

SF2-Junior-KIT

Использование интерфейса USB в проектах пользователя.

**Разработка устройства с интерфейсом USB,
работающим в режиме виртуального коммуникационного порта (VCP)**

Введение

В данном руководстве описано создание проекта системы на кристалле (СнК) и встроенного программного обеспечения (ВПО) для реализации устройства на основе СнК SmartFusion2 взаимодействующего с персональным компьютером через интерфейс USB в режиме виртуального коммуникационного порта (VCP).

Необходимое программное обеспечение

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

- среда разработки [Microsemi Libero SoC v11.8](#);
- 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 v3.4 или более поздняя, которая может быть установлена как часть пакета программ Microsemi Libero SoC или отдельно;
- программа-оболочка HyperTerminal или аналогичное программное обеспечение (PuTTy или TeraTerm).

Необходимое аппаратное обеспечение

Вам понадобится отладочный набор [SF2-Junior-KIT](#), включающий 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 9B 1A;
- 7) Кабель USB 2.0 A-male to mini-B.

Особенности реализации USB контроллера в СнК SmartFusion2

Микроконтроллерная подсистема СнК SmartFusin2 имеет в своем составе аппаратно реализованный контроллер шины USB 2.0 OTG с интерфейсом ULPI для подключения к микросхеме - драйверу физического уровня. Характеристики контроллера:

- Поддерживает скорости работы High-speed и Full-speed;
- Поддерживает режимы Host/peripheral, соединение point-to-point multi-point;
- Соответствует стандарту USB 2.0 для High Speed и One-To-Go операций.

Описание проекта

В качестве примера создадим проект с использованием аппаратного контроллера USB микроконтроллерной подсистемы SmartFusion2 для отладочного набора SF2-Junior-KIT с чипом M2S010-TQ144. С помощью приложения встроенного программного обеспечения продемонстрируем работу контроллера USB в режим виртуального коммуникационного порта (Virtual Com Port, VCP) интерфейса USB. Наше приложение будет принимать коды символов нажатых клавиш, приходящих по интерфейсам MMUART_0 и USB и посыпать принятые коды обратно в оба указанных интерфейса (рис. 1).

Microsemiⁱ предлагает целый ряд примеров встроенного программного обеспечения, демонстрирующего функционирование интерфейса USB в различных режимах работы. При разработке пользовательского проекта для сокращения времени проектирования будем использовать готовый пример Microsemi, для чего изменим код примера в соответствии с логикой работы разрабатываемого приложения.

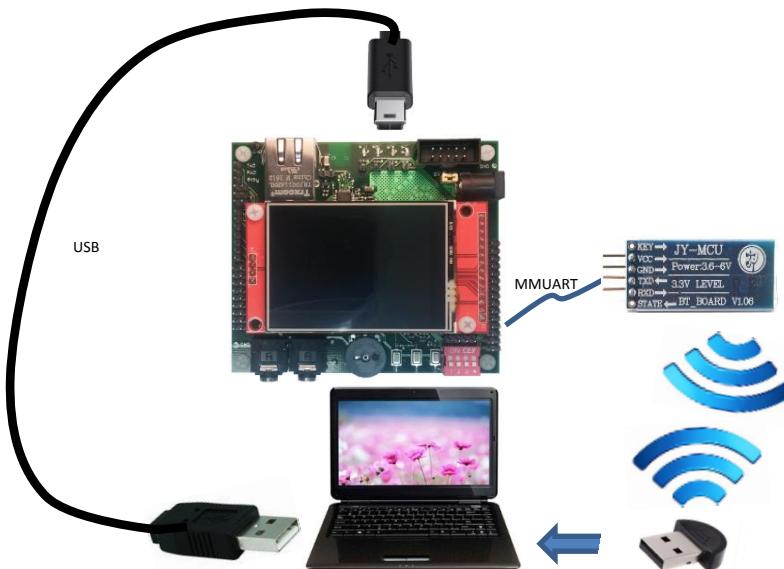


Рис. 1.

Разработка проекта системы-на-кристалле

Запустите приложение Libero SoC 11.8, дважды кликнув на ярлычок на рабочем столе или на аналогичный в меню Пуск > Все программы > Microsemi > Libero SoC v11.8.

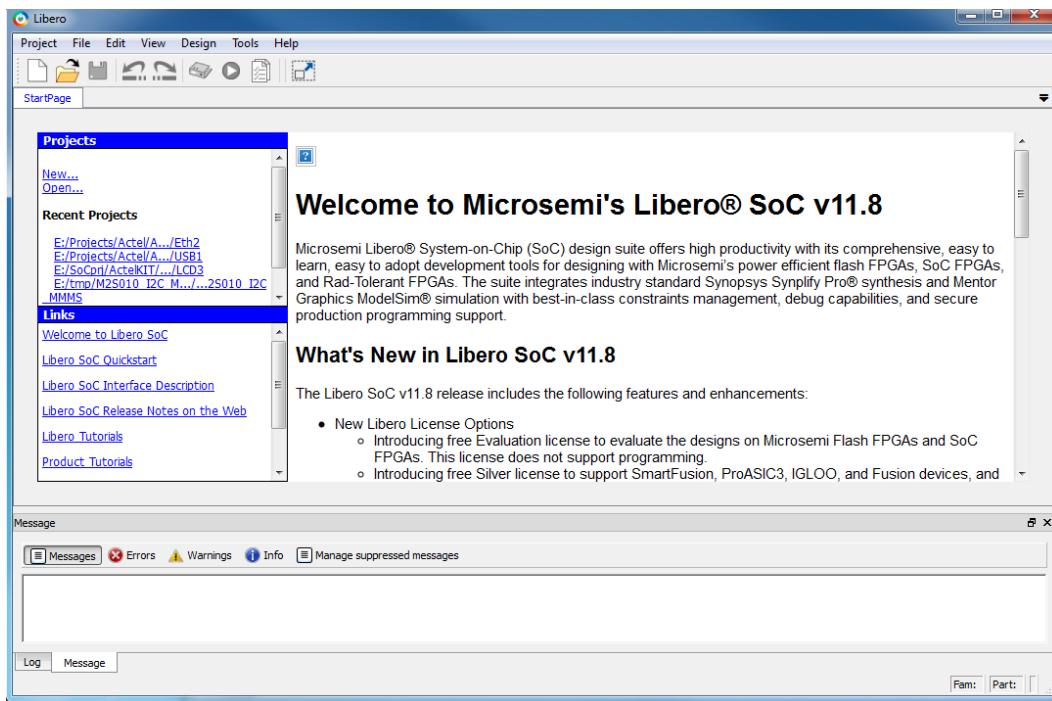


Рис. 2.

В главном меню Libero SoC (рис. 1) выполните команду **Project/New Project**, запустится мастер создания нового проекта. В появившемся окне укажите название проекта, например M2S010_USB_vcp, место расположения нашего проекта на диске и предпочтаемый язык проектирования аппаратуры – Verilog или VHDL. Галочку Enable block creation устанавливать не нужно. Нажмите кнопку «Next» (рис. 3).

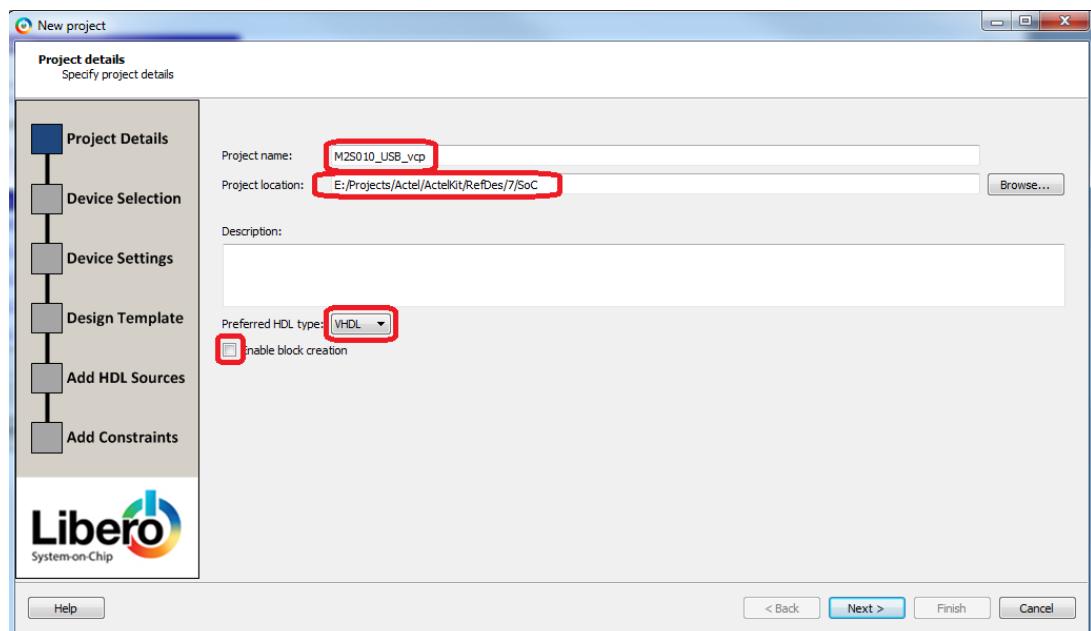


Рис. 3.

В появившемся окне «Device selection» выбором желаемых параметров в выпадающих списках укажите PartNumber микросхемы, с которой будем работать. При работе с отладочным комплектом SF2-Junior-KIT необходимо выбрать вариант M2S010-TQ144. После чего нажмите «Next».

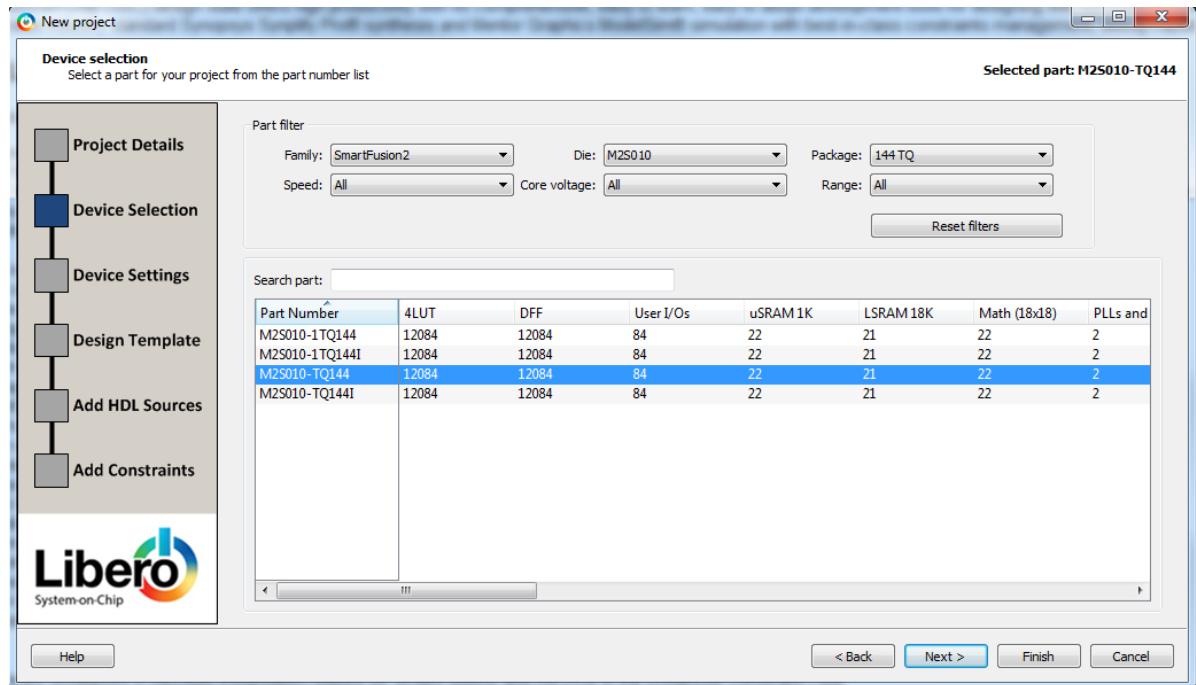


Рис. 4.

В следующем окне выберите настройки стандарта ввода-вывода по умолчанию LVCMOS 2.5V, напряжение питания PLL 2.5 V и задержку старта микросхемы после сигнала Reset 100 ms. Нажмите кнопку «Next» (рис. 5).

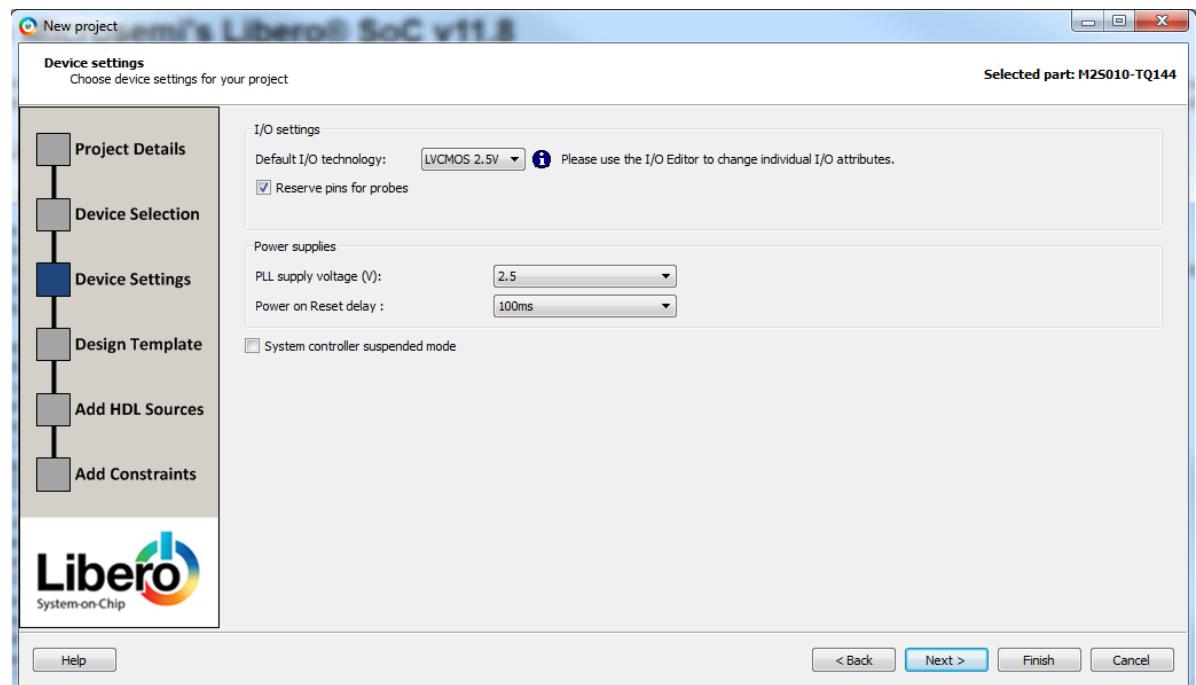


Рис. 5.

В следующем появившемся окне предлагается выбрать тип создаваемого проекта (с микроконтроллерной системой или без нее) и мастер, который будет использоваться для

настройки микроконтроллерной подсистемы. Выберем пункт «Create a microcontroller (MSS) based design», после чего нажмем «Next» (рис. 6).

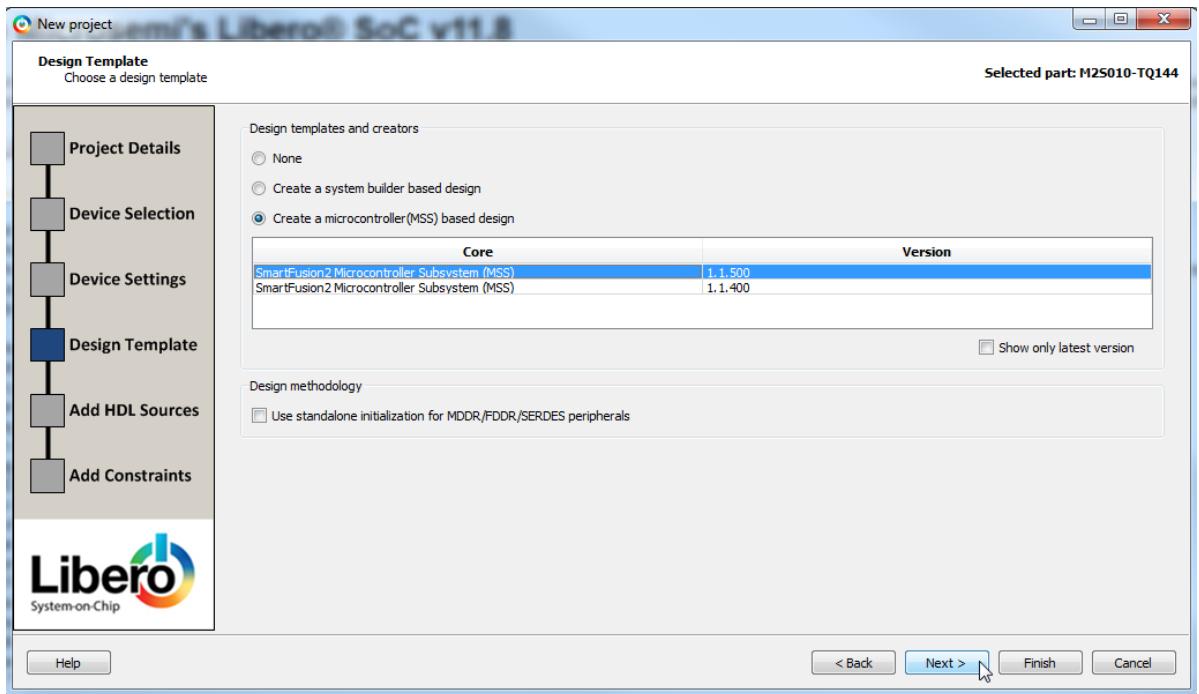


Рис. 6.

В следующих появляющихся окнах ничего не меняем, просто нажимаем «Next» до появления окна показанного на рис. 7. При выборе способа установки проектных ограничений нажмите кнопку «Use Enhanced Constraint Flow».

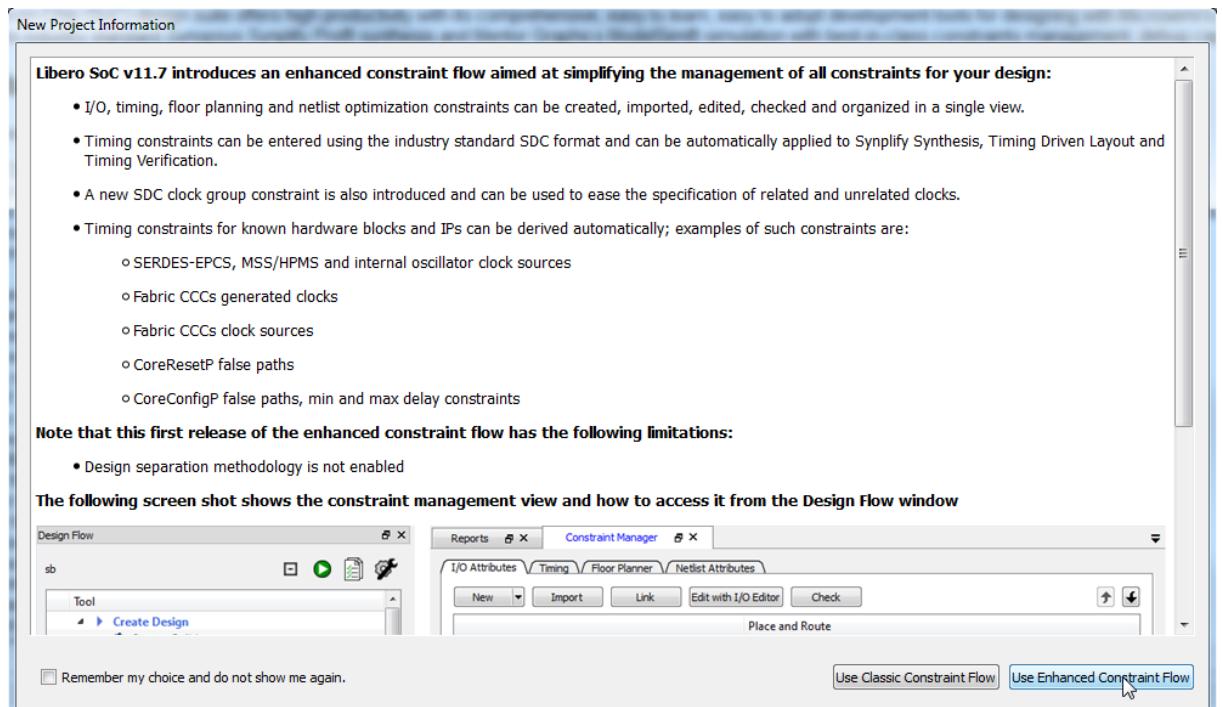


Рис. 7.

В результате выполнения описанных действий появится окно утилиты SmartDesign, в котором будет находиться один компонент M2S010_USB_vcp_MSS_0 – микроконтроллерная подсистема СнК SmartFusion2 (рис. 8).

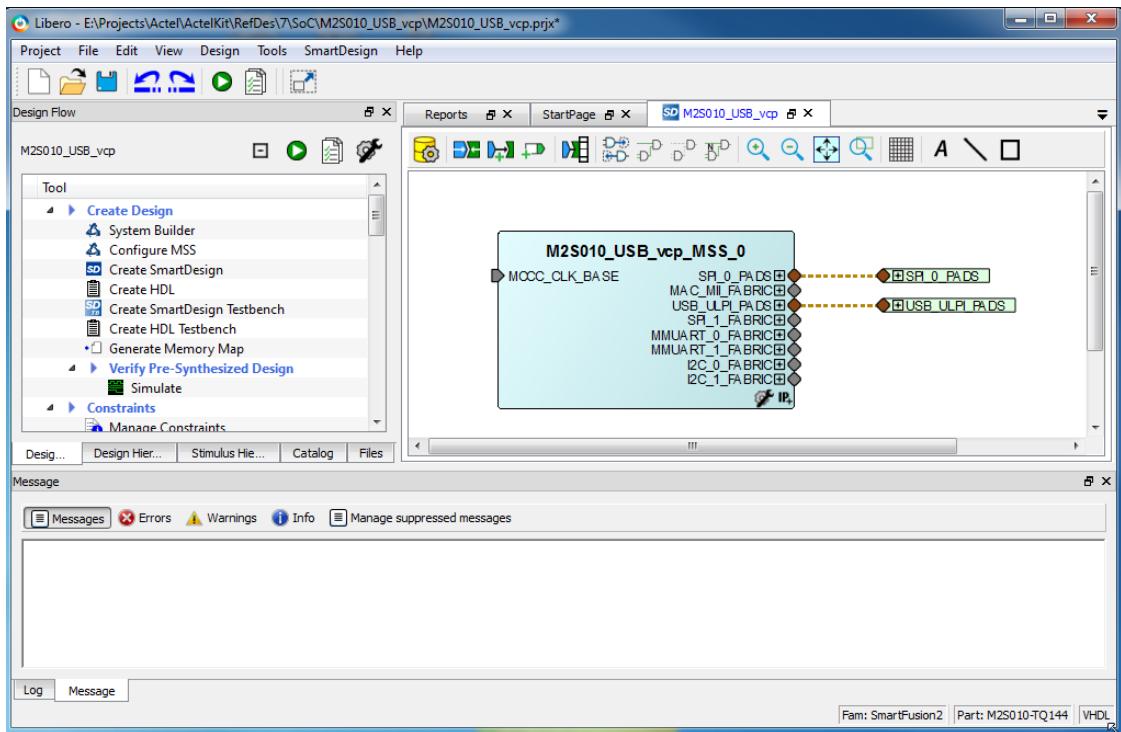


Рис. 8.

Созданный компонент M2S010_USB_vcp_MSS_0 имеет настройки микроконтроллерной подсистемы проинициализированые мастером «по умолчанию». Необходимо изменить эти настройки в соответствии с задачами, решаемыми нашим приложением.

Настроим блоки архитектуры MSS, для этого дважды щелкнем на компоненте M2S010_USB_vcp_MSS_0. Откроется окно настроек микроконтроллерной подсистемы (рис. 9).

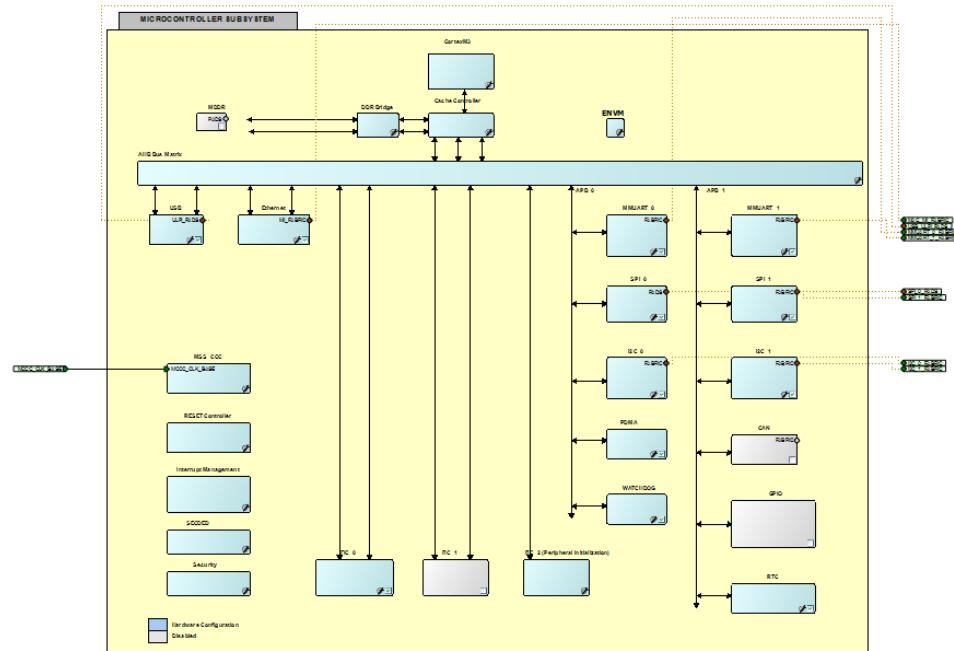


Рис. 9.

Из всего ряда доступных компонент оставим включенными блоки MMUART_0 и USB. Все остальные компоненты MSS, а именно Ethernet, MMUART_1, SPI_0, SPI_1, I2C_0, I2C_1, PDMA, CAN, WatchDog, GPIO, RTC, FIC_1 в разрабатываемом проекте задействованы не будут, их

необходимо отключить, т. е. снять галочку в правом нижнем углу перечисленных компонентов (рис. 10).

Теперь настроим используемые в нашем проекте блоки микроконтроллерной подсистемы. Начнем с контроллера сброса. Для этого дважды щелкнем на блоке RESET Controller. Выберем опции, представленные на рис. 11.

Настроим контроллер универсального приемопередатчика MMUART_0. В появившемся окне выберем следующие опции (рис. 12):

Duplex Mode: Full Duplex.

Async/Sync Mode: Asynchronous.

Use Modem Interface: снять галочку

RHD: Fabric

TXD: Fabric

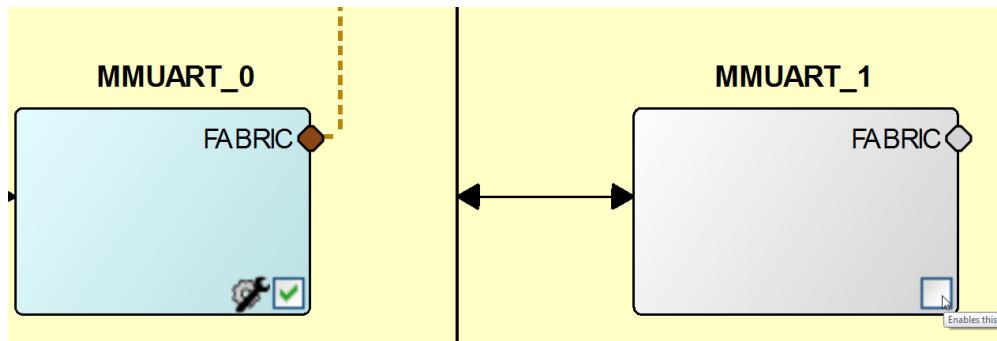


Рис. 10.

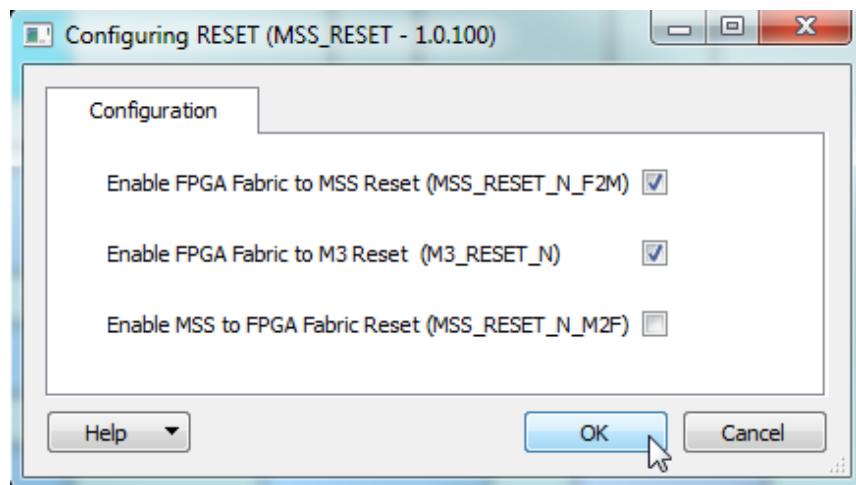


Рис. 11.

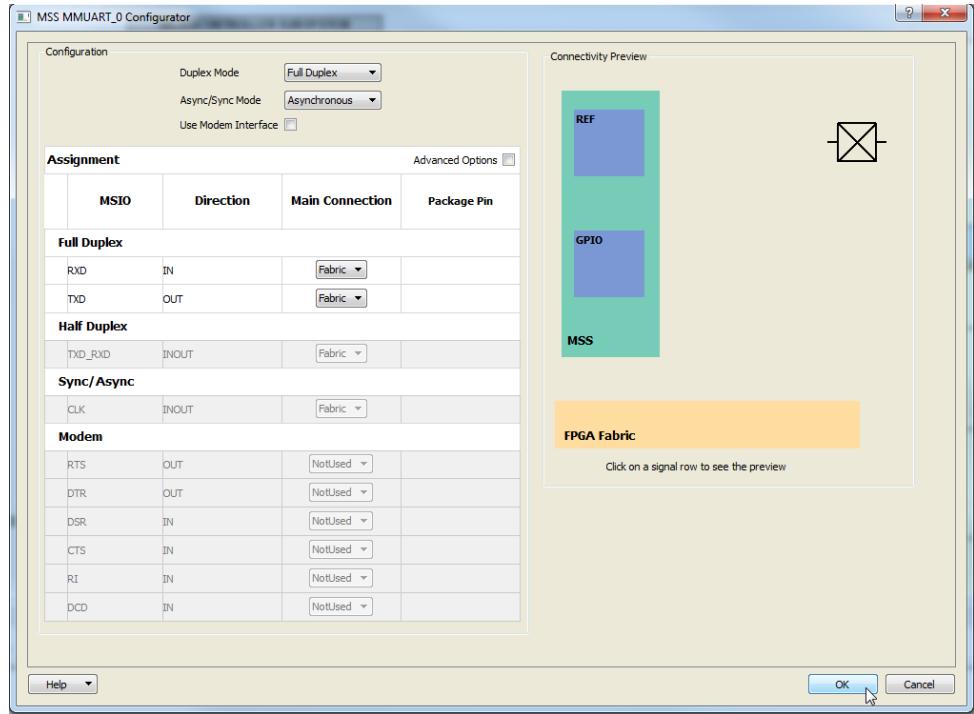


Рис. 12.

Настройки системы формирования сигналов тактирования микроконтроллерной подсистемы СнК SmartFusion2 представлены на рис. 13.

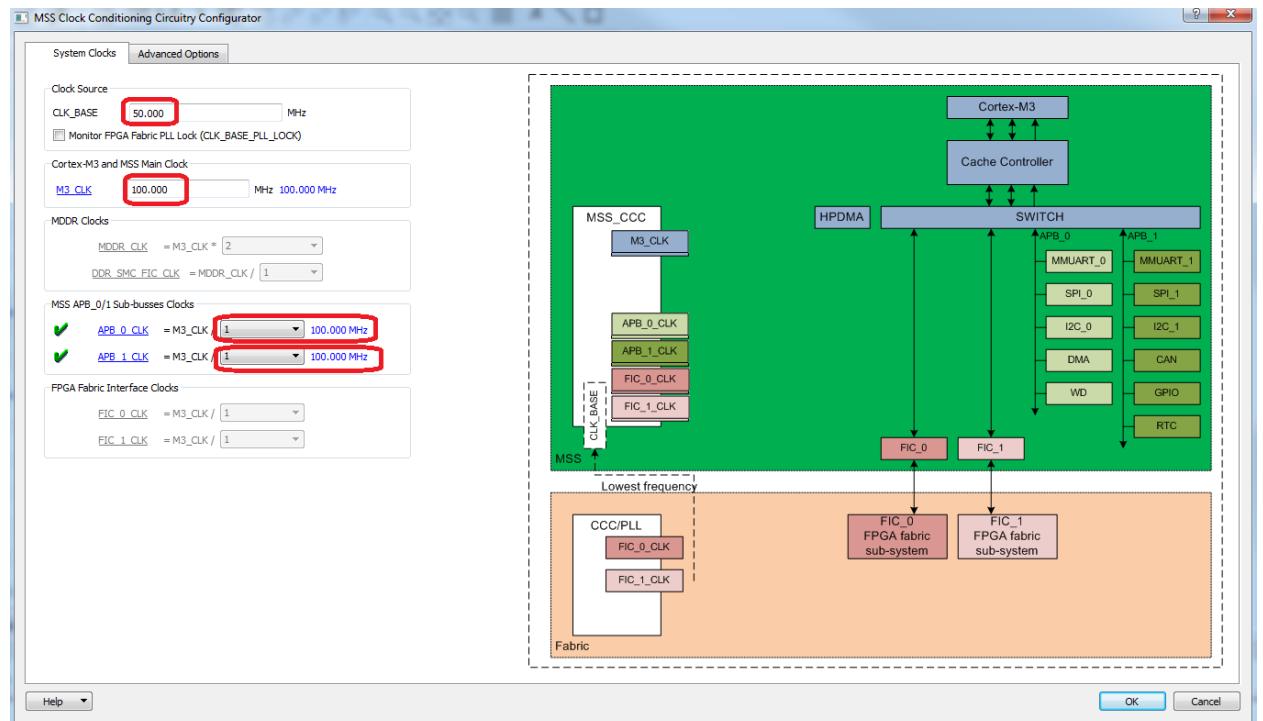


Рис. 13.

Окно настроек блока USB микроконтроллерной подсистемы представлено на рис. 14. На плате отладочного набора в качестве драйвера физического уровня интерфейса USB установлена микросхема USB3300-EZK имеющая интерфейс подключения ULPI. Необходимо выбрать соответствующий интерфейс в окне настроек. После выбора интерфейса в окне появится таблица с названиями цепей интерфейса ULPI, которым будет сопоставлен номер контакта на корпусе микросхемы. Особенностью данного интерфейса является жесткое закрепление контактов за цепями интерфейса, без возможности замены номеров контактов используя трассировочные ресурсы FPGA Fabric.

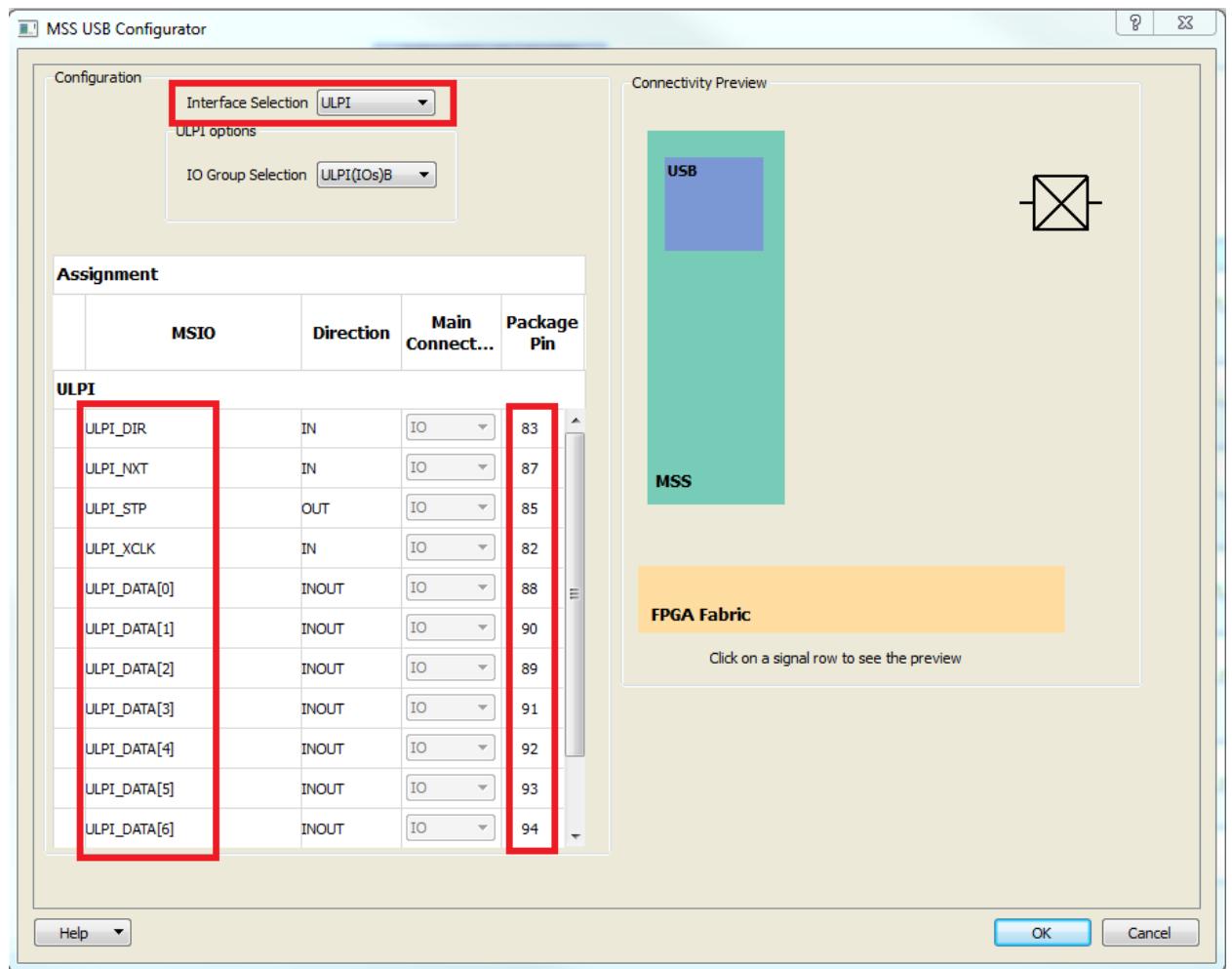


Рис. 14.

В результате описанных действий окно настроек микроконтроллерной подсистемы примет вид, представленный на рис. 15.

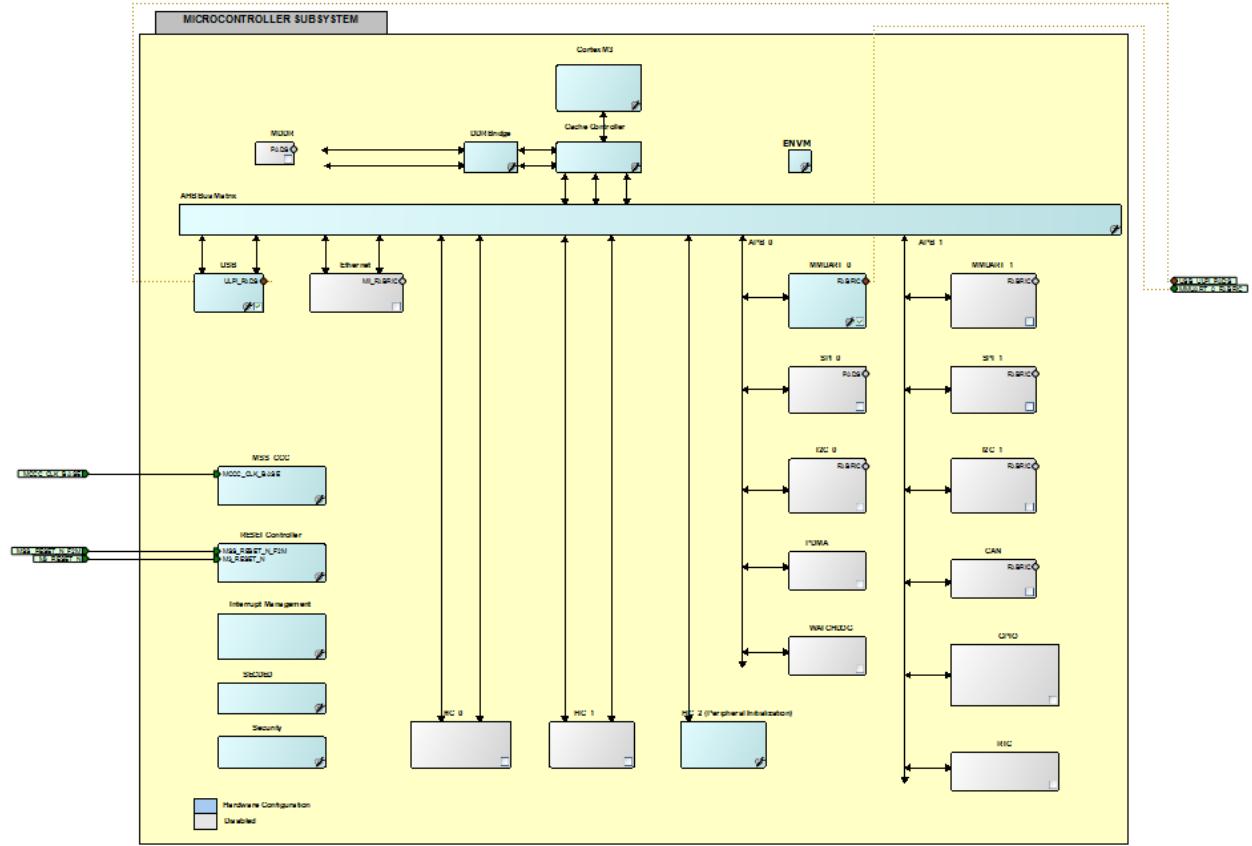


Рис. 15.

Сохраним изменения и вернемся во вкладку M2S010_USB_vcp редактора SmartDesign. Внешний вид компонента микроконтроллерной подсистемы изменился – в правом верхнем углу появился восклицательный знак на желтом фоне. Это означает, что свойства компонента изменились и его необходимо обновить. Для обновления необходимо щелкнуть по нему правой кнопкой мыши и в появившемся меню выбрать команду **Update Instance(s) with Latest Component...** (рис. 16).

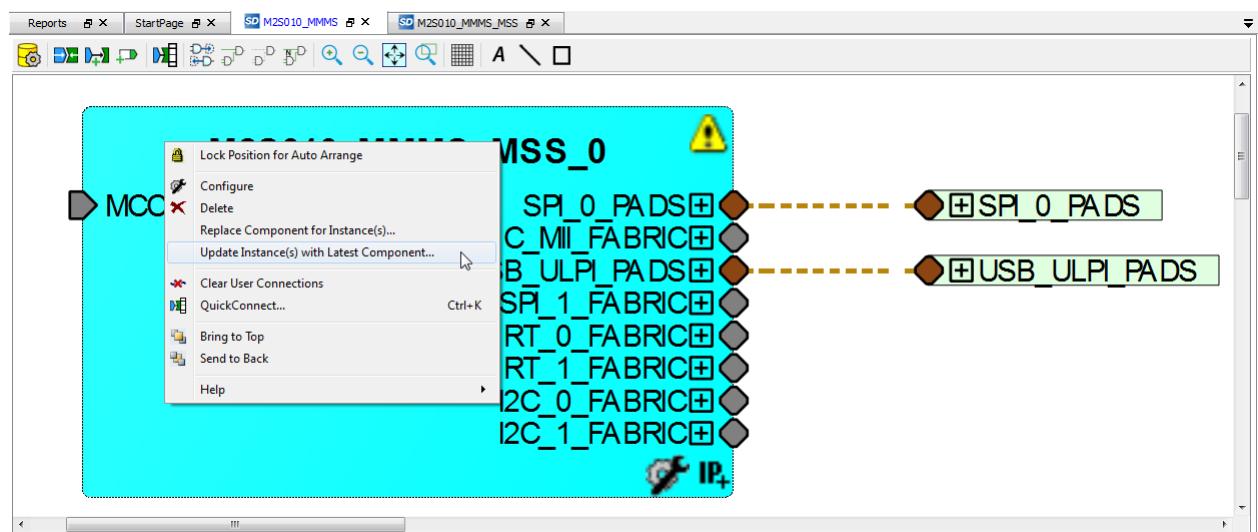


Рис. 16.

После обновления компонент должен принять вид, подобный представленному на рис. 17.

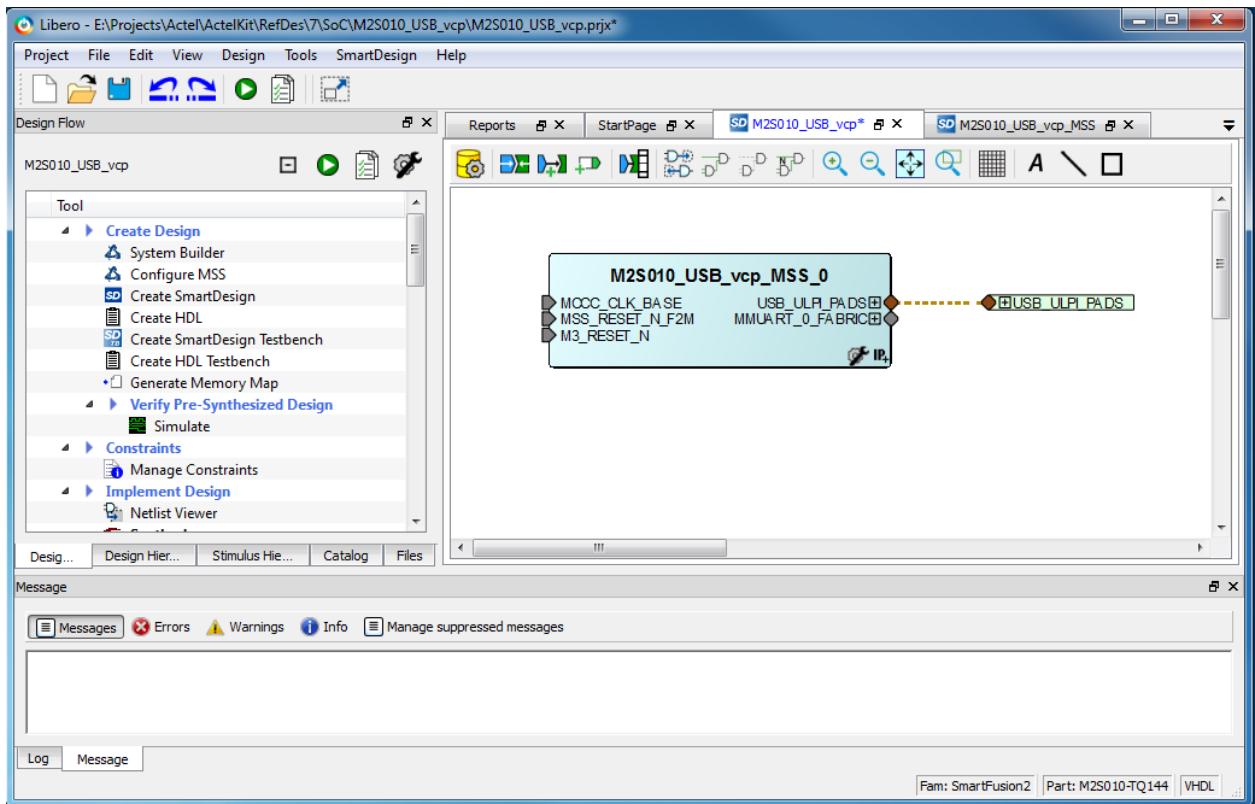


Рис. 17.

Теперь в дополнение к компоненту микроконтроллерной подсистемы из стандартного каталога Libero SoC на рабочее поле проекта нужно добавить компоненты, отвечающие за системный сброс и тактирование системы.

Для реализации описанной выше функциональности проекта необходимы компоненты стандартной библиотеки Libero SoC, указанные в таблице 1.

Таблица 1.
Ядра и компоненты стандартного каталога Libero SoC, используемые в проекте.

№ п/п	Раздел стандартного каталога Libero SoC	Название ядра/компонента в каталоге Libero SoC	Название в проекте	Количество в проекте	Назначение
1	Processors	SmartFusion2 Microcontroller Subsystem (MSS)	M2S010_int_MSS_0	1	Конфигуратор микроконтроллерной подсистемы MSS SmartFusion2
2	Macro Library	SYSRESET	SYSRESET_0	1	Генератор сигнала «Сброс»
3	Clock & Management	Chip Oscillators	OSC_0	1	Источник сигнала тактирования
4	Macro Library	AND2	AND2_0	1	Сброс системы по нажатию кнопки

Для использования нужного компонента в проекте необходимо перейти во вкладку Catalog, раскрыть нужный раздел каталога и мышью перетащить компонент на рабочее поле проекта (рис. 18).

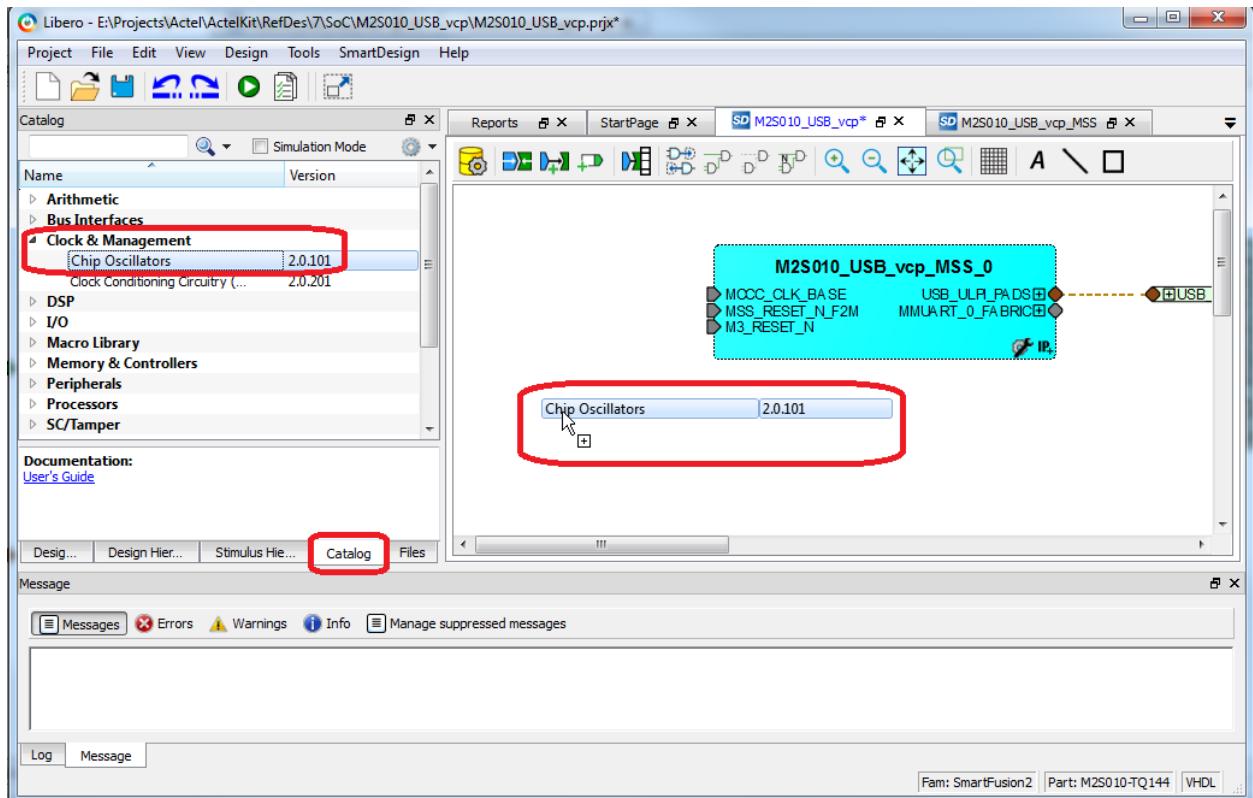


Рис. 18.

Параметры настройки источника сигнала тактирования приведены на рис. 19. Будем использовать внутренний тактовый RC-генератор опорных колебаний на 50 МГц.

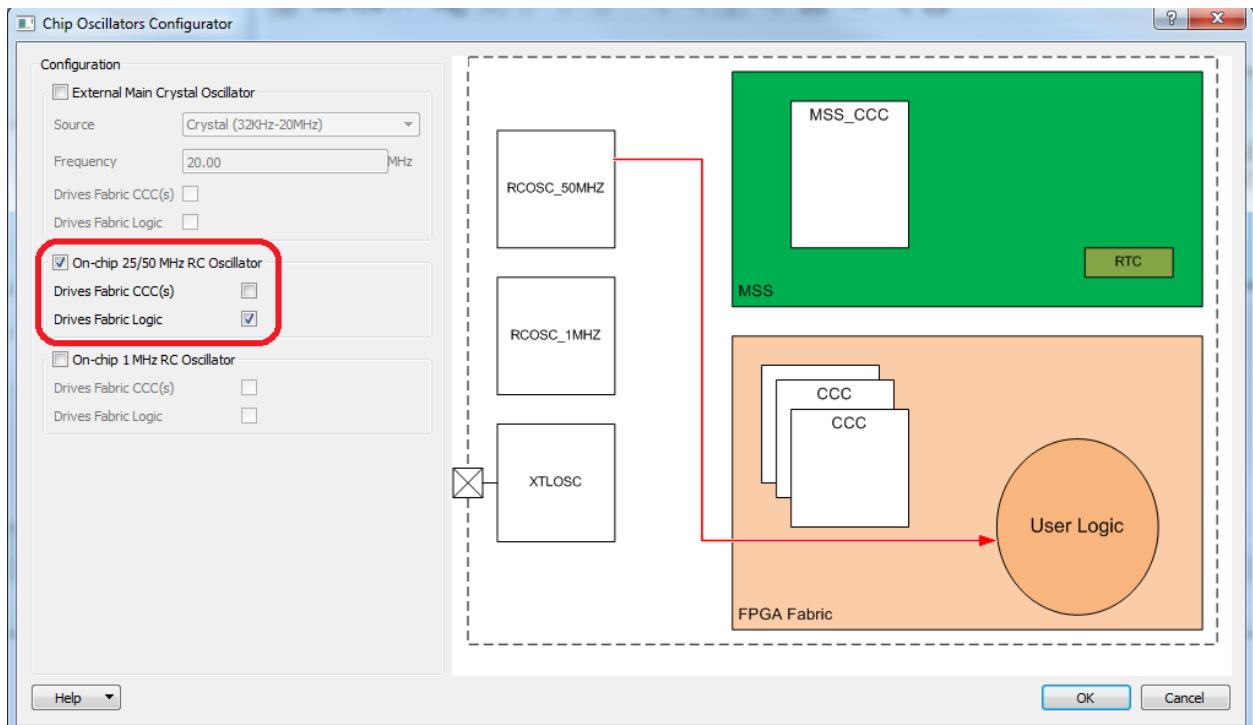


Рис. 19.

Выполнить соединения контактов компонентов в Libero SoC можно несколькими способами:

- 1) При нажатой клавише <Ctrl> левой кнопкой мыши выделить два или более соединяемых общей цепью контактов, щелкнуть правой кнопкой мыши на одном из контактов и в появившемся меню выбрать Connect (рис. 20);
- 2) Используя быструю клавишу Connection Mode (рис. 21);
- 3) Используя мастер соединений QuickConnect (рис. 22.).

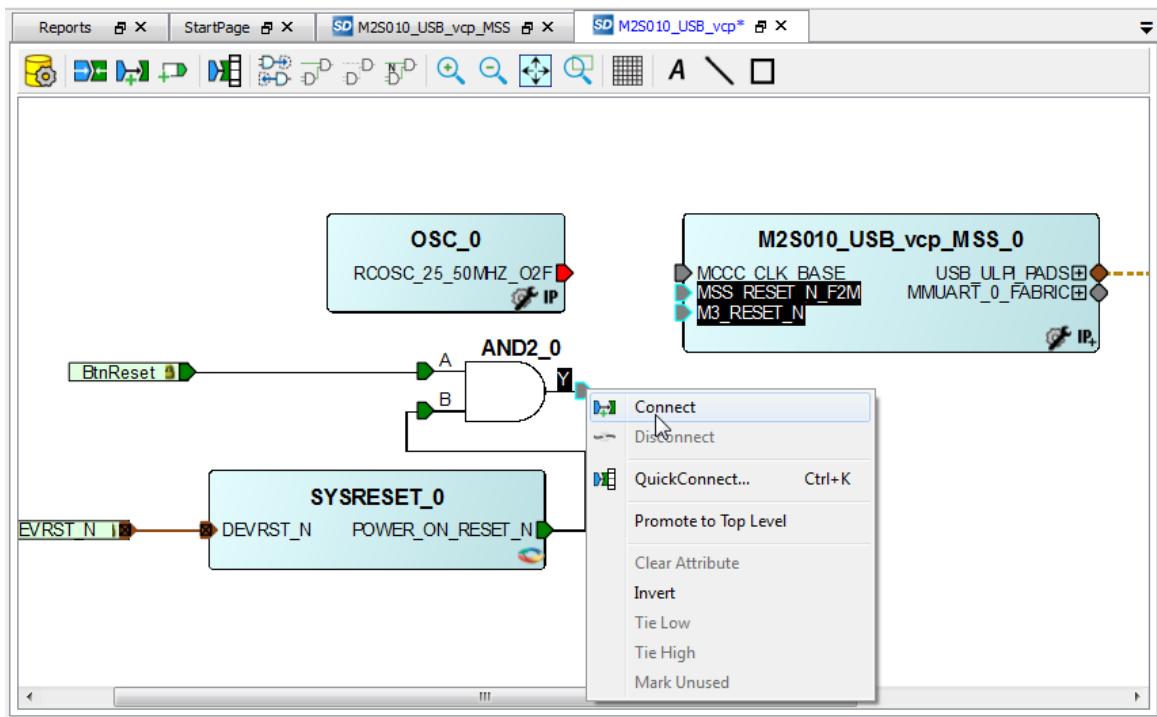


Рис. 20.

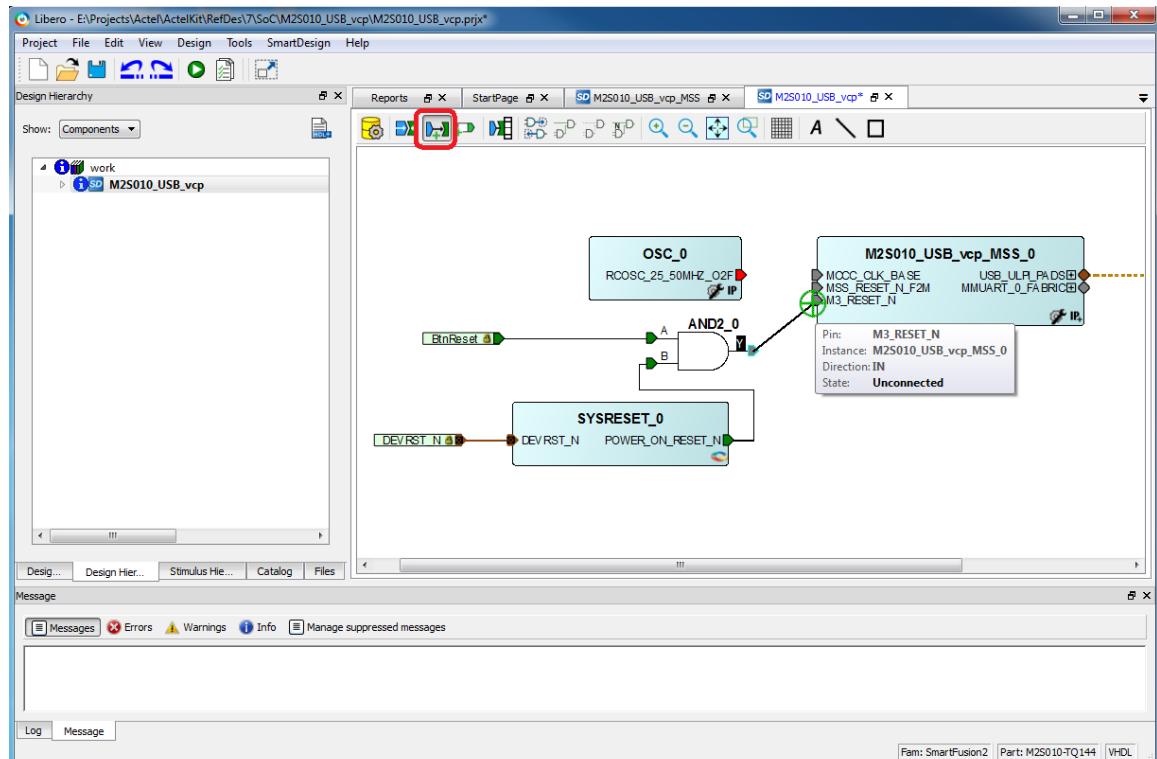


Рис. 21.

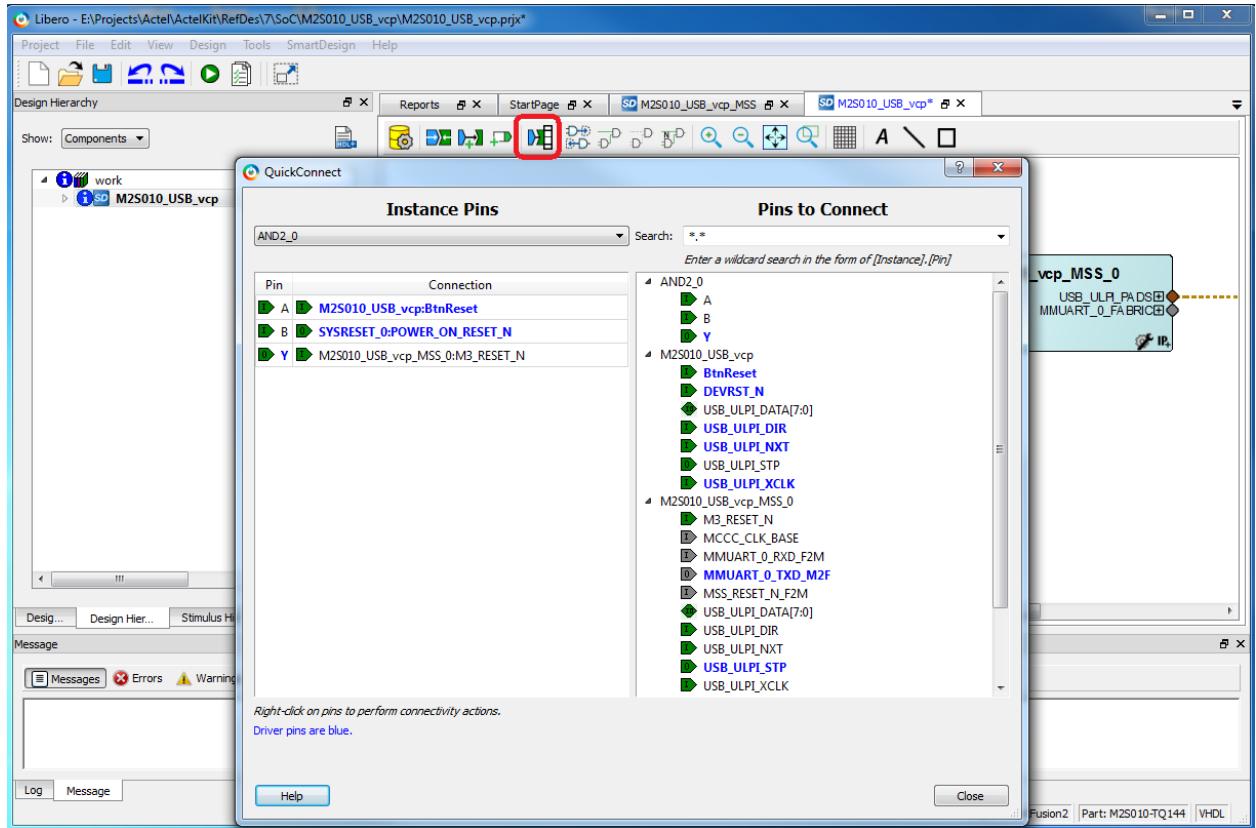


Рис. 22.

Используя описанные выше способы, выполним соединения. Ниже приведен список соединений проекта в формате Компонент.Контакт – Компонент.Контакт:

1. OSC_0.RCOSC_25_50MHZ_O2F – M2S010_USB_vcp_MSS_0.MCCC_CLK_BASE.
2. SYSRESET_0.POWER_ON_RESET_N – AND2_0.B.
3. AND2_0.Y – M2S010_USB_vcp_MSS_0.M3_RESET_N.
4. AND2_0.Y – M2S010_USB_vcp_MSS_0.MSS_RESET_N_F2M.

Контакты M2S010_USB_vcp_MSS_0.MMUART_0_FABRIC, AND2_0.A выводим на верхний уровень, т.е. щелкаем на контакте правой кнопкой мыши и в появившемся меню выбираем команду **Promote to Top Level**.

Контакты M2S010_USB_vcp_MSS_0.ULPI_PADS выведены на верхний уровень проекта и выделены коричневым цветом. В данном случае коричневый цвет означает, что используемые контакты микросхемы жестко закреплены за интерфейсом ULPI и назначить другие контакты для этого интерфейса не получится.

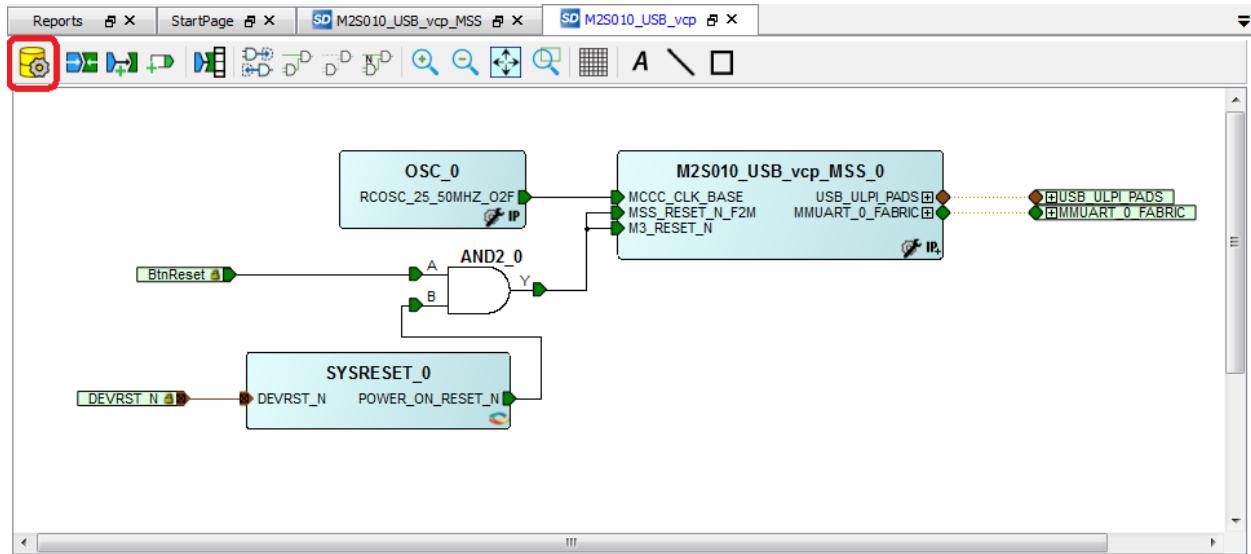


Рис. 23.

Сохраняем изменения и выполняем команду **Generate Component**.

Выполним синтез проекта (рис. 24) и назначим контакты микросхемы входным и выходным сигналам нашего проекта для чего выполним команду **Manage Constraints** (рис. 25). В появившейся вкладке **Constraints Manager** выполним команду **Edit > Edit with I/O Editor** (рис. 26). В открывшемся окне утилиты I/O Editor назначаем контакты микросхемы M2S010-TQ144 входящим и выходящим сигналам нашего проекта (рис. 27). Сохраним изменения и закроем I/O Editor.

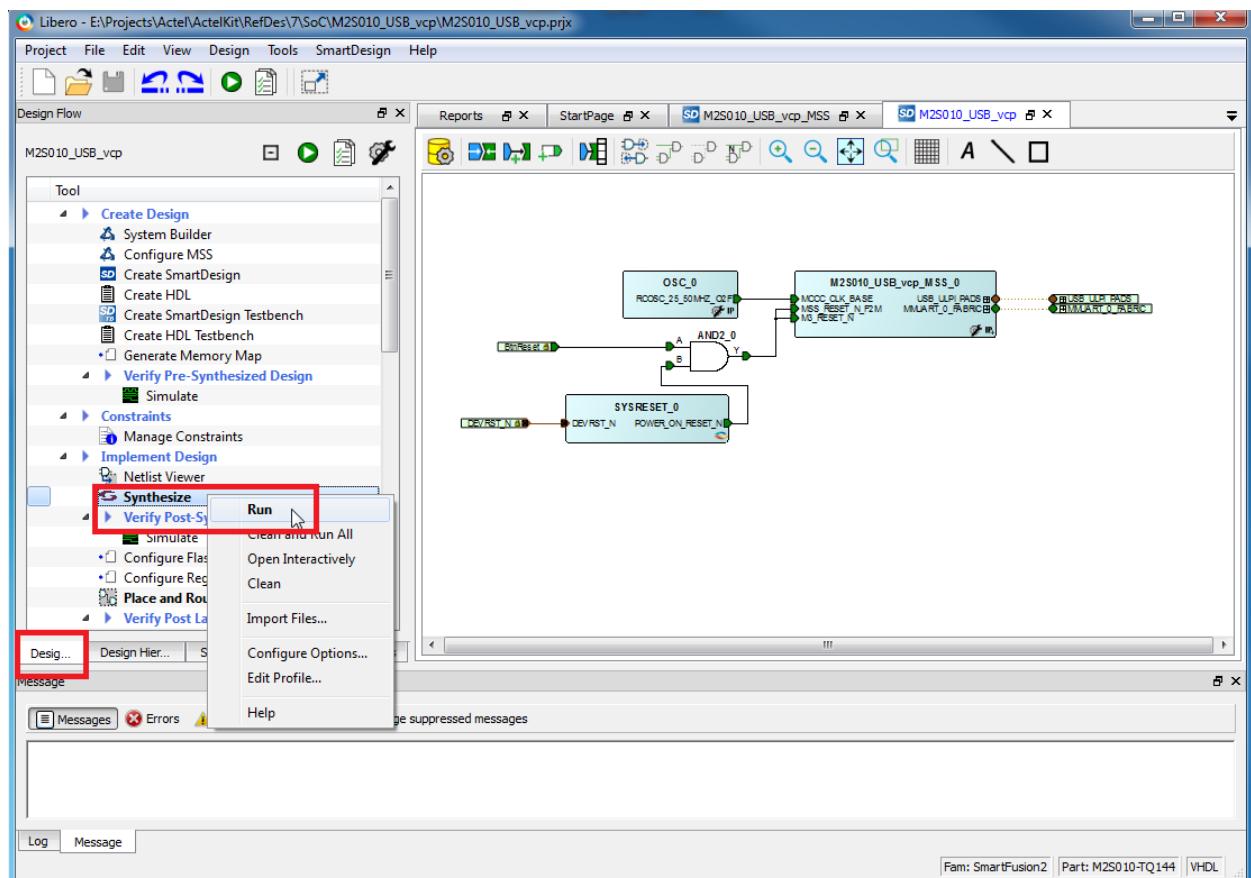


Рис. 24.

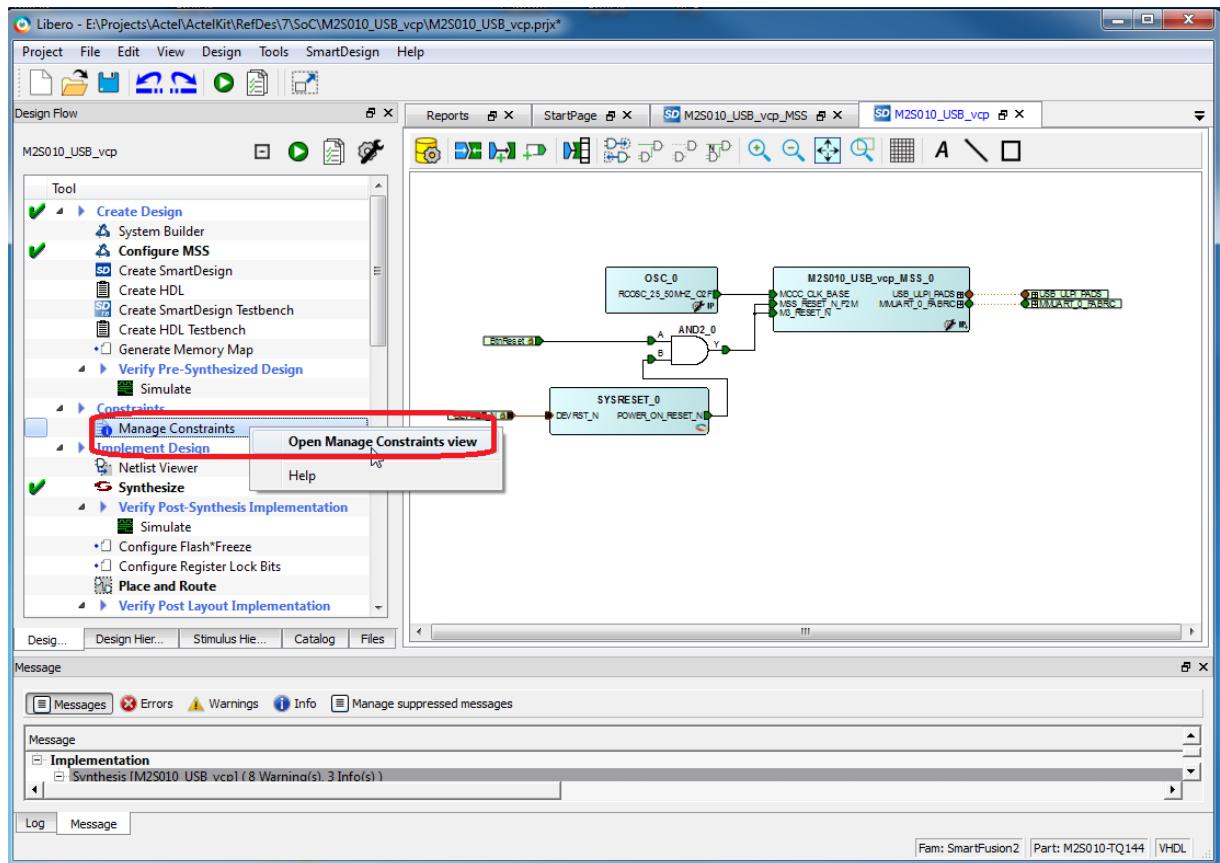


Рис. 25.

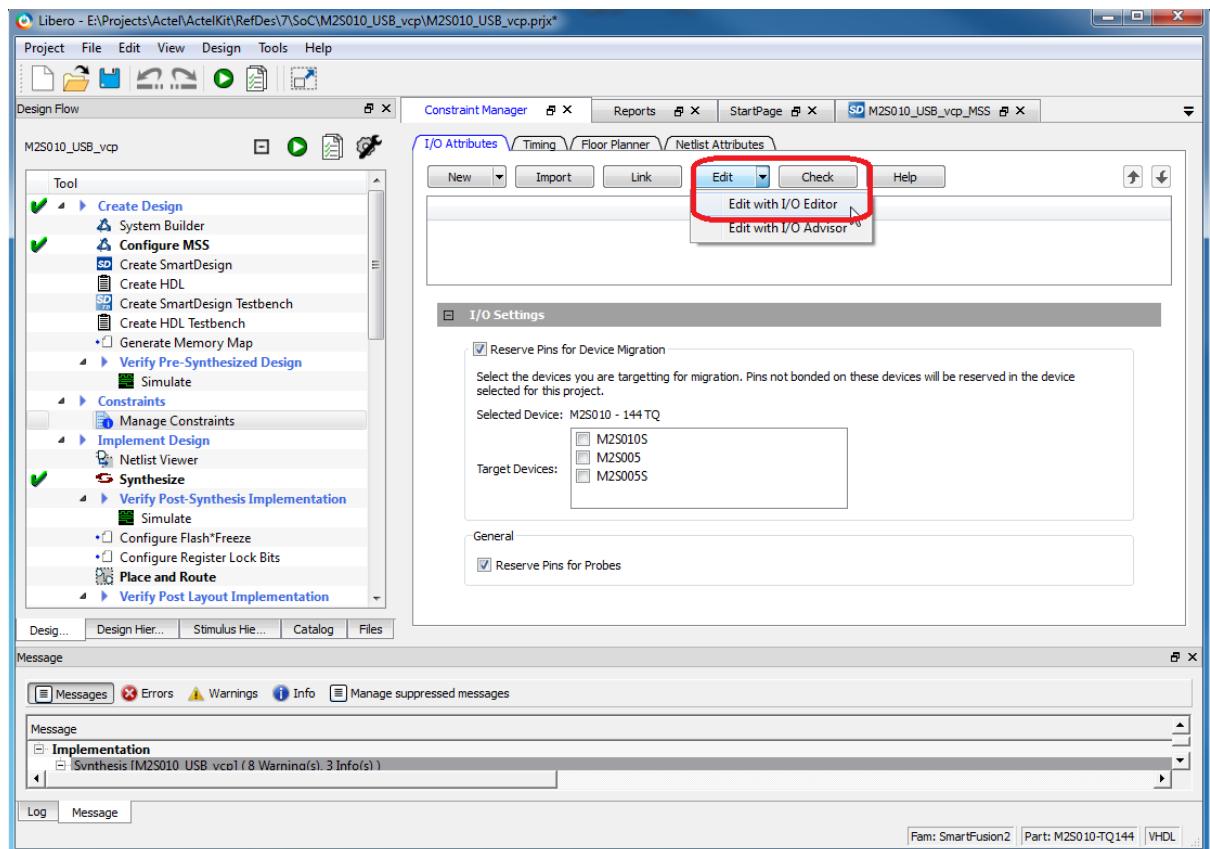


Рис. 26.

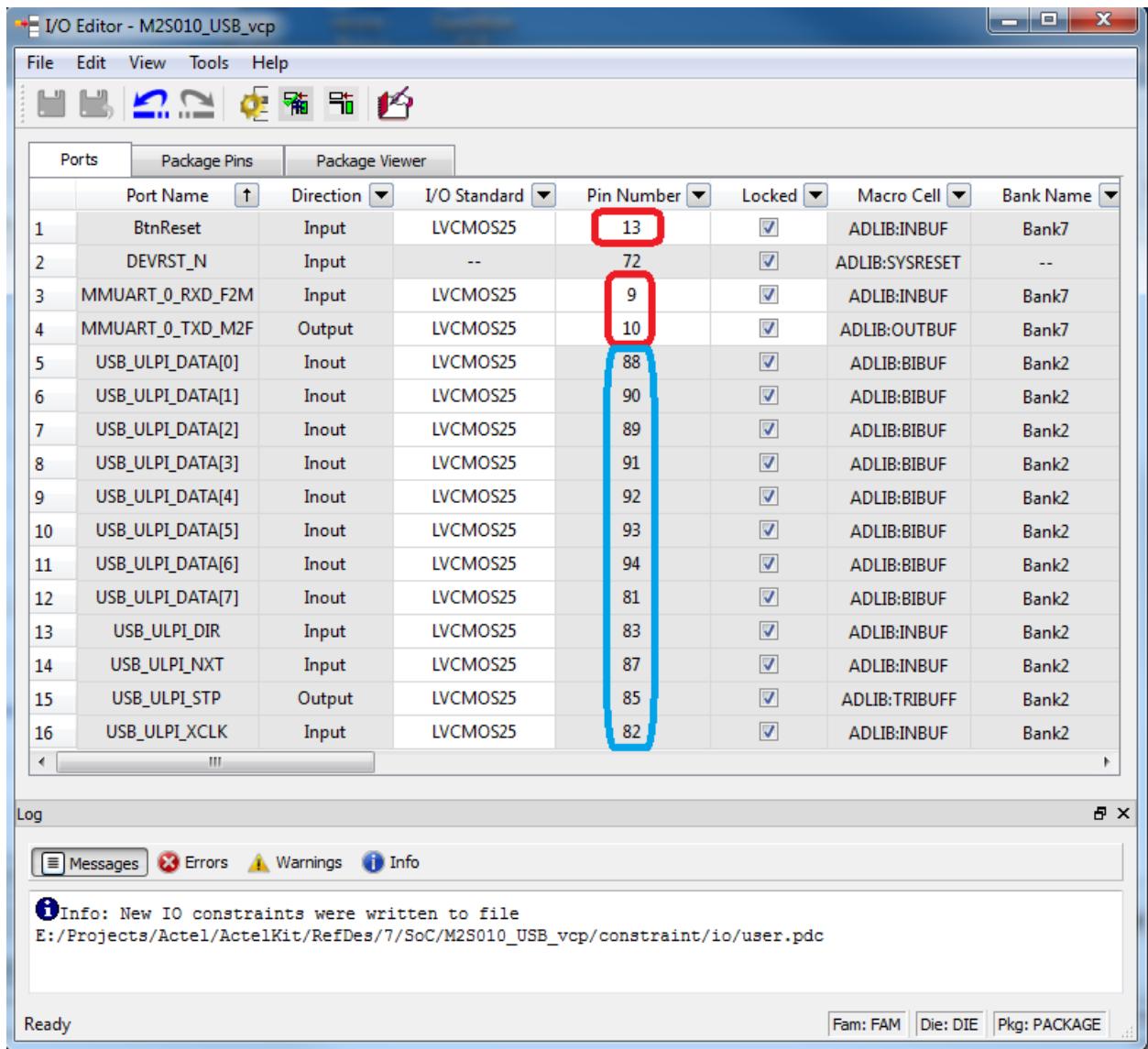


Рис. 27.

Фактически назначить номера контактов необходимо только трем цепям: BtnReset, MMUART_RXD_F2M и MMUART_TXD_F2M, т.к. физические контакты цепям интерфейса ULPI уже выделены и поменять их на другие не получится.

Выполним команду **Configure Firmware Cores** во вкладке **Design Flow** (рис. 28).

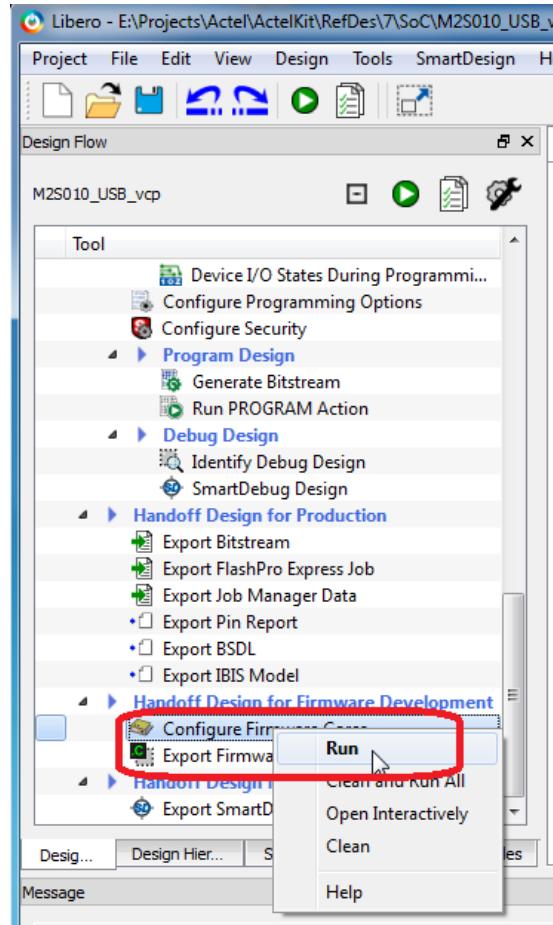


Рис. 28.

В случае если во вкладке **DESIGN_FIRMWARE** один или несколько драйверов отображены курсивом, необходимо нажать кнопку **Download All Firmware** (рис. 29) для загрузки отсутствующих драйверов по сети Internet с сайта производителя (для успешной загрузки компьютер должен быть подключен к сети Интернет).

Теперь создадим проект встроенного программного обеспечения (ВПО) процессора ARM Cortex-M. Воспользуемся готовым примером проекта встроенного программного обеспечения, для чего нажмем правую кнопку мыши на драйвере USB и выберем интересующий нас проект ВПО по работе USB в режиме виртуального ком-порта (рис. 30). В результате описанных действий в папке проекта СнК появится директория \SoftConsole в которой окажутся файлы проекта ВПО.

По умолчанию пример проекта ВПО создается для чипа M2S090TS-FG484 отладочного набора [SmartFusion2 Security Evaluation KIT](#). В случае использования иного набора с другим чипом необходимо заменить драйверы и файлы настроек конфигурации ВПО на аналогичные, созданные для проекта с актуальным для данного набора чипом. В данном руководстве рассматривается создание проекта для отладочного набора [SF2-Junior-KIT](#) с чипом M2S010-TQ144. Информацию о выбранном при создании проекта чипе можно получить, выполнив команду основного меню **Project > Project Settings** (рис. 31). Для получения файлов драйверов и файлов аппаратной конфигурации выбранного чипа необходимо выполнить команду **Export Firmware** (Рис. 32). В появившемся окне **Export Firmware** выбрать предпочтительную среду разработки и пункт **Export hardware configuration and firmware drivers**, говорящий о том, что нам нужны только файлы аппаратной конфигурации и драйверов, а создание проекта ВПО на данном этапе не входит в наши планы (рис. 33).

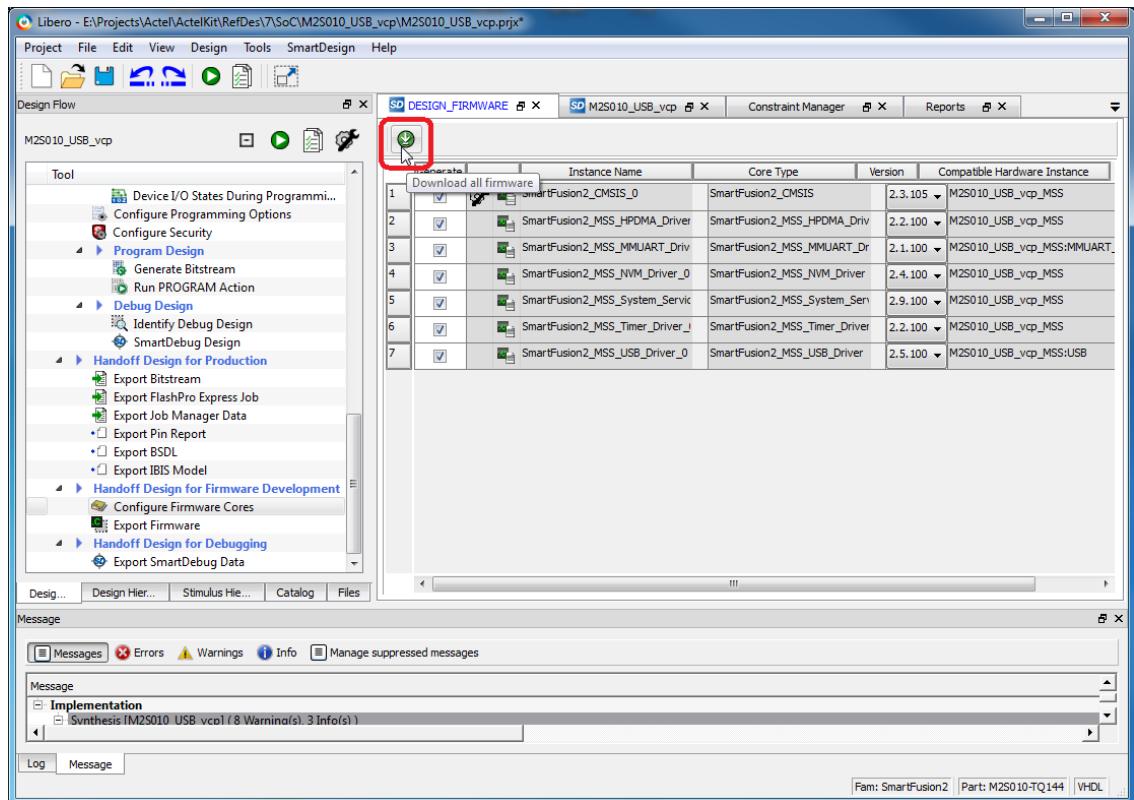


Рис. 29.

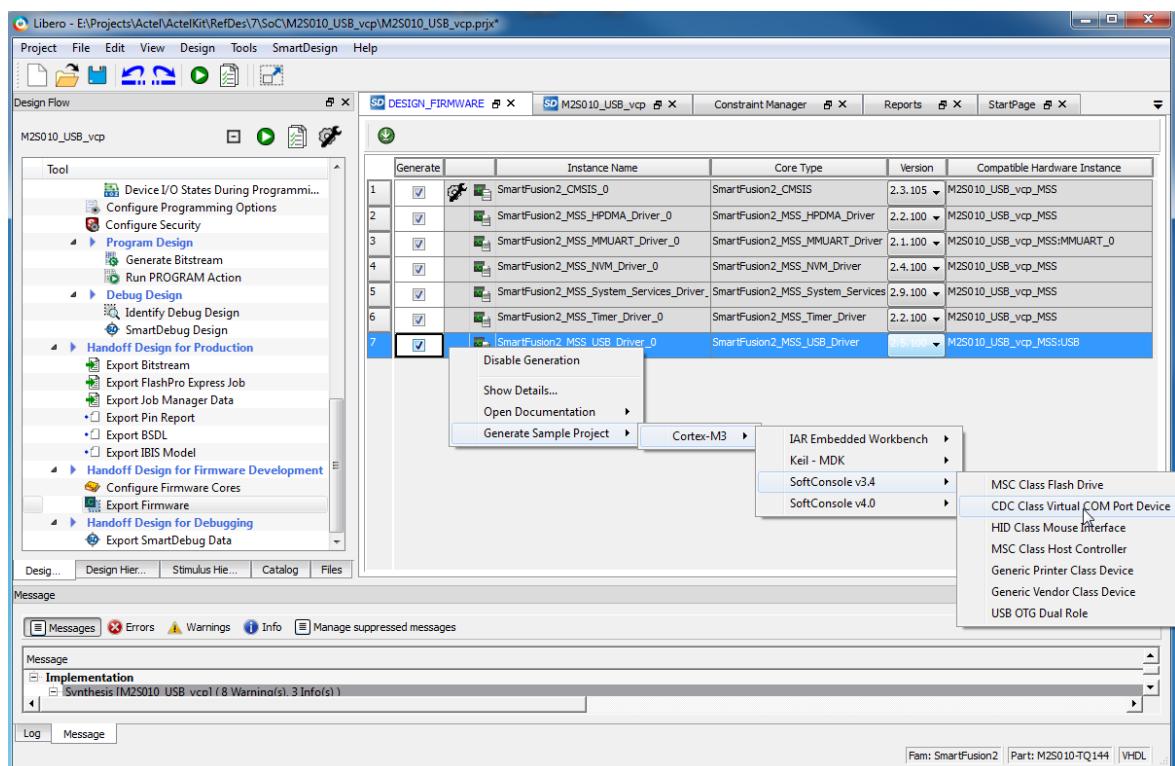


Рис. 30.

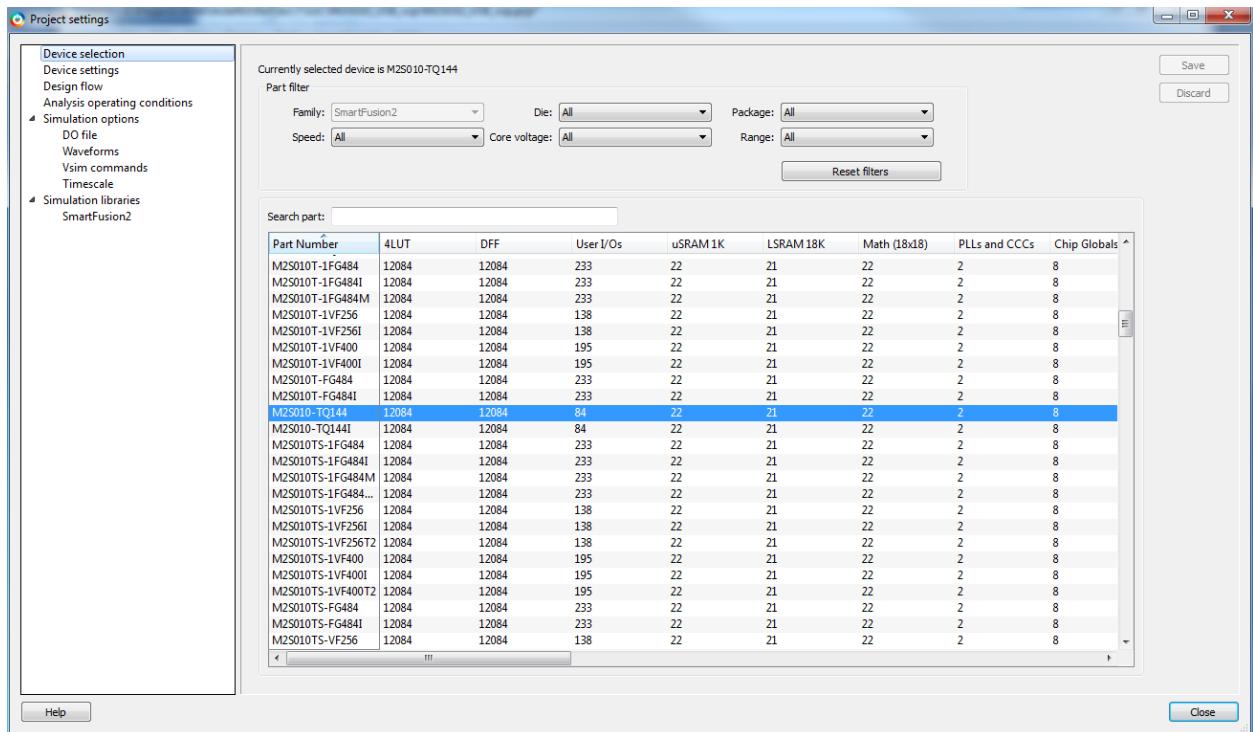


Рис. 31.

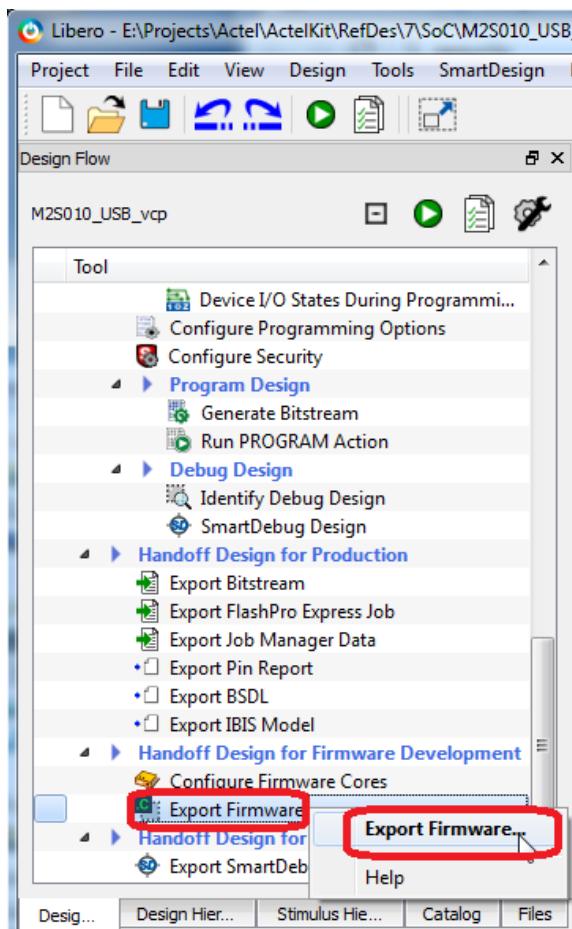


Рис. 32.

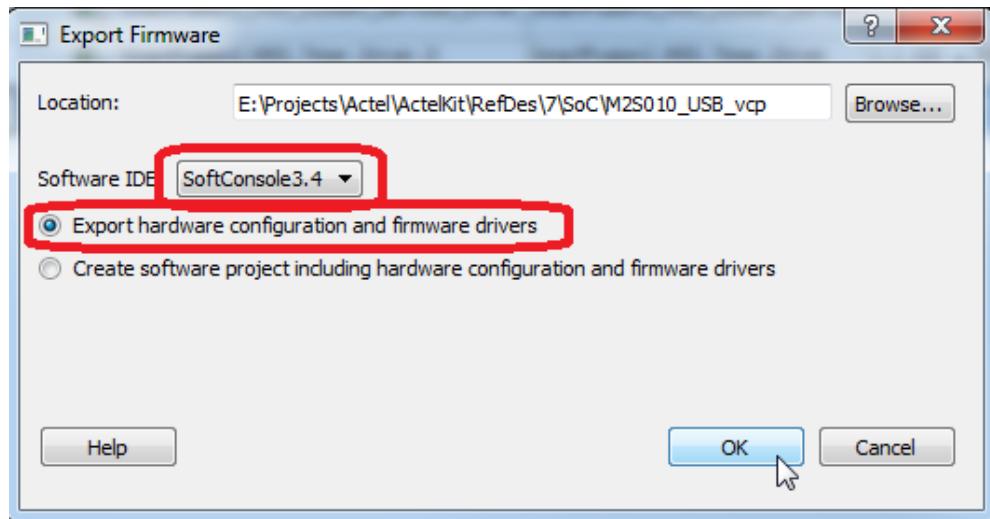


Рис. 33.

В результате выполнения команды в папке проекта СнК появится каталог **\firmware**, в котором окажутся все необходимые нам файлы драйверов и аппаратной конфигурации (рис. 34).

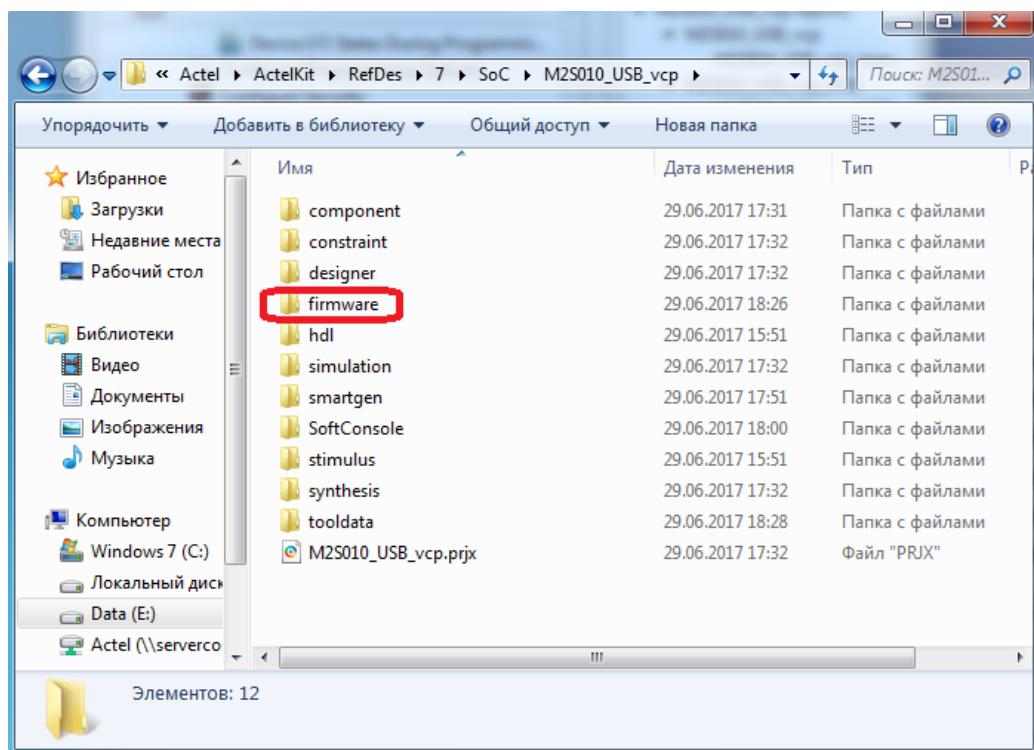


Рис. 34.

Заменим файлы и каталоги ВПО нашего проекта **\SoftConsole\CM3_GNU_SC3_usb_device_vcp** аналогичными из каталога **\firmware** тем самым получим пример проекта ВПО от Microsemi, адаптированный для выбранного нами чипа M2S010-TQ100.

Модификация кода встраиваемого приложения

Переходим к написанию кода встраиваемого приложения. Встроенное программное обеспечение нашего проекта должно выполнить следующие шаги:

- 1) Отправить приветственные сообщения двум окнам программы-терминала (первое окно связано с MMUART_0 микроконтроллерной подсистемы, второе связано с интерфейсом USB работающим в режиме VCP);
- 2) Принимать коды нажатых клавиш от двух окон программ-терминалов;
- 3) Отображать символы нажатых клавиш в двух окнах программ-терминалов.

Запустим среду разработки SoftConsole v. 3.4 (рис. 35).

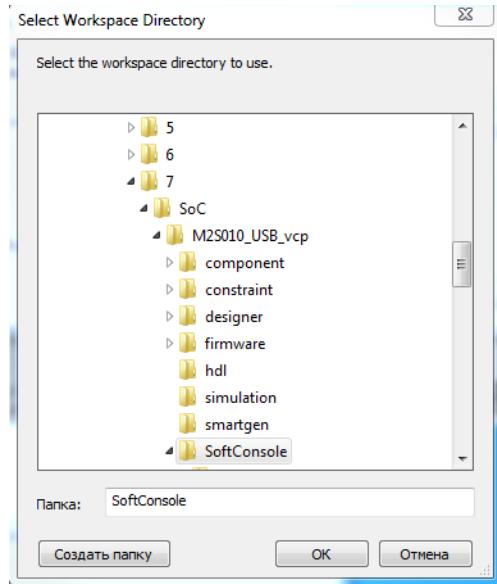


Рис. 35.

Если при открытии SoftConsole главное окно окажется пустым (Рис. 36), необходимо выполнить команду основного меню **File>Import**, в появившемся окне выбрать опцию **General>Existing Projects into Workspace**

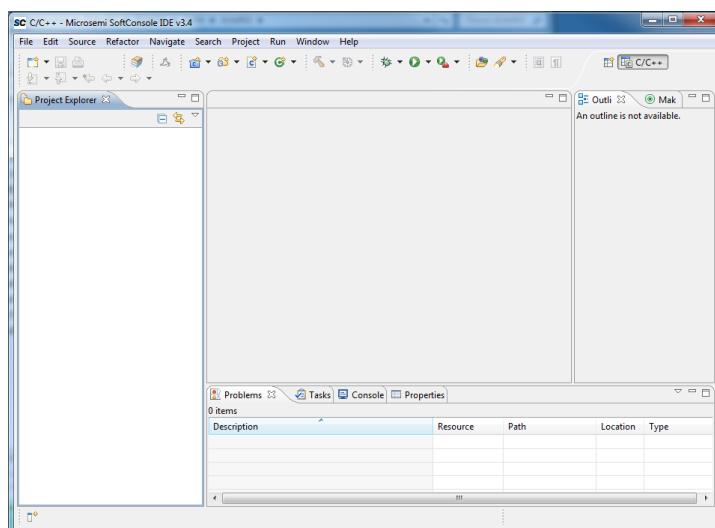


Рис. 36.

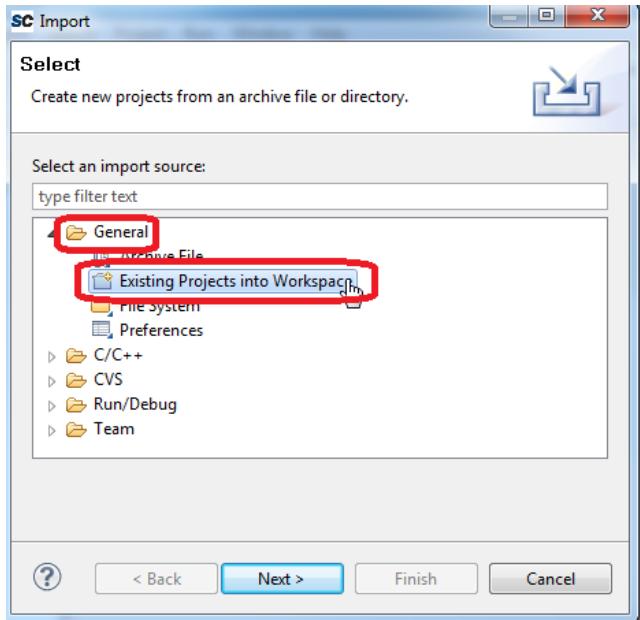


Рис. 37.

И в следующем появившемся окне еще раз указать путь к проекту ВПО (рис. 38).

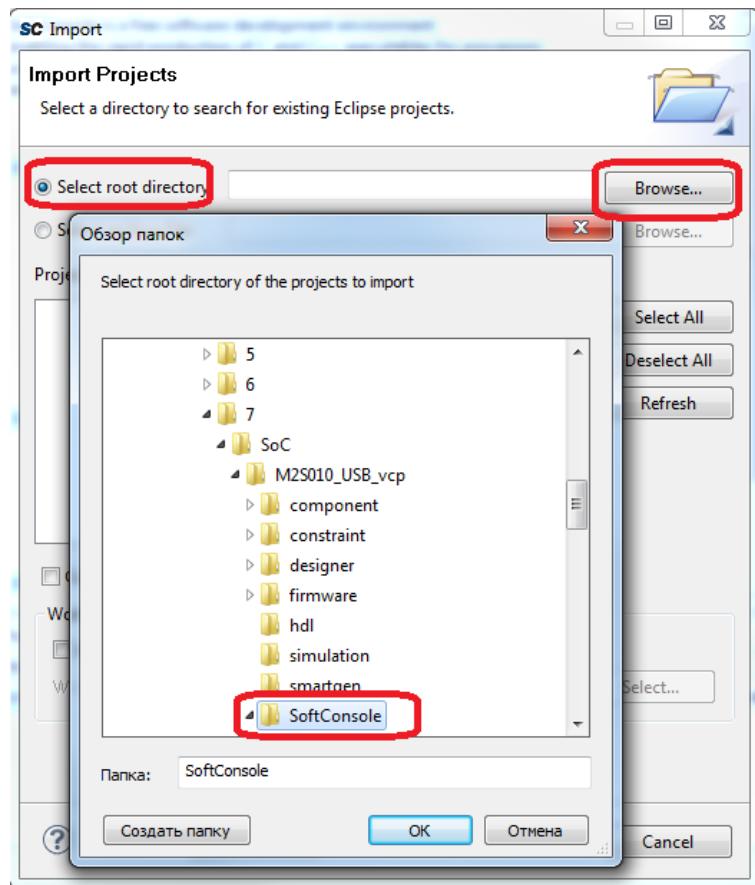


Рис. 38.

В результате демонстрационный проект ВПО окажется загруженным в среду разработки SoftConsole v3.4, главное окно примет вид аналогичный показанному на рис. 39.

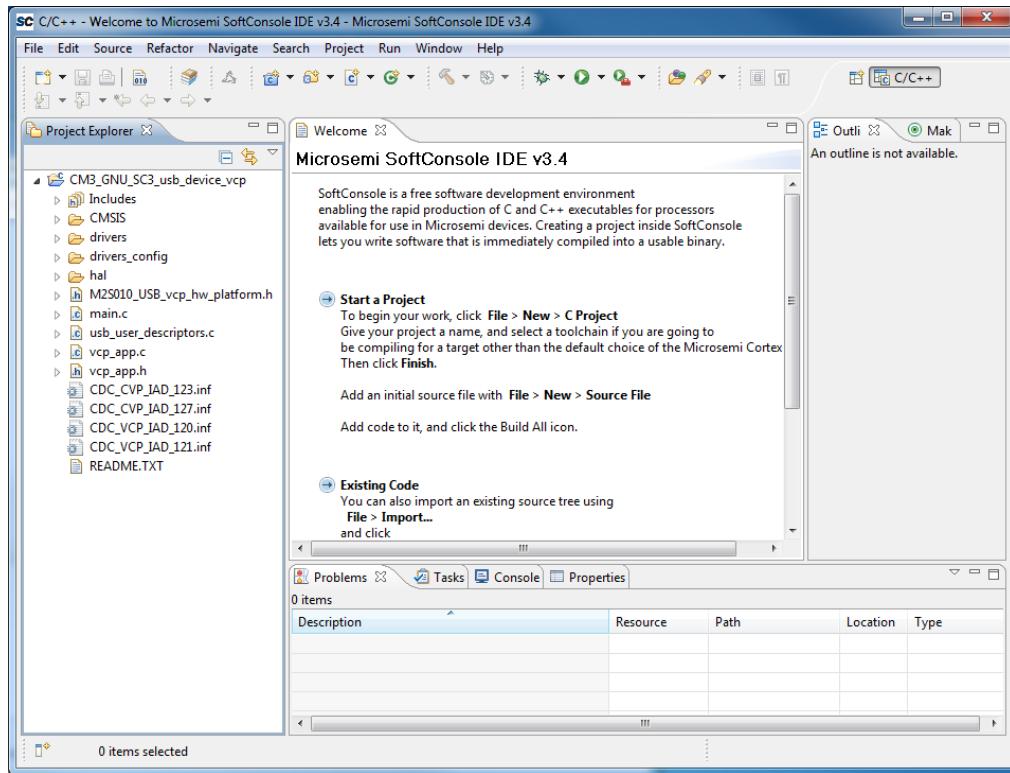


Рис. 39.

Отредактируем содержимое файла main.c в соответствии с задачами нашего приложения. Код главной процедуры приложения представлен в листинге ниже:

```
int main()
{
    size_t rx_size;
    uint8_t rx_buf[1];

    VCP_init(MSS_USB_VCP_9600_BAUD, MSS_USB_VCP_ONE_STOP_BIT,
              MSS_USB_VCP_NO_PARITY, MSS_USB_VCP_DATA_8_BITS);

    MSS_UART_init(&g_mss_uart0, MSS_UART_115200_BAUD,
                  MSS_UART_DATA_8_BITS | MSS_UART_NO_PARITY |
                  MSS_UART_ONE_STOP_BIT);

    while(VCP_HOST_APP_READY != VCP_configuration_state());
    VCP_rx_prepare(receive_buf, sizeof(receive_buf), receive_data_handler);
    VCP_tx(g_greeting, sizeof(g_greeting));

    MSS_UART_polled_tx_string(&g_mss_uart0, (const uint8_t*)" Hello, World!!
\ n\r \ n\r") ;

    for(;;)
    {
        if(g_rx_count)
        {
```

```

        VCP_tx(receive_buf, g_rx_count);
        MSS_UART_polled_tx(&g_mss_uart0, receive_buf, g_rx_count);
        g_rx_count=0;
    }
    if((1 == VCP_tx_done()))
    {
        VCP_rx_prepare(receive_buf, sizeof(receive_buf),
        receive_data_handler);
    }
    rx_size = MSS_UART_get_rx(&g_mss_uart0, rx_buff, sizeof(rx_buff));
    if( rx_size > 0 )
    {
        MSS_UART_polled_tx(&g_mss_uart0, rx_buff, sizeof(rx_buff));
        VCP_tx(rx_buff, rx_size);
    }
}

```

Полностью исходный код приложения доступен в папке \Source в архиве с файлами проекта.

Теперь из исходного кода на языке «C» необходимо получить исполняемый процессором код приложения и объединив его с битстримом FPGA Fabric ChK SmartFusion2 получить итоговую прошивку микросхемы.

Установим активной конфигурацию Release (рис. 40).

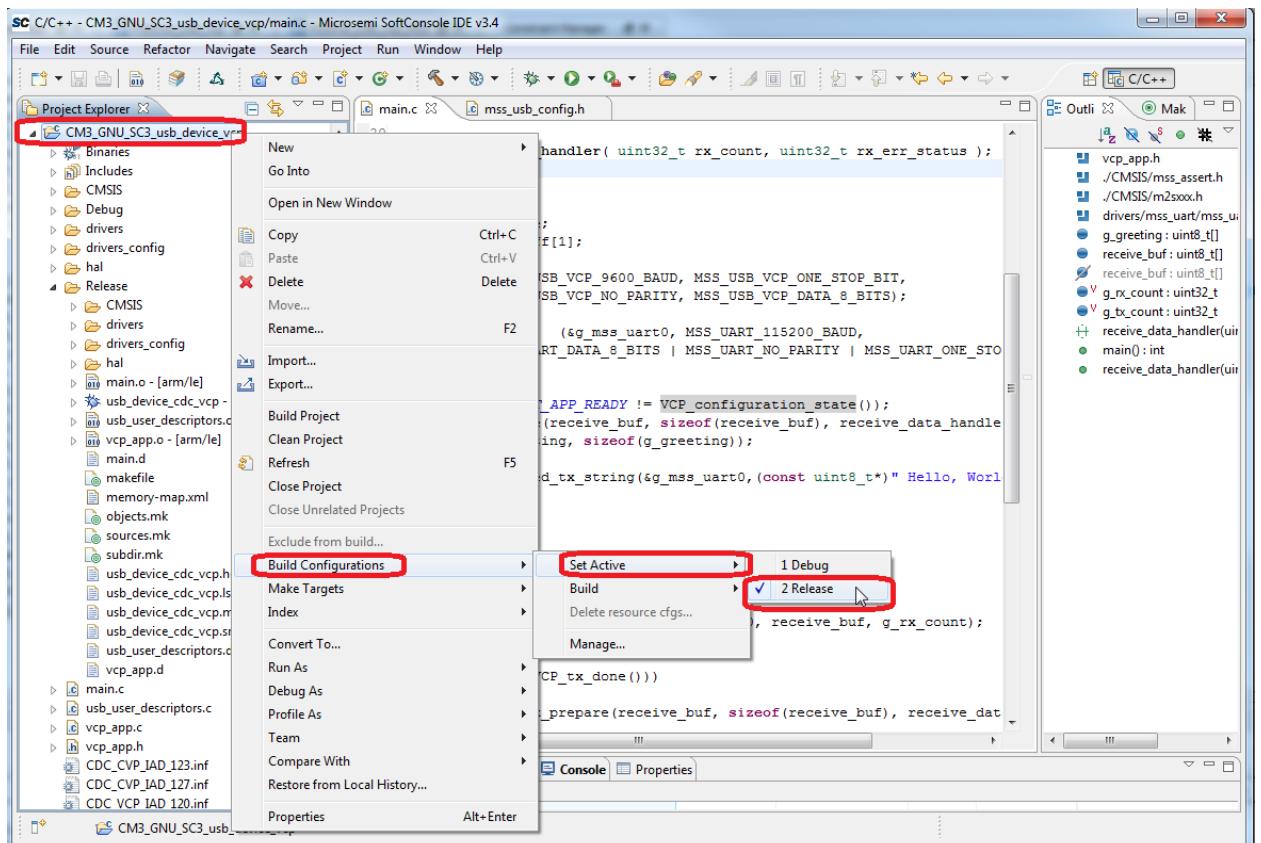


Рис. 40.

Выделив проект CM3_GNU_usb_device_vcp, отредактируем его свойства (рис. 41). Во вкладке C/C++ Build > Settings перейдем в окно Tool Settings и в настройках GNU C Linker > Miscellaneous (рис. 42) изменим значение параметра Linker flags на

-T\${workspace_loc:/M2S010_int_MSS_CM3_hw_platform/CMSIS/startup_gcc/production-smartfusion2-execute-in-place.ld}

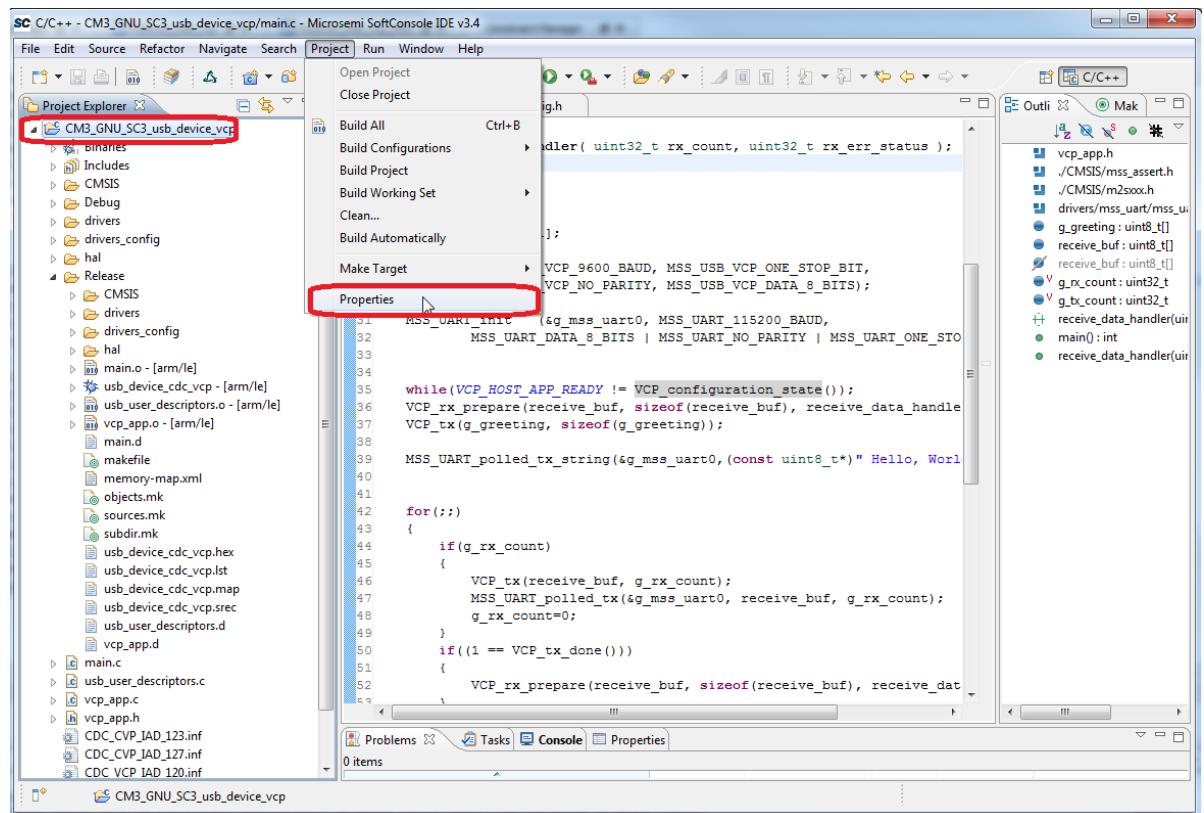


Рис. 41.

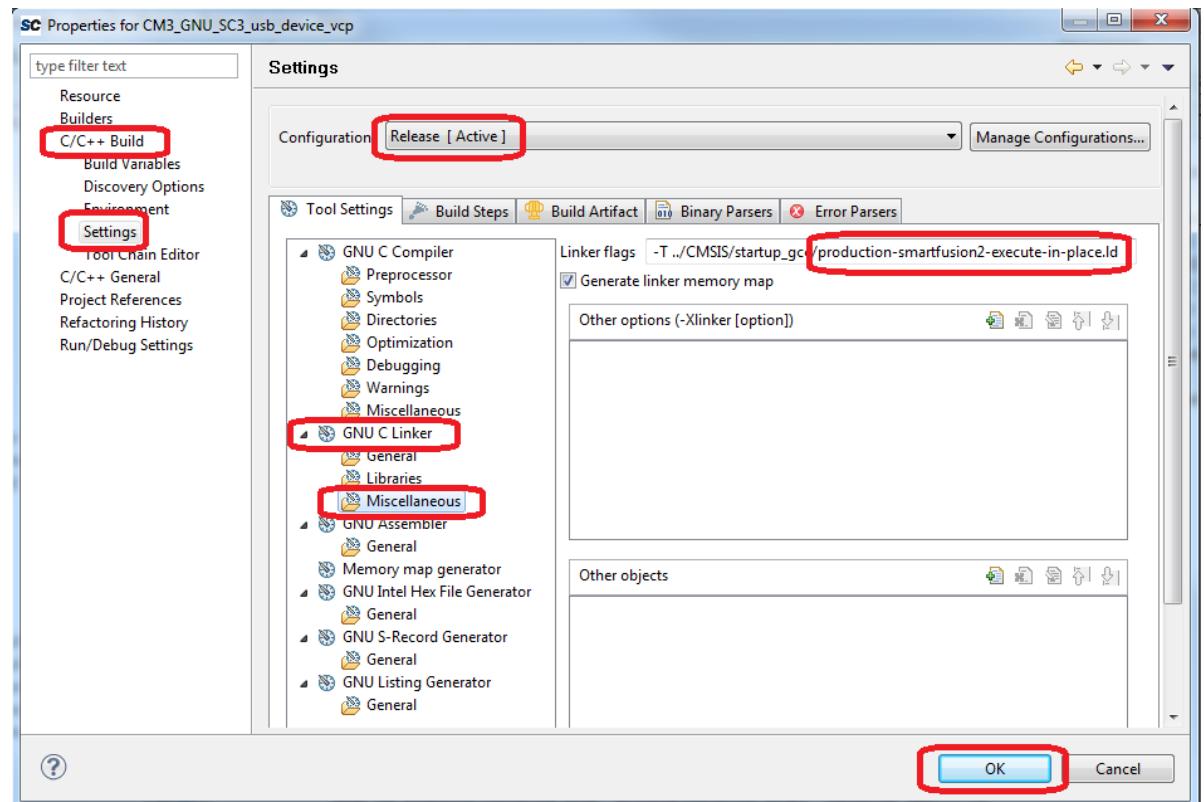


Рис. 42.

Интерфейс USB микроконтроллерной подсистемы СиК SmartFusion2 способен работать в различных режимах: как USB-host, USB-device и USB OTG. Драйвер USB интерфейса созданный мастером по умолчанию сконфигурирован для работы как USB-host. В нашем проекте хостом является USB интерфейс персонального компьютера, а USB-интерфейс СиК SmartFusion2 играет роль USB-device и должен быть сконфигурирован соответствующим образом. Настроим интерфейс USB чипа нашего отладочного набора для работы как USB device. Для этого в каталоге \drivers\mss_usb проекта ВПО откроем файл mss_usb_config.h и раскомментируем строку 29.

```
#define MSS_USB_PERIPHERAL_MODE
```

И закомментируем строку 26 устанавливающую активной конфигурацию MSS_USB_OTG_HOST_MODE (рис. 43).

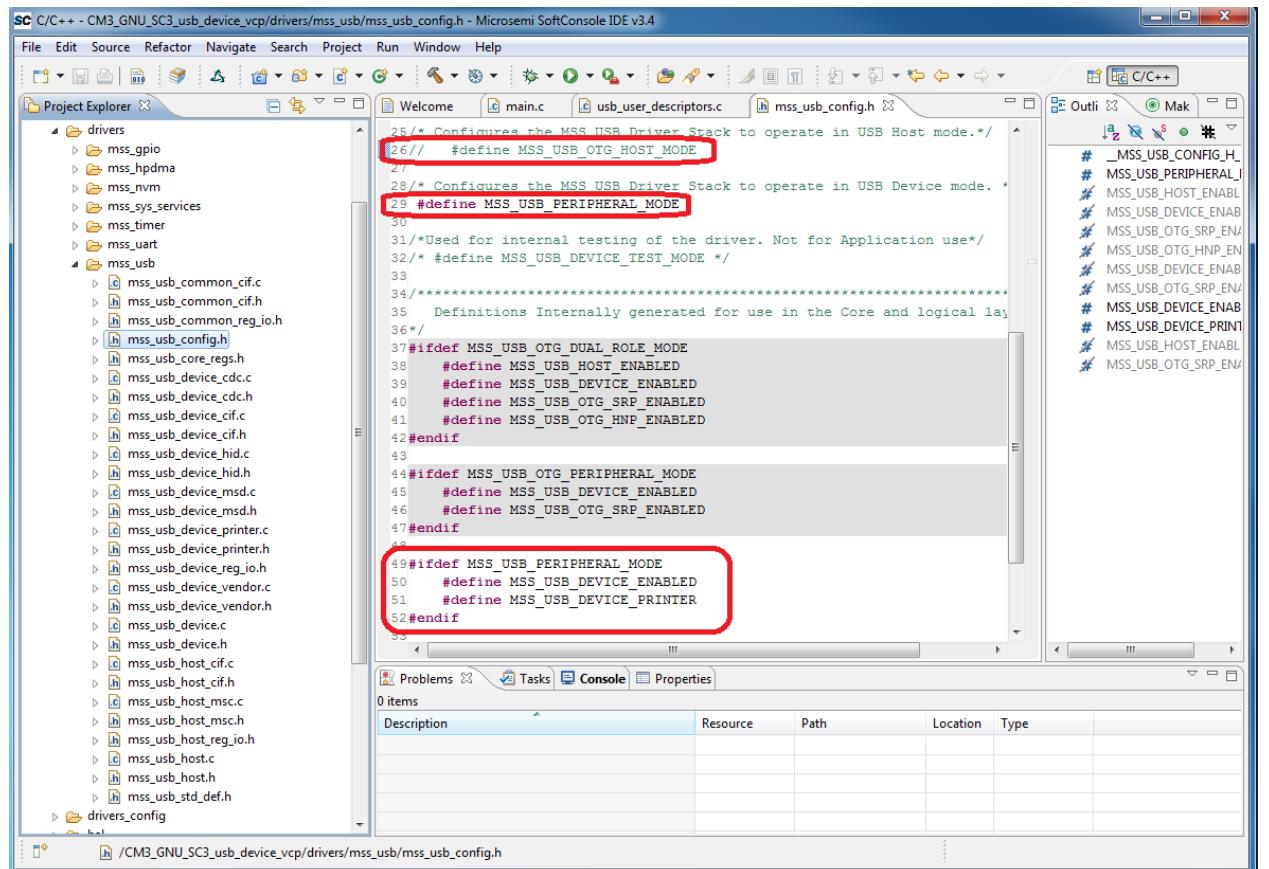


Рис. 43.

Выполним команду основного меню **Project > Clean** и получим файл с исполнимым кодом нашего приложения в папке Release проекта ВПО (рис. 44).

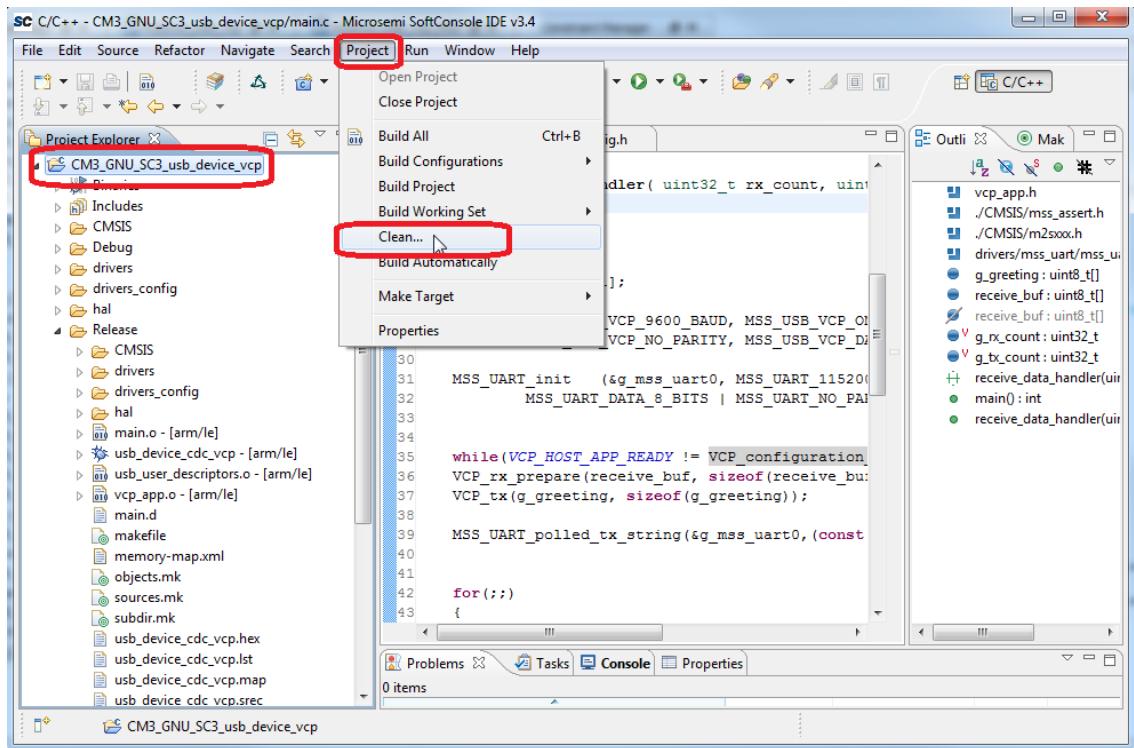


Рис. 44.

В результате выполнения команды в папке проекта ВПО появится каталог Release, а в нем файл образа встраиваемого программного обеспечения процессора ARM Cortex-M3, не содержащий отладочной информации, который может быть использован для создания единого файла прошивки СнК, включающего битстримы FPGA Fabric и ВПО (рис. 45).

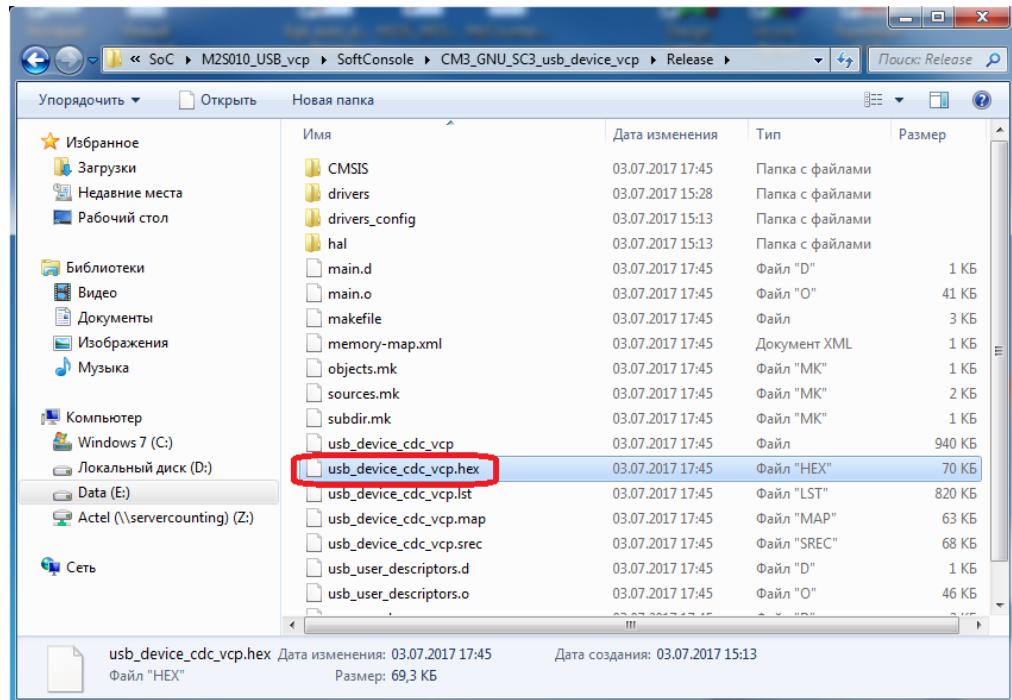


Рис. 45.

Создание файла конфигурационной последовательности

Теперь создадим файл конфигурационной последовательности, включающий битстрим матрицы FPGA Fabric и исполняемый образ встроенного программного обеспечения процессора ARM Cortex-M3. Для этого в проекте СнК в среде Libero SoC 11.8 в окне настроек микроконтроллерной подсистемы отредактируем свойства блока eNVM, для чего дважды щелкнем на соответствующем элементе графического интерфейса (рис. 46).

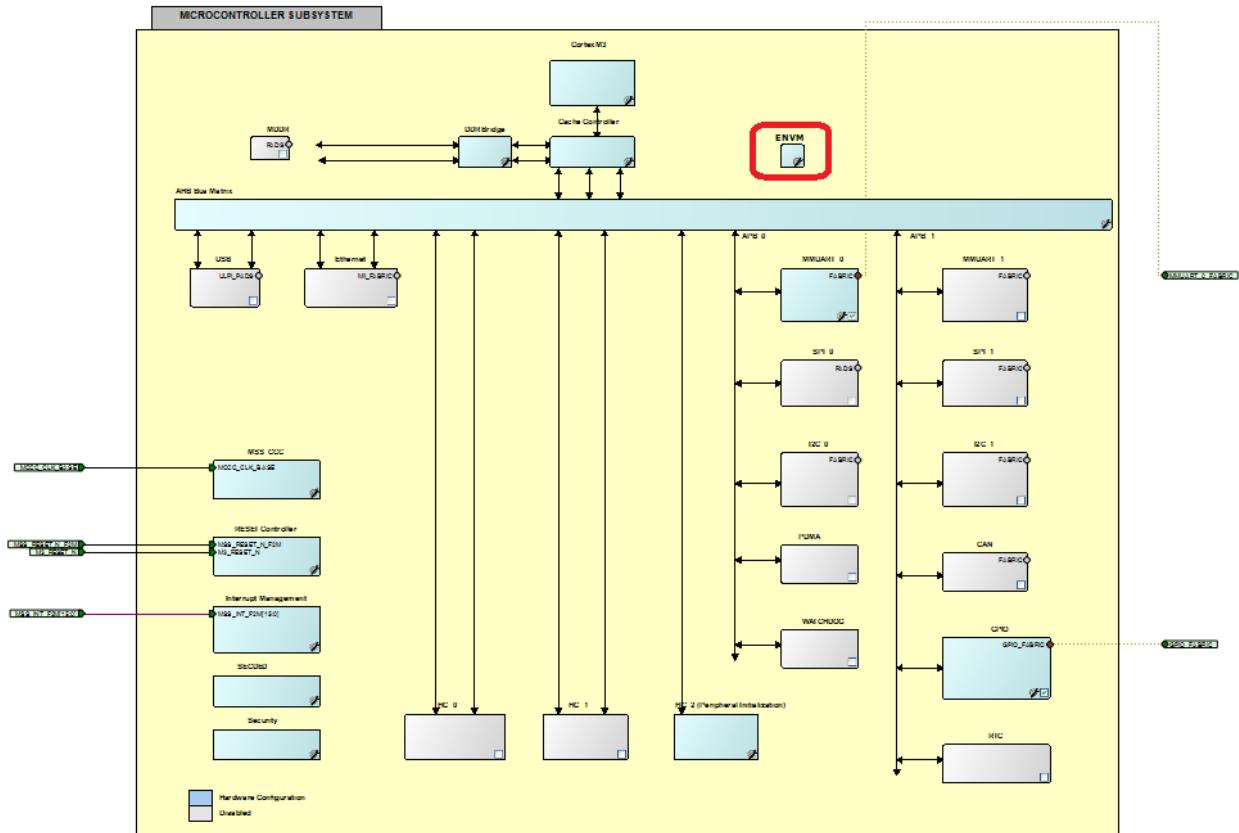


Рис. 46.

В появившемся окне **eNVM Configurator** дважды щелкнем на пункте **Data Storage** и в следующем появившемся окне **Add Data Storage Client** укажем условное название нашего приложения, а также пути к hex-файлу образа ВПО `usb_device_cdc_vcp.hex` (рис. 47).

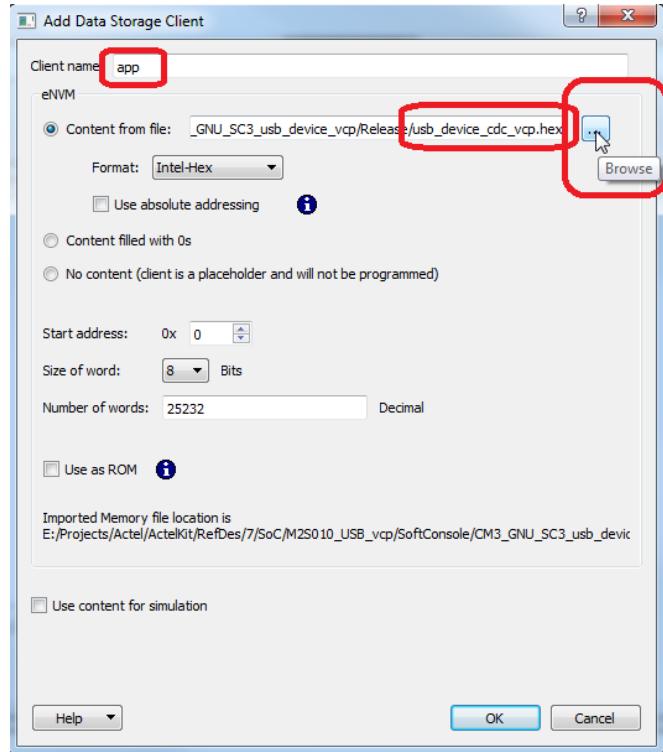


Рис. 47.

После чего окно **eNVM Configurator** примет вид, представленный на рис. 48. В окне отобразится строка с основными характеристиками загруженного файла.

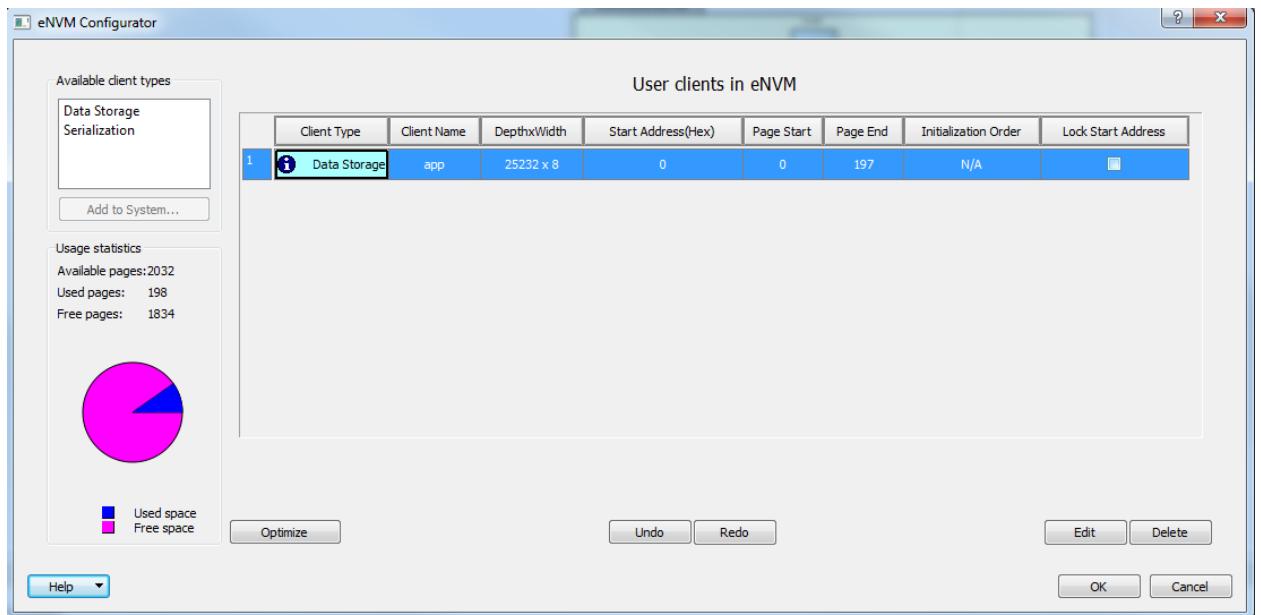


Рис. 48.

Сохраним изменения, выполним процедуру генерации компонента (рис. 49).

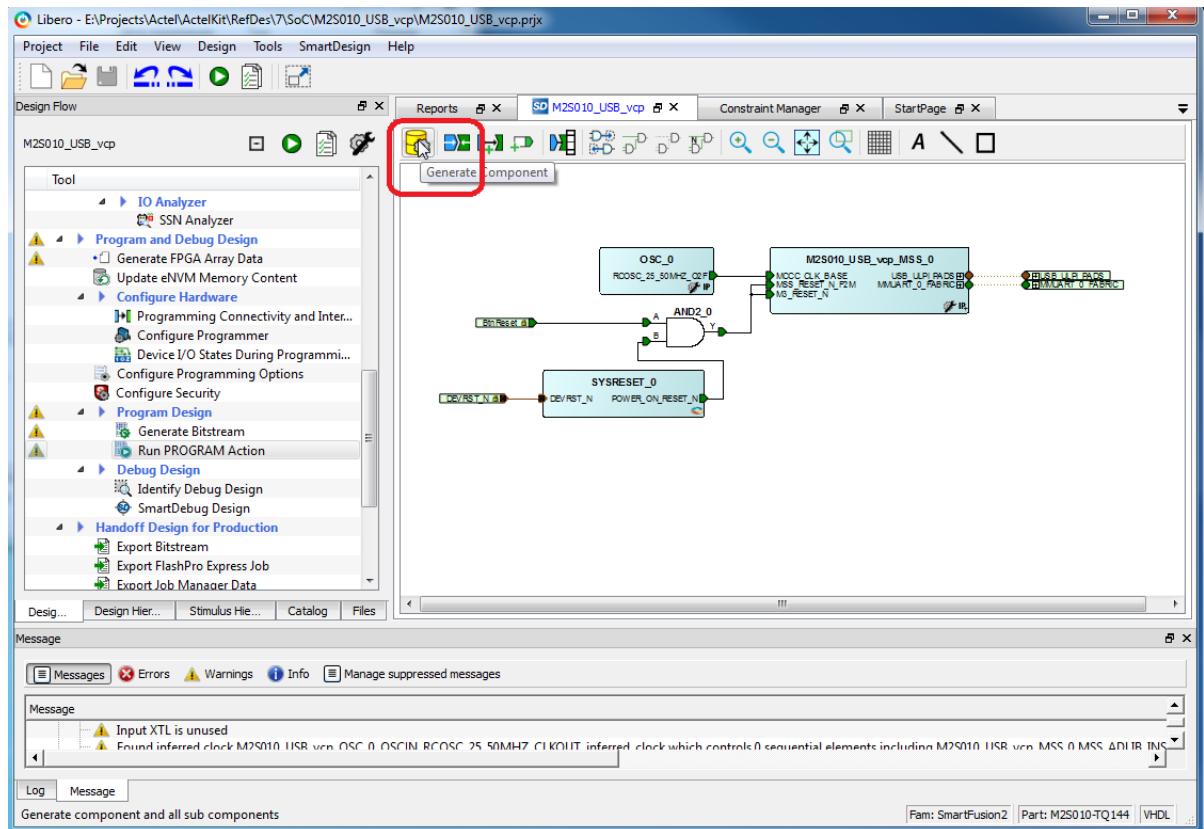


Рис. 49.

Подключим программатор к отладочному набору и персональному компьютеру, запустим процесс генерации файла прошивки и программирования (рис. 50).

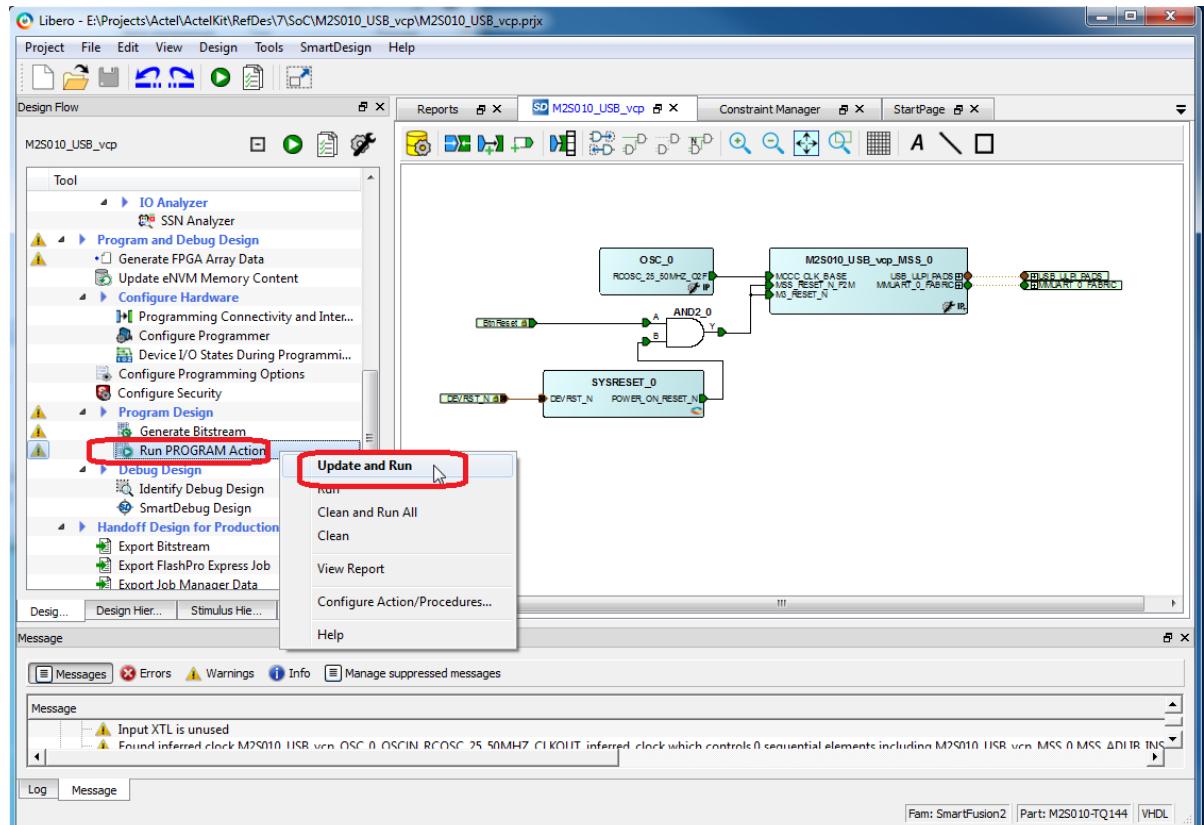


Рис. 50.

На плате отладочного набора SF2-Junior-KIT располагаются два разъема ввода-вывода общего назначения (GPIO) имеющие позиционные обозначения X9, X10 для подключения сигналов внешних устройств к контактам M2S010-TQ144. В данном проекте контакты микросхемы M2S010-TQ144 назначены таким образом, чтобы вывести сигналы интерфейса UART (UART_Rx, UART_Tx) на разъем X10.

В состав набора SF2-Junior-KIT входят приемопередатчики USB – Bluetooth dongle и модуль UART- Bluetooth HC-06. Для связи СнК SmartFusion2 с персональным компьютером (ПК) через интерфейс MMUART_0 необходимо в USB разъем ПК включить USB – Bluetooth dongle, а к разъему X10 отладочного набора в соответствии с таблицей 2 подключить модуль HC-06.

Таблица 2. Подключение модуля HC-06 к разъему X10 отладочного набора SF2-Junior-KIT

№ контакта разъема X10 SF2-Junior-KIT	Номер контакта M2S010-TQ144	Название цепи проекта СнК / принципиальной схемы	Название цепи модуля HC-06
5	9	MMUART_0_RXD_F2M	RXD
6	10	MMUART_0_TXD_M2F	TXD
25	-	+3.3V	VCC
29	-	GND	GND

Для связи СнК SmartFusion2 с ПК через интерфейс USB микроконтроллерной подсистемы СнК SmartFusion2, сконфигурированный для работы в режиме VCP, подключаем отладочный набор к персональному компьютеру с помощью кабеля USB-B – mini_USB.

Прошьем СнК конфигурационной последовательностью нашего проекта.

Для отображения результатов работы нашего приложения на экране ПК воспользуемся программой – терминалом TeraTerm. Необходимо запустить два экземпляра программы: один – для связи с портом USB, работающим в режиме VCP, другой для связи с MMUART_0. Узнать какие номера последовательных портов присвоены виртуальным устройствам нашего проекта можно с помощью стандартной утилиты «Диспетчер устройств» (рис. 51).

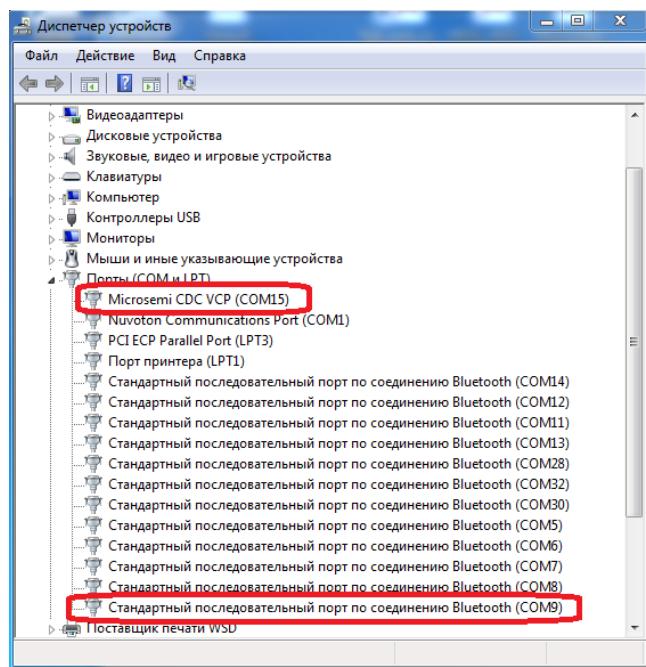


Рис. 51.

Запускаем два экземпляра программы терминала, открываем в них необходимые порты, в нашем случае это порты №9 и №15, наблюдаем корректную работу созданного приложения (рис. 52).

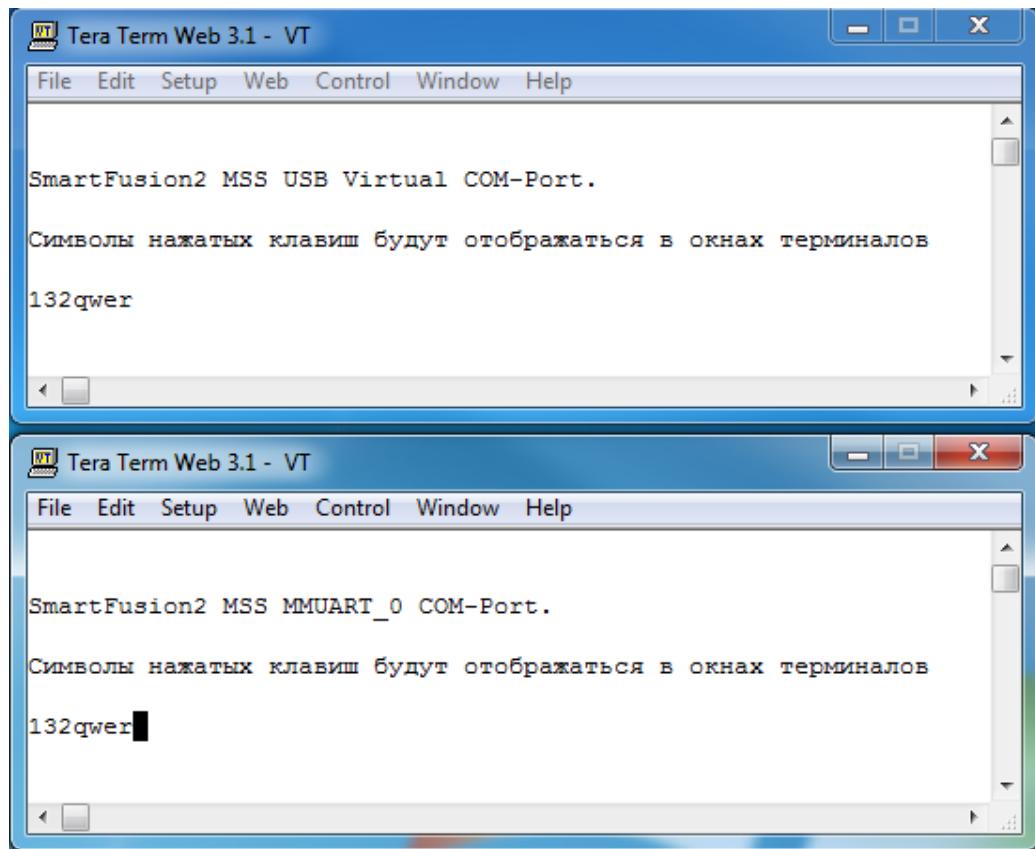


Рис. 52.

Проекты Снк и ВПО, разработка которых описана в данном руководстве, файлы с исходным кодом встроенного программного обеспечения можно скачать по [ссылке](#).

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