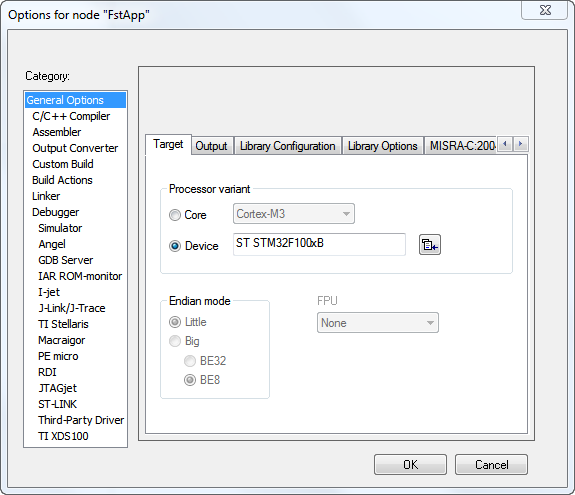
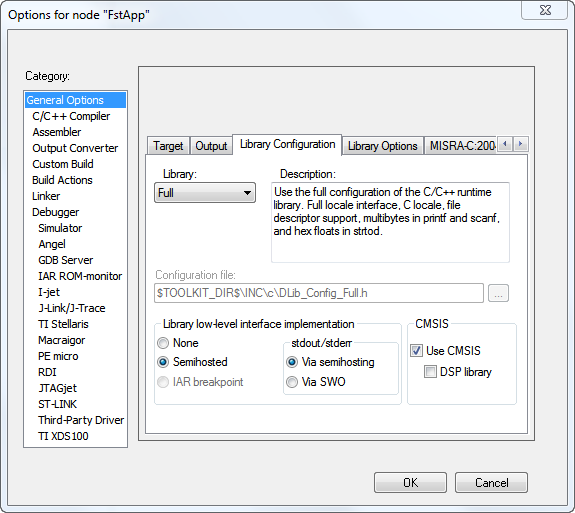
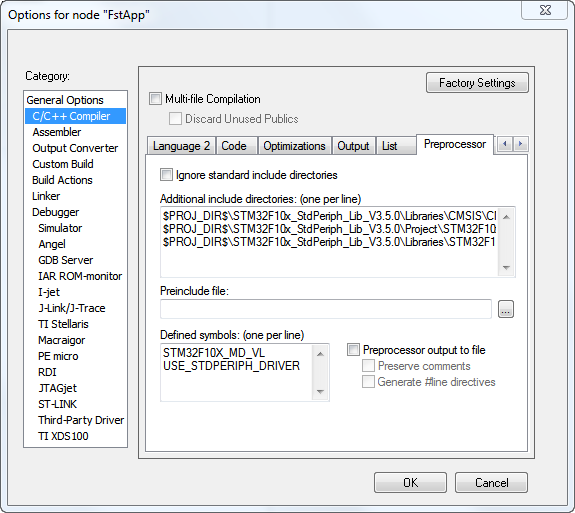
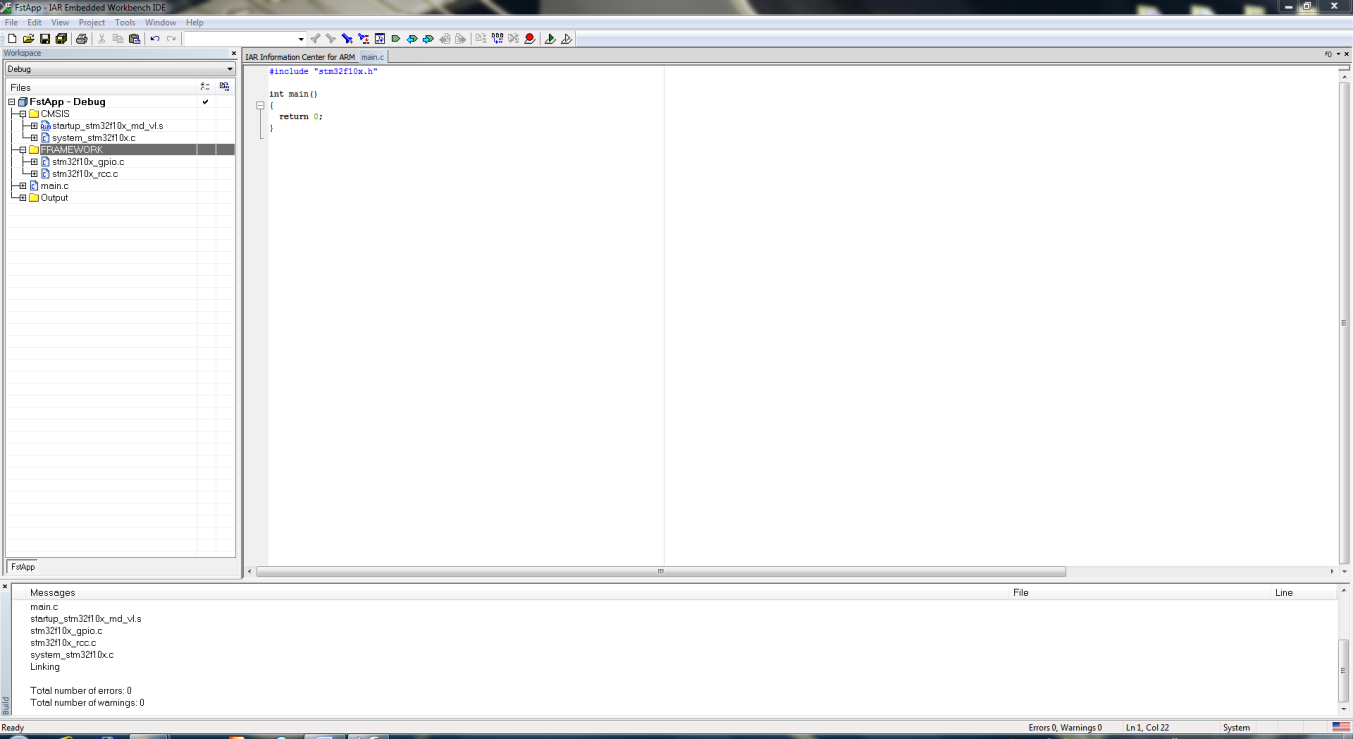
**Создание проекта в среде IAR Systems 6.40.**

Создаваемый проект является шаблонным, имеет имя FstPrj, содержится в каталоге Dropbox/Learn STM32.

Предполагается использование стандартной библиотеки управления периферией, в этом проекте последняя на данный момент версия – 3.5.0, содержится в каталоге проекта. Последовательность действий, необходимая для создания проекта:

1. Создаем проект. Меню Project -> Create New Project. В появившемся окне выбираем «C», затем «main». Вводим имя проекта «FstPrj».
2. В обозревателе workspace проект представлен в виде иерархического дерева с корнем FstPrj. Теперь нужно выбрать требуемый микроконтроллер. Щелкаем по корневому узлу проекта правой кнопкой мыши, жмем General Options, открываем вкладку Target и выбираем Device: STM32F100xB (или какой там у вас).   
   
3. Открываем вкладку «Library Configuration», в секции «CMSIS» ставим галочку «use CMSIS», тем самым мы подключаем «родную» поддержку уровня ядра Cortex M3, устанавливаемую вместе с IAR Embedded Workbench. Можно, по идее, использовать CMSIS из стандартной библиотеки периферии, однако при сборке проекта компилятор посылает вас куда подальше. Я не знаю с чем это связано! Пока что будем использовать встроенную поддержку. Кстати, в списке «Library» наверное лучше выбрать «Full».   
   
4. Далее прописываем пути, по которым располагаются требуемые заголовочные файлы, включаемые препроцессором. Для этого в этом же окне опций проекта выбираем категорию «C/C++ Compiler», открываем вкладку «Preprocessor». В текстовом ящике построчно прописываем пути. Кстати здесь можно использовать переменную «$PROJ\_DIR$», означающую текущий каталог проекта. Итак, нам понадобится:
   1. Поддержка уровня устройства CMSIS, файлы «stm32f10x.h» и «system\_stm32f10x.h», расположенные в каталоге: *$PROJ\_DIR$\STM32F10x\_StdPeriph\_Lib\_V3.5.0\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x*.
   2. Файл конфигурации библиотеки «stm32f10x\_conf.h», расположенный в каталоге:   
      *$PROJ\_DIR$\STM32F10x\_StdPeriph\_Lib\_V3.5.0\Project\STM32F10x\_StdPeriph\_Template*.
   3. API периферии, т.е. заголовочные файлы драйверов:   
      *$PROJ\_DIR$\STM32F10x\_StdPeriph\_Lib\_V3.5.0\Libraries\STM32F10x\_StdPeriph\_Driver\inc*.
5. Далее надо выбрать семейство, к которому относится ваш микроконтроллер. Делается это с помощью директивы препроцессора #define, которую вписываем в текстовый ящик под лэйблом «Defined symbols». В нашем случае это STM32F10X\_MD\_VL. Кроме того, нужно указать о намерении использовать стандартные драйверы периферии. Соответствующий макрос - USE\_STDPERIPH\_DRIVER.   
   
6. С настройками проекта закончили, жмем «ОК».
7. Добавляем файлы исходного кода. В обозревателе проекта создаем подкаталог CMSIS. Здесь будут расположены файл поддержки уровня устройства system\_stm32f10x.c – отвечает за инициализацию микроконтроллера, настройку системной частоты и пр, а также, зависимый от компилятора код стартапа startup\_stm32f10x\_md\_vl.s.
8. Осталось прописать в файле main.c директиву включения файла stm32f10x.h и можно запускать на компиляцию. В итоге получается что-то вроде этого:   
   
9. Теперь можно писать приложение. При использовании функций драйверов периферии нужно подключать соответствующие файлы исходного кода в проект.
10. Да, и самое главное. Если вы используете отладочную плату vldiscovery, то в настройках проекта нужно выбрать в категории отладчика «Debugger», на вкладке «Setup» программатор ST-LINK, и на вкладке «Download» поставить галочки «Verify download» и «Use flash loader»!!! Кроме того, в категории ST-LINK выбрать интерфейс SWD.

Примечание: содержимое данной заметки – результат мучений автора в попытке запустить отладочную плату STM32VLDISCOVERY и помигать светодиодом.