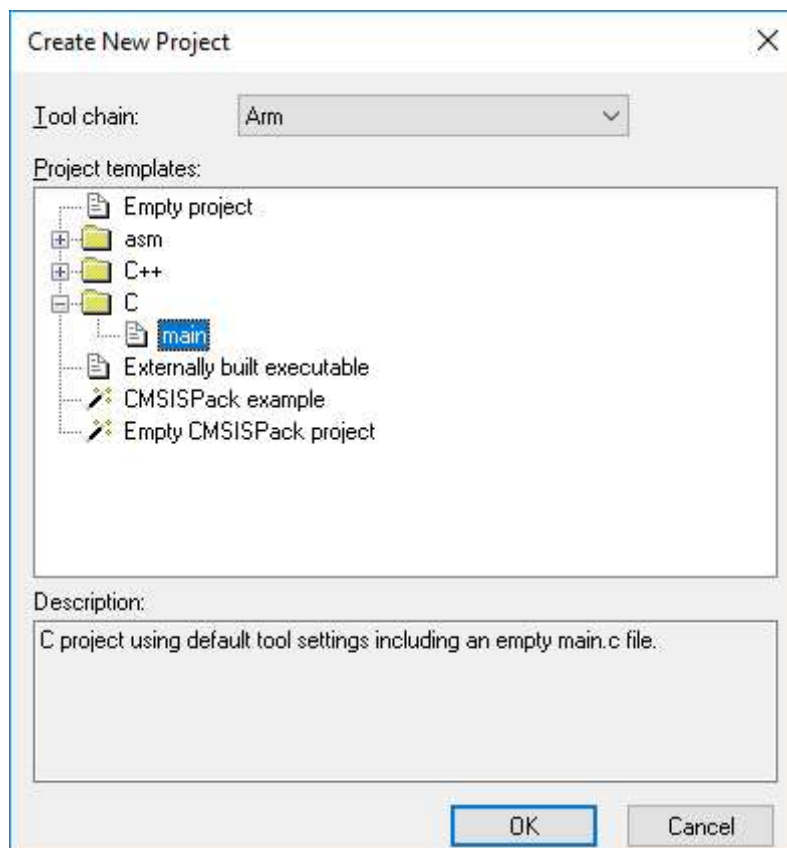


IAR Embedded Workbench

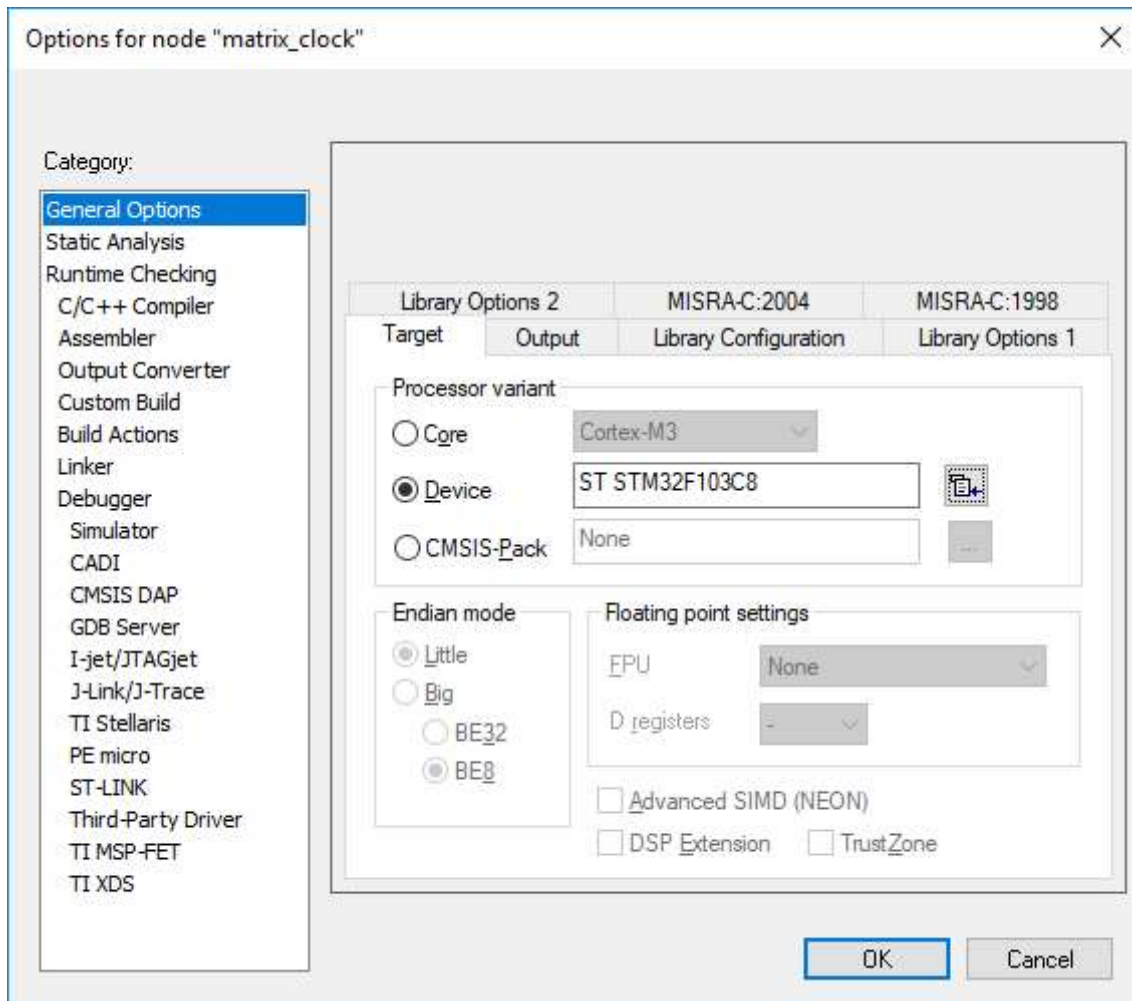
Другая достаточно популярная среда называется IAR Embedded Workbench. Она проприетарна и требует лицензирования (в случае если ваш код превышает 32 Кб), однако её вполне хватает для домашнего использования (kickstart-лицензия). Среда на много скуднее TrueStudio, но по ощущениям бинарный файл получается меньше (ввиду своей реализации компилятора). Для полноты картины рассмотрим процесс настройки данной среды.

Во-первых, нужно создать рабочее пространство. Для этого перейдите в меню **File** ⇒ **New Workplace**. Далее создайте проект, меню **Project** ⇒ **Create New Project...** и выберите C-проект.

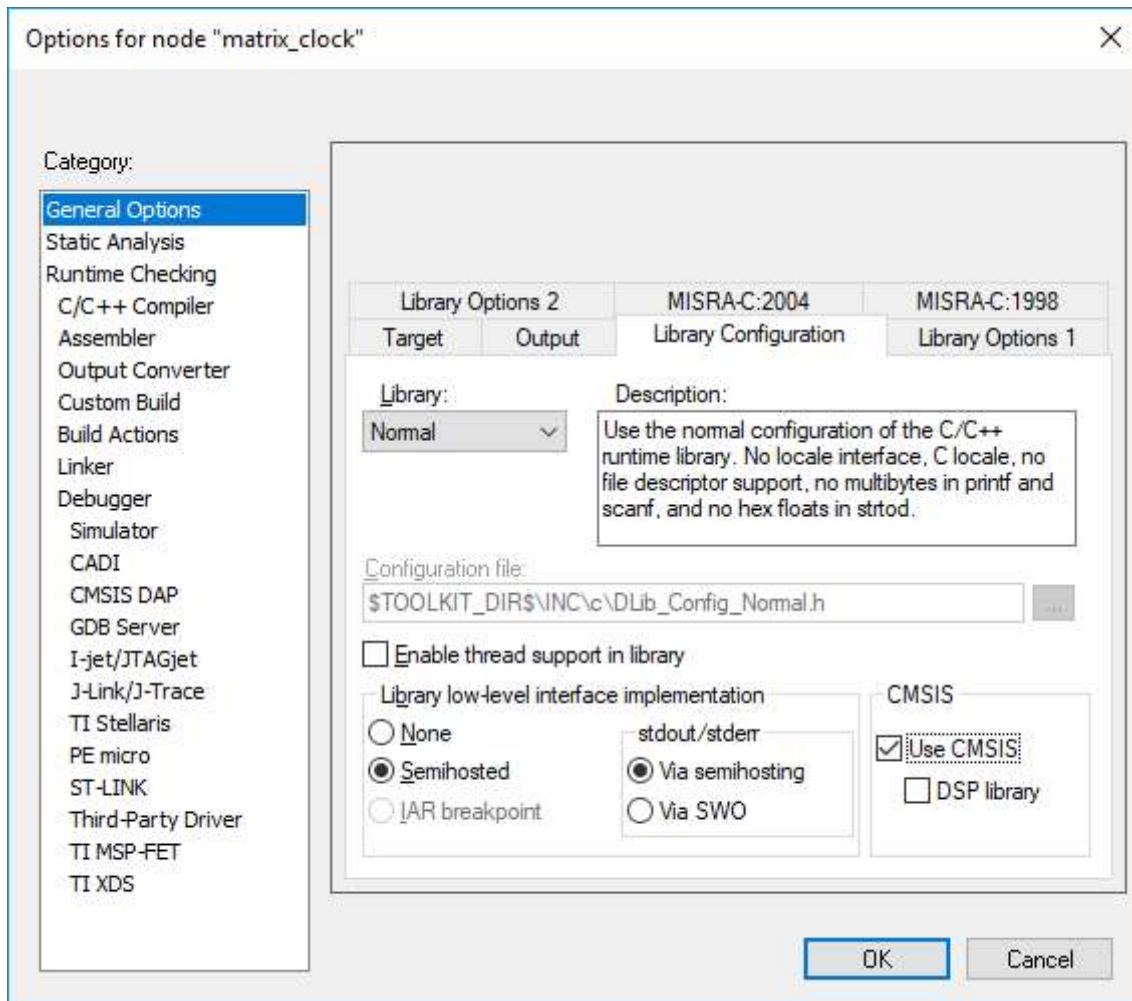


Далее нужно настроить новосозданный проект. Щёлкните правой кнопкой мыши по проекту в менеджере проектов или перейдите в меню **Project** ⇒ **Options...**

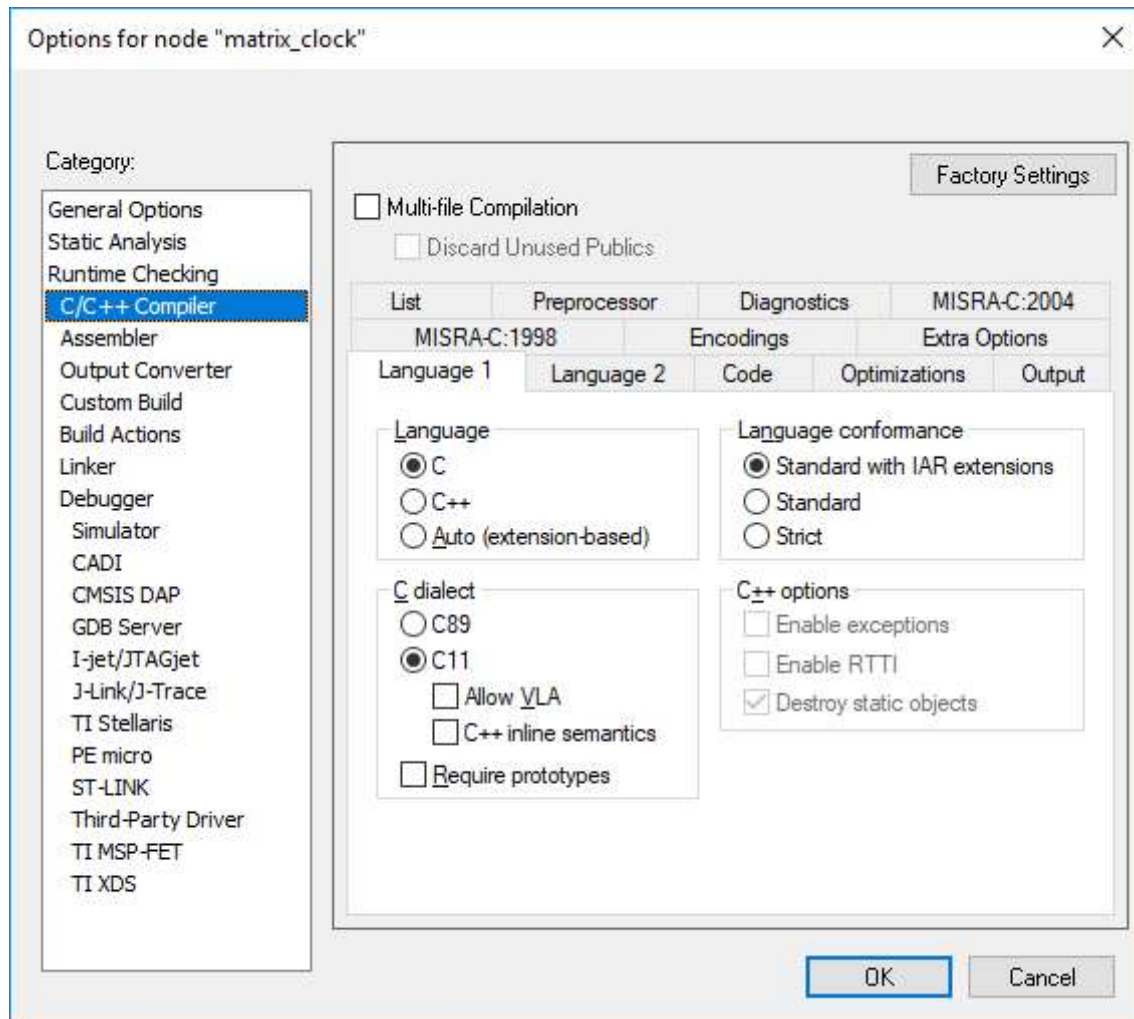
Во вкладке **Target (General Options)** укажите целевой микроконтроллер, в нашем случае это **ST STM32F03C8**.



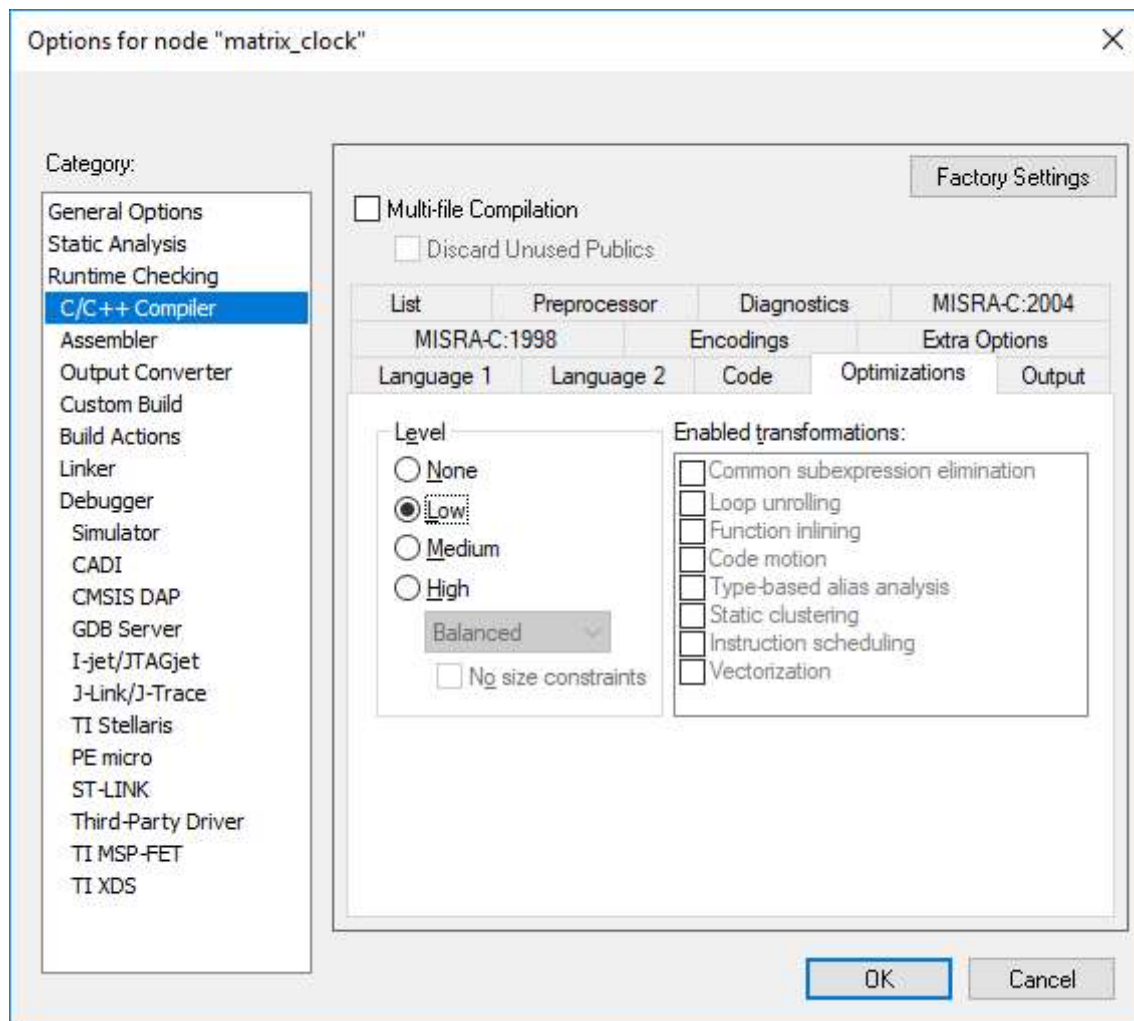
Во вкладке **Library Configuration** поставьте галочку **CMSIS**. IAR автоматически подключит библиотеку (вендернезависимую часть) к проекту.



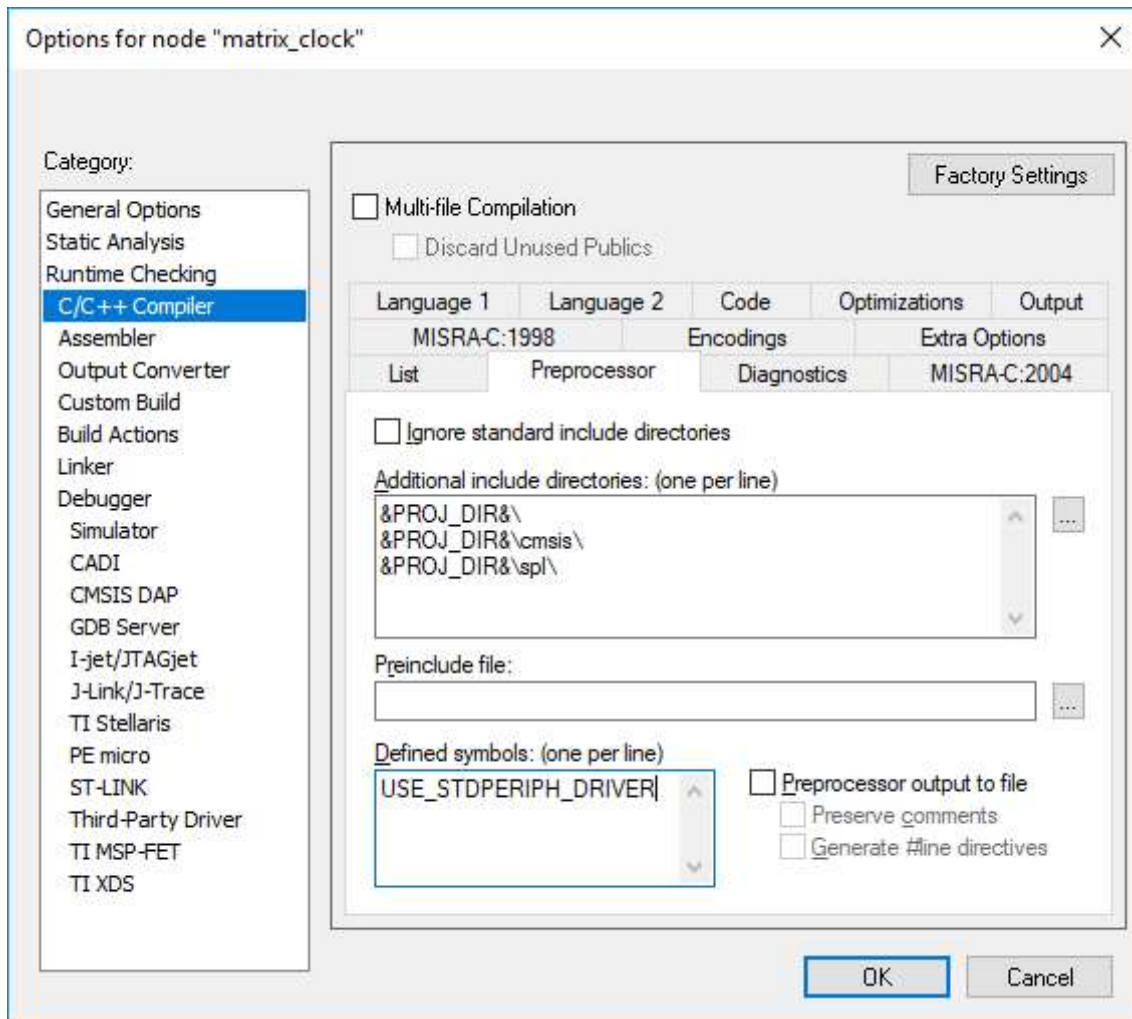
Во вкладке **Language 1 (C/C++ Compiler)** можно выбрать диалект языка, оставьте `c11`.



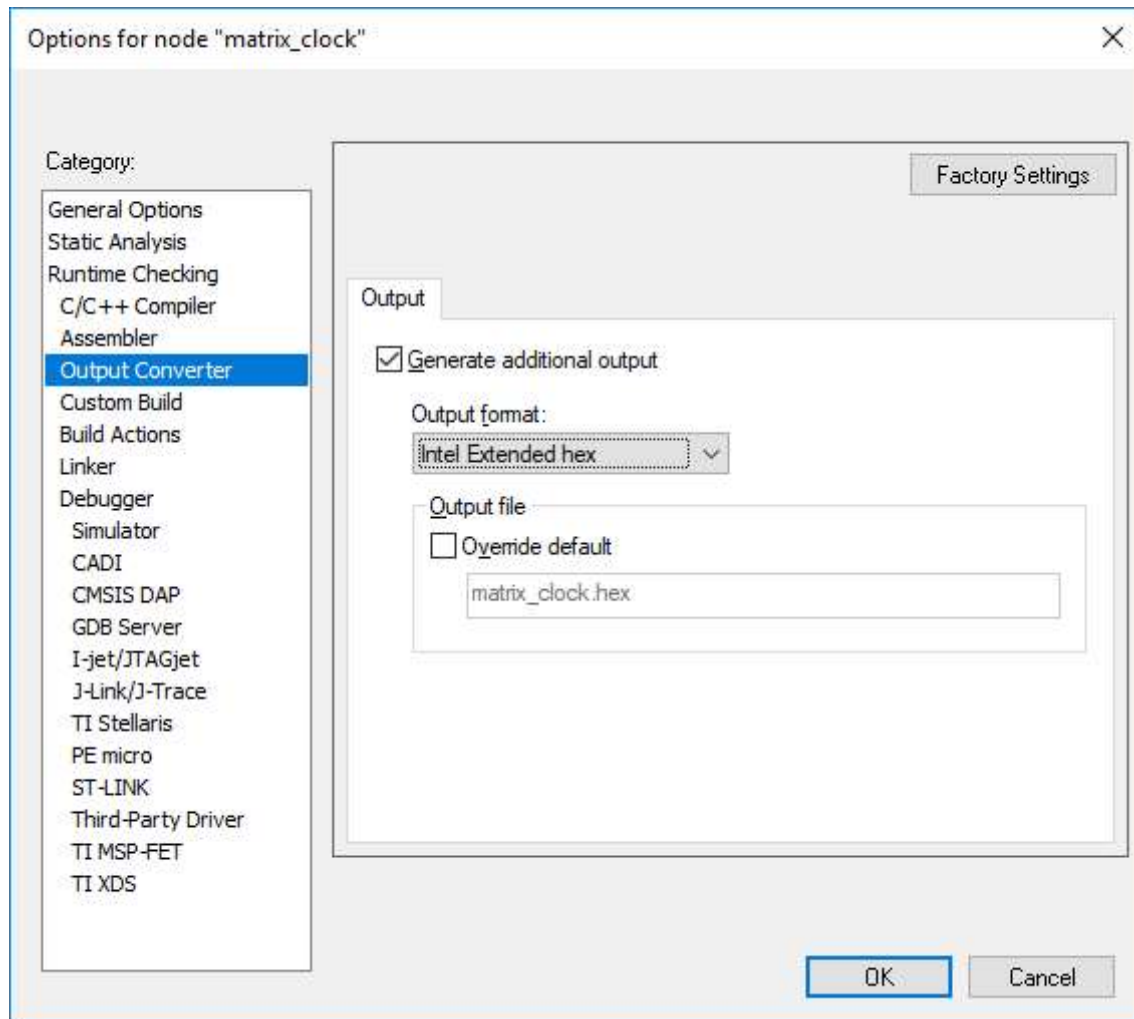
Если необходимо включить оптимизацию, перейдите во вкладку **Optimization** и выберите **Medium** или **High**. По-умолчанию стоит **Low**, мы не будем его менять.



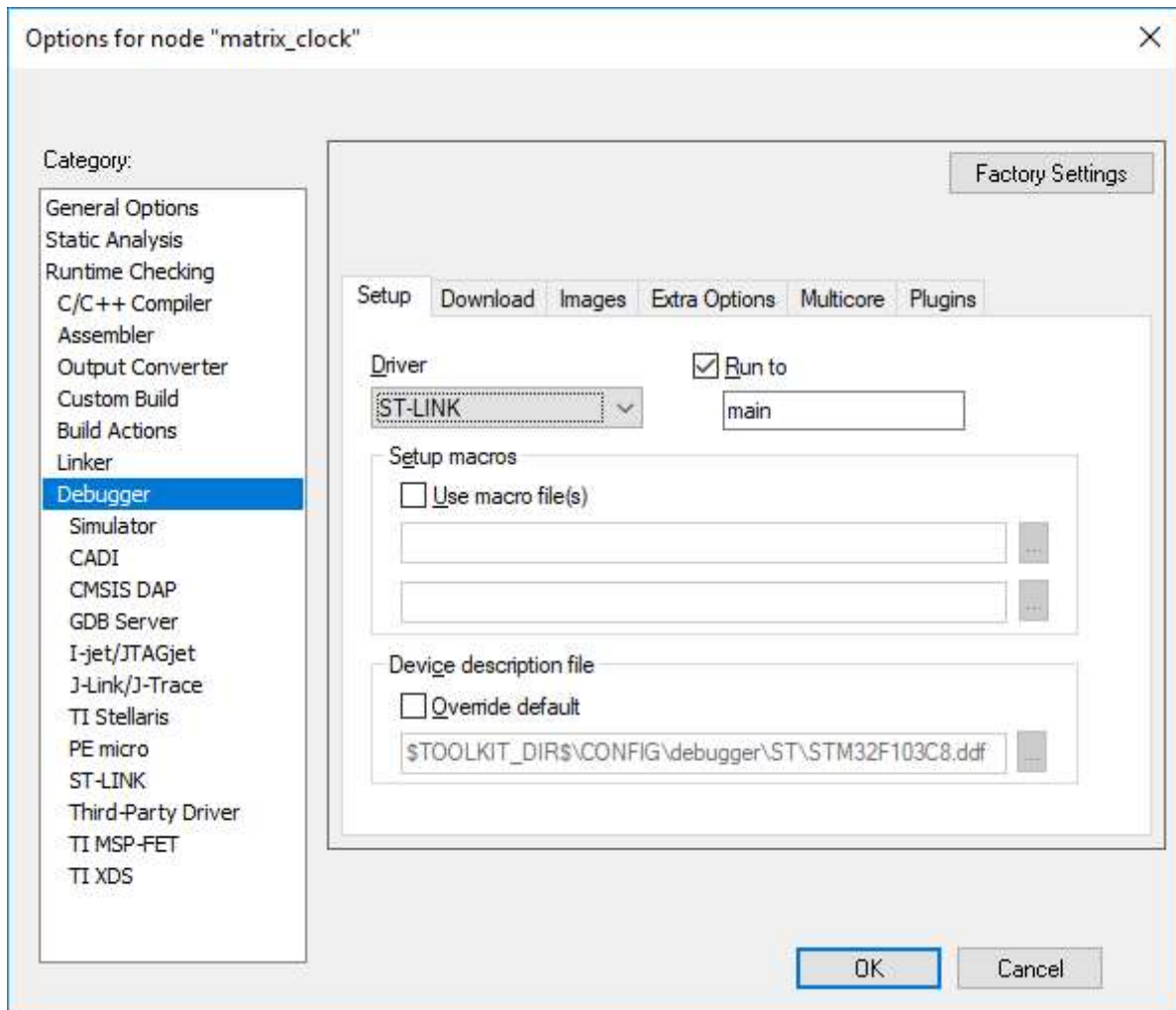
Следующее что необходимо сделать, это указать директории в которых хранятся библиотеки, а так же определить маркеры для препроцессора. Перейдите во вкладку **Preprocessor** и перепишите содержимое полей. Здесь `&PROJ_DIR&` заменяется на положение проекта в системе. Мы скопируем нужные файлы в конце.



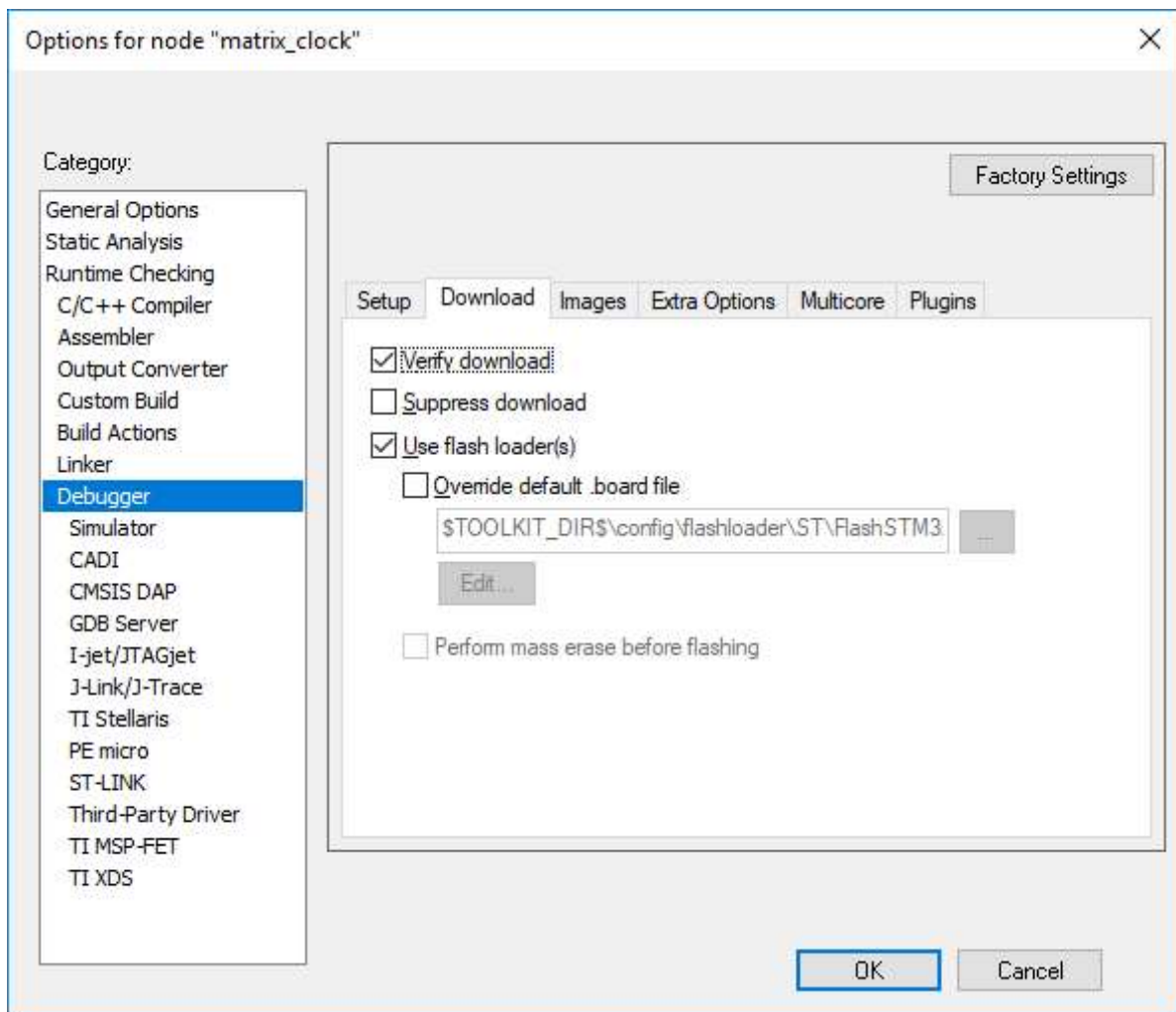
Что бы включить генерацию бинарного файла, откройте вкладку **Output (Output converter)**, поставьте галочку **Generate additional output** и выберите **Intel Extended hex**.



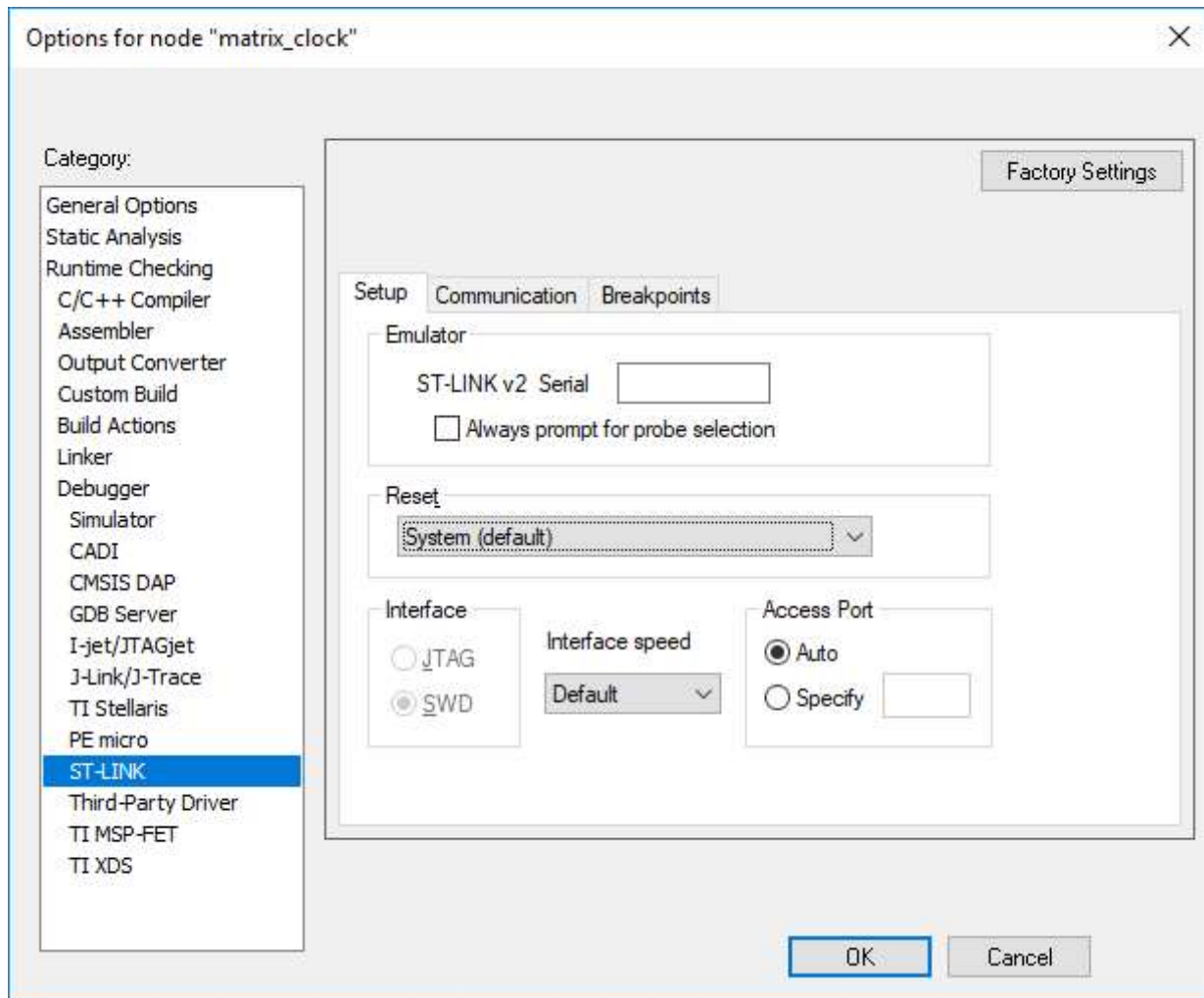
Последнее что нужно сделать, это выбрать отладчик, который мы будем использовать. IAR работает с разными контроллерами, отладчик ST называется ST-Link. Перейдите в пункт **Debugger**.



Во вкладке **Download** поставьте галочку **Verify download**, она заставит среду проверять успешность заливки прошивки.

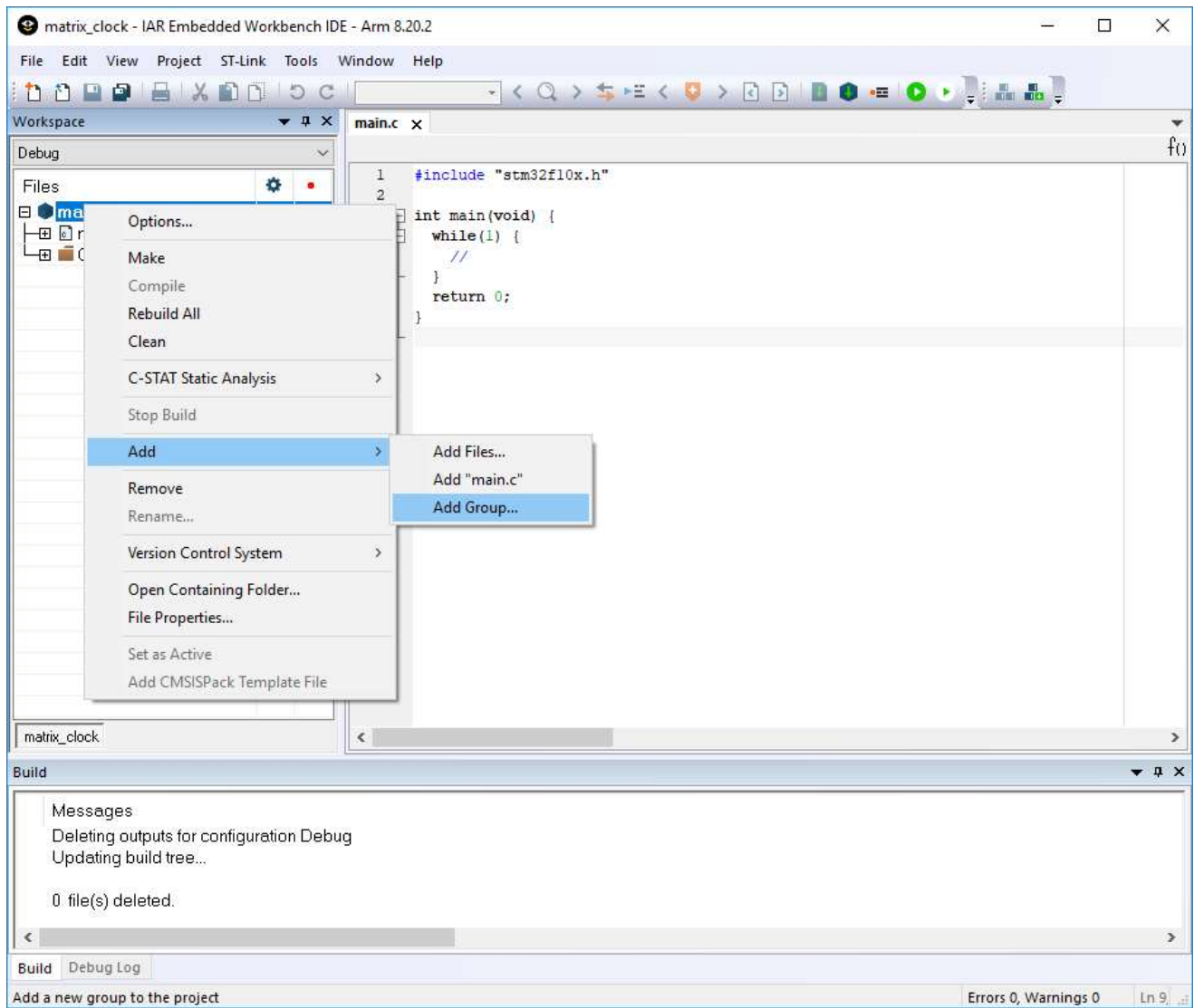


В пункте ST-LINK ничего менять не нужно, но все равно зайдите в него и изучите параметры при котором будет работать наш отладчик.

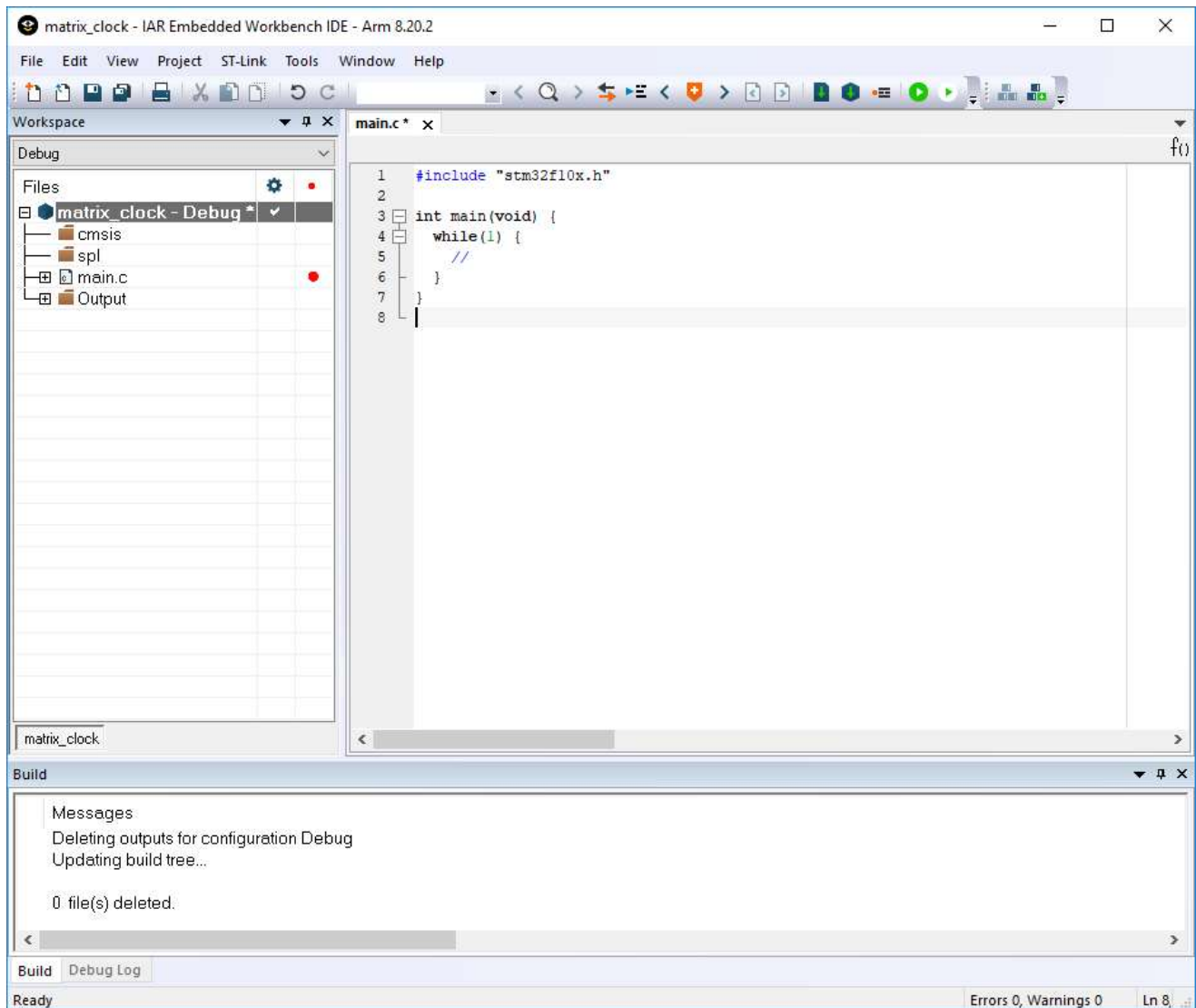


Нажмите OK для принятия изменений.

IAR позволяет создать виртуальные папки (группы) для файлов в проекте. Чтобы файлы библиотек вам не мешались, создайте под них такие группы. Для этого нажмите правой кнопкой мыши по проекту и выберите **Add ⇒ Add Group...**



Ваш проект должен выглядеть примерно следующим образом.



Далее скачайте файлы библиотек с сайта ST и добавьте их к проекту через то же меню.

Рассмотрим процесс настройки для библиотеки SPL.

С сайта ST Microelectronics нужно будет скачать архив STM32F10x standard peripherals library — в нём содержатся:

- **CMSIS** (Cortex Microcontroller Software Interface Standard) — библиотека, определяющая работу ядра, того самого Cortex M3 (о ней мы поговорим позже);
- **SPL** (Standart Peripheral Library) — стандартная периферийная библиотека, которая сэкономит вам много сил в конфигурации той или иной периферии;
- **Examples** — если нужно изучить, как работает новая для вас периферия, то в этой папке лежит множество примеров на все случаи жизни.

Так как среда IAR уже включает в себя CMSIS и она новее, чем та что в библиотеке, может возникнуть конфликт.

```
Warning[25] : Label 'xxxxx' is defined pubweak in a section implicitly declared root
```

Проблема решается заменой некоторых строк во встроенной библиотеке CMSIS: в местах упоминания `CODE:REORDER` добавьте `NOROOT(1)`. Например:

```
PUBWEAK NMI_Handler  
SECTION .text:CODE:REORDER:NOROOT(1)  
NMI_Handler
```

Сгенерировать файлы проекта с другими библиотеками можно в CubeMX, о котором мы поговорим чуть позже.

[Назад](#) | [Оглавление](#) | [Дальше](#)