

SF2-Junior-KIT

**Разработка встраиваемого приложения
в среде SoftConsole 4.0.**

Введение

Данное руководство описывает процесс создания проекта встраиваемого программного обеспечения (ВПО) процессора ARM Cortex-M3 системы-на-кристалле (СнК) Microsemi SmartFusion2 в среде SoftConsole v4.0.

Необходимое программное обеспечение

Для изучения материала, изложенного в данном руководстве необходимо следующее программное обеспечение:

- среда разработки [Microsemi Libero SoC v11.8](#) с пакетом обновления [SP1](#);
- утилита для программирования микросхем СнК и ПЛИС FlashPro v11.8 или более поздняя версия, которая может быть установлена как часть пакета программ Microsemi Libero SoC и может быть запущена внутри Libero SoC или отдельно;
- среда разработки встраиваемого программного обеспечения SoftConsole v4.0;
- программа-оболочка TeraTerm или аналогичное программное обеспечение (PuTTY или HyperTerminal);
- драйверы шины USB, которые можно установить, пройдя по ссылке:
http://www.microsemi.com/document-portal/doc_download/131593-usb-uart-driver-files

Необходимое аппаратное обеспечение

Вам понадобится отладочный набор [SF2-Junior-KIT](#), который включает следующие компоненты:

- 1) Модуль SF2-Junior-KIT;
- 2) Жидкокристаллический дисплей 320x240 с интерфейсом SPI и сенсорной панелью (touchscreen);
- 3) Программатор FlashPro4;
- 4) USB – Bluetooth донгл;
- 5) Модуль приемопередатчика Bluetooth – UART;
- 6) Преобразователь напряжения AC-DC 9В 1А;
- 7) Кабель USB 2.0 A-male to mini-B.

Создание проекта системы-на-кристалле

Перед созданием проекта встроенного программного обеспечения необходимо создать проект системы-на-кристалле (СнК) в среде разработки Libero SoC. Поскольку процесс создания проектов СнК в среде разработки Libero SoC уже описан в [Руководстве быстрого старта](#), пропустим этот этап и перейдем непосредственно к этапу разработки ВПО в среде SoftConsole v4.0. В качестве проекта СнК воспользуемся [проектом](#), разработка которого описана в Руководстве быстрого старта.

Создание проекта ВПО в среде SoftConsole v4.0

В процессе создания проекта ВПО в среде разработки SoftConsole v3.4 в папке проекта СнК ../MyFirstProject были созданы папки /firmware и /SoftConsole. Поскольку форматы файлов проектов ВПО созданных в SoftConsole v3.4 и SoftConsole v4.0 не совместимы, содержимое указанных папок в рамках рассматриваемой темы оказывается не востребованным. Чтобы не потерять полученные ранее наработки, просто переименуем ранее созданные папки /firmware и /SoftConsole с ВПО созданным в SoftConsole 3.4 в папки с именами /firmware_34 и /SoftConsole_34 соответственно.

В папке проекта СнК создадим папку /SC40. В ней будут храниться файлы проекта ВПО созданные в SoftConsole 4.0. Теперь откроем проект СнК в Libero SoC и убедимся, что все необходимые для нашего проекта драйверы скачаны и доступны. Для этого выполняем команду **Configure Firmware Cores** в окне Design Flow (рис. 1).

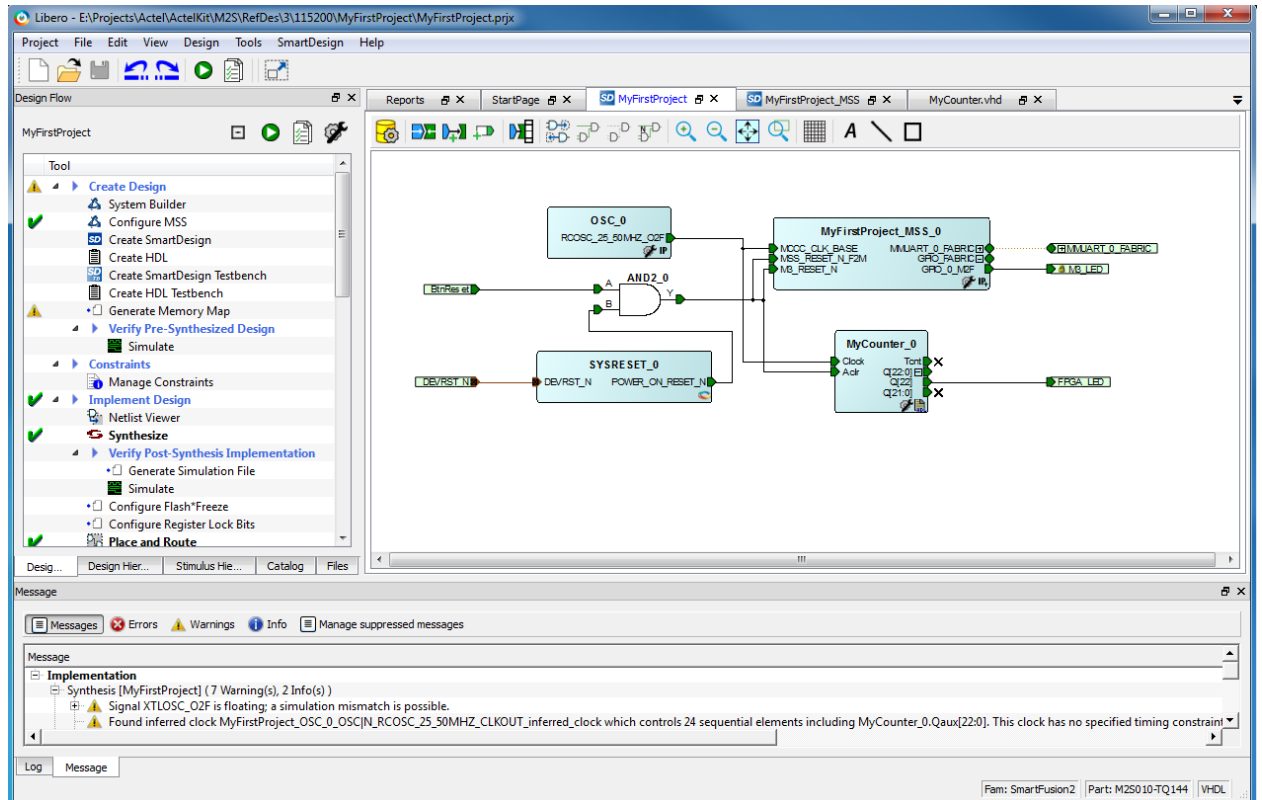
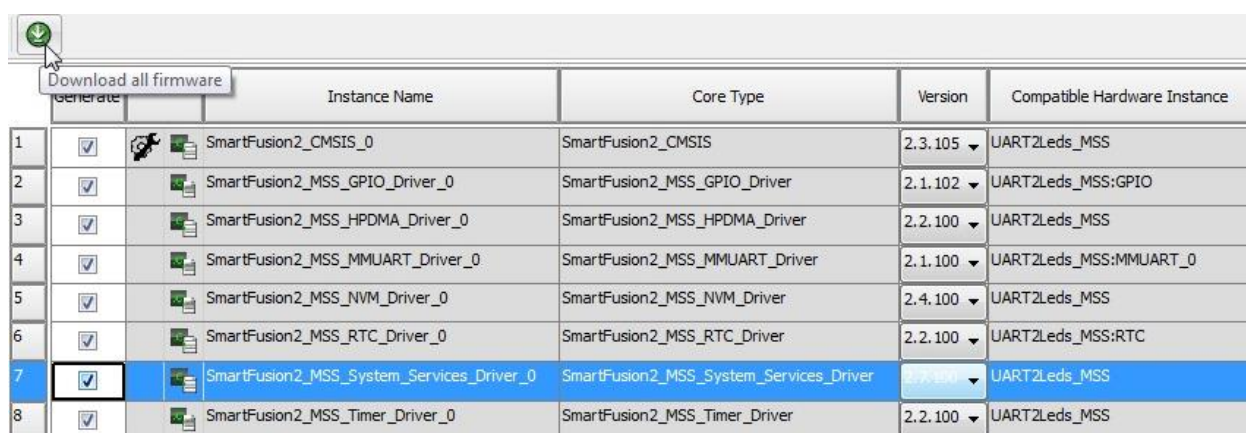


Рис. 1.

Если какой-либо драйвер отображается курсивом необходимо нажать кнопку «Download All Firmware» как показано на рис. 2.



	Generate	Instance Name	Core Type	Version	Compatible Hardware Instance
1	<input checked="" type="checkbox"/>	SmartFusion2_CMSIS_0	SmartFusion2_CMSIS	2.3.105	UART2Leds_MSS
2	<input checked="" type="checkbox"/>	SmartFusion2_MSS_GPIO_Driver_0	SmartFusion2_MSS_GPIO_Driver	2.1.102	UART2Leds_MSS:GPIO
3	<input checked="" type="checkbox"/>	SmartFusion2_MSS_HPDM_A_Driver_0	SmartFusion2_MSS_HPDM_A_Driver	2.2.100	UART2Leds_MSS
4	<input checked="" type="checkbox"/>	SmartFusion2_MSS_MMUART_Driver_0	SmartFusion2_MSS_MMUART_Driver	2.1.100	UART2Leds_MSS:MMUART_0
5	<input checked="" type="checkbox"/>	SmartFusion2_MSS_NVM_Driver_0	SmartFusion2_MSS_NVM_Driver	2.4.100	UART2Leds_MSS
6	<input checked="" type="checkbox"/>	SmartFusion2_MSS_RTC_Driver_0	SmartFusion2_MSS_RTC_Driver	2.2.100	UART2Leds_MSS:RTC
7	<input checked="" type="checkbox"/>	SmartFusion2_MSS_System_Services_Driver_0	SmartFusion2_MSS_System_Services_Driver	2.2.100	UART2Leds_MSS
8	<input checked="" type="checkbox"/>	SmartFusion2_MSS_Timer_Driver_0	SmartFusion2_MSS_Timer_Driver	2.2.100	UART2Leds_MSS

Рис. 2.

После этого выполняем команду **Export Firmware**. В появившемся окне Export Firmware выбираем пункт **Software IDE: SoftConsole4.0**, как показано на рис. 3. Нажимаем кнопку OK.

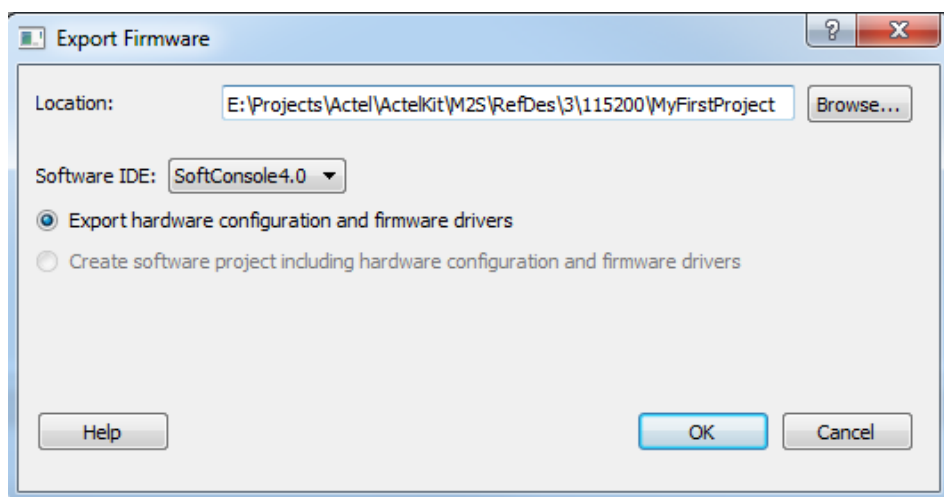


Рис. 3.

В результате в каталоге проекта появится папка /firmware с драйверами блоков микроконтроллерной подсистемы.

Теперь запустим SoftConsole 4.0 и создадим проект ВПО. В качестве каталога хранения файлов проекта укажем ранее созданный нами каталог ../SC40 в папке проекта (рис. 4). Откроется главное окно графического интерфейса пользователя SoftCinsole 4.0. Поскольку мы еще не создали ни одного проекта все окна графического интерфейса будут пустыми (рис. 5). Выполним команду **File > New > C Project**. В появившемся окне свойств создаваемого проекта заполним поля:

- Project Name: SC40_project
- Project type: Executable / Empty Project
- Toolchains: Cross ARM GCC

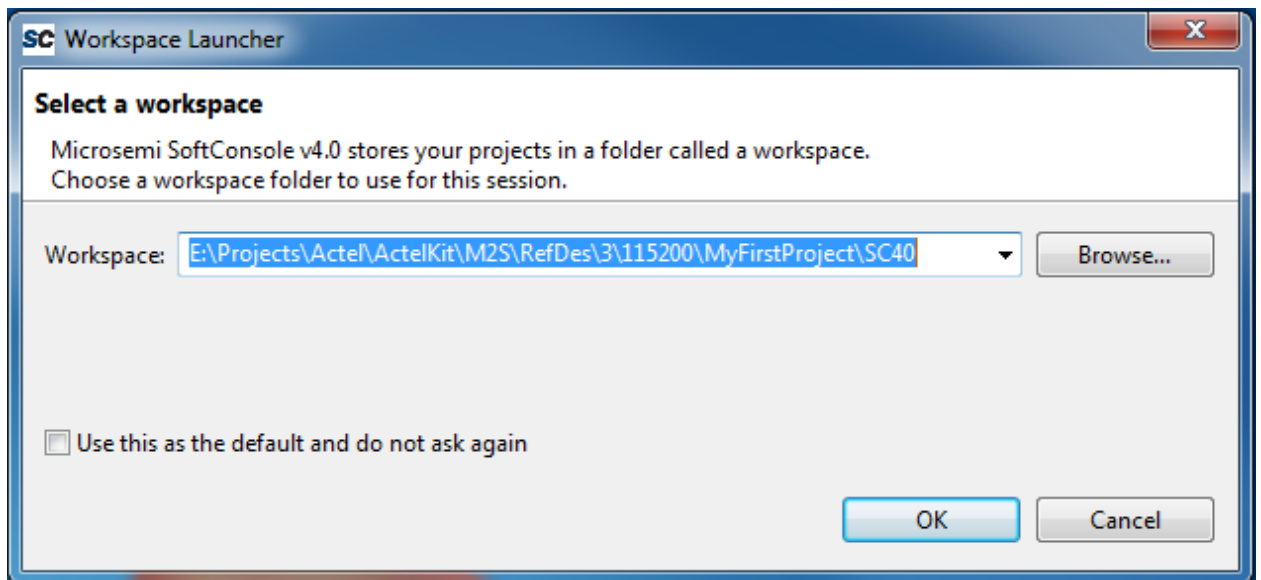


Рис. 4.

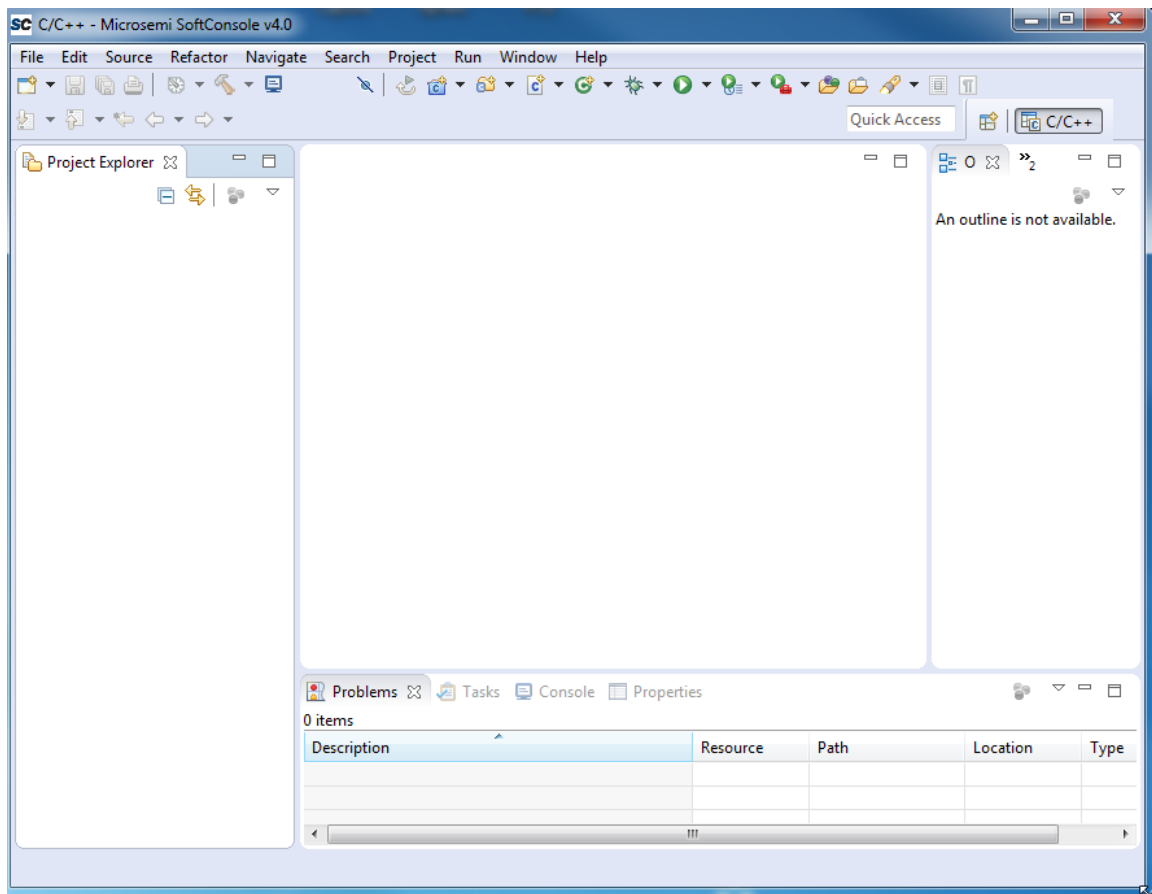


Рис. 5.

Опцию **Use default location** оставим выбранной (рис. 6). Принимаем изменения, нажимаем кнопку **Next**. На следующем шаге согласимся с предложенными по умолчанию настройками и также нажимаем **Next**. Во вкладке **Cross GNU ARM Toolchain** контролируем значения параметров:

- Toolchain name: GNU Tools for ARM Embedded Processors (arm-none-eabi-gcc)

- Toolchain path: `${eclipse_home}/../arm-none-eabi-gcc/bin` и нажимаем **Finish**.

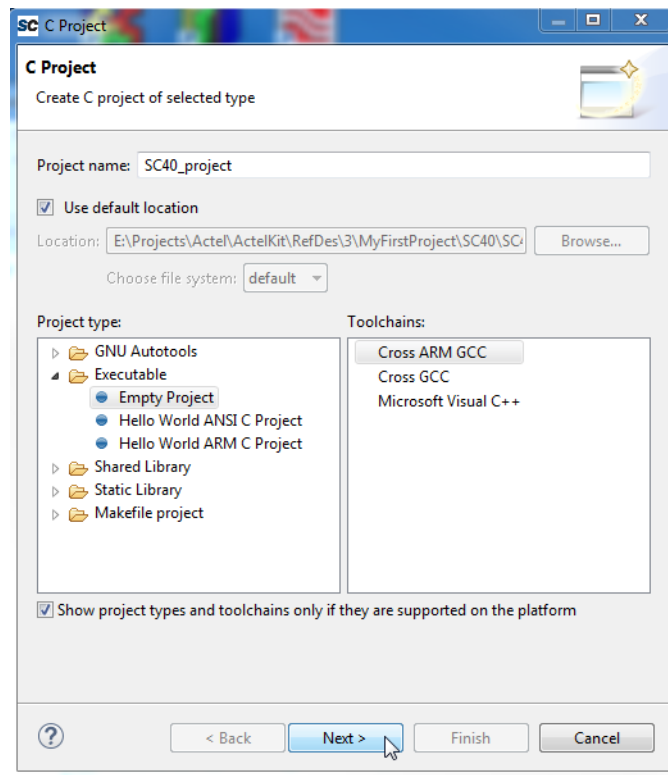


Рис. 6.

В результате выполнения описанных шагов будет создан новый проект встраиваемого программного обеспечения, а главное окно графического интерфейса SoftConsole 4.0 примет вид, показанный на рис. 7.

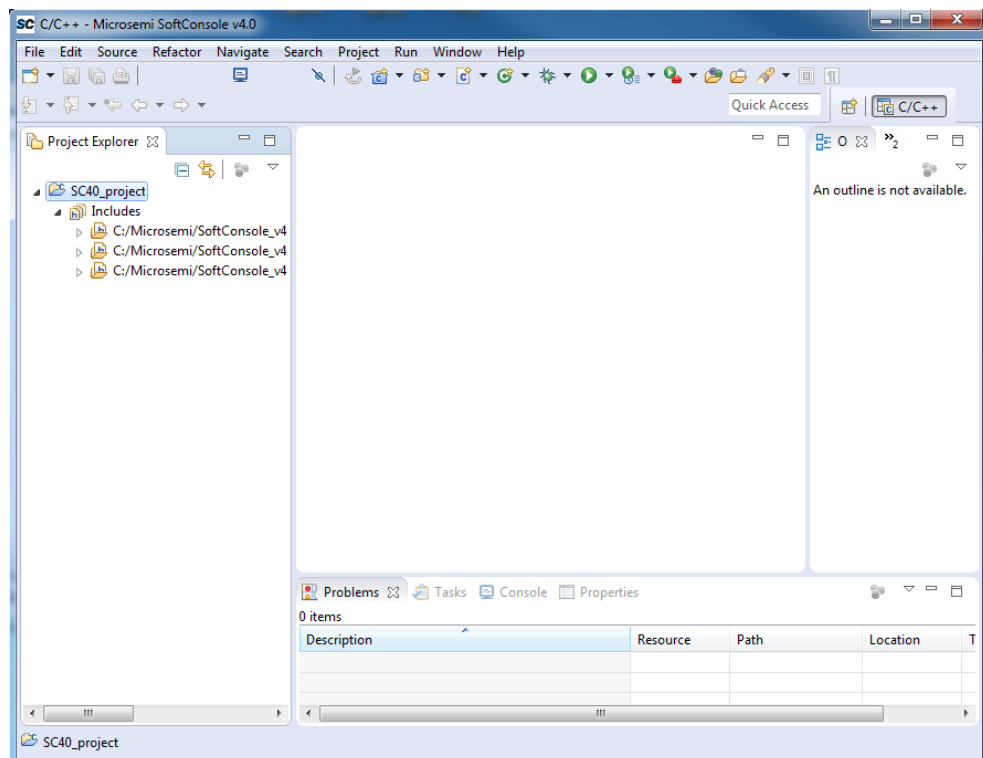


Рис. 7

Теперь необходимо импортировать в проект драйверы используемых периферийных блоков MSS SmartFusion2 и стандартные библиотеки, используемые для разработки приложений. Для этого выполним команду меню **File > Import**, в появившемся окне раскроем в качестве источника выберем **General > File System** и нажмем кнопку **Next**. В следующем открывшемся укажем каталог с нужными файлами /Firmware нашего проекта и выберем импортируемые опции в левом окне:

- CMSIS
- drivers
- drivers_config
- hal

В правом окне выберем файл **MyFirstProject_hw_platform.h**. В качестве места назначения выбираем папку нашего проекта (рис. 8).

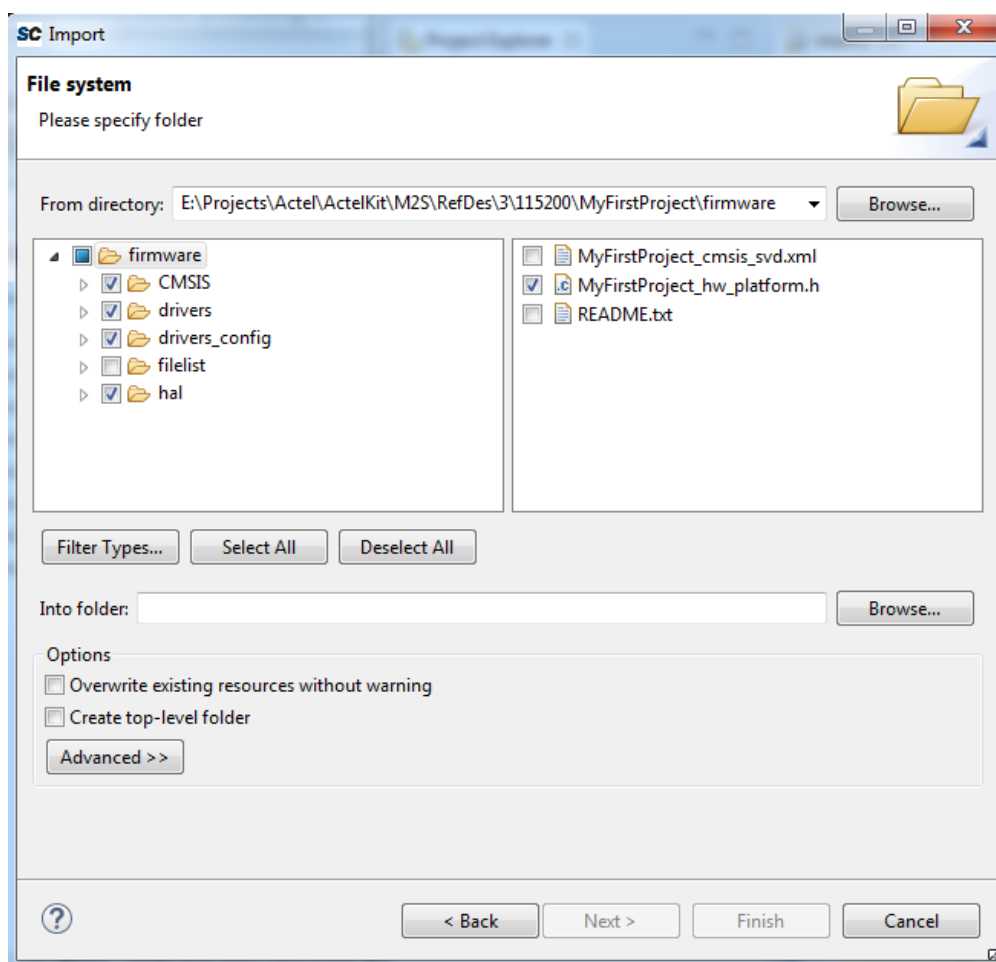


Рис. 8.

Файл с исходным кодом нашего приложения на языке «С» нами уже создан. Он находится в каталоге с файлами ВПО SoftConsole v3.4. Остается импортировать его в проект по уже пройденной нами схеме, то есть выполнить команду **File > Import, General > File System**. Указываем каталог, в котором находится файл: `..MyFirstProject\SoftConsole_34\MyFirstProject_MSS_CM3\MyFirstProject_MSS_CM3_app`. Ставим галочку напротив нужного нам файла **main.c** (рис. 9). В результате описанных действий файл с

исходным кодом на языке С окажется виден в окне Project Explorer. Откроем его, дважды щелкнув на нем левой кнопкой мыши. Результат представлен на рис. 10.

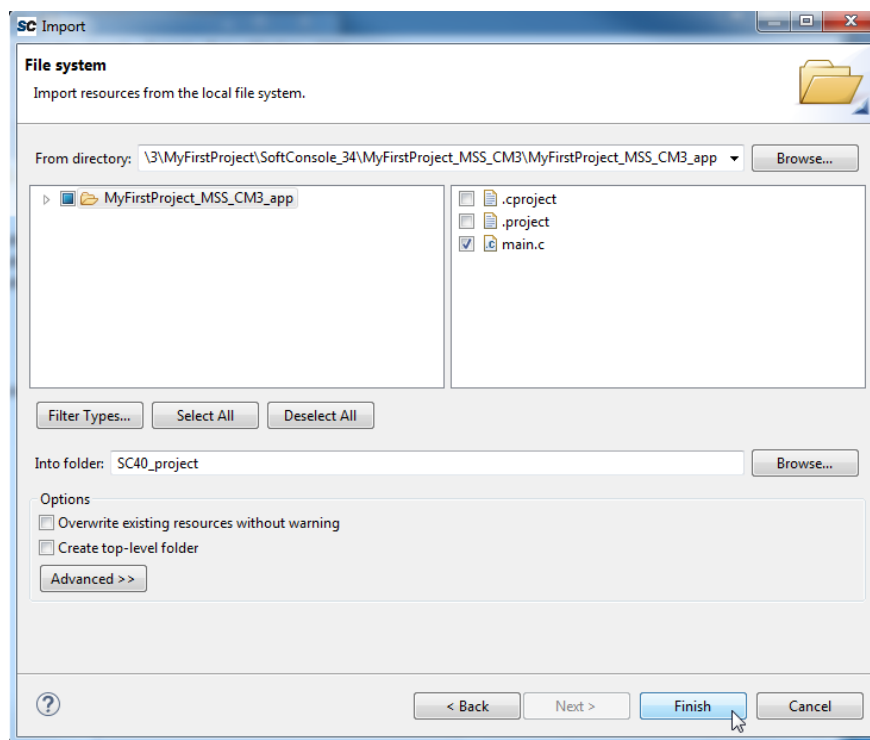


Рис. 9.

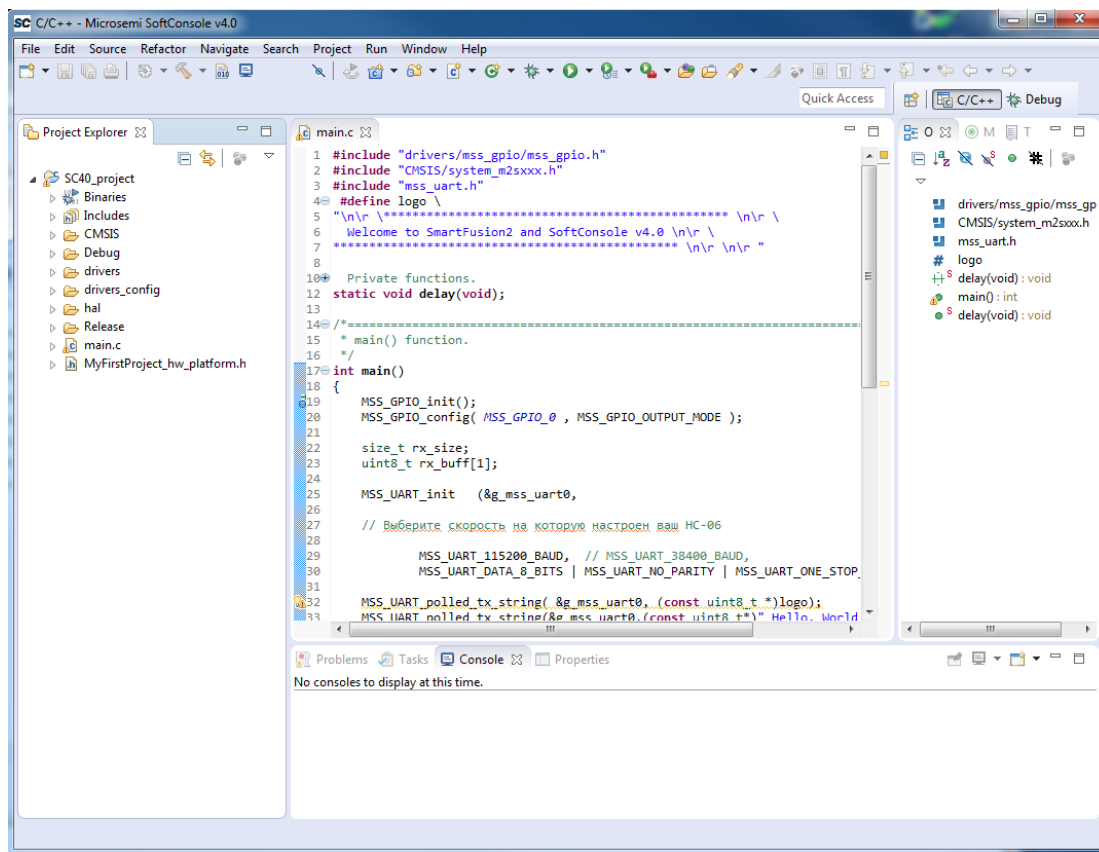


Рис. 10.

Теперь необходимо настроить среду разработки для работы с проектом. Для этого в окне **Project Explorer** щелкнем мышкой на корневой папке проекта **first_SC40_project**, затем выполним команду **File > Properties**. В появившемся окне выберем опции **C/C++ Build > Settings**. В выпадающем списке **Configuration:** выберем **All configurations**. Во вкладке **Tool Settings** выберем **Cross ARM C Compiler > Miscellaneous**. В поле **Other compiler flags** необходимо ввести значение: `--specs=cmsis.specs` (рис. 11). Во вкладке **Cross ARM C Compiler > Includes** добавим пути `../drivers/mss_uart/`, `../drivers/mss_gpio/`, `../CMSIS/`.

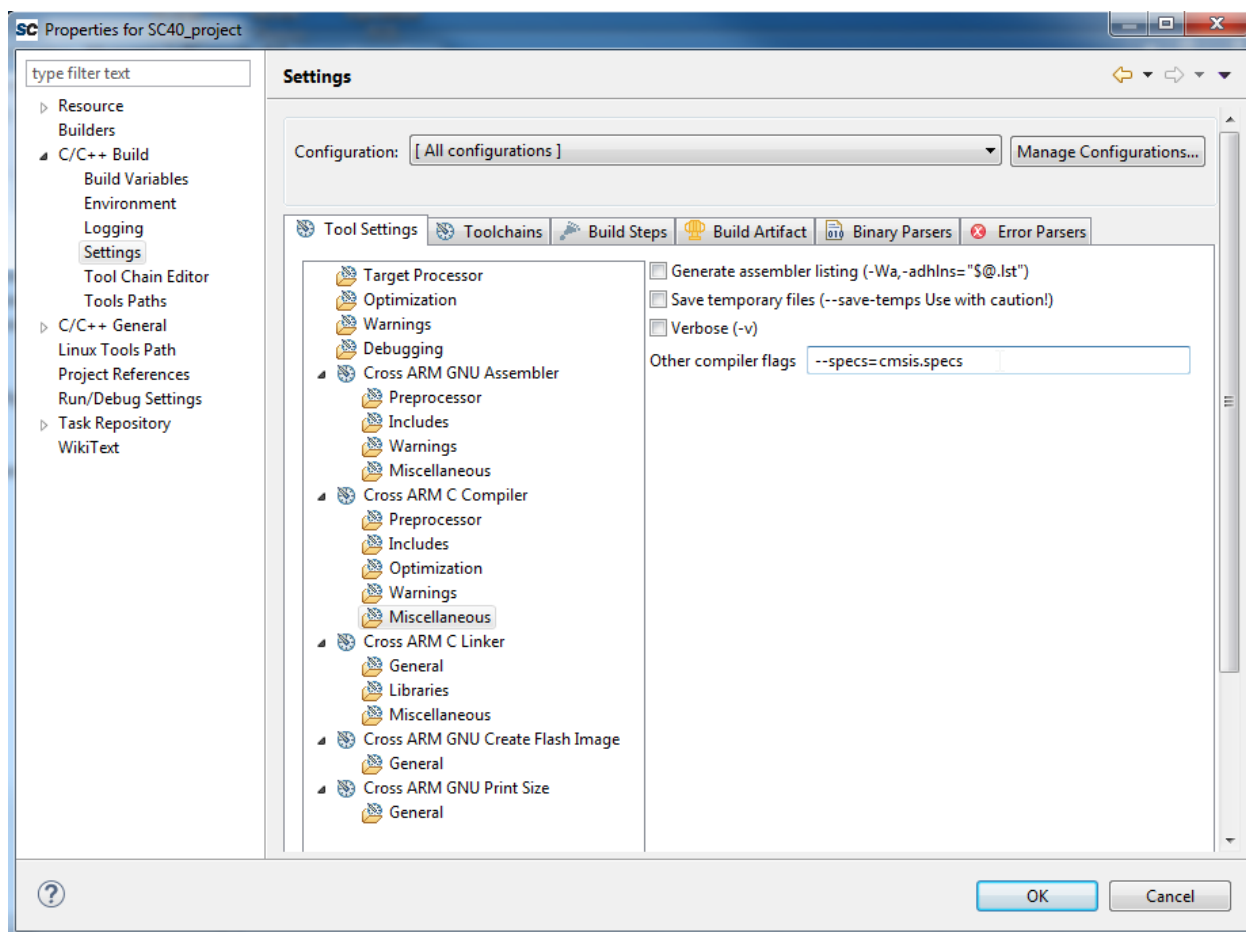


Рис. 11.

Следующим шагом настроим специфические параметры конфигураций **Debug** и **Release**. Начнем с конфигурации **Debug**. Из выпадающего списка **Configuration:** выберем **Debug [Active]**. Во вкладке **Tool Settings** выбираем **Cross ARM C Linker > General**. В окне **Script files (-T)** справа нажимаем кнопку **Add..**. В появившемся окне указываем значение параметра:

`../CMSIS/startup_gcc/debug-in-microsemi-smartfusion2-esram.ld` (рис. 12).

Переходим во вкладку **Cross ARM C Linker > Miscellaneous**. В ней необходимо выбрать пункт **Use newlib-nano (--spcs=nano.specs)**.

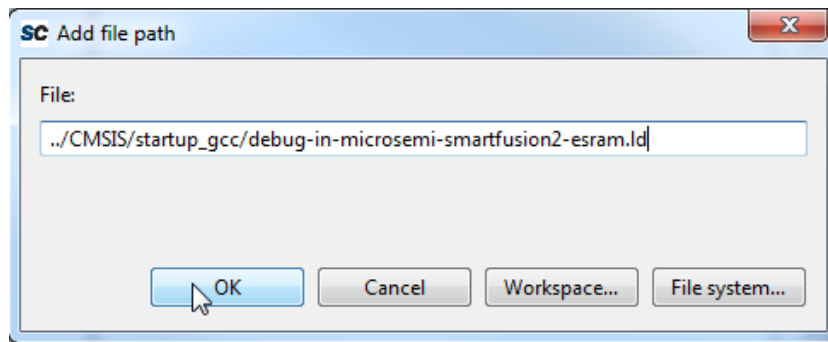


Рис. 12.

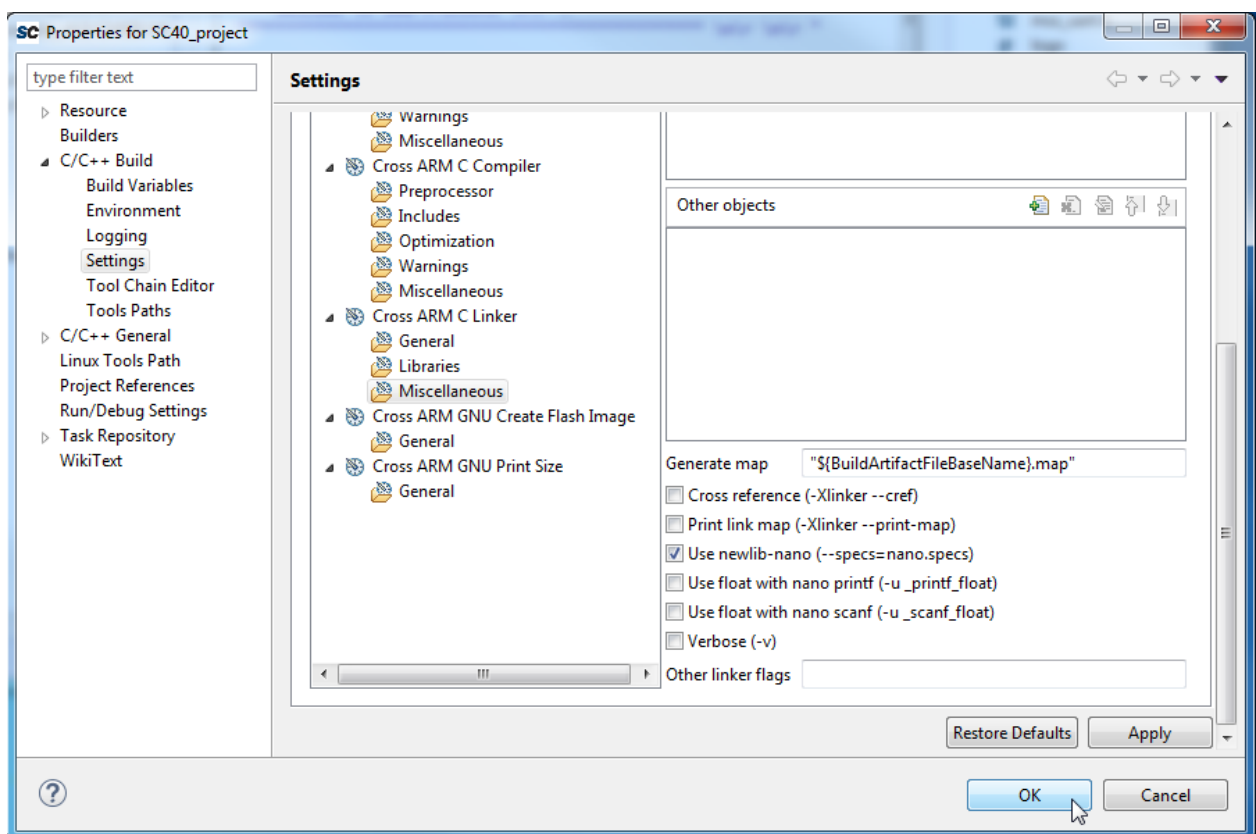


Рис. 13.

Настроим конфигурацию Release, для этого из выпадающего списка **Configuration:** выберем **Release**. Во вкладке **Tool Settings** выбираем **Cross ARM C Linker > General**. В окне **Script files (-T)** справа нажимаем кнопку **Add..** В появившемся окне указываем значение параметра:

```
../CMSIS/startup_gcc/debug-in-microsemi-smartfusion2-envm.ld
```

для отладки из eNVM или

```
../CMSIS/startup_gcc/production-smartfusion2-execute-in-place.ld.
```

для получения исполнимого *.hex файла ВПО и создания единого файла прошивки СнК. Во вкладке **Cross ARM C Linker > Miscellaneous** необходимо выбрать пункт

Use **newlib-nano** (**--spcs=nano.specs**).

Далее необходимо выбрать активную конфигурацию щелкнув в окне Project Explorer на папке проекта **SC40_project**. В появившемся меню выбрать **Build Configuration > Set Active > Debug** или **Release**. После чего собрать проект, выполнив команду основного меню **Project > Build Project**.

Пошаговая отладка проекта в SoftConsole v4.0

В результате выполнения команды меню **Project > Build Project** при выбранной активной конфигурации в конфигурации **Debug** в каталоге проекта появляется папка **\Debug** а в ней файл **SC40_project.hex** (рис. 14) который содержит выполняемый процессором программный код и символы отладочной информации, которые необходимы для выполнения программы в пошаговом режиме.

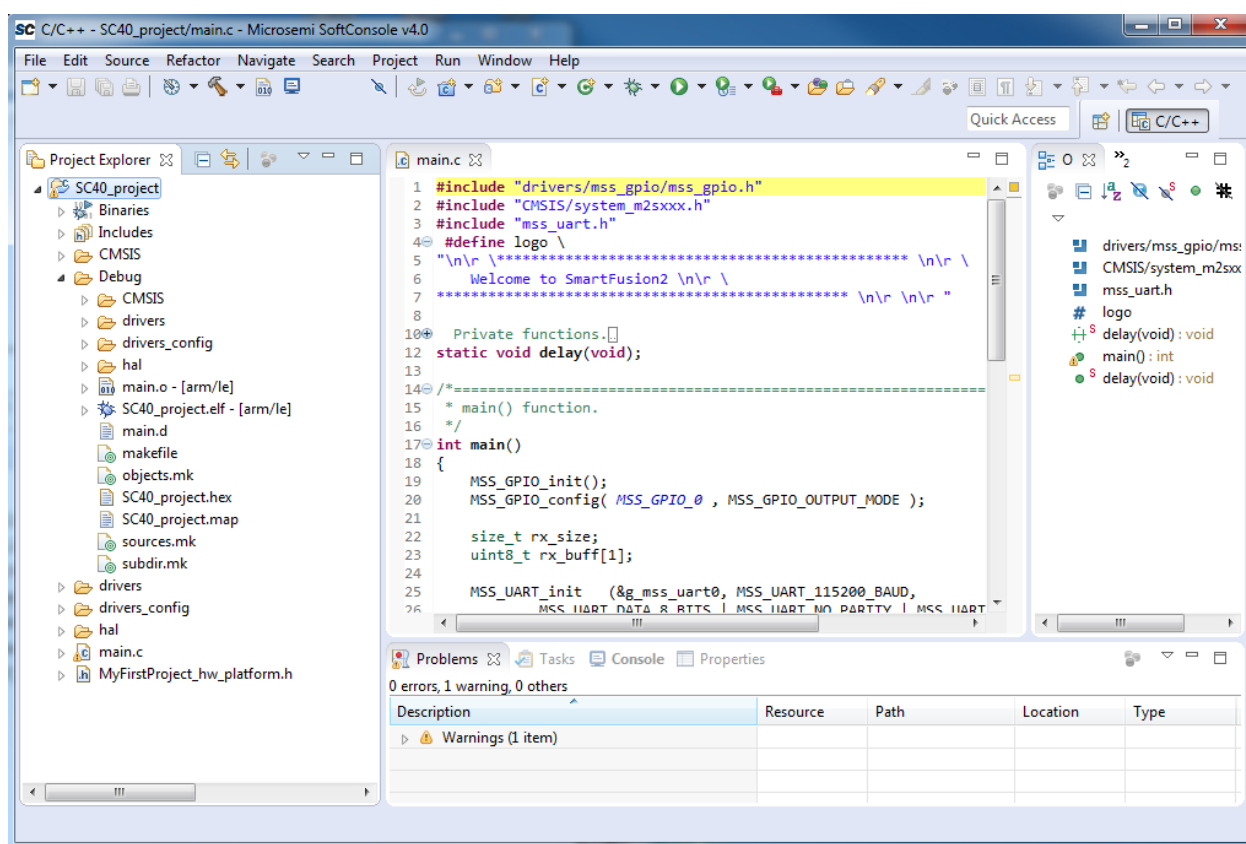


Рис. 14.

Для выполнения отладки необходимо подключить программатор к отладочному набору и подать питание. Также не забываем прошить микросхему конфигурационной последовательностью содержащей описание конфигурации микроконтроллерной подсистемы и логику работы ПЛИС.

Настроим оставшиеся параметры конфигурации отладчика SoftConsole 4.0 и запустим процесс пошаговой отладки. Для этого выполним команду основного меню **Run > Debug Configuration** (рис. 15). В открывшемся окне двойным щелчком активизируем команду **GDB OpenOCD Debugging**. Результат работы команды представлен на рис. 16.

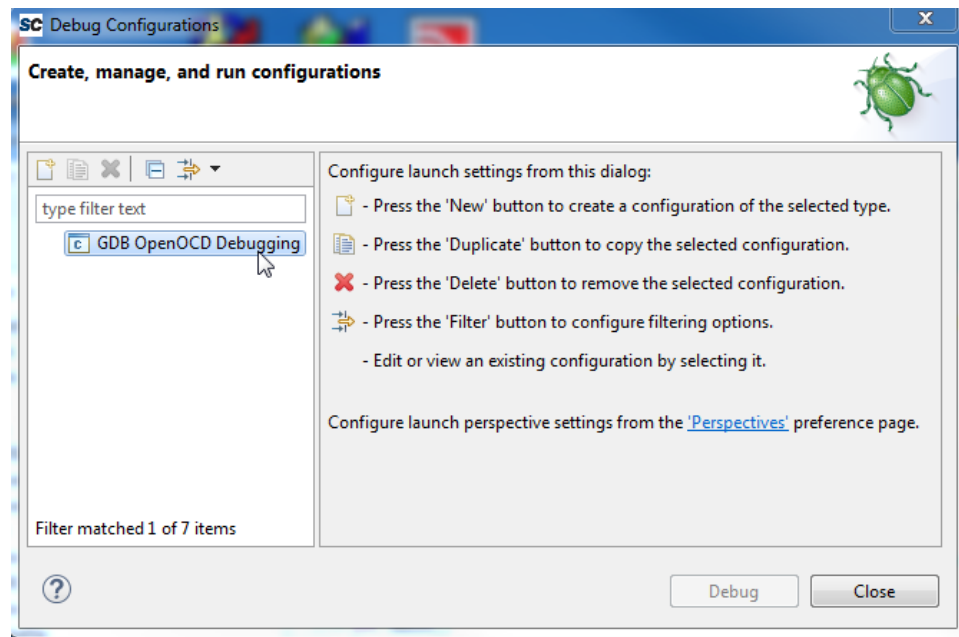


Рис. 15.

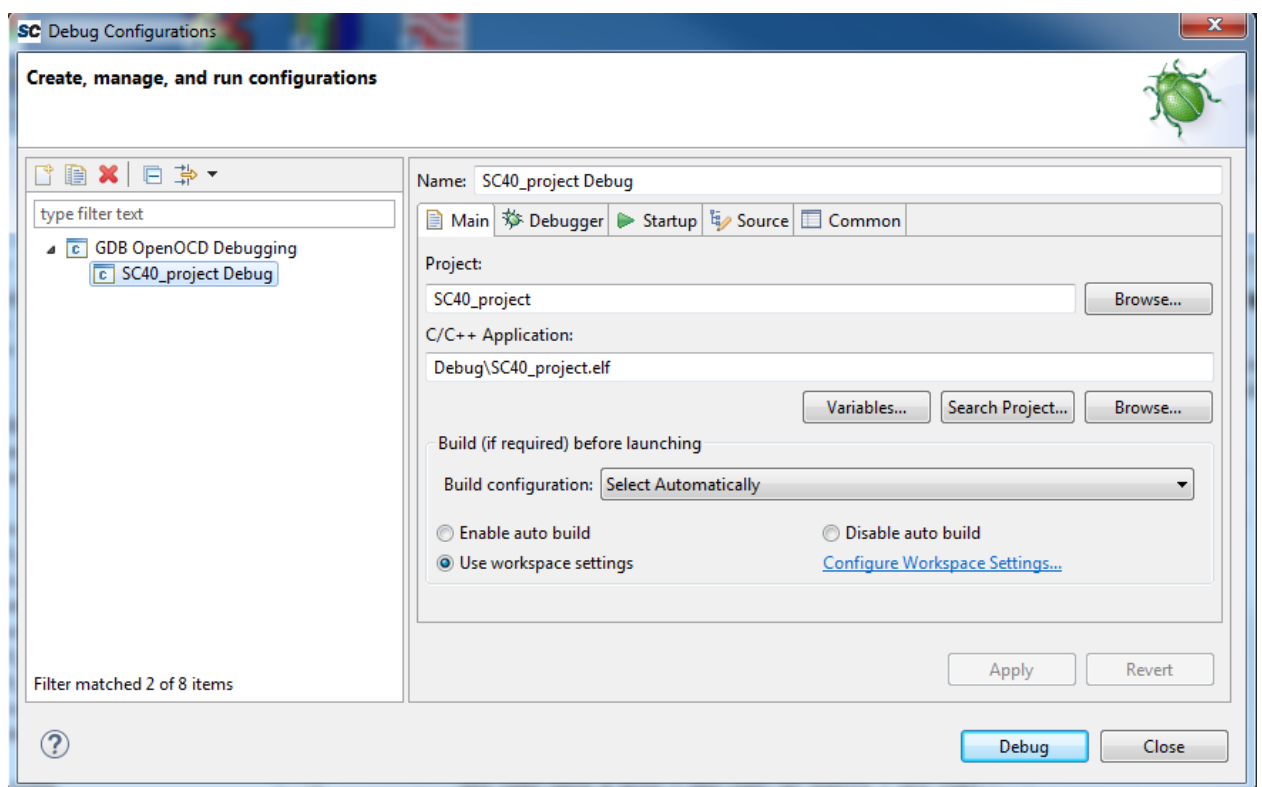


Рис. 16.

Теперь необходимо перейти во вкладку **Debugger** и скорректировать значение параметра **Config options** в соответствии с наименованием микросхемы с которой планируется работать. По умолчанию данное поле имеет значение

```
--command "set DEVICE M2S090" --file board/microsemi-cortex-m3.cfg
```

На нашем отладочном наборе [SF2-Junior-KIT](#) установлена микросхема M2S010-TQ144, таким образом, значение параметра **Config options** необходимо изменить на:

--command "set DEVICE M2S010" --file board/microsemi-cortex-m3.cfg

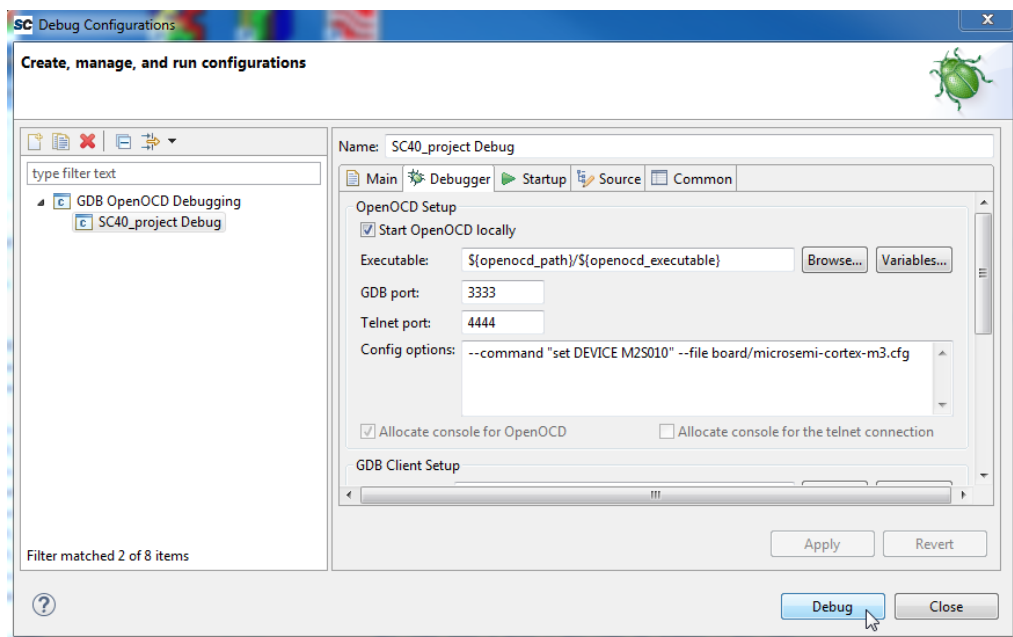


Рис. 17.

После нажатия кнопки Debug исполняемый код и отладочная информация загружается в eSRAM память SmartFusion2 и начинается процесс пошаговой отладки. Курсор выполнения программы устанавливается на первую выполнимую строку основной процедуры (рис. 18).

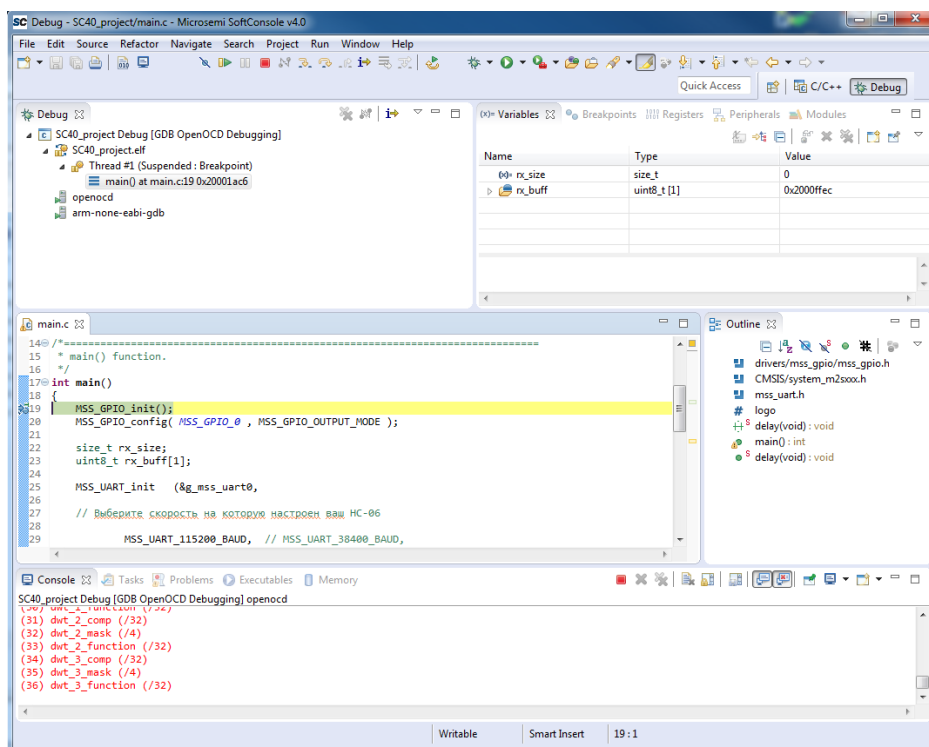


Рис. 18.

Запускаем программу программу-терминал и убеждаемся успешном выполнении разработанного нами приложения (рис. 19).

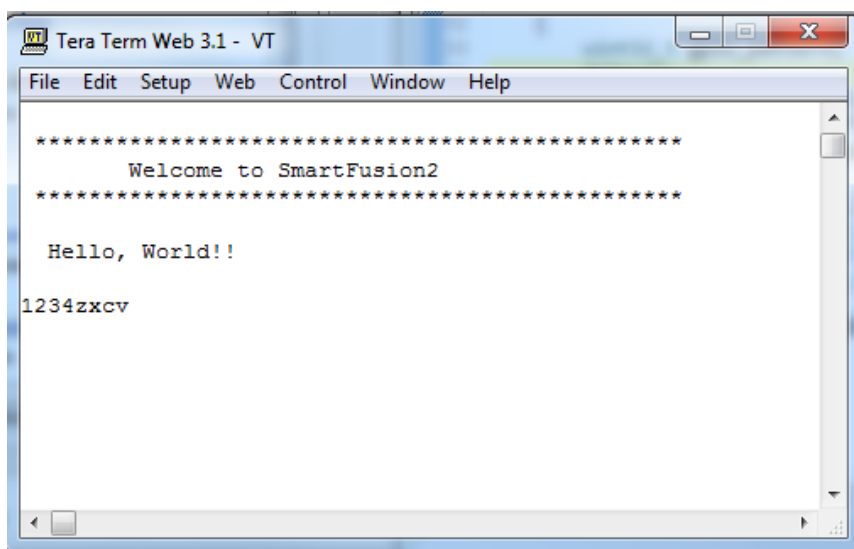


Рис. 19.

Убедившись в корректной работе приложения в режиме отладки, переходим к созданию Release-версии прошивки включающей конфигурационную последовательность ПЛИС и оптимизированный код нашего приложения. Для этого в среде SoftConsole 4.0 сделаем активной конфигурацию Release (рис. 20).

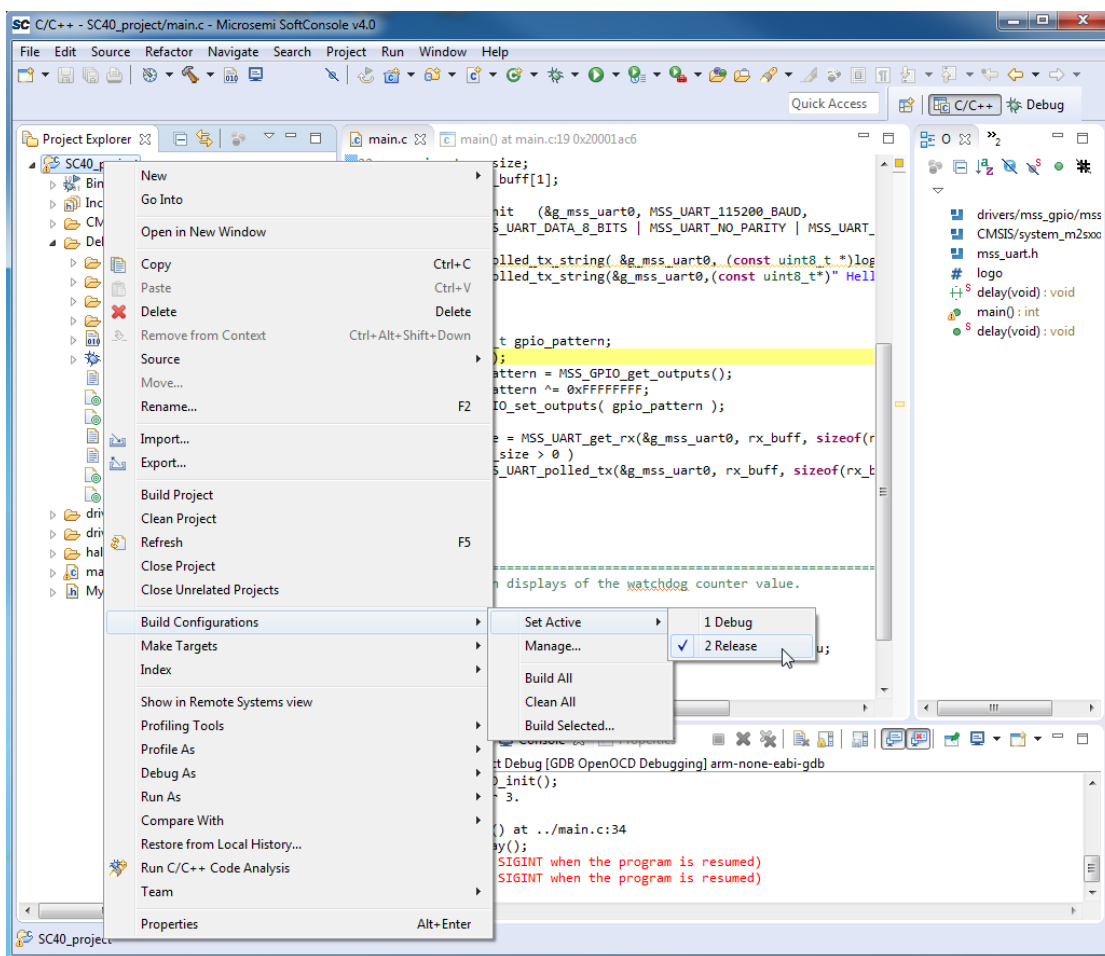


Рис. 20.

Теперь выполним команду основного меню **Project > Build All**, то есть создадим файл с оптимизированным кодом нашего приложения. В результате выполнения команды в рабочей папке проекта ВПО появится каталог **\Release**, а в нем файл **SC40_project.hex** с кодом нашего приложения не содержащим отладочной информации (рис. 21).

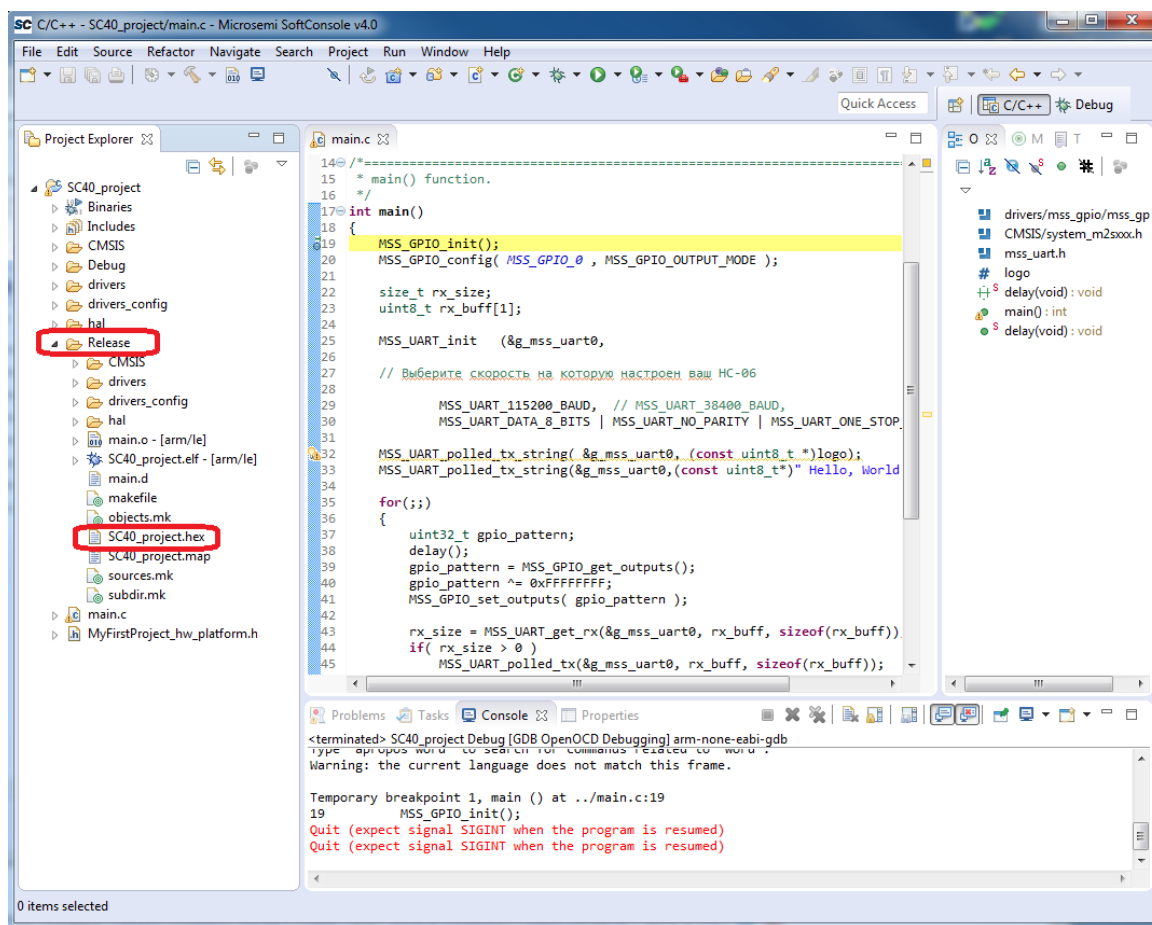


Рис. 21.

Откроем проект системы на кристалле в среде Libero SoC. Двойным щелчком левой кнопки мыши по компоненту **MyFirstProject_MSS_0** откроем окно редактирования свойств микроконтроллерной подсистемы (рис. 22).

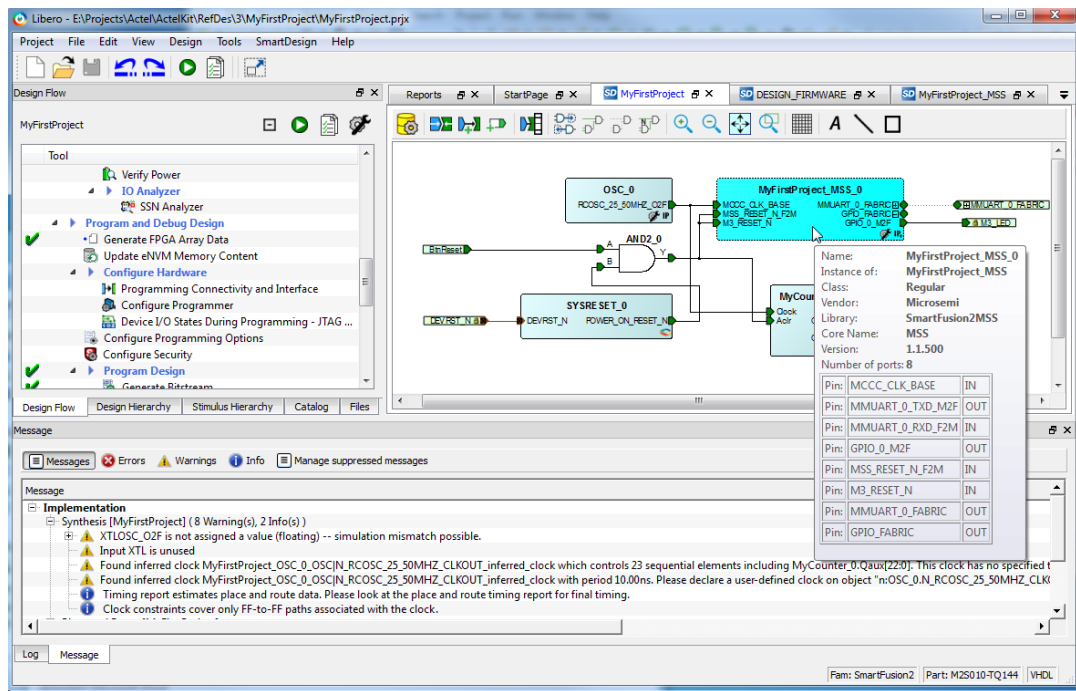


Рис. 22.

В открывшемся окне двойным щелчком по компоненту **ENVM** откроем окно конфигурации eNVM памяти Microsemi SmartFusion2 (рис. 23).

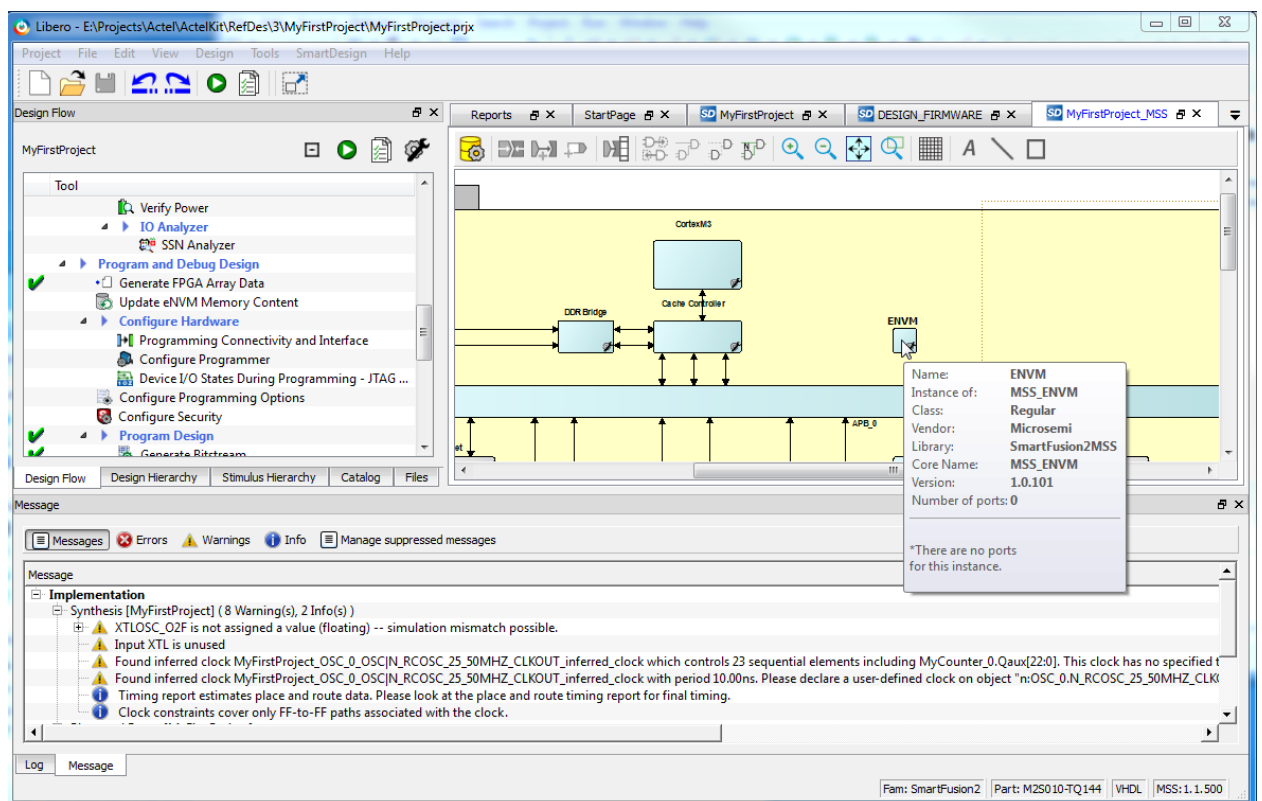


Рис. 23.

Добавим клиент **Data Storage**. В появившемся окне Add Data Storage Client укажем имя клиента и путь к файлу Release версии приложения (рис. 24).

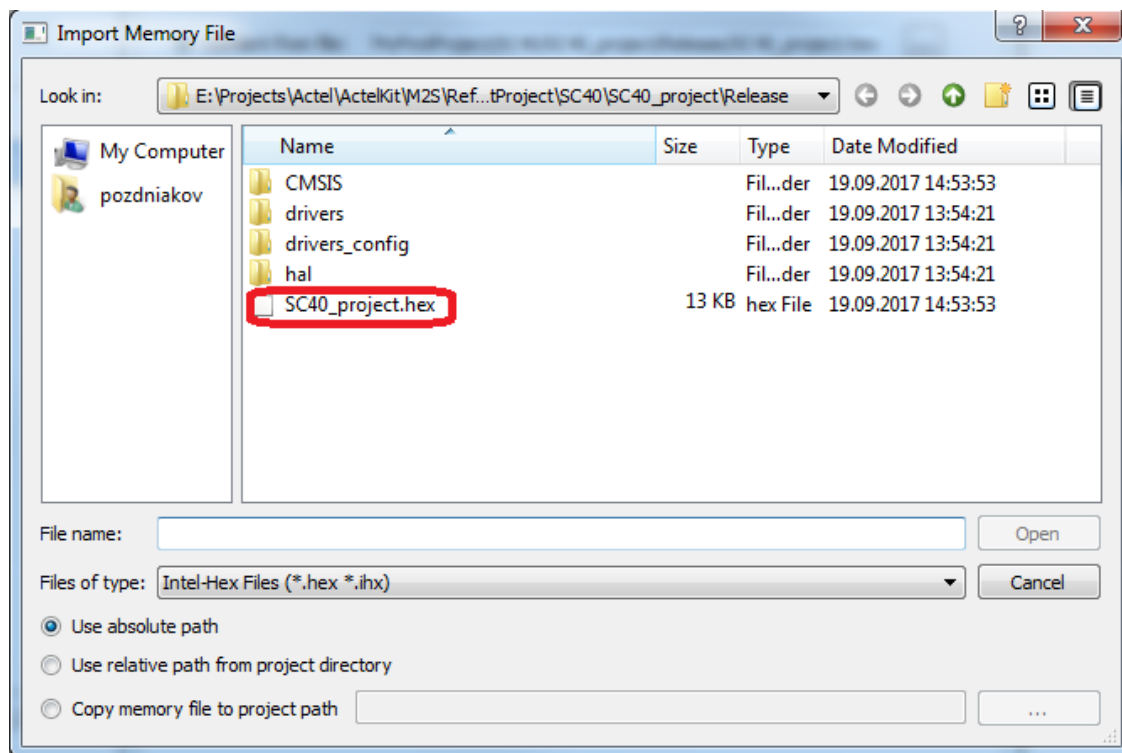


Рис. 24.

Сохраним изменения, выполним процедуру генерации компонента (рис. 25), и запустим процесс генерации файла прошивки и программирования (рис. 26).

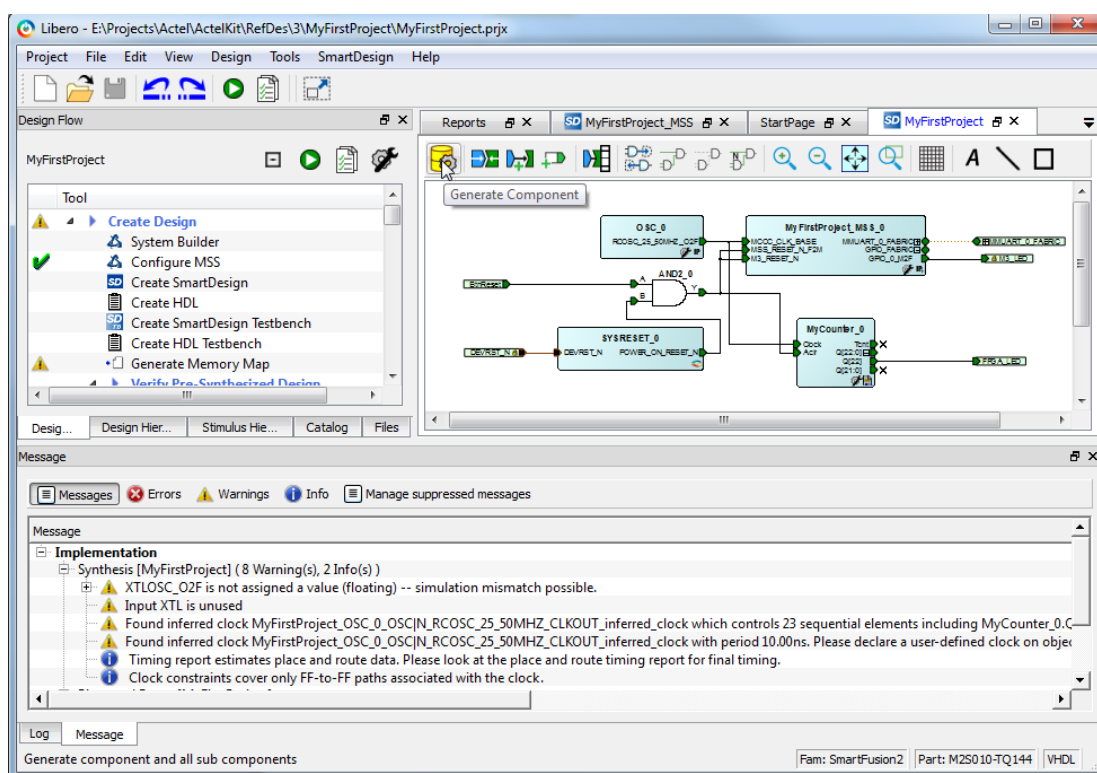


Рис. 25.

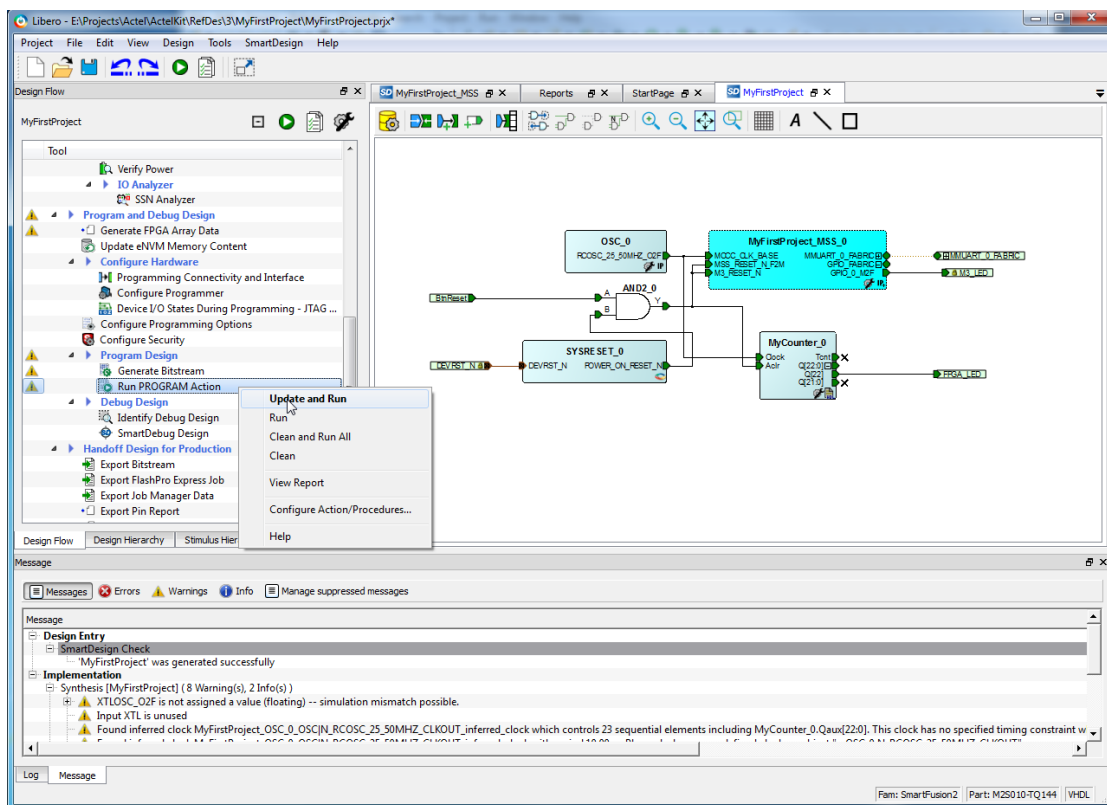


Рис. 26.

Запустим программу терминал и убедимся в корректной работе приложения.

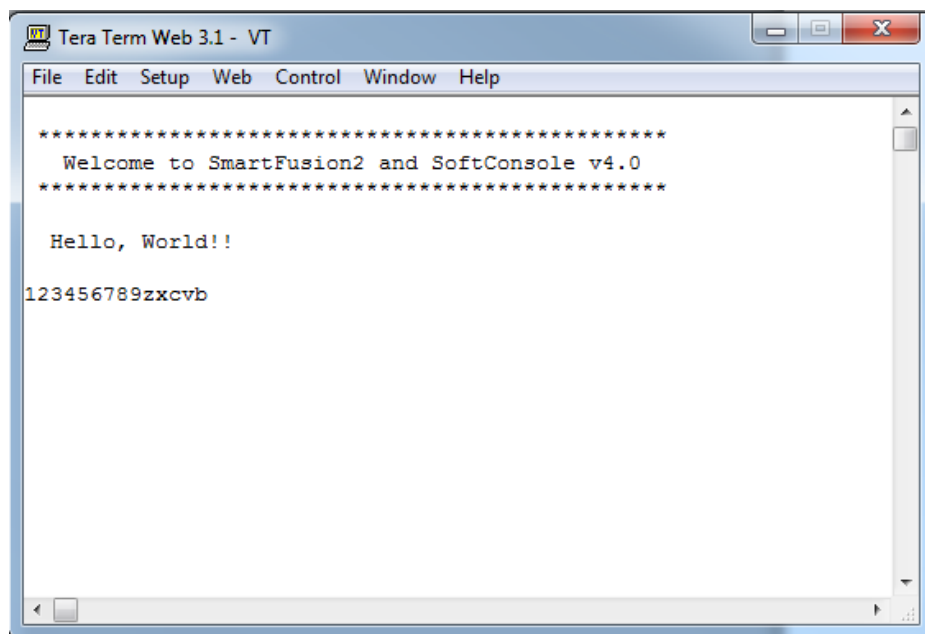


Рис. 27.

Вопросы по материалу, изложенному в данном руководстве, можно задать сотрудникам службы технической поддержки компании ООО «ПРС Актел» по телефону +7 (812) 740-60-09, или по электронной почте **support@actel.ru**.