

第一章 新建 nRF52832 工程（MDK）

前面讲了 4 种开发 nRF52 的环境，我选择使用 MDK 来写教程，源码一样，不要受 IDE 的限制。

1. 本章必备

- 《nRF5 SDK 简介》

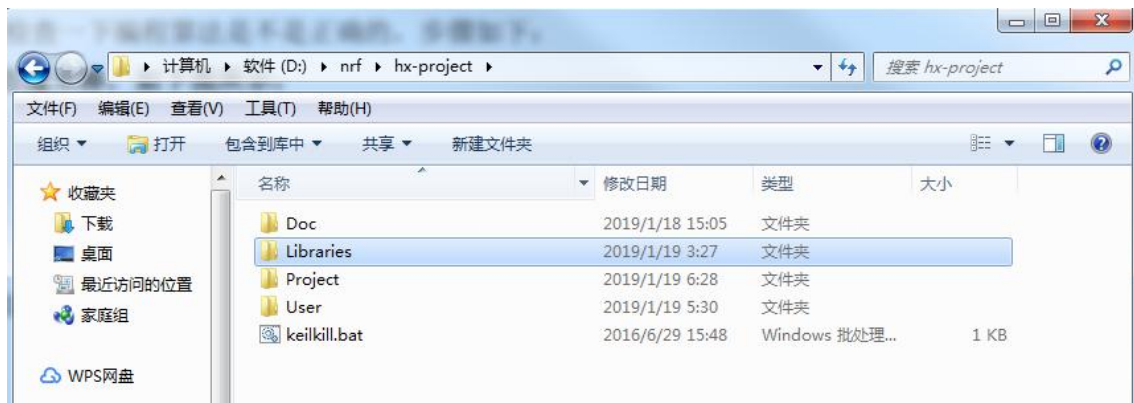
2. 目标

- 掌握 MDK 新建 nRF52 工程，编译和下载到开发板。

3. MDK 新建工程步骤

3.1. 工程结构安排，可根据自己个性修改

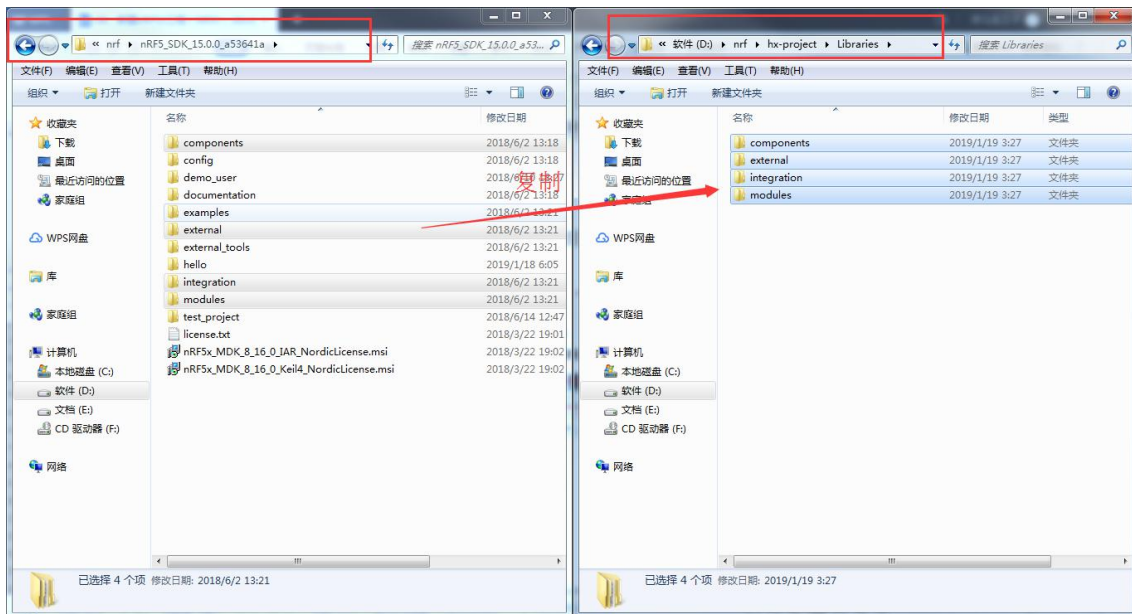
新建文件夹“hx-project”，内部放入 4 个文件夹和 1 个批处理文件（可修改），如下图所示，后期源码结构可能和这个不一样，此处主要用于学习搭建。



文件夹名	功能
Doc	工程说明，版本升级记录，用户来维护
Libraries	存放的是官方库文件
Project	用来存放工程
Project/Output	存放编译产生的调试信息、hex 文件、预览信息、封装库等
USER	用户编写的驱动文件，包括主函数、配置文件等
keilkill.bat	批处理文件，打包程序前使用，清除编译生成的无用文件，方便保存共享

3.2. 复制必须文件到工程的文件夹内

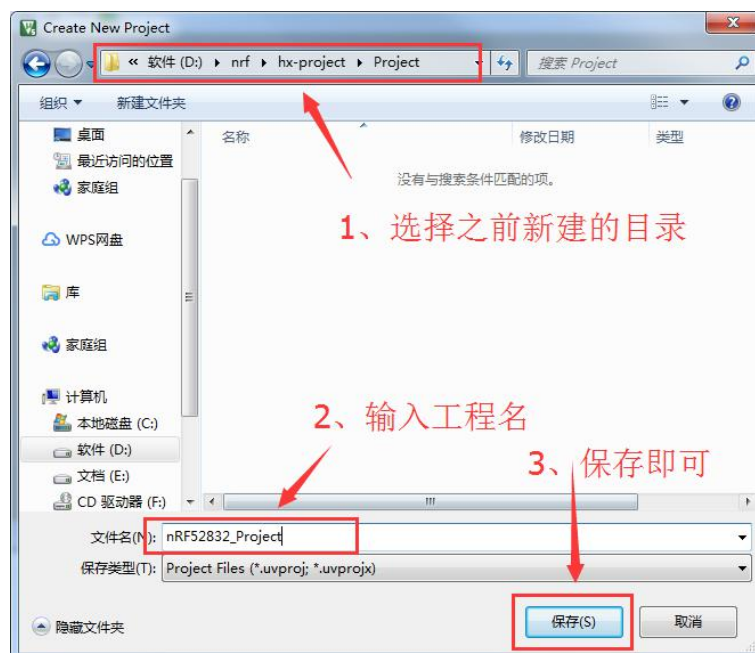
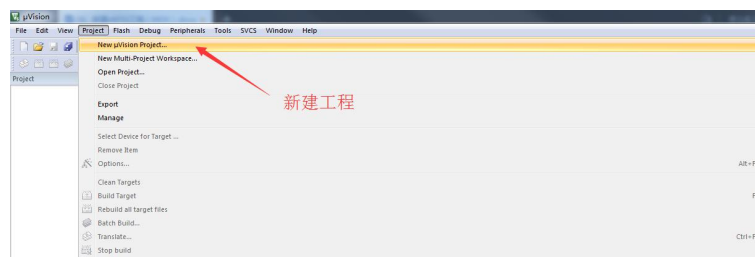
打开 SDK 文件夹，将所需的官方库全部复制到工程的 Libraries 文件夹下，如下图所示：



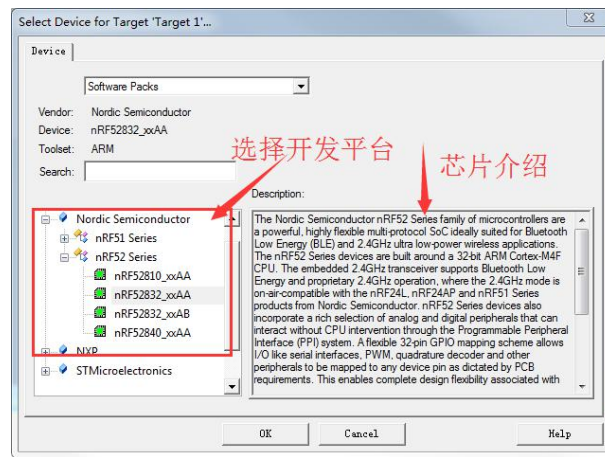
注：官方的库文件较大，一般 200M+，如果觉得受不了，可以裁剪不用的部分，学习阶段不建议裁剪，否则除了问题，一脸懵逼。等进化升级为老鸟后再裁剪不用的部分。

3.3. 新建工程

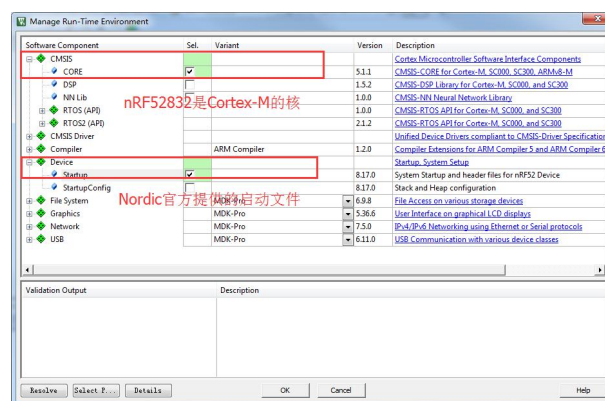
打开 MDK5，新建和保存工程



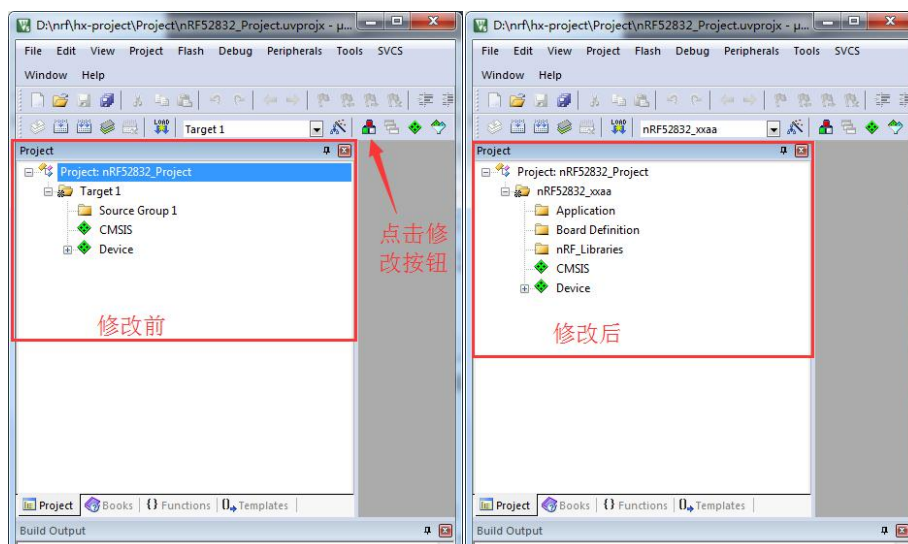
开发平台选择：nRF52832_xxAA



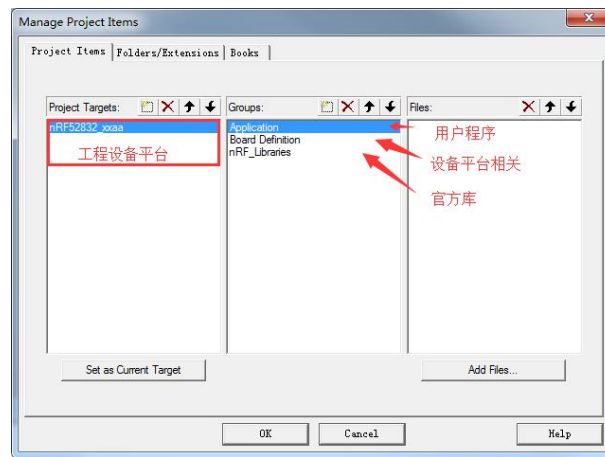
运行环境选择：必须选择 **Cortex_M** 核和官方启动文件，此处只有一个版本可选，后面可修改，不用紧张。



工程目录搭建：参考官方 SDK 例程的目录结构。

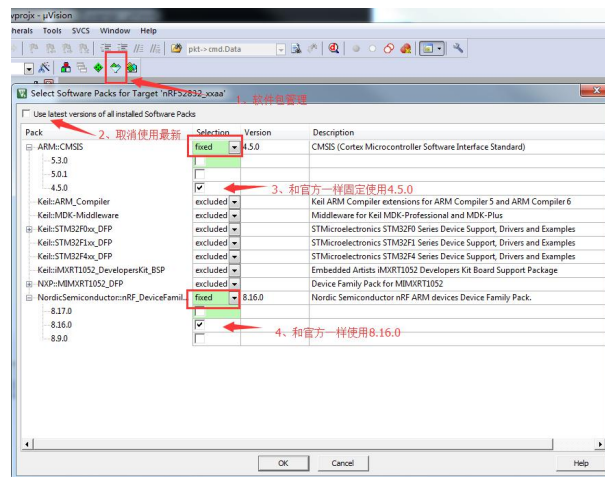


工程目录修改过程，随便点点摸索。

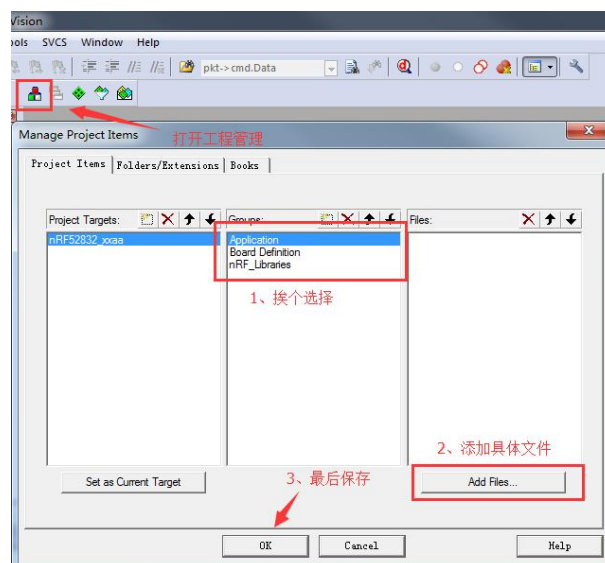


Cortex-M 核和 Device 启动的软件包管理，如下所示：

自己的工程最好把**自动使用最新**的软件包关掉，并 **Fixed** 固定某个版本。谁也不想，做好的东西，软件包更新后，编译不通过。

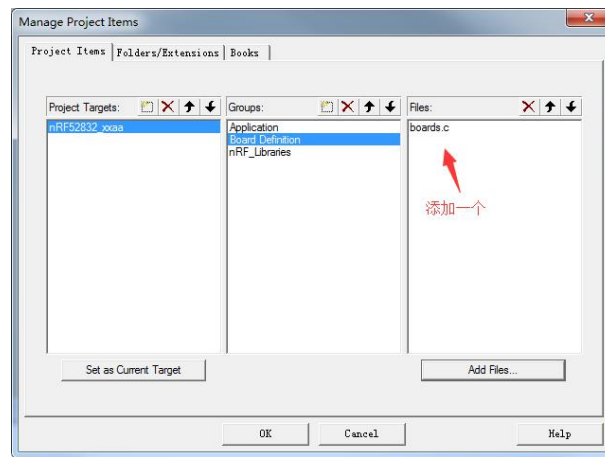


向工程文件夹中添加具体文件的方法



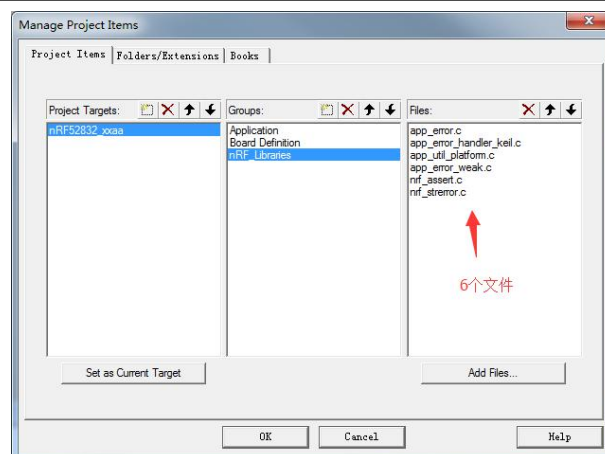
添加板级支持文件

boards.c	D:\nrf\hx-project\Libraries\components\boards
----------	---



添加错误处理文件

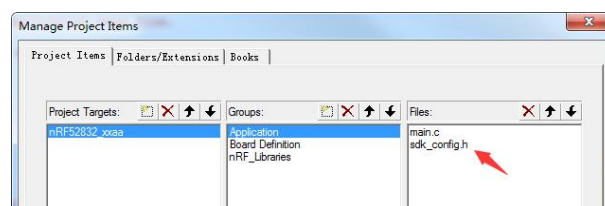
app_error.c	hx-project\Libraries\components\libraries\util
app_error_handler_keil.c	hx-project\Libraries\components\libraries\util
app_error_weak.c	hx-project\Libraries\components\libraries\util
app_util_platform.c	hx-project\Libraries\components\libraries\util
nrf_assert.c	hx-project\Libraries\components\libraries\util
nrf_strerror.c	hx-project\Libraries\components\libraries\strerror



复制 SDK 中的 Blink 工程的 main.c 和 sdk_config.h 文件,放在新建工程的 User 文件夹内



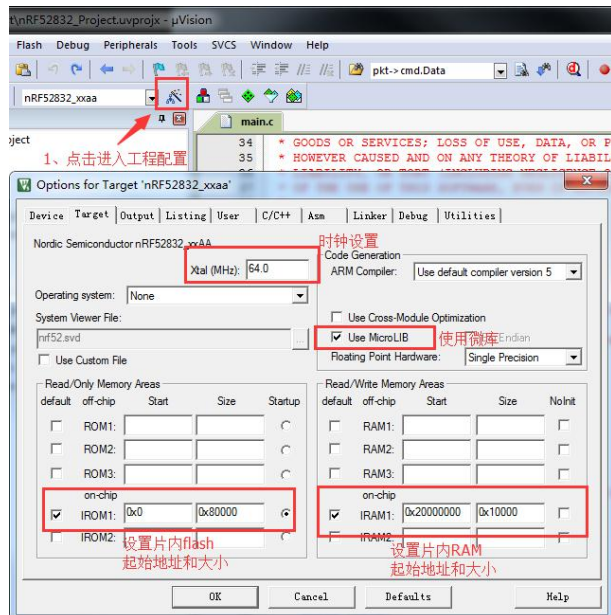
添加 main.c 和 sdk_config.h 到工程



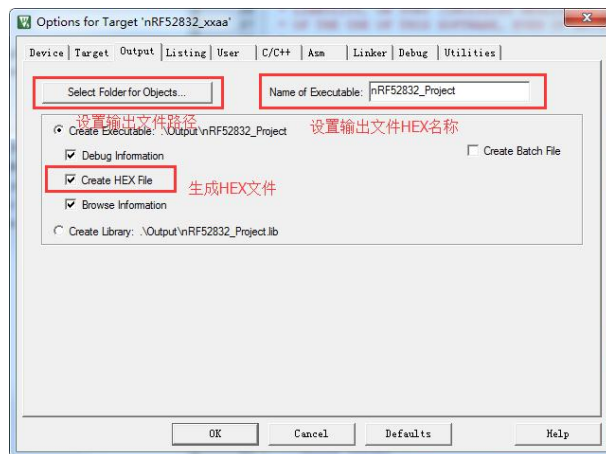
3.4. 配置工程

目标平台设置，以下是 nRF52832 参数，由硬件决定了。

平台	Flash 大小	RAM 大小
nRF52832	512K (0x80000)	64K (0x10000)



输出文件目录为 hx-project\Project\Output

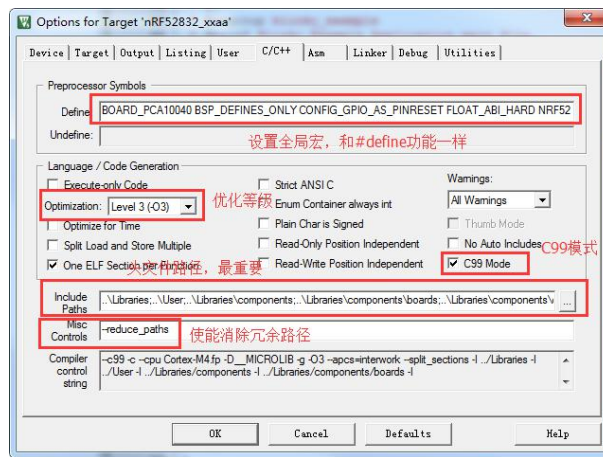


C/C++设置，大部分目录文件找不到都是这个设置的问题。参考官方设置，复制过来。

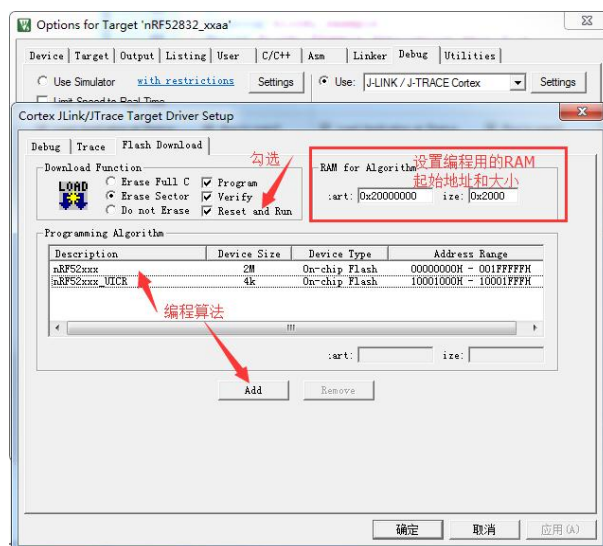
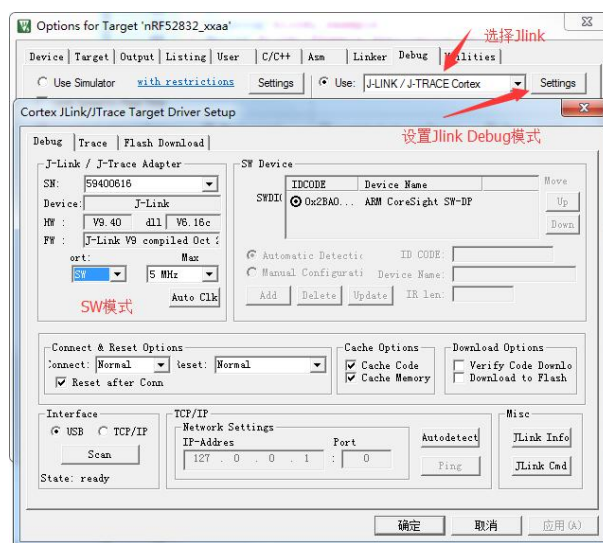
宏介绍

宏介绍	功能
BOARD_PCA10040	官方 52832 开发板，板载文件加载，会包含“pca10040.h”
BSP_DEFINES_ONLY	BSP 使用开启
CONFIG_GPIO_AS_PINRESET	RST 引脚配置宏
__HEAP_SIZE=8192	堆大小
__STACK_SIZE=8192	栈大小

优化等级：0 不优化，值越大，优化等级越高，优化越高编译后的代码越小。



Debug 设置，前提接上硬件，安装成功 Jlink 驱动。参考前章安装。

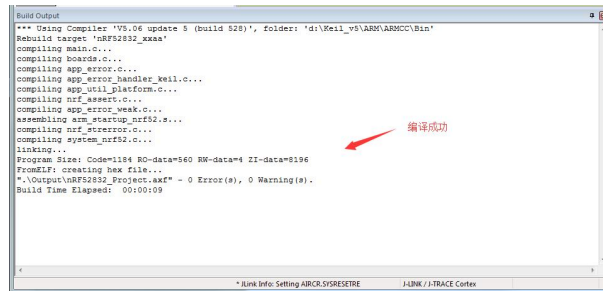


3.5. 编译工程



三个编译按钮分别是：

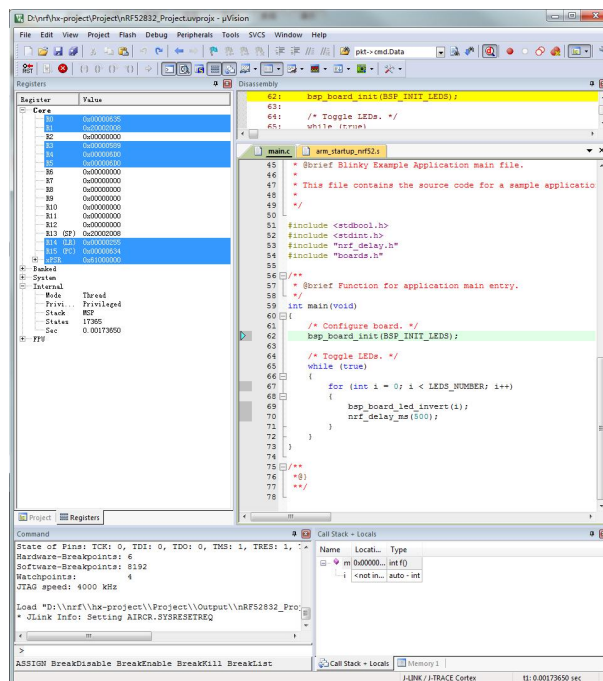
- Translate (按钮 1) : 编译当前打开的文件
- Build (按钮 2) : 增量编译, 编译整个工程中修改的文件及其依赖文件, 生成可执行程序。
- Rebuild (按钮 3) : 全部编译



3.6. 下载程序 流水灯测试 ok。



3.7. Debug 调试



4. MDK 新建工程总结

- 使用过 MDK 搭建过环境的人觉得千篇一律的简单, 没搭建过得人仔细观察即可
- 搭建过程中有问题, 多看几遍, 再交流。
- 我们教程的源码将裁剪官方库, 以便于传输分享。
- 源码: <https://github.com/HX-IoT>