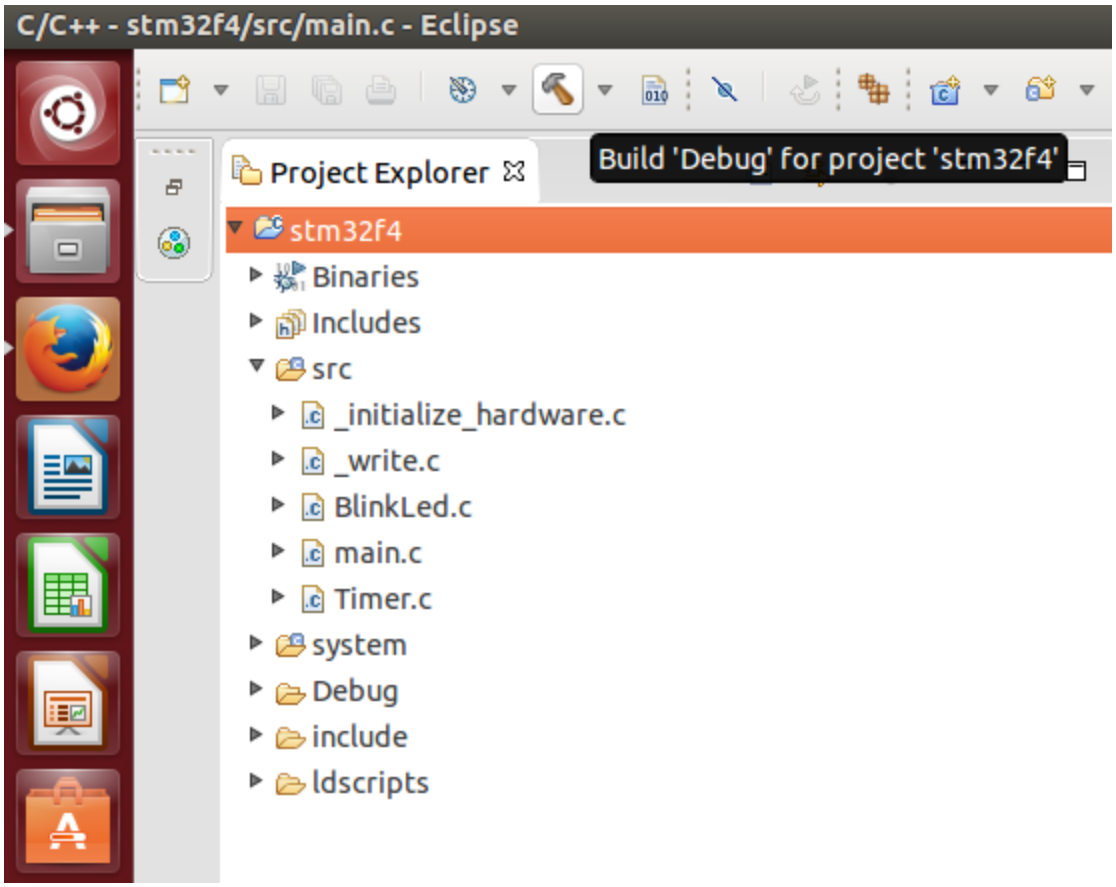


7. 单击Next按钮，直至如下界面，确保工具链路径正确：

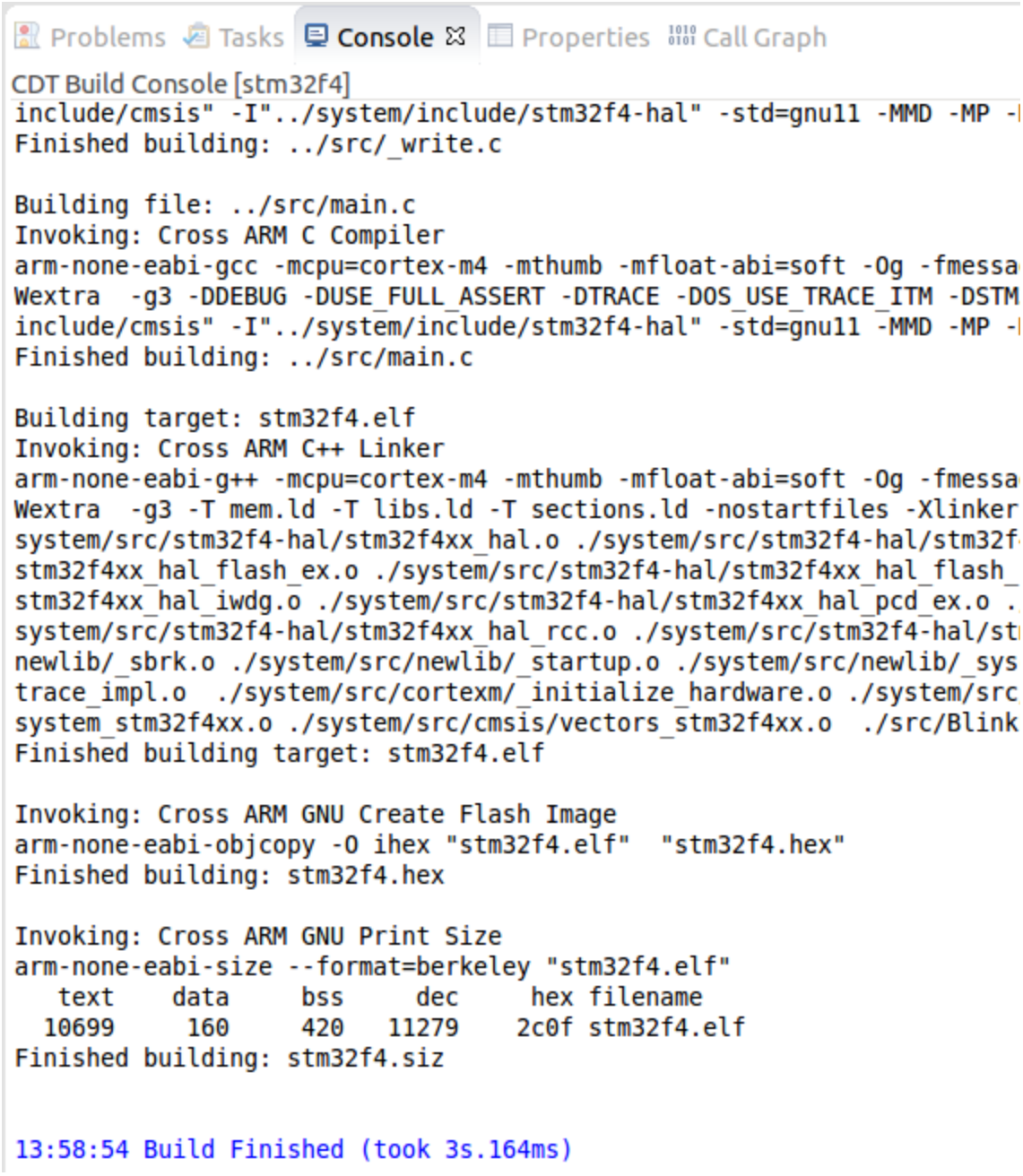


没有问题，就可以点击Finish完成工程的创建。

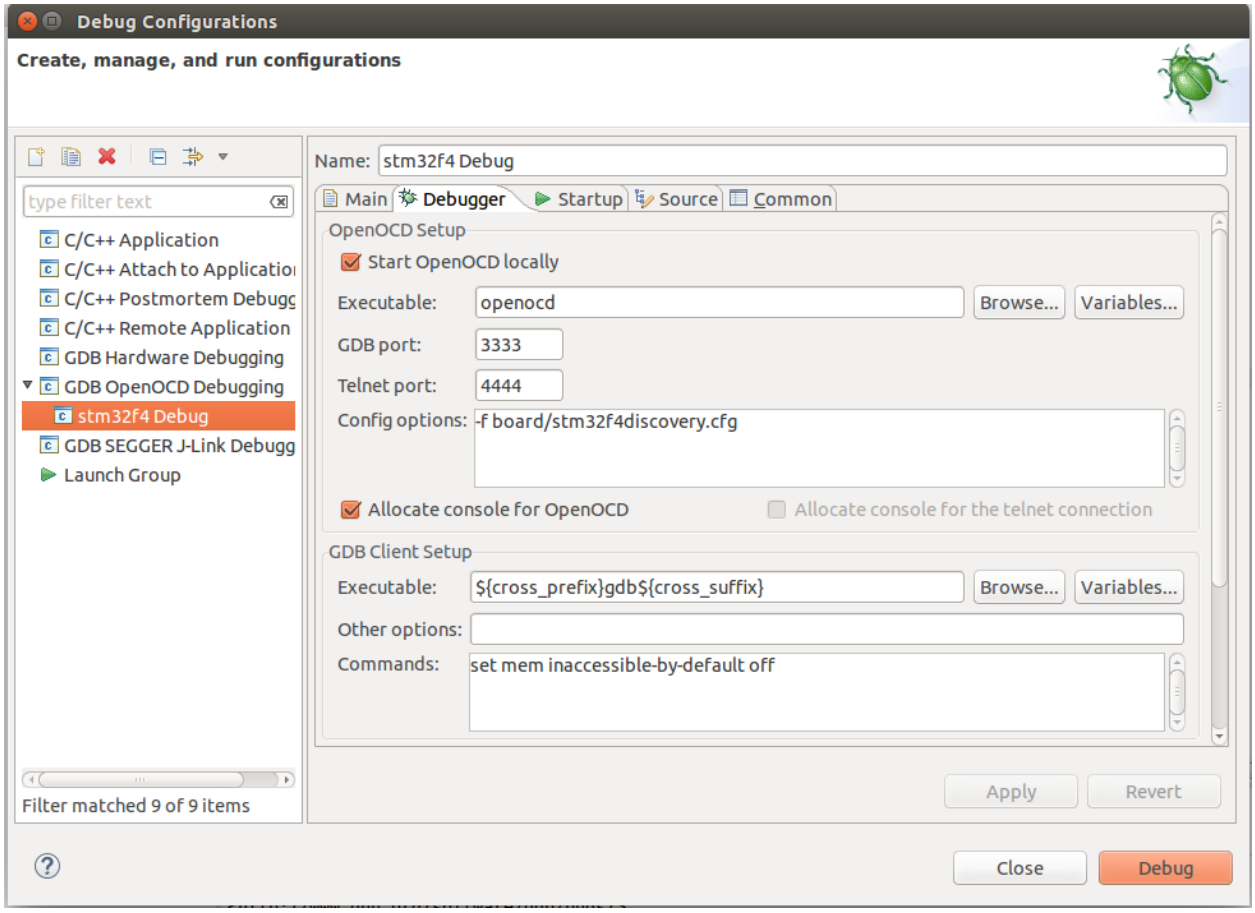
8. 单击工具栏中的锤子按钮来编译我们的工程



9. 通过Console我们可以看到全部的编译链接过程，最终如下图所示，我们成功编译链接了我们的工程



10. 在这里我们选用OpenOCD作为我们的调试服务程序，在Ubuntu 14.04中，OpenOCD已经被接受作为官方的一个软件包，可以通过下述命令来直接安装：
    sudo apt-get install openocd
    安装后的openocd，版本是0.7。如果想使用最新的0.8版本，可以参考OpenOCD官网安装最新版本。
11. 回到Eclipse主界面，在Project Explore栏中，选中我们的工程，单击右键，选择Debug As，接着选择Debug Configurations，我们来到调试配置窗口，双击左侧列表中的GDB OpenOCD Debugging，Eclipse会自动为我们创建一个调试项stm32f4 Debug，选中该项目，接着选中右侧窗口中的Debugger标签，做如下配置：



使用USB线连接STM32F4开发板，然后点击Debug按钮，开始调试我们的工程。单击Debug按钮后，OpenOCD会把工程结果加载到开发板，所以此时也可以立即终止调试，通过重启开发板来直接观察工程运行情况。我们这个示例工程是让板载LED灯持续闪烁，我已经看到了期望的结果，你呢？

除了这些来自开源社区的集成开发环境和组件外，在ARM公司创建的mbed.org网站，你会发现更多更有意思的微控制器开发设备，项目和代码。近些年来开源软件有了长足的发展，得到了越来越多人的认可，甚至已经变成了一种生活方式和理念，很多开源软件的质量已经完全可以媲美或者超越商业软件。本文抛砖引玉，希望大家能够更多更好的利用开源的力量实现自己的梦想，同时也能开源自己的东西，帮助别人，让我们一起努力，让世界变得更加美好!