

IAR 简介

IAR 公司是一家全球领先的嵌入式系统开发工具和服务供应商，1983 年成立于瑞典，主要开发 C/C++ 编译器和仿真调试工具等产品。IAR 的 Embedded Workbench 系列适用于开发基于 8 位、16 位以及 32 位微处理器的嵌入式系统

IAR 几乎支持所有常见的内核

The development tools support the following targets: 78K, 8051, ARM, AVR, AVR32, CR16C, Coldfire, H8, HCS12, M16C, M32C, MSP430, Maxim MAXQ, R32C, R8C, RH850, RL78, RX, S08, SAM8, STM8, SuperH, V850, ARM(ARM7 / ARM9 / ARM10 / ARM11 / Cortex M0 / M0+ / M1 / M3 / M4 / M7 / M23 / M33, Cortex R4 / R5 / R7 / Cortex A5 / A7 / A8 / A9 / A15 / A17).

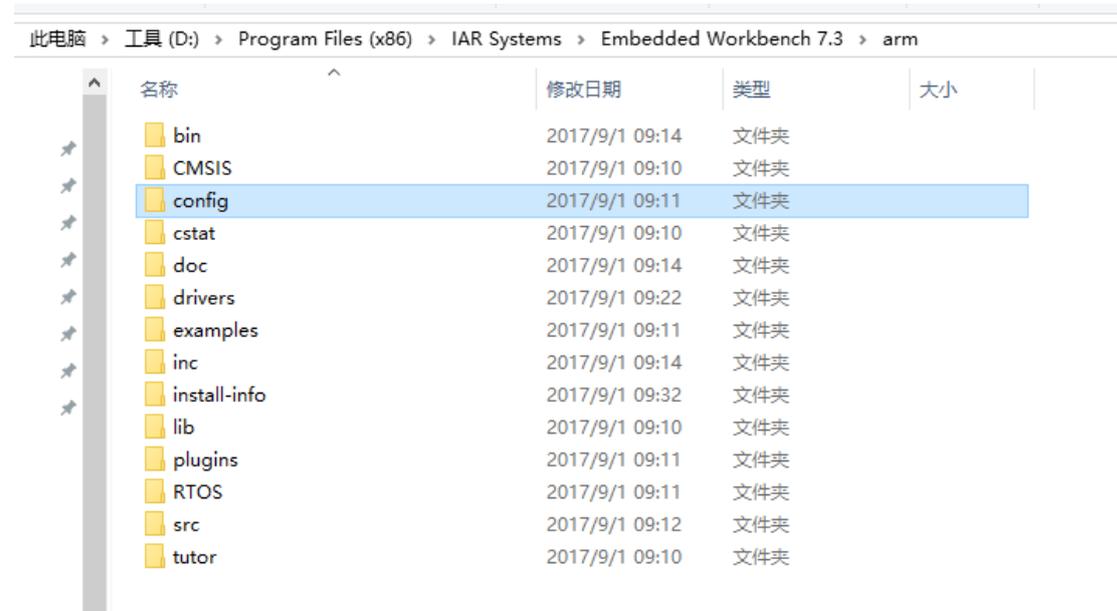
与我们常用的 Keil 相比，IAR 有如下区别：

- 芯片启动文件有所不同
- IAR 不支持 ULINK2，需要使用 Jlink 下载和仿真
- IAR 不同版本间兼容性较差，容易丢失配置

IAR 对 FM33A0 系列芯片的支持

FM33A0 系列芯片使用的是 ARM 的 Cortex-M0 内核，通过简单的配置即可让 IAR 支持 FM33A0 系列芯片，具体方法如下，具体见（IAR 安装复旦微 ARM 芯片插件说明）：

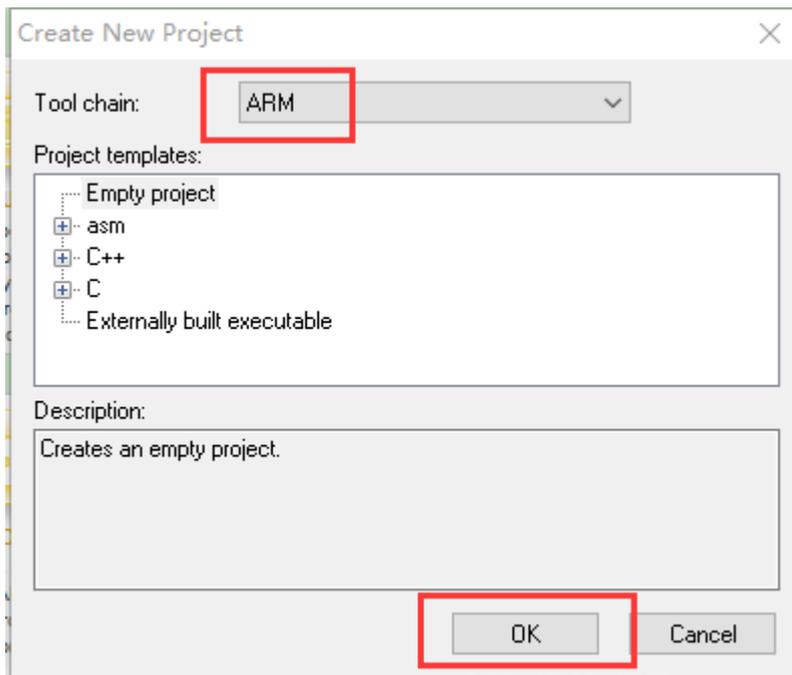
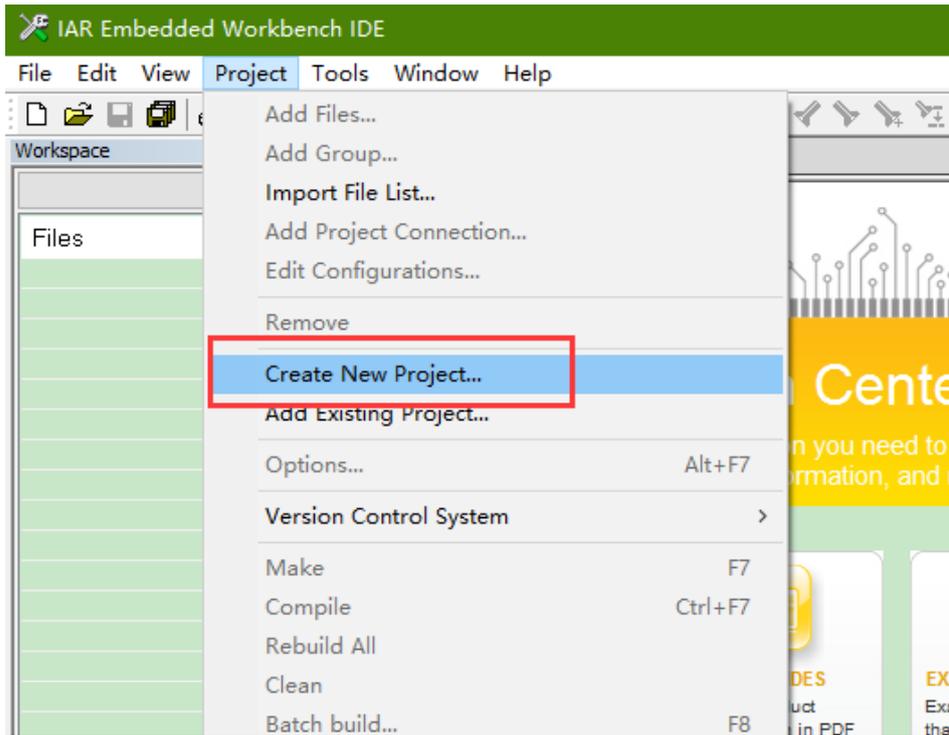
将复旦微提供的 Config 文件夹下的文件拷贝到 IAR 的安装路径 “IAR Systems\Embedded Workbench 7.3\arm\config” 中，即可让 IAR 支持 FM33A0 系列芯片。

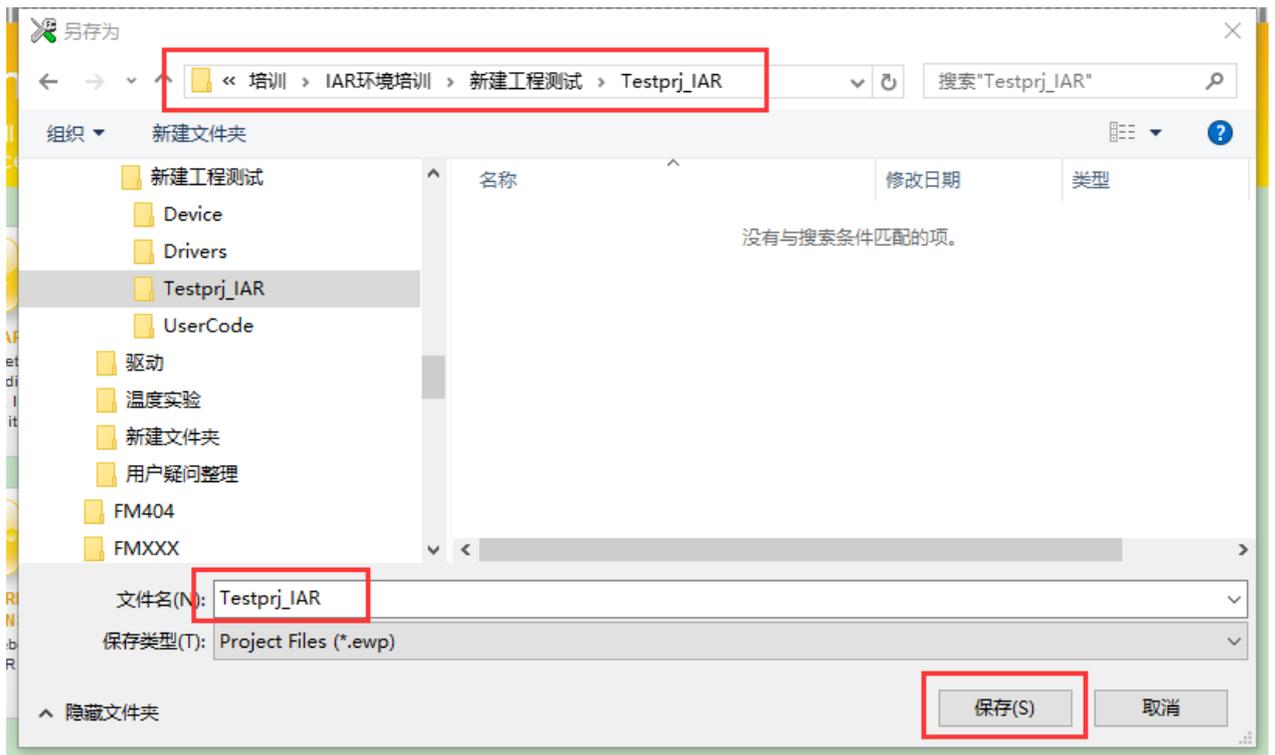


IAR 使用简介

配置完 IAR 后即可建立 FM33A0XX 芯片的工程，具体过程如下：

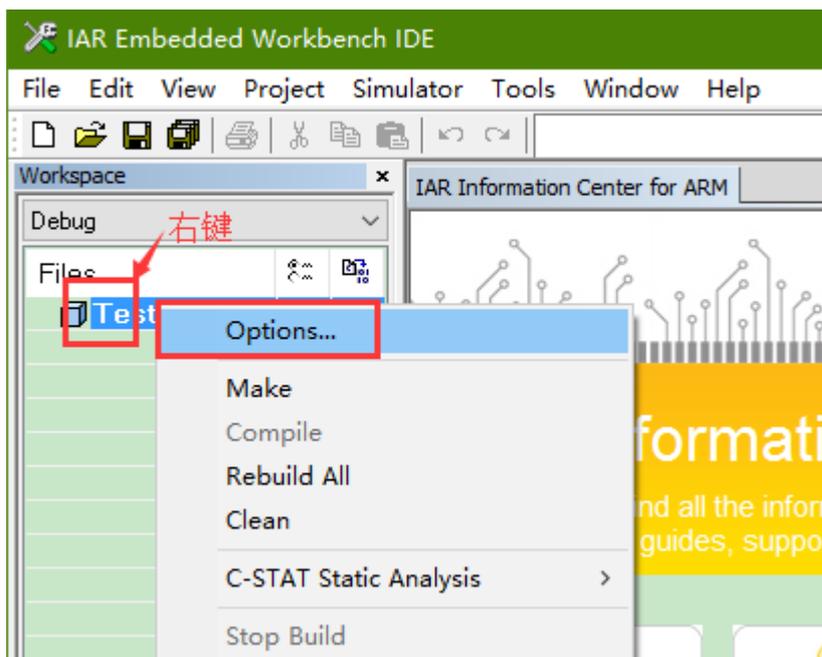
1. 建立一个基于 ARM 的新工程



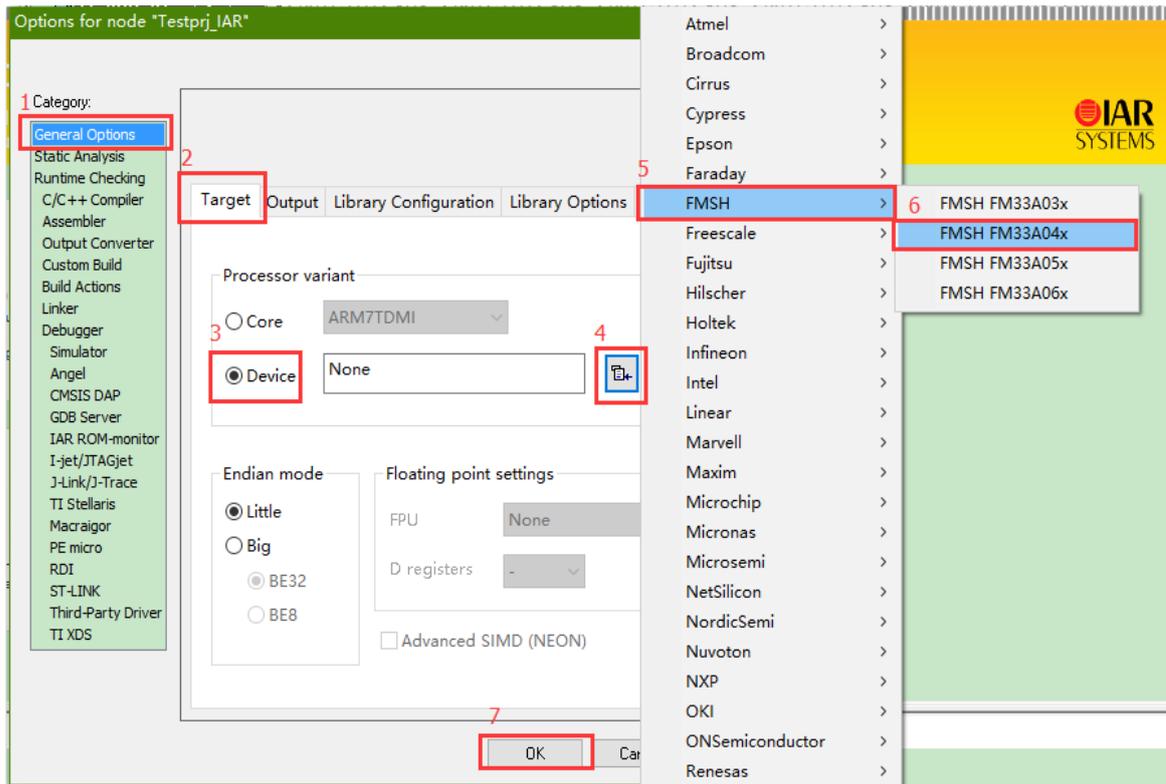


2.配置工程

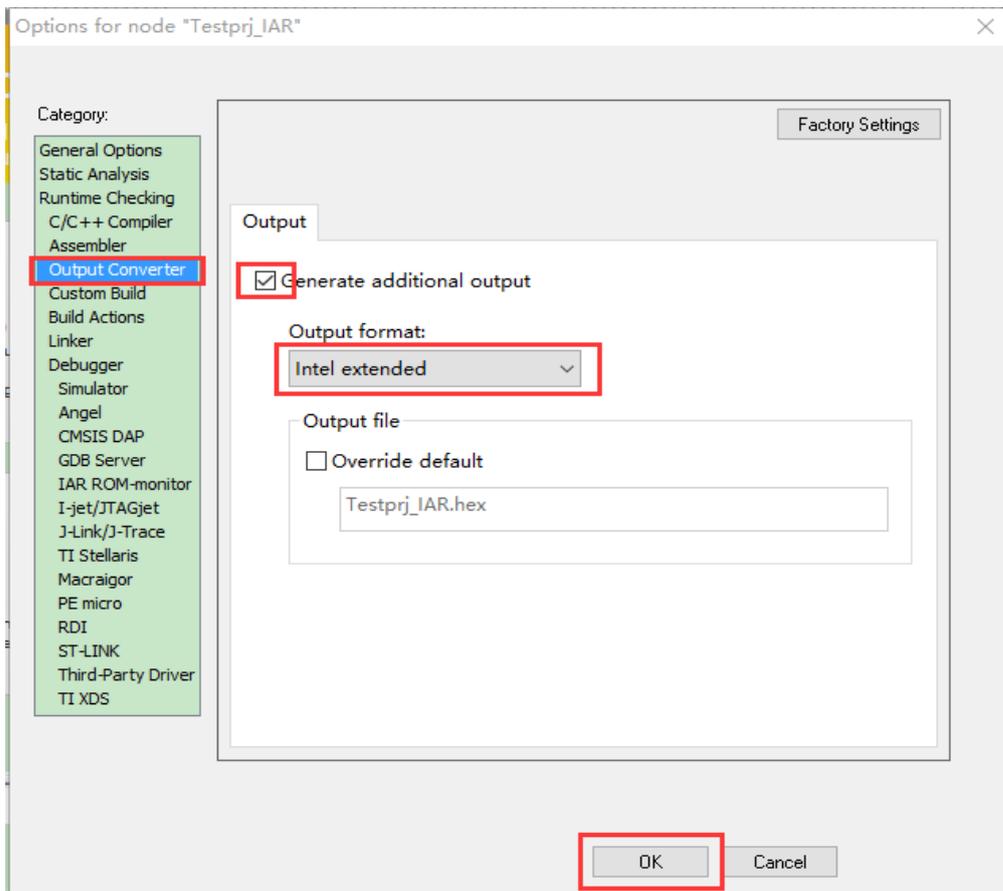
一般情况下使用默认配置即可，但是不同版本间的 IAR 打开工程可能会丢失配置，导致不能编译或者不能下载程序，如果出现异常不能下载或者仿真，可以尝试将工程重新配置一遍。



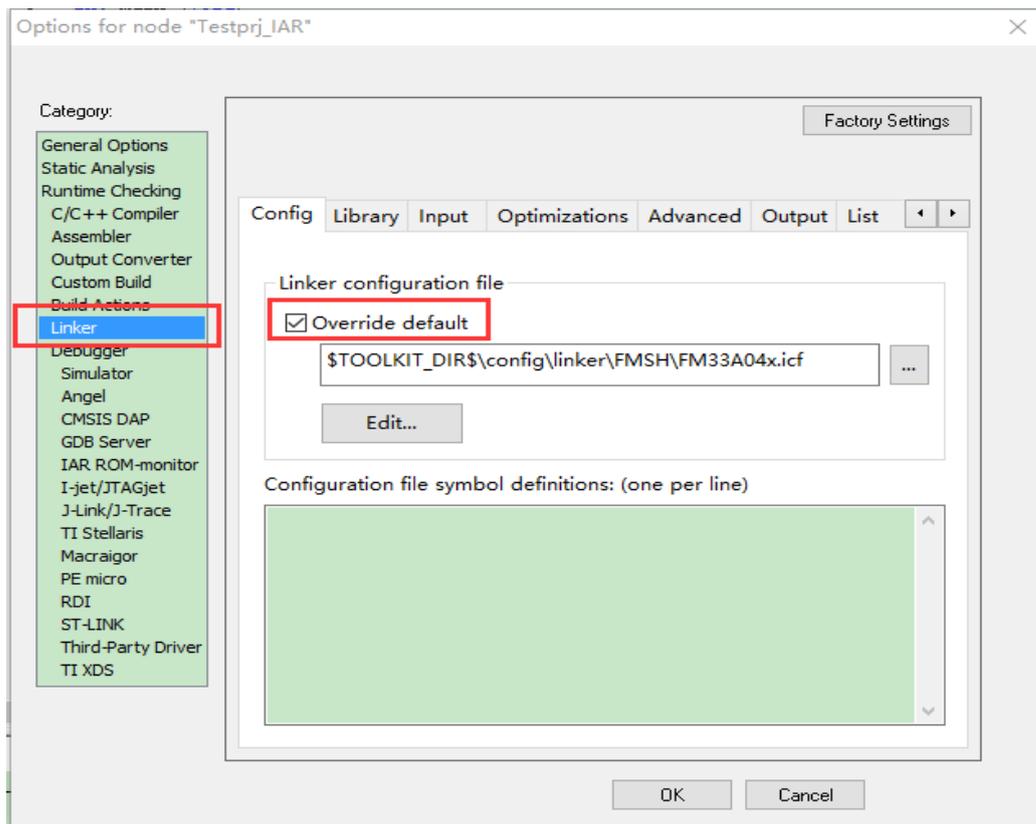
选择芯片型号



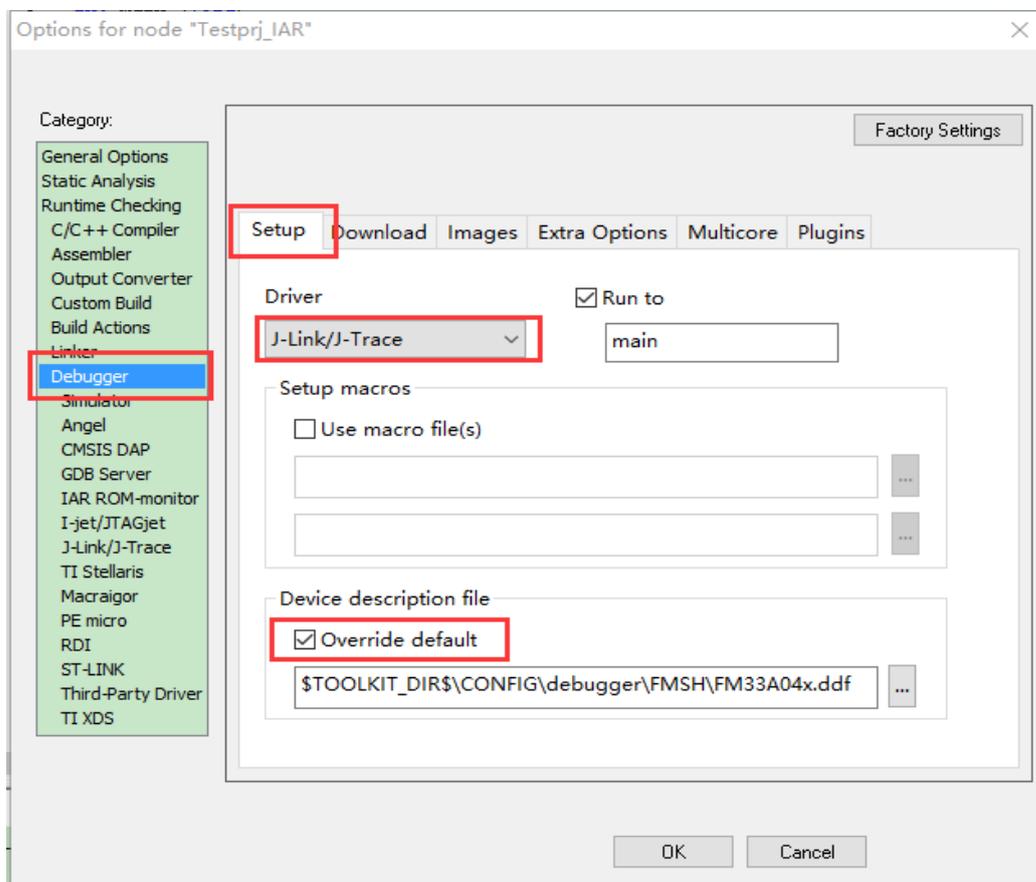
配置输出 HEX 文件



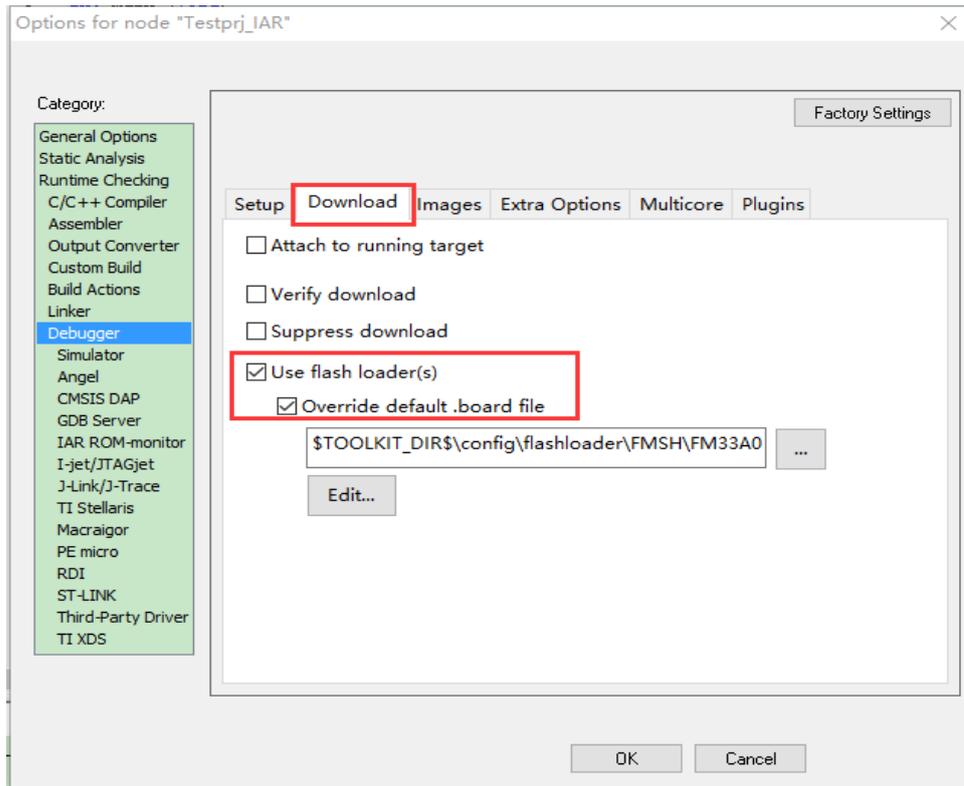
连接控制文件配置



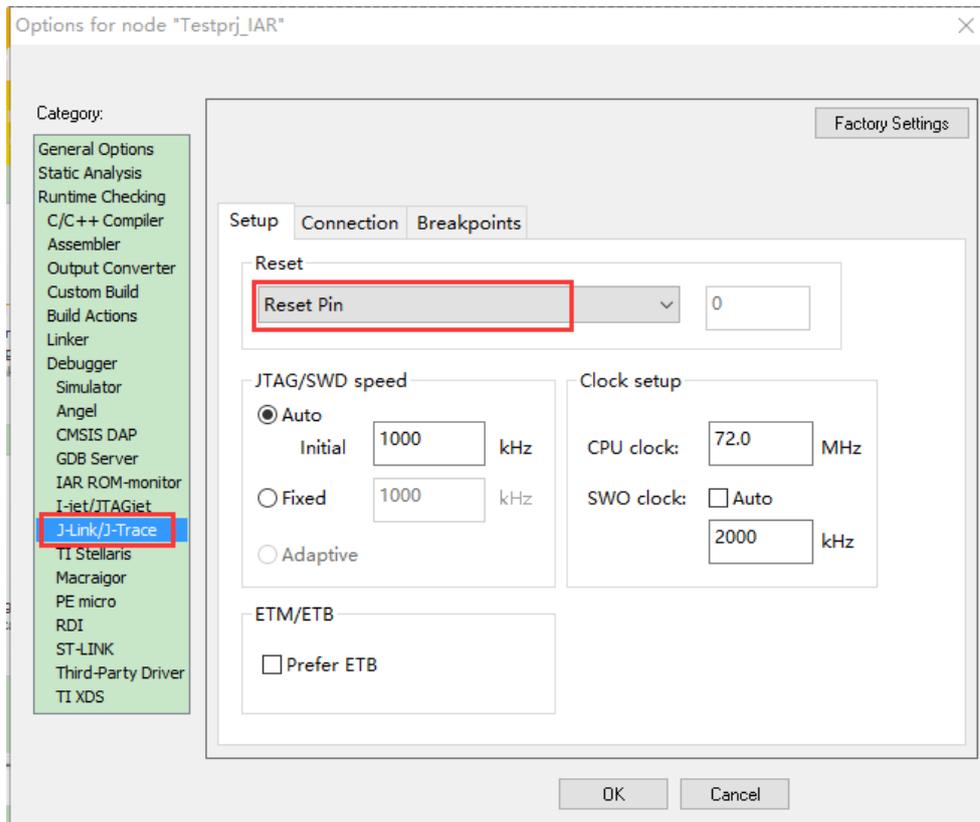
配置仿真器



配置下载驱动

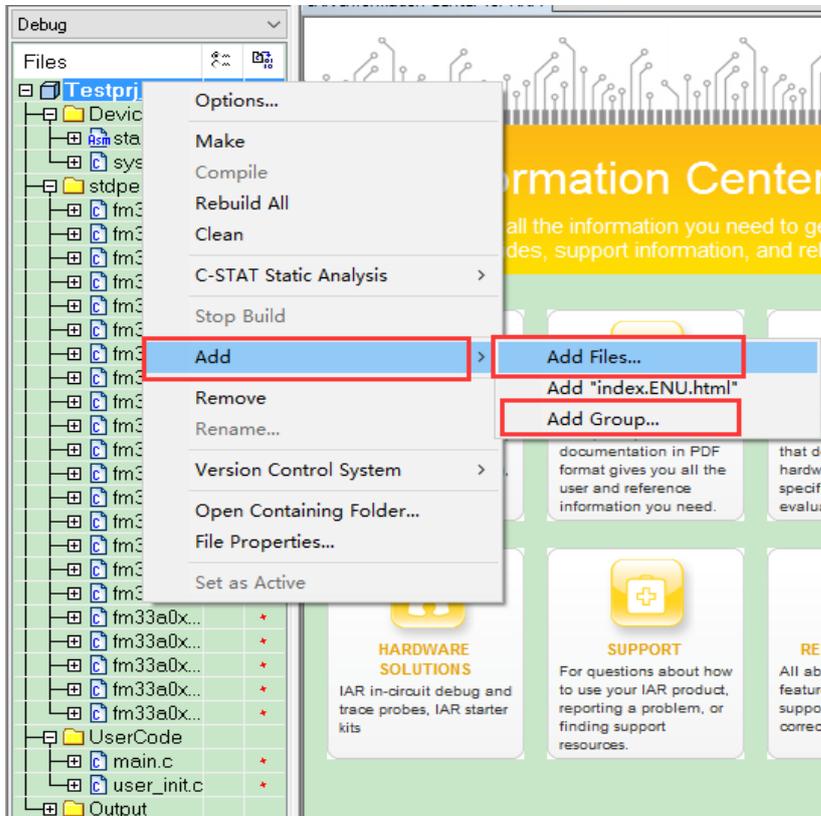


仿真器设置，仅需选择 Reset Pin 即可

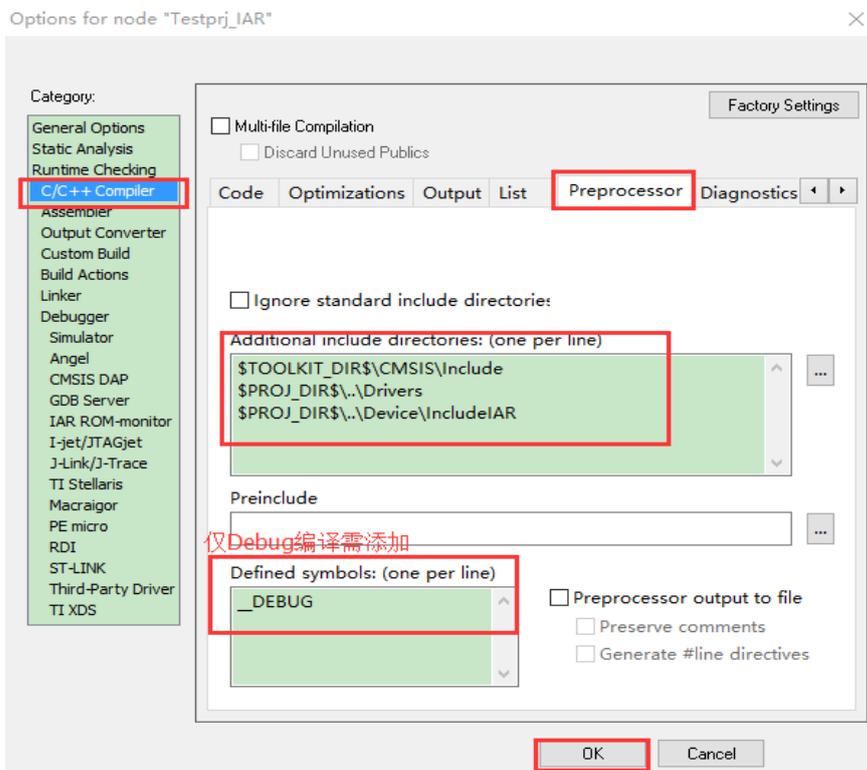


3. 添加代码文件

添加程序文件



添加头文件路径



4. 编译



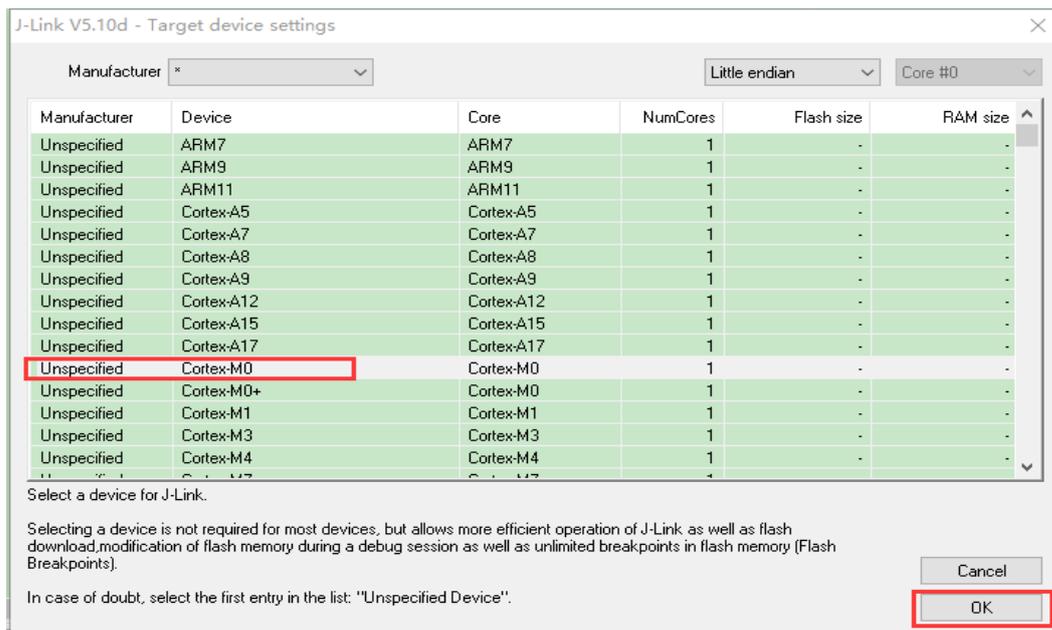
Make:编译, 连接当前工程 (常用)。(编译只编译有改动文件, 或者设置变动的文件, 工程窗口文件右边会有个*号)

compile: 只编译当前源文件。 (不管文件是否改动, 或者设置是否变动)

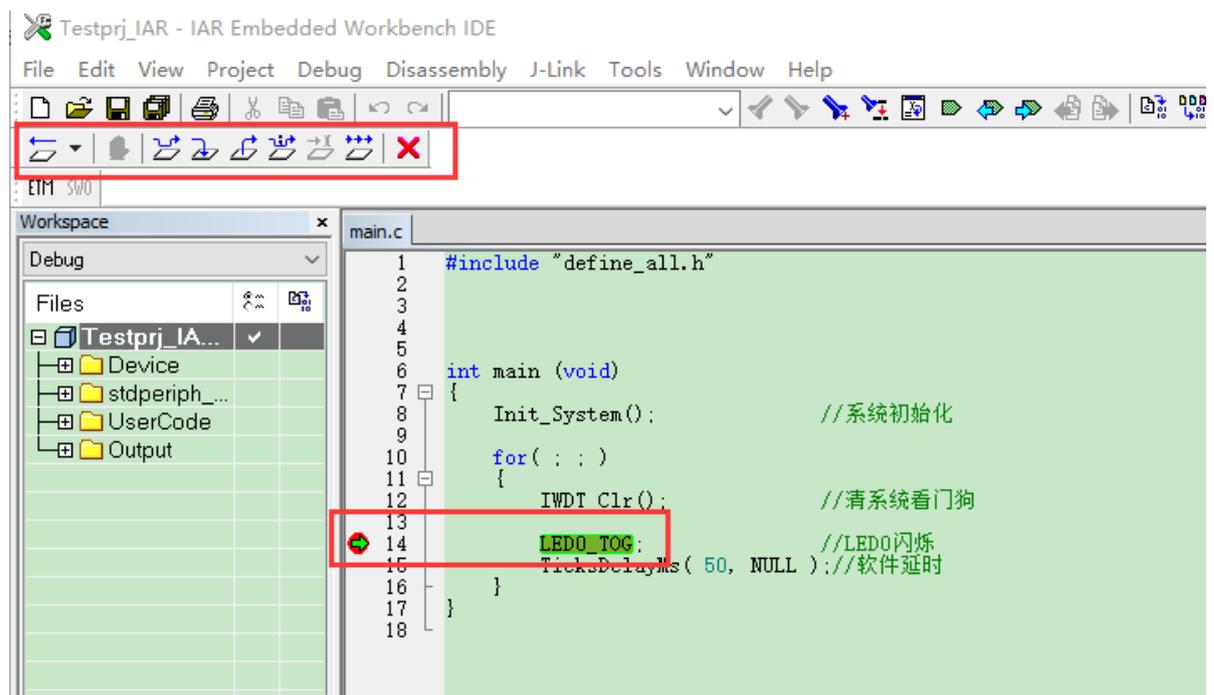
rebuild all: 编译, 连接当前工程。 (不管文件是否改动, 或者设置是否变动)

5 下载与仿真

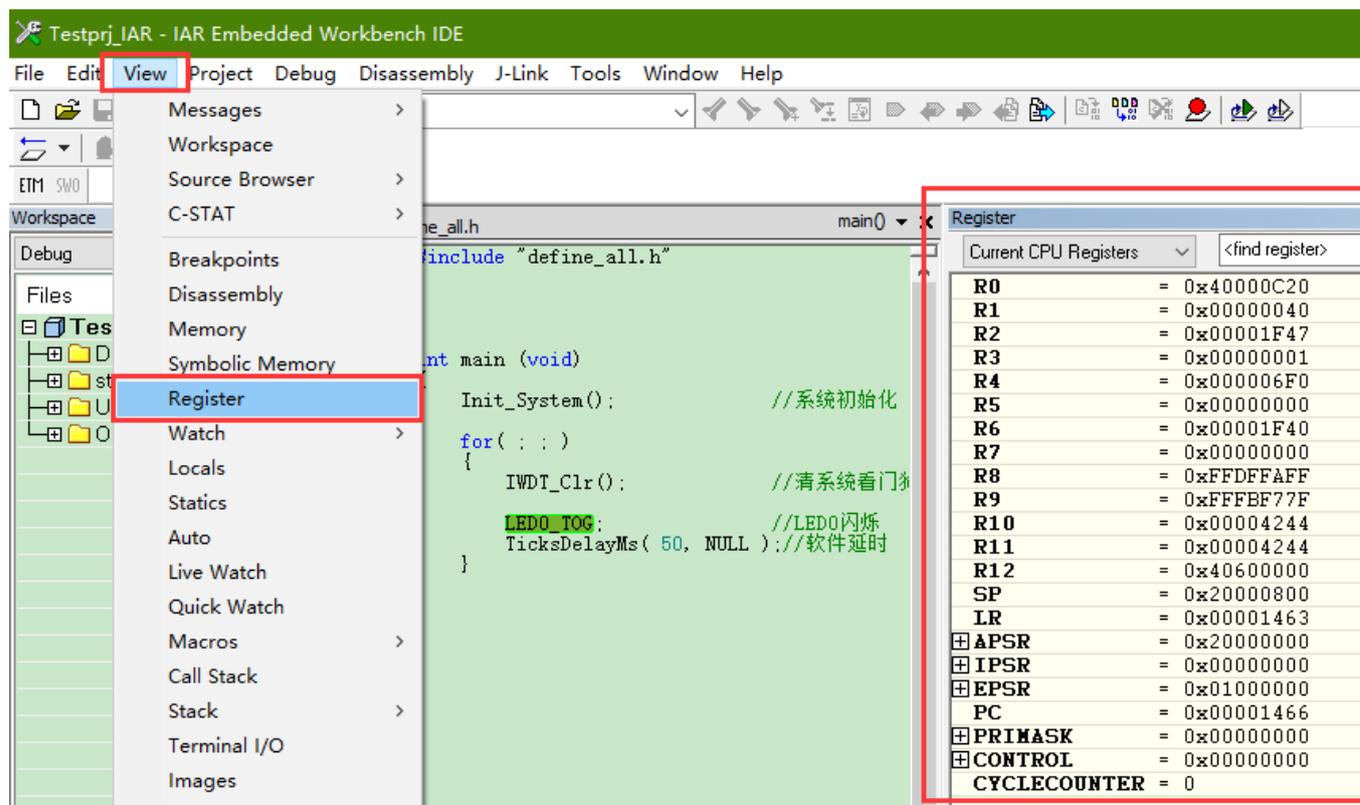
由于 Jlink 驱动中没有 FM33A0 系列芯片的信息, 新工程第一次下载会弹出如下提示, 点击 OK 后手动选择 Cortex-M0 内核并确定即可正常下载与仿真。



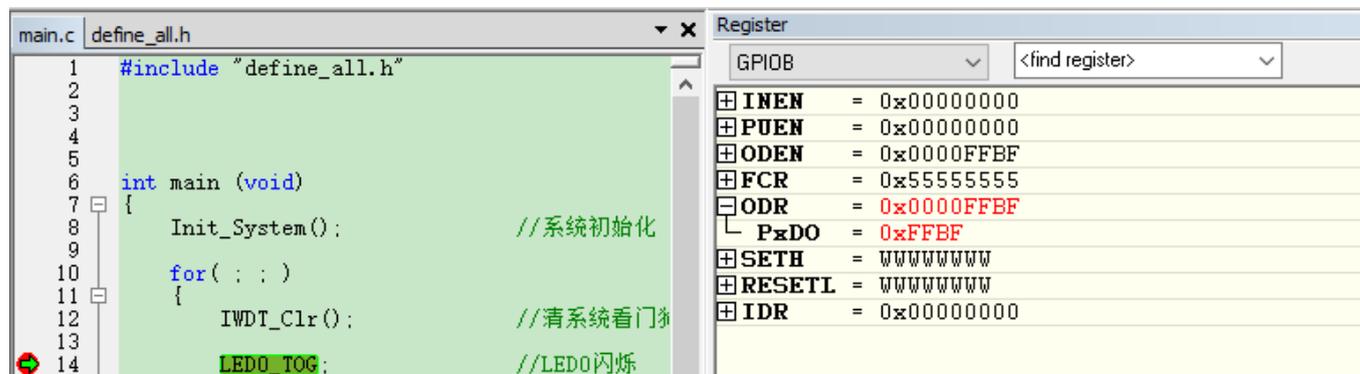
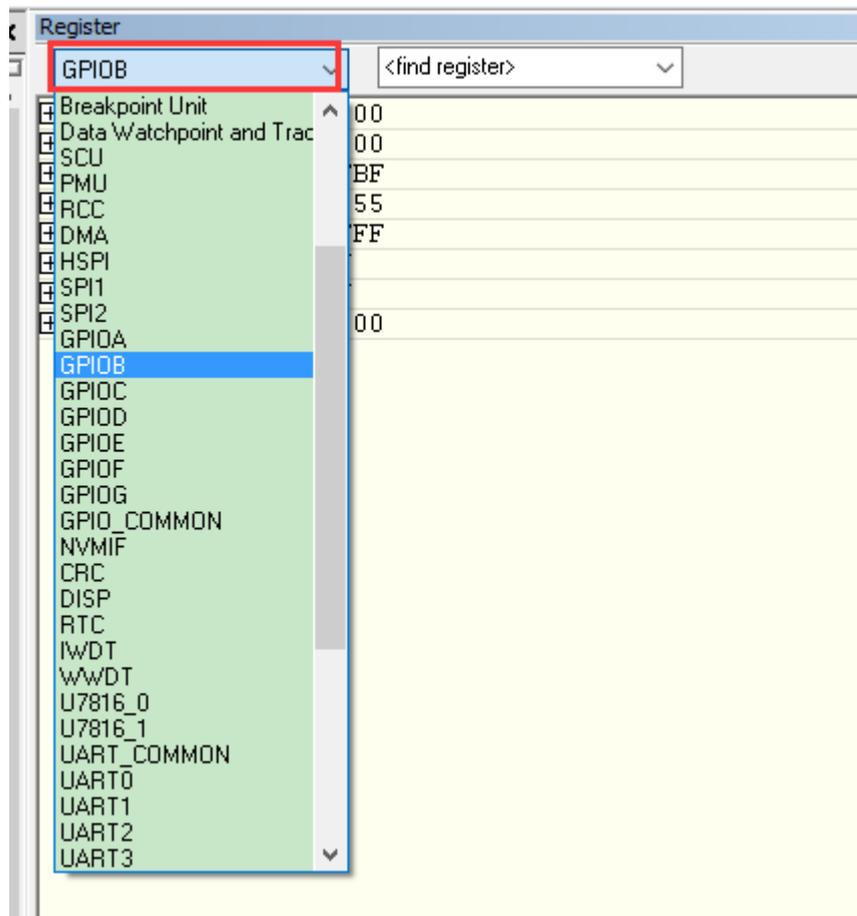
进入仿真界面后可在快捷操作区域选择全速运行，单步运行，复位等操作。



此外，还可通过 View->Register 打开寄存器观察窗观察和修改寄存器



寄存器观察窗还可以观察和修改外设的寄存器，可通过下拉菜单选择需要调试的外设模块，如下图所示。



6. 仿真注意

在使用 Register 窗口观察外设寄存器时，每次单步运行，仿真器都会将被观察模块的寄存器数据全部读回刷新一次，这样会导致部分读后清零的标志，如 uart 的接收中断标志位，spi 的接收中断标志位都会被清掉，导致程序运行不正常，在调试外设时需要注意。