

# **SF2-Junior-KIT**

**Проект измерительной системы на основе СиК SmartFusion2 и  
10-параметрического датчика MPU-9255**

## **Введение**

Общие принципы построения проекта системы на кристалле (СнК) с использованием контроллера I2C рассмотрены в [описании](#) к проекту «Использование интерфейса I2C в проектах пользователя».

В данном руководстве описан пример создания СнК реализующей функцию преобразователя интерфейсов, то есть чтение данных с 10-параметрического датчика по шине I2C и выдачу прочитанных данных по интерфейсу UART в персональный компьютер для дальнейшей обработки и визуализации результатов измерений.

## **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 3D-viewer.exe поставляемая в архиве с файлами проекта СнК;

## **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.
- 8) Модуль GY-91 с 10-параметрическим сенсором MPU-9255.

## Описание проекта

В проекте СнК, разработка которого описана в данном руководстве, задействован один контроллер I2C, работающий в режиме мастера шины, осуществляющий чтение данных с 10-параметрического сенсора MPU-9255 установленного на модуле GY-91 которые подключается к отладочному набору через разъем X10 с помощью соединительных проводов, входящих в комплект набора. СнК SmartFusion2 по интерфейсу I2C осуществляет чтение данных измерений из регистров датчика MPU-9255, формирует из прочитанных данных массив и передает сформированный массив в ПК по интерфейсу UART для дальнейшей обработки и визуализации результатов измерений.

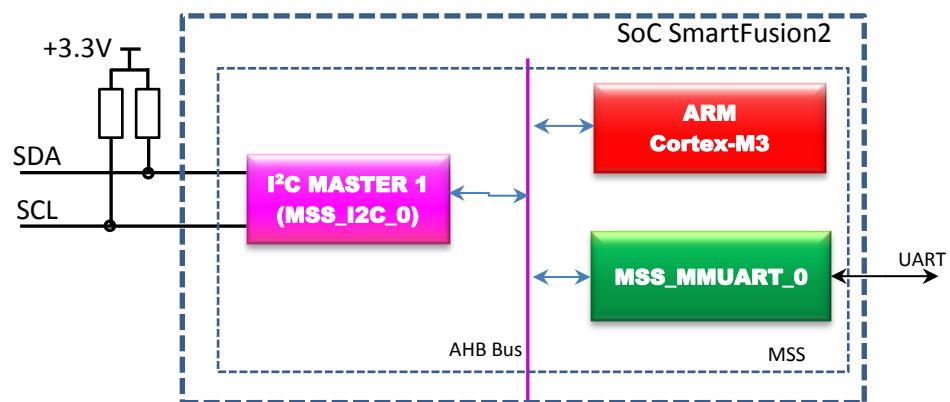


Рис. 1.

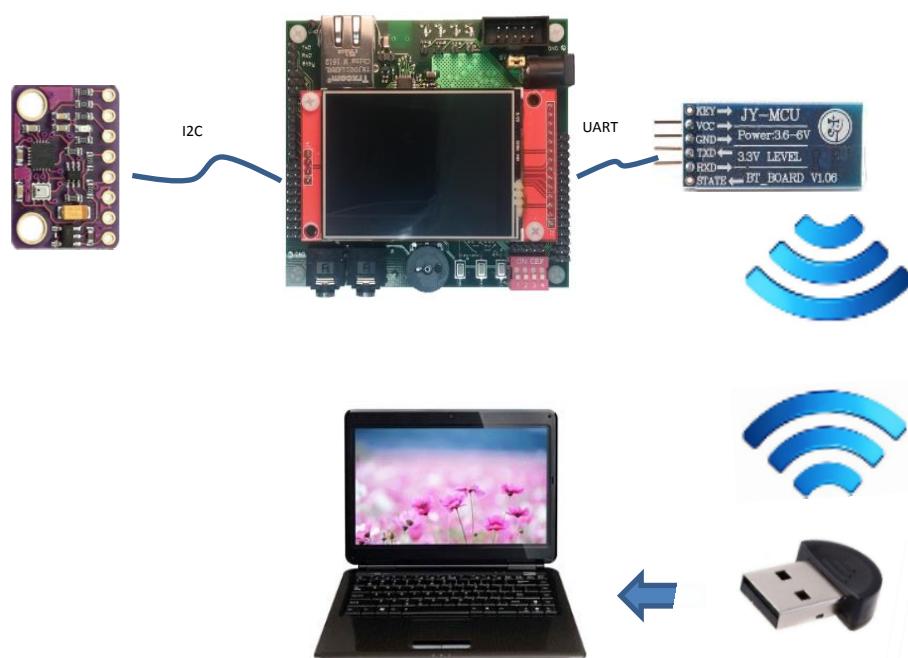


Рис. 2.

Микросхема MPU-9255 представляет собой 10-параметрический датчик, в миниатюрном корпусе которого находятся 3-х осевой акселерометр, 3-осевой гироскоп, 3-осевой компас, датчик температуры кристалла, процессор для предварительной обработки измерений (Digital Motion Processor, DMP), два последовательных интерфейса, один из которых предназначен для коммуникации с хостом, второй – auxMaster - для работы с дополнительными датчиками системы.

Структурная схема MPU-9255 представлена на рис.3.

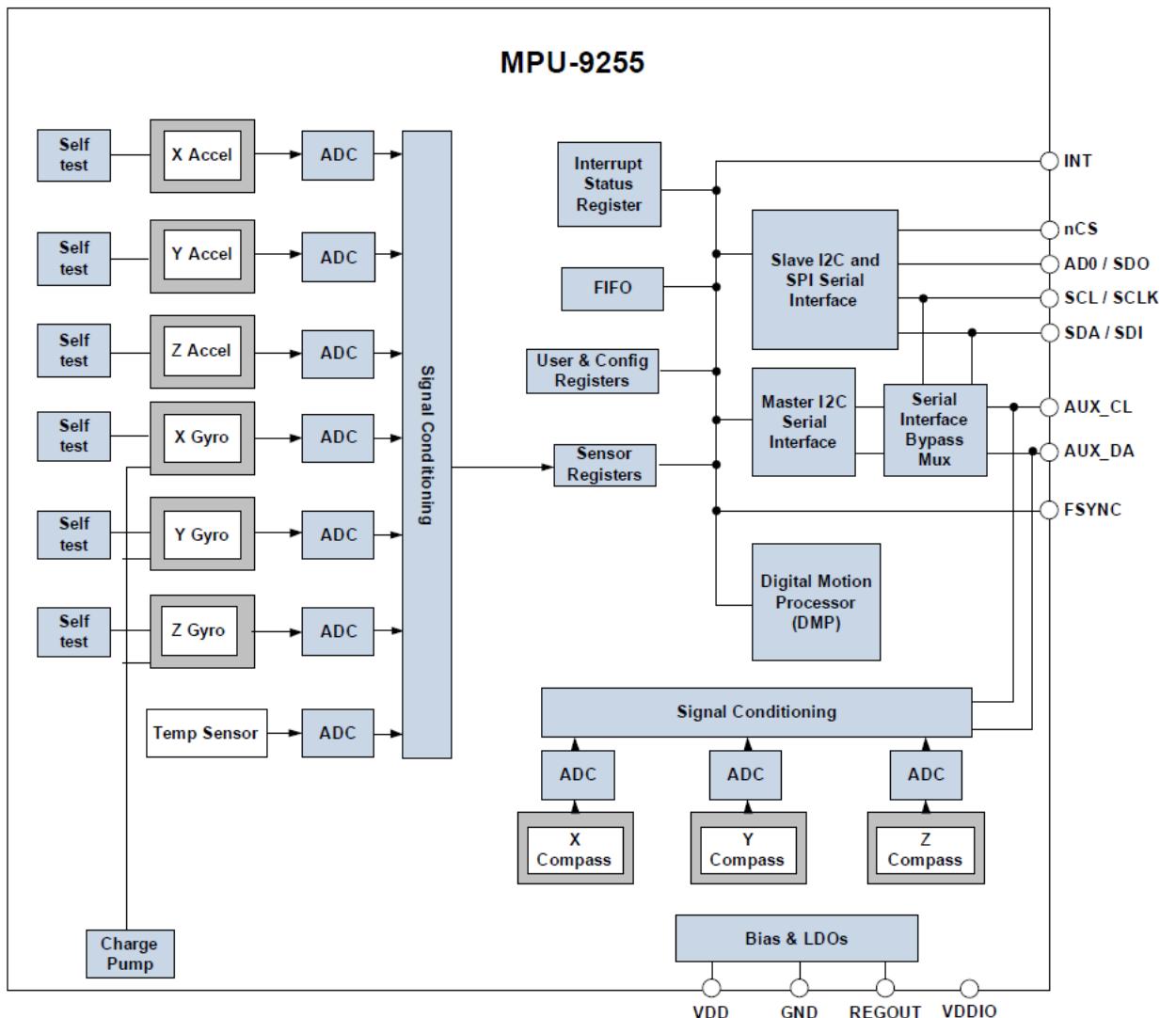


Рис. 3.

С точки зрения внутреннего устройства микросхема MPU-9255 представляет собой мультикристальную сборку, в которой в качестве инерциального сенсора (акселерометр и гироскоп) взят кристалл аналогичный MPU-6555 InvenSense Inc, а в качестве компаса-магнетометра используется кристалл AK8963 производства Asahi Kasei Microdevices Corporation. Кристаллы инерциального сенсора и магнетометра имеют отличия во временной диаграмме работы, которые необходимо учесть при разработке кода встраиваемого программного обеспечения.

Кроме MPU-9255 на печатную плату модуля GY-91 установлен датчик давления BMP-280.

Принципиальная схема модуля GY-91 представлена на рис. 4.

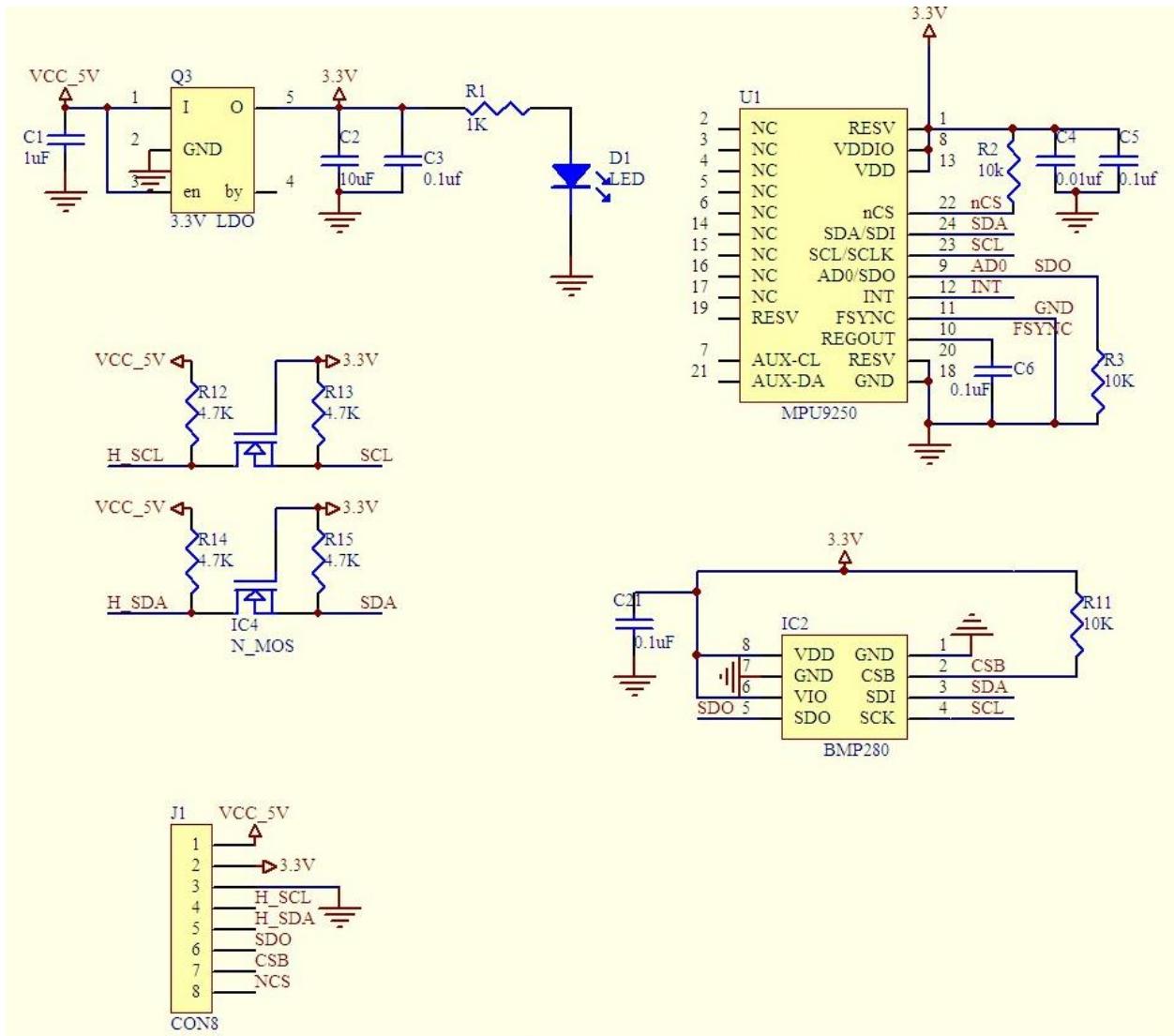


Рис. 4.

# 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.

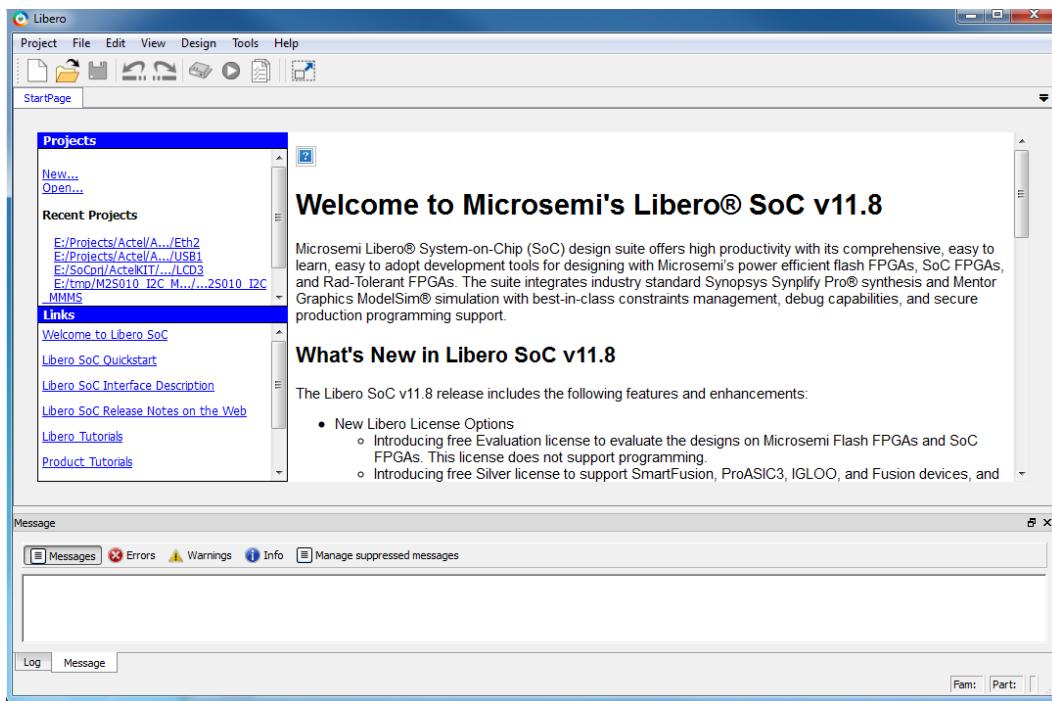


Рис. 5.

В главном меню Libero SoC (рис. 5) выполните команду **Project/New Project**, запустится мастер создания нового проекта. В появившемся окне укажите название проекта, например M2S010\_I2C\_UART, место расположения нашего проекта на диске и предпочтаемый язык проектирования – Verilog или VHDL. Опцию Enable block creation устанавливать не нужно. Нажмите кнопку «Next» (рис. 6).

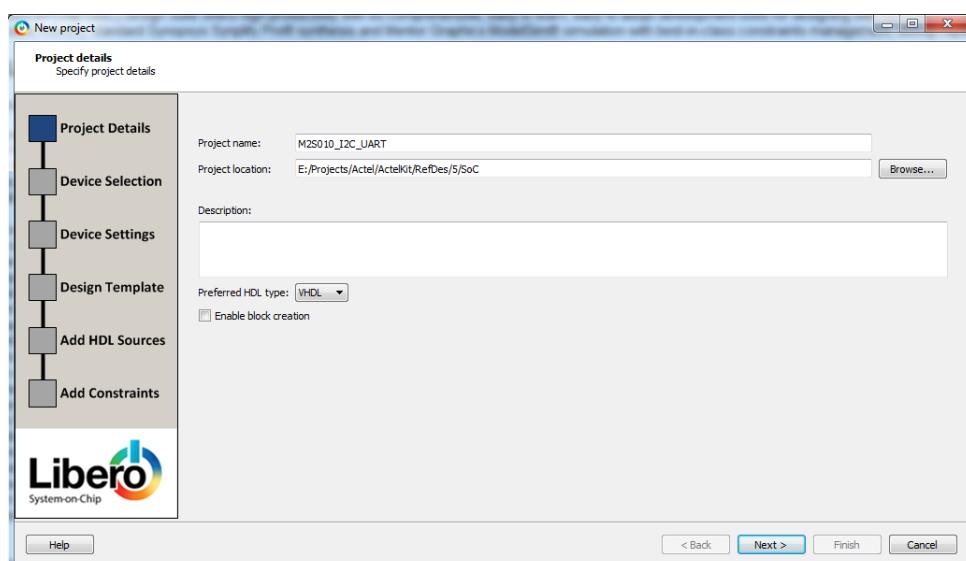


Рис. 6.

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».

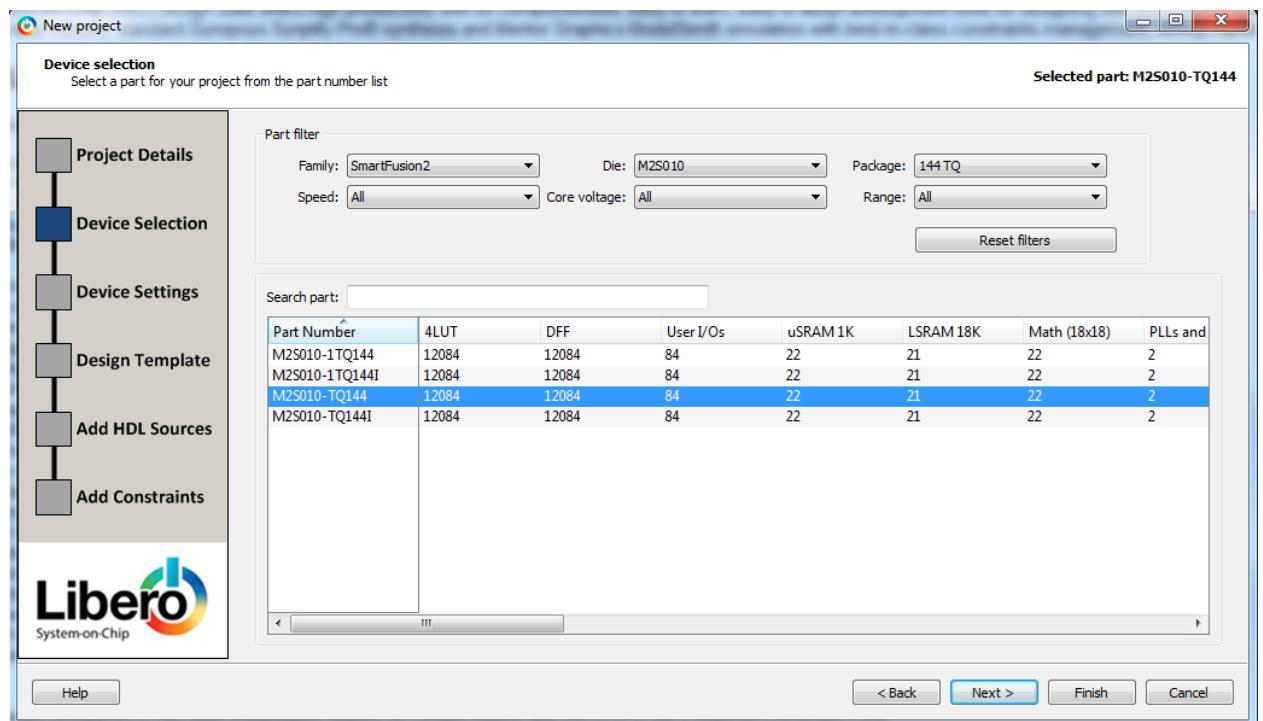


Рис. 7.

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. 8).

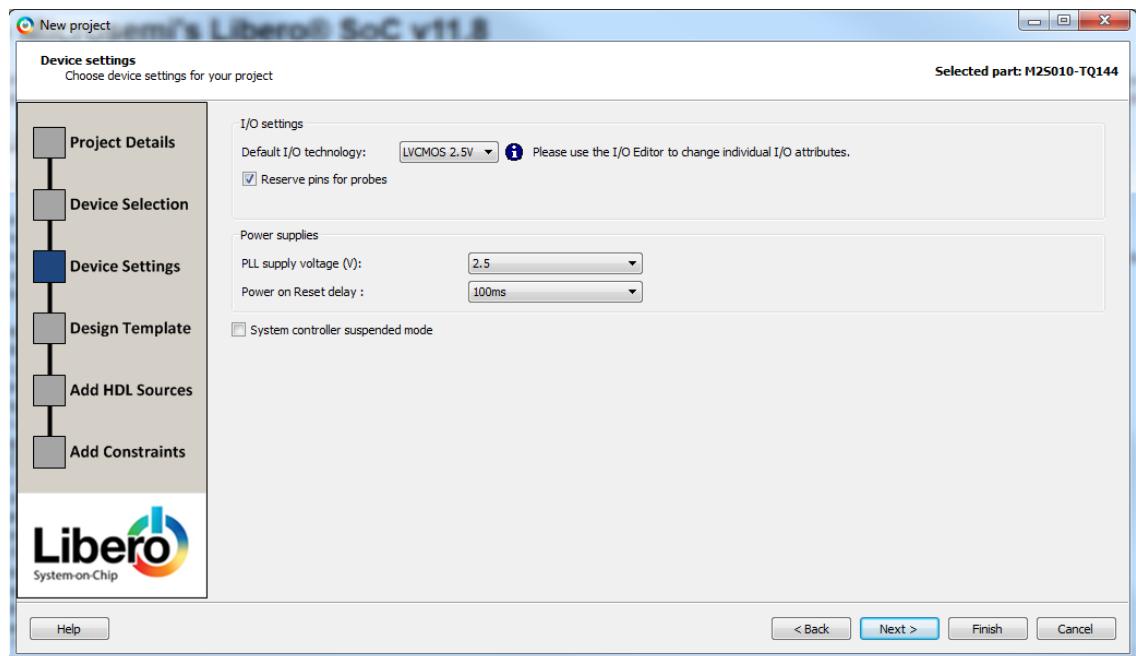


Рис. 8.

В следующем появившемся окне предлагается выбрать мастер, который будет использоваться для настройки микроконтроллерной подсистемы. Выберем пункт «Create a microcontroller (MSS) based design», после чего нажмем «Next» (рис. 9).

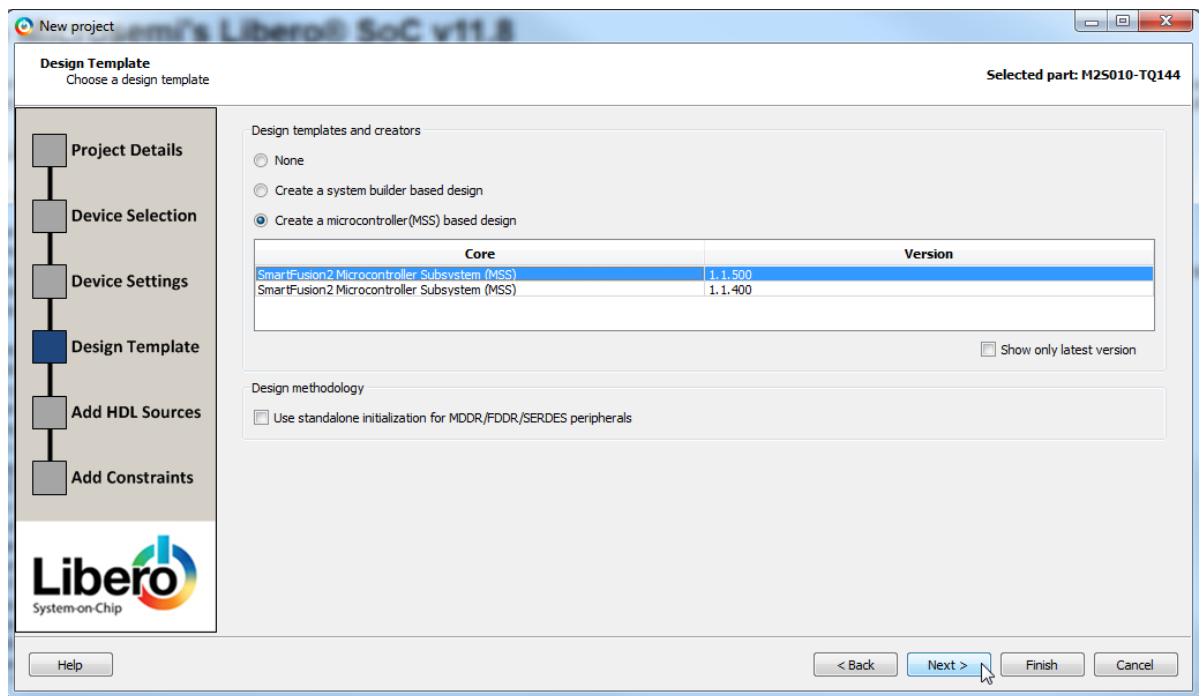


Рис. 9.

В следующих появляющихся окнах ничего не меняем, просто нажимаем «Next» до появления окна изображенного на рис. 10. При выборе способа установки проектных ограничений нажмите кнопку «Use Enhanced Constraint Flow».

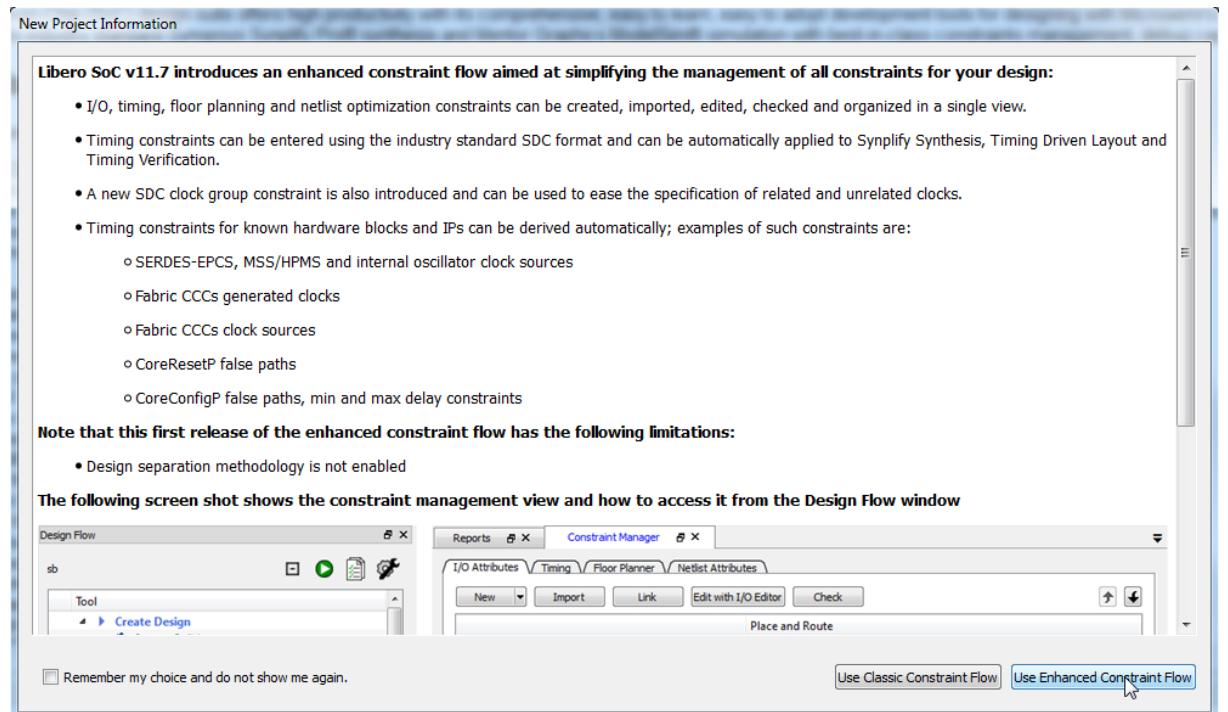


Рис. 10.

В результате выполнения описанных действий появится окно утилиты SmartDesign, в котором будет находиться один компонент M2S010\_I2C\_UART\_MSS\_0 – микроконтроллерная подсистема СнК SmartFusion2 (рис. 11).

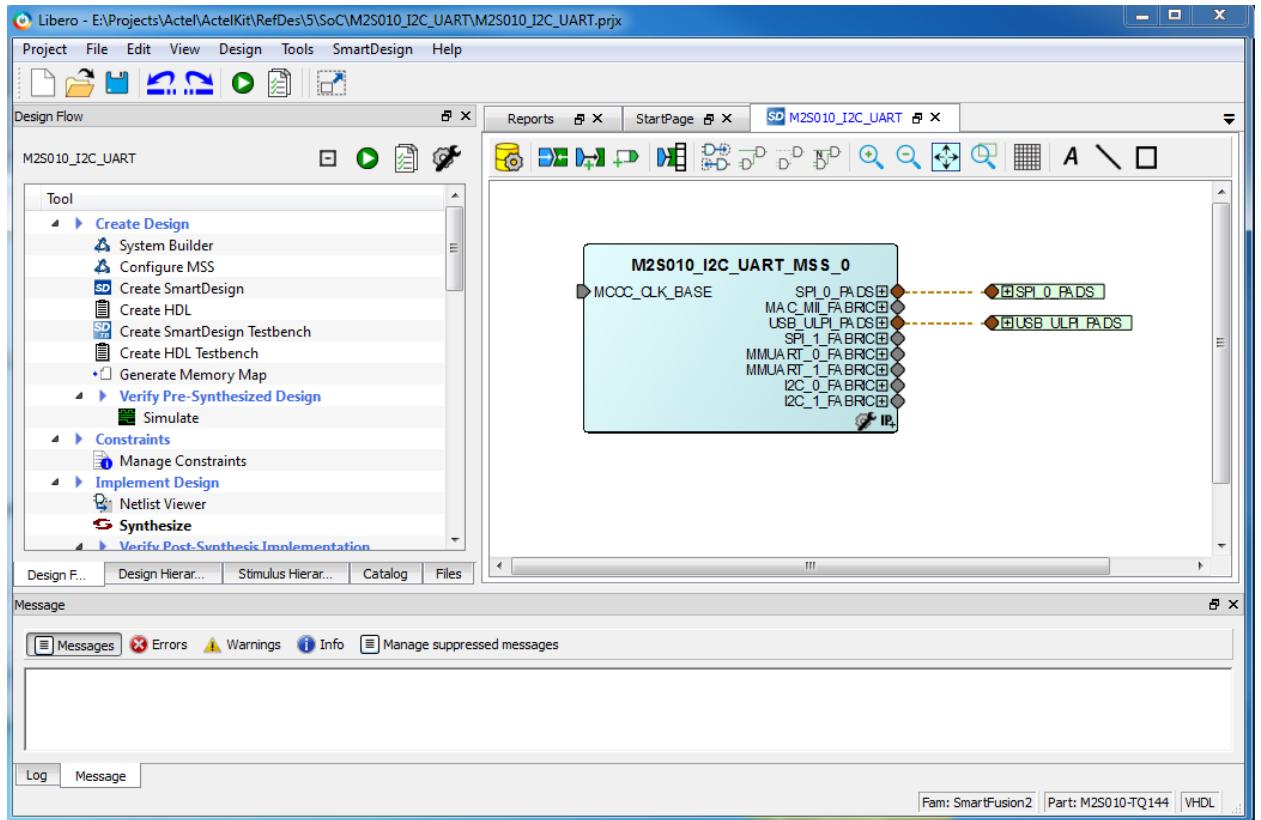


Рис. 11.

Созданный мастером компонент M2S010\_I2C\_UART\_MSS\_0 отражает состояние настроек микроконтроллерной подсистемы «по умолчанию». Необходимо изменить эти настройки в соответствии с задачами, решаемыми нашим приложением.

Настроим блоки архитектуры MSS, для этого дважды щелкнем на компоненте M2S010\_I2C\_UART\_MSS\_0. Откроется окно настроек микроконтроллерной подсистемы (рис. 12).

Из всего ряда доступных компонентов оставим включенными блоки I2C\_0, UART\_0. Все остальные компоненты MSS, а именно USB, Ethernet, MMUART\_1, SPI\_0, SPI\_1, 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. 13).

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. 14.

Настроим контроллер универсального приемопередатчика MMUART\_0. В появившемся окне выберем следующие опции (рис. 15):

Duplex Mode: Full Duplex.

Async/Sync Mode: Asynchronous.

Use Modem Interface: снять галочку

RHD: Fabric

TXD: Fabric

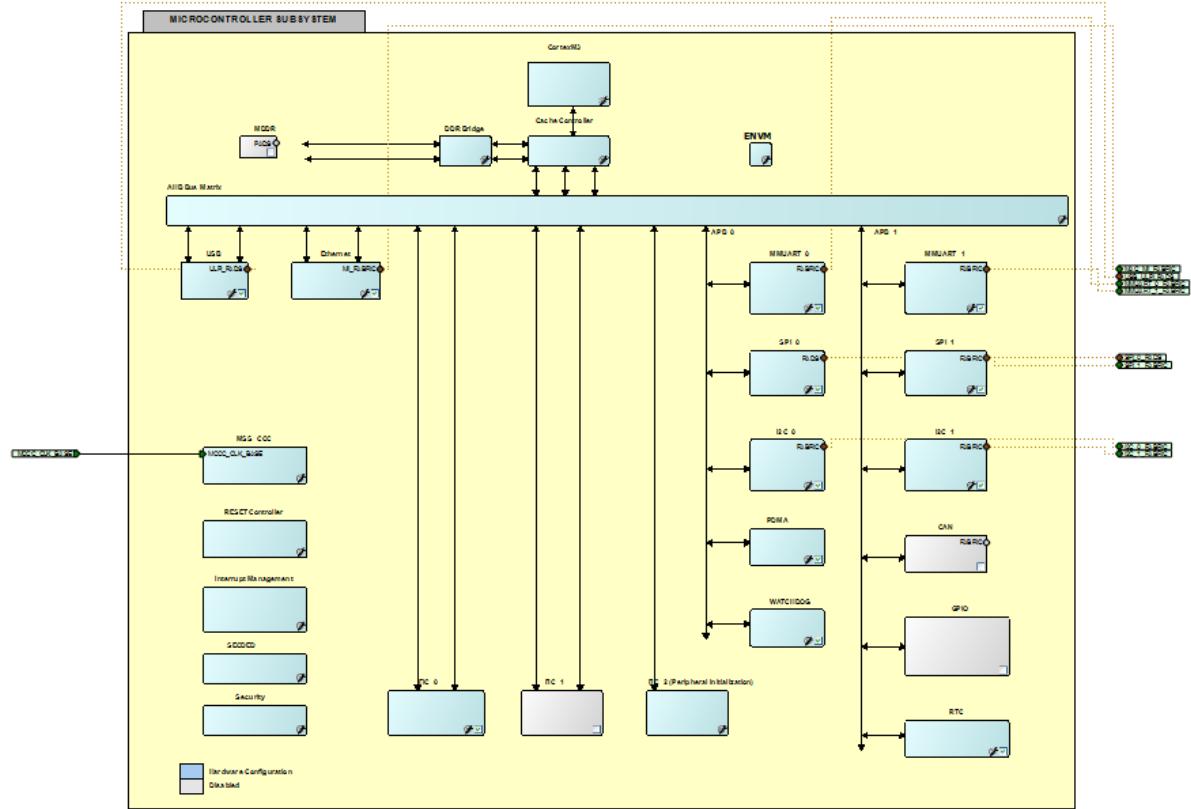


Рис. 12.

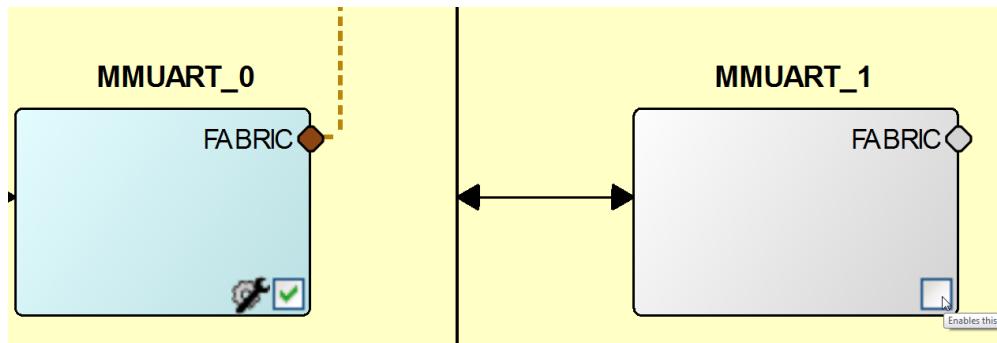


Рис. 13.

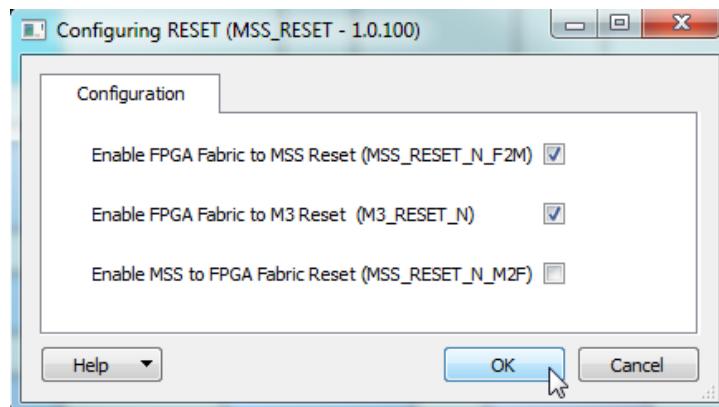


Рис. 14.

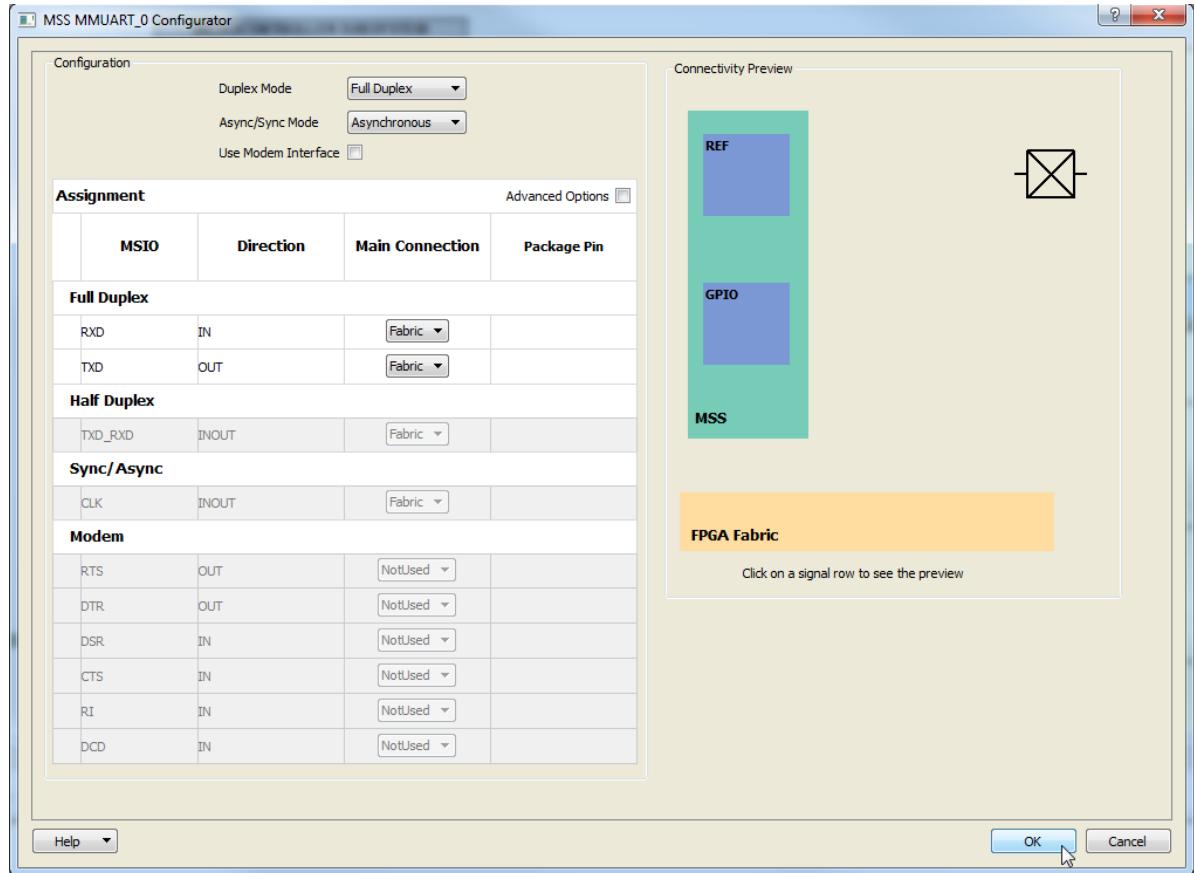


Рис. 15.

Настройки аппаратного контролера шины I2C остаются без изменений (рис. 16).

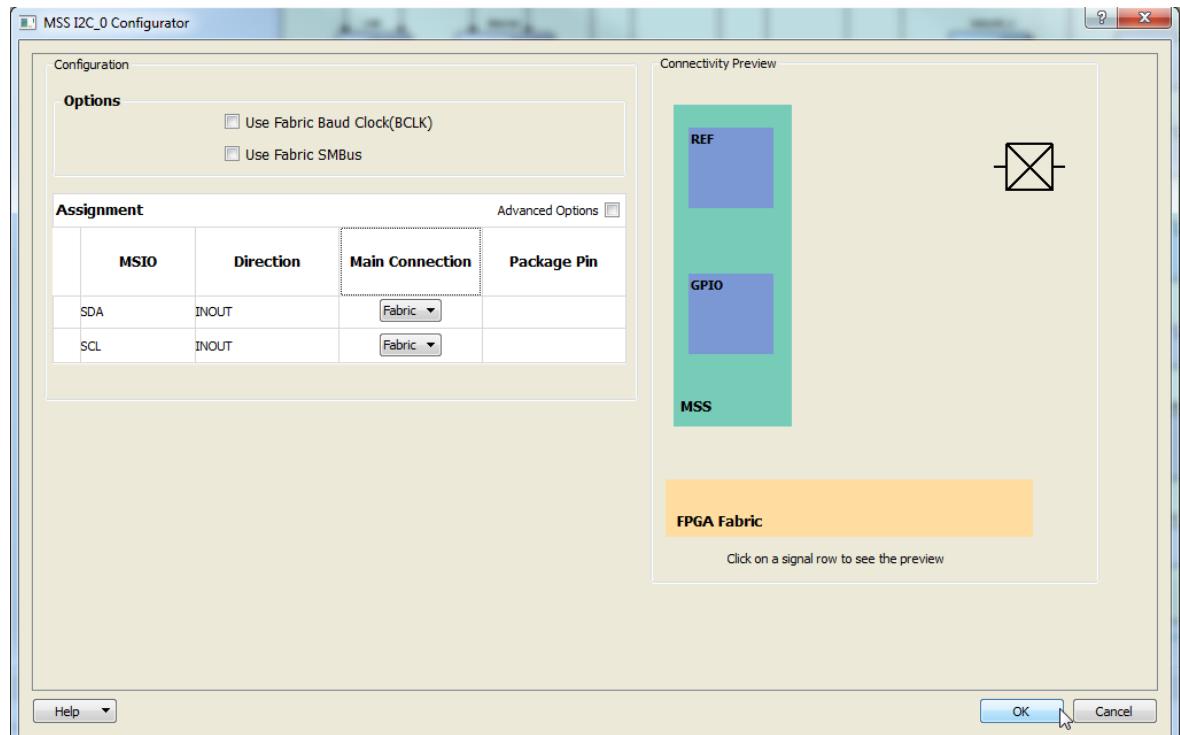


Рис. 16.

Настройки системы формирования сигналов тактирования микроконтроллерной подсистемы СнК SmartFusion2 представлены на рис. 17.

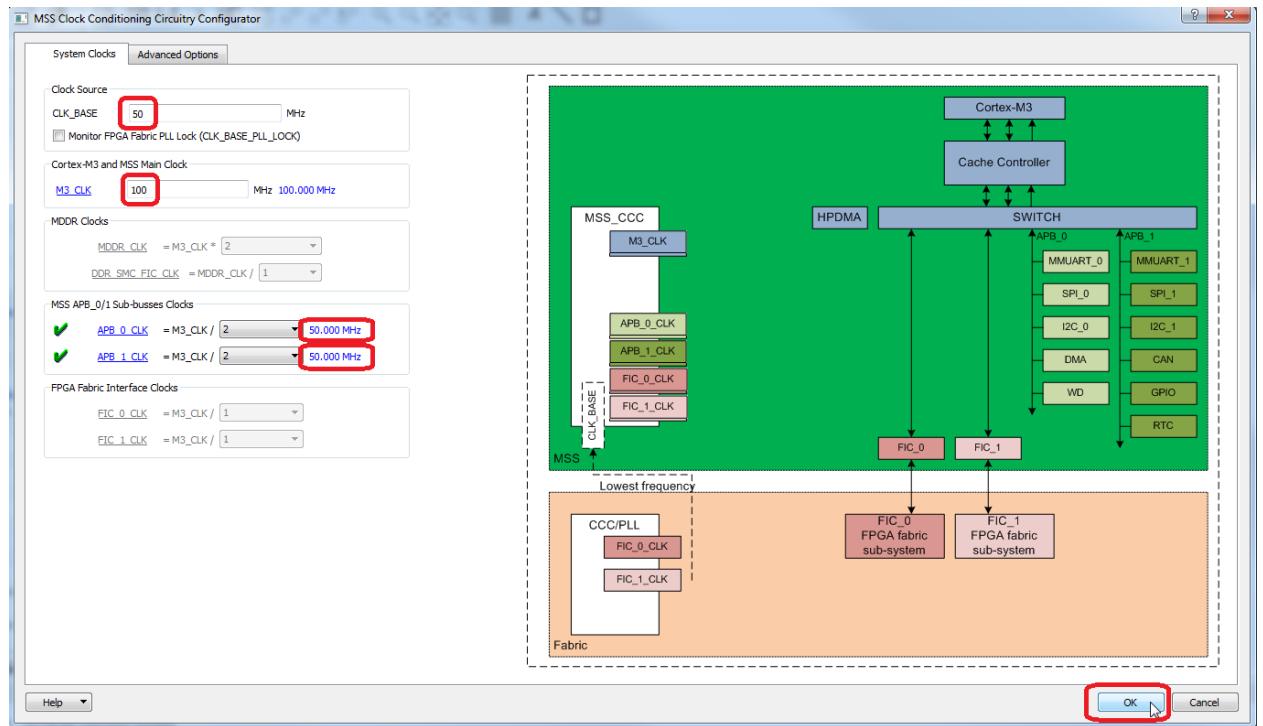


Рис. 17.

В результате описанных действий окно настроек микроконтроллерной подсистемы примет вид, аналогичный представленному на рис. 18.

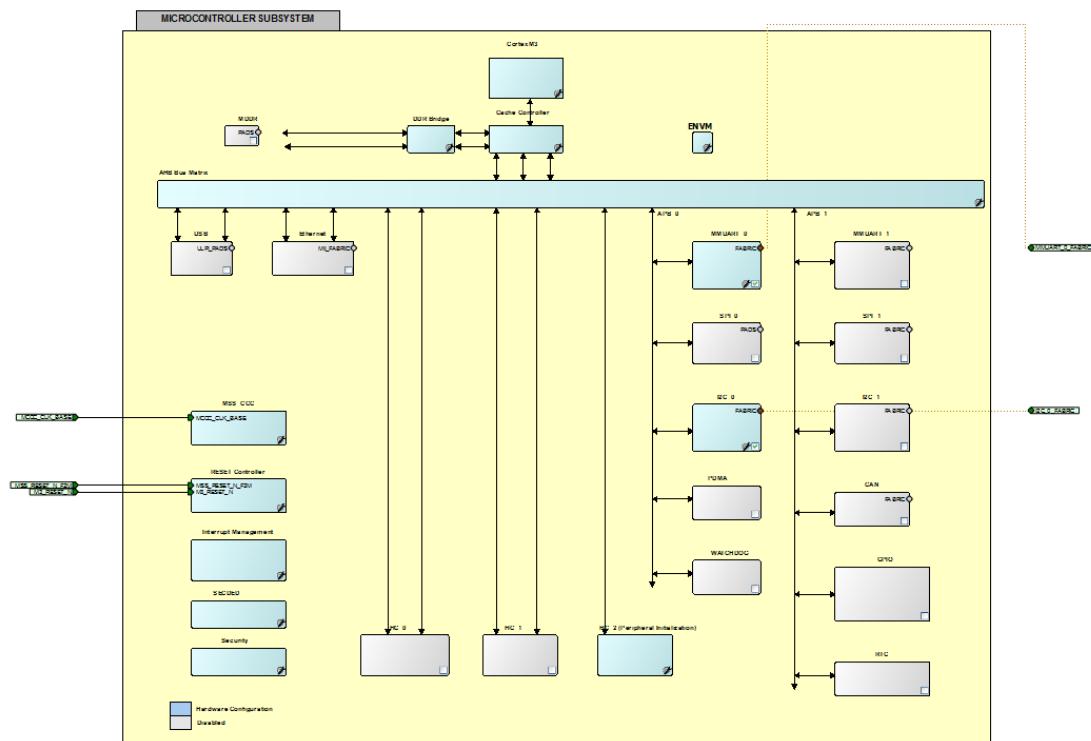


Рис. 18.

Сохраним изменения и вернемся во вкладку M2S010\_MMMS 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. 19).

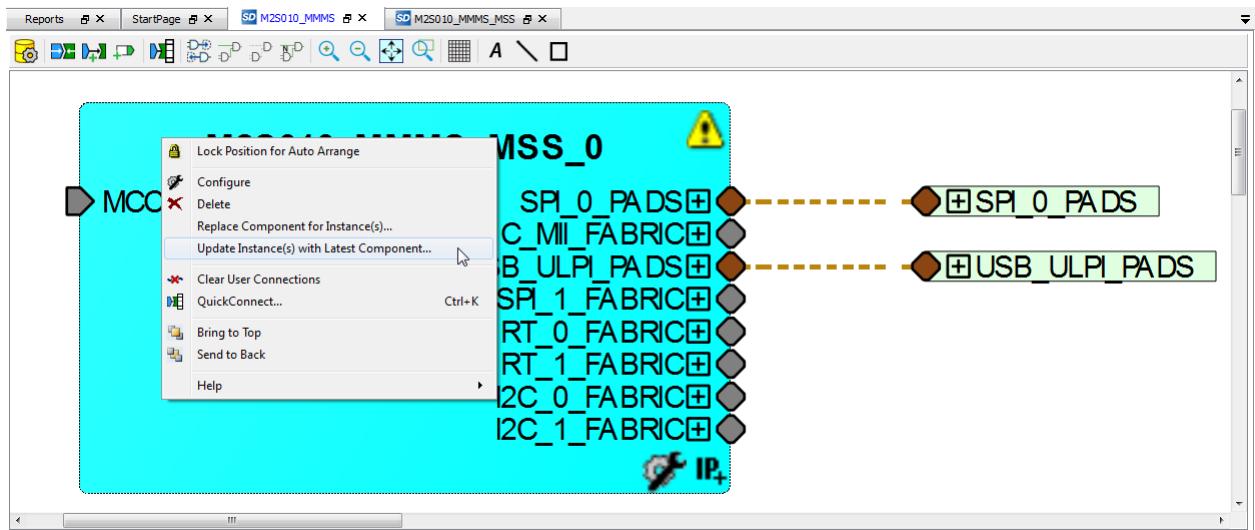


Рис. 19.

После обновления компонент должен принять вид, подобный представленному на рис. 20.

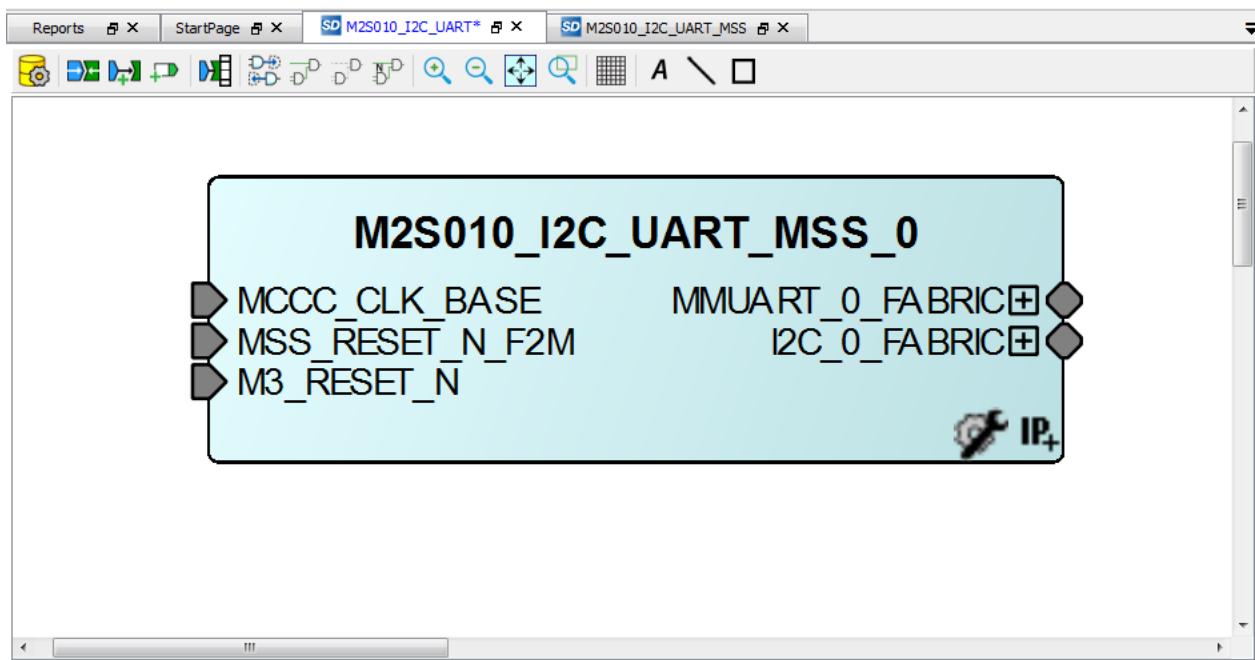


Рис. 20.

Теперь в дополнение к компоненту микроконтроллерной подсистемы из стандартного каталога Libero SoC на рабочее поле проекта нужно добавить IP-ядра и компоненты, отвечающие за тактирование, системный сброс и обмен по шине I2C.

Для реализации описанной выше функциональности проекта необходимы ядра и компоненты стандартной библиотеки компонент 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/компоненты в каталоге Libero SoC	Название в проекте	Количество в проекте	Назначение
1	Processors	SmartFusion2 Microcontroller Subsystem (MSS)	M2S010_I2C_UART_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	Clock & Management	Clock Conditioning Circuitry (CCC)	FCCC_0	1	Формирование сетки тактовых частот
5	Macro Library	BIBUF	BIBUF_0, BIBUF_1	2	Двунаправленный буфер ввода-вывода
6	Macro Library	AND3	AND3_0	1	Логический элемент AND на 3 входа

Для использования нужного компонента в проекте необходимо перейти во вкладку Catalog, раскрыть нужный раздел каталога и мышью перетащить компонент на рабочее поле проекта (рис. 21).

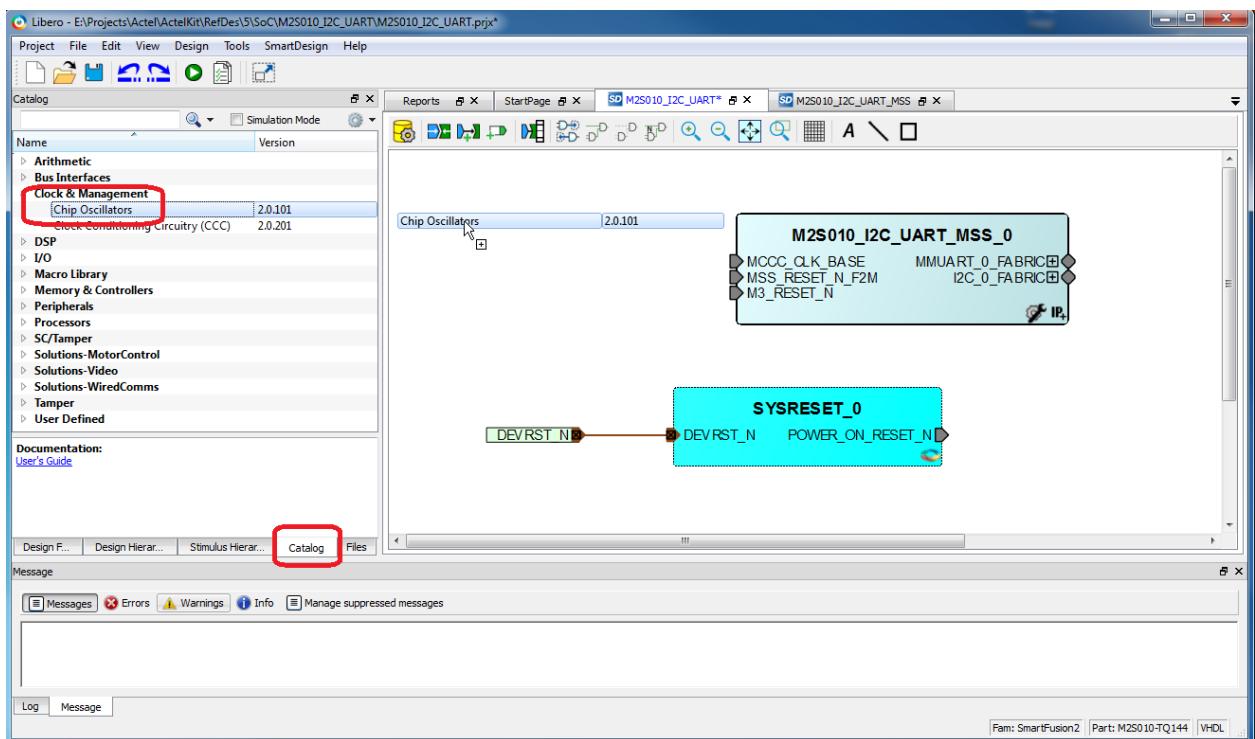


Рис. 21.

Параметры настройки источника сигнала тактирования приведены на рис. 22. Будем использовать внутренний тактовый RC-генератор опорных колебаний на 50 МГц.

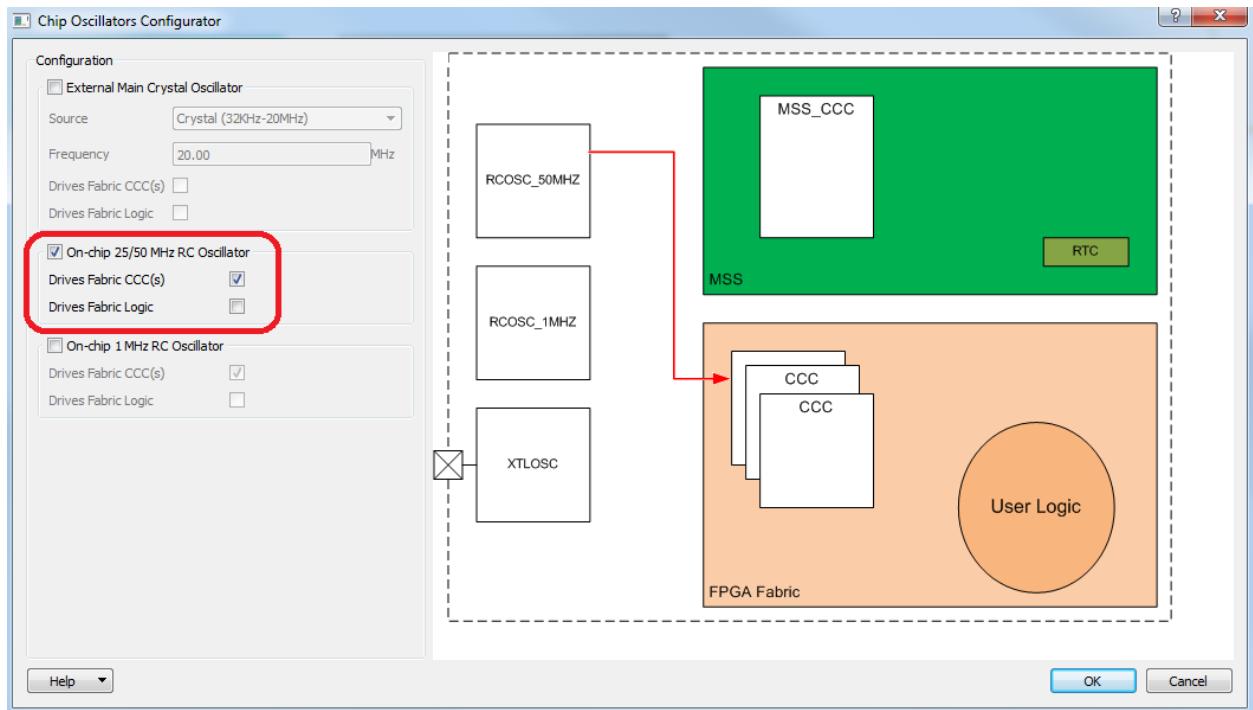


Рис. 22.

Параметры настройки схемы формирования сигналов тактовых частот представлены на рис. 23.

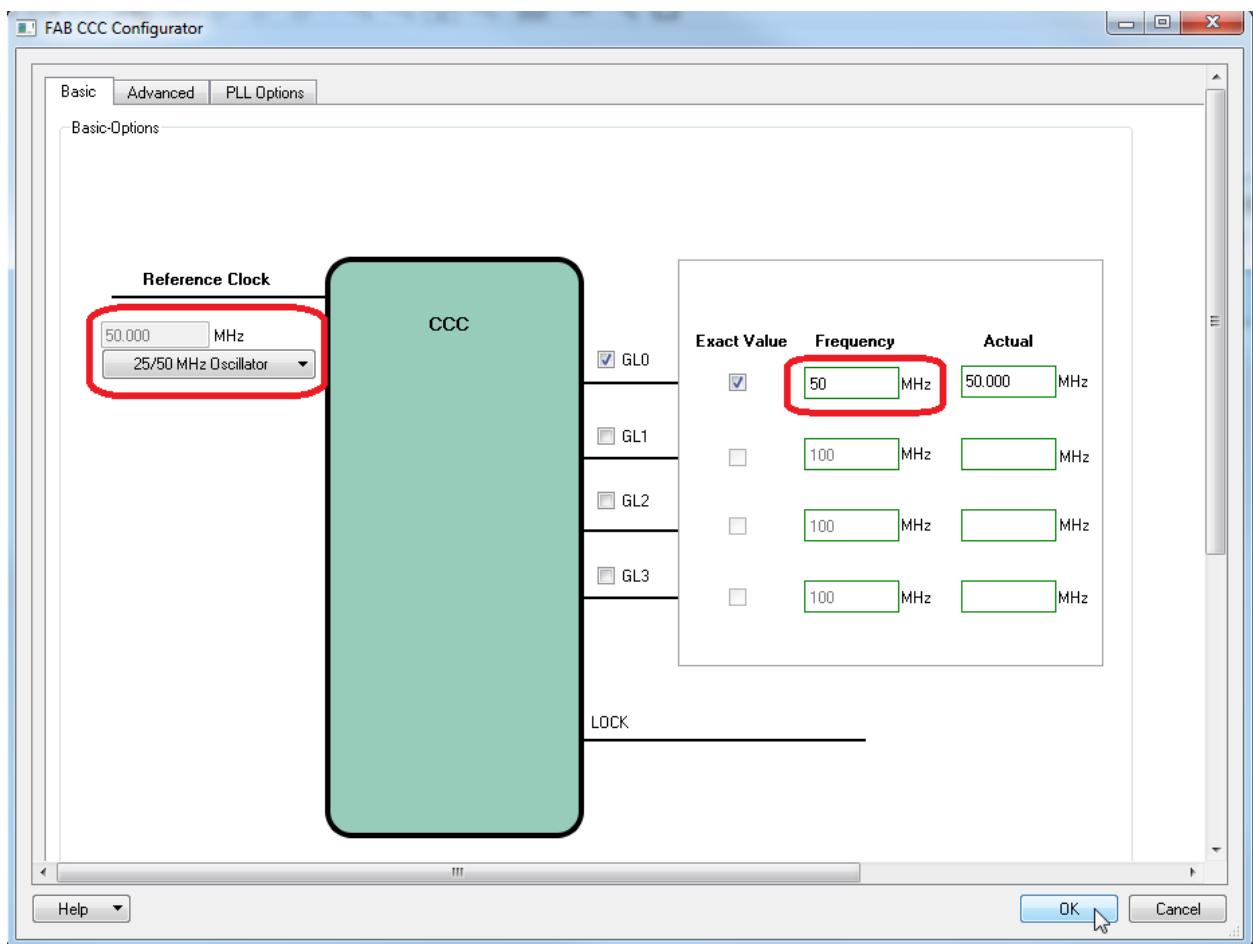


Рис. 23.

Необходимо отметить, что в данном случае показаны настройки системы формирования сигналов тактирования на ЧнК (**FAB CCC**). В проекте пользователя ЧнК SmartFusion2 может быть до четырех таких блоков. На рис. 17 показаны параметры настройки системы формирования сигналов тактирования микроконтроллерной подсистемы (**MSS CCC**). В проекте пользователя ЧнК SmartFusion2 может использоваться только одна MSS CCC (по числу микроконтроллерных подсистем SmartFusion2). Вместе, FAB CCC и MSS CCC, позволяют создать в проекте гибкую систему тактирования с множеством входных сигналов тактирования (внутренние RC-генераторы, внешние кварцевые резонаторы, внешние сигналы тактирования приходящие на контакты ЧнК SmartFusion2) и множеством внутренних тактовых частот используемых различными функциональными блоками проекта (тактовых доменов).

Теперь выполним соединения. Ниже приведен список соединений проекта в формате IP-ядро.Контакт – IP-ядро.Контакт:

1. OSC\_0.RCOSC\_25\_50MHZ\_CCC\_OUT – FCCC\_0.RCOSC\_25\_50\_CCC\_IN.
2. FCCC\_0.GL0 – M2S010\_I2C\_UART\_MSS\_0.MCCC\_CLK\_BASE.
3. FCCC\_0.LOCK – AND3\_0.A.
4. SYSRESET\_0.POWER\_ON\_RESET\_N – AND3\_0.C
5. AND3\_0.Y – M2S010\_I2C\_UART\_MSS\_0.MSS\_RESET\_N\_F2M.
6. AND3\_0.Y – M2S010\_I2C\_UART\_MSS\_0.M3\_RESET\_N.
7. M2S010\_I2C\_UART\_MSS\_0.I2C\_0\_SDA\_F2M – BIBUF\_0.Y.
8. M2S010\_I2C\_UART\_MSS\_0.I2C\_0\_SDA\_M2F – BIBUF\_0.D
9. M2S010\_I2C\_UART\_MSS\_0.I2C\_0\_SDA\_M2F\_OE – BIBUF\_0.E
10. M2S010\_I2C\_UART\_MSS\_0.I2C\_0\_SCL\_F2M – BIBUF\_1.Y
11. M2S010\_I2C\_UART\_MSS\_0.I2C\_0\_SCL\_M2F – BIBUF\_1.D
12. M2S010\_I2C\_UART\_MSS\_0.I2C\_0\_SCL\_M2F\_OE – BIBUF\_1.E

Контакты M2S010\_MMMS\_MSS\_0.MMUART\_0\_FABRIC и AND3\_0.B выводим на верхний уровень, т.е. последовательно позиционируем на них указатель мыши, нажимаем правую кнопку и выбираем команду **Promote to Top Level**.

Переименуем контакты «PAD» блоков BIBUF в соответствии с их функциональным назначением для улучшения читабельности схемы. Верхний уровень проекта после выполнения всех подключений представлен на рис. 24.

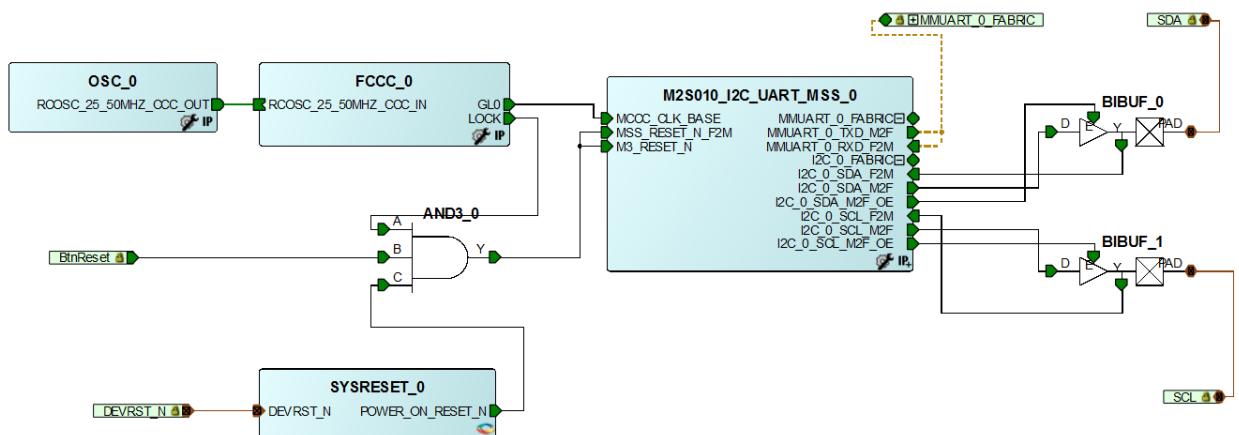


Рис. 24.

Сохраняем настройки и выполняем команду **Generate Component** (рис. 25).

Выполним синтез проекта (рис. 26) и назначим контакты микросхемы входным и выходным сигналам нашего проекта для чего выполним команду **Manage Constraints** (рис. 27), в

появившейся вкладке **Constraints Manager** выполним команду **Edit > Edit with I/O Editor** (рис. 28). В открывшемся окне утилиты I/O Editor назначаем контакты микросхемы M2S010-TQ144 входящим и выходящим сигналам нашего проекта (рис. 29). Сохраним изменения и закроем I/O Editor.

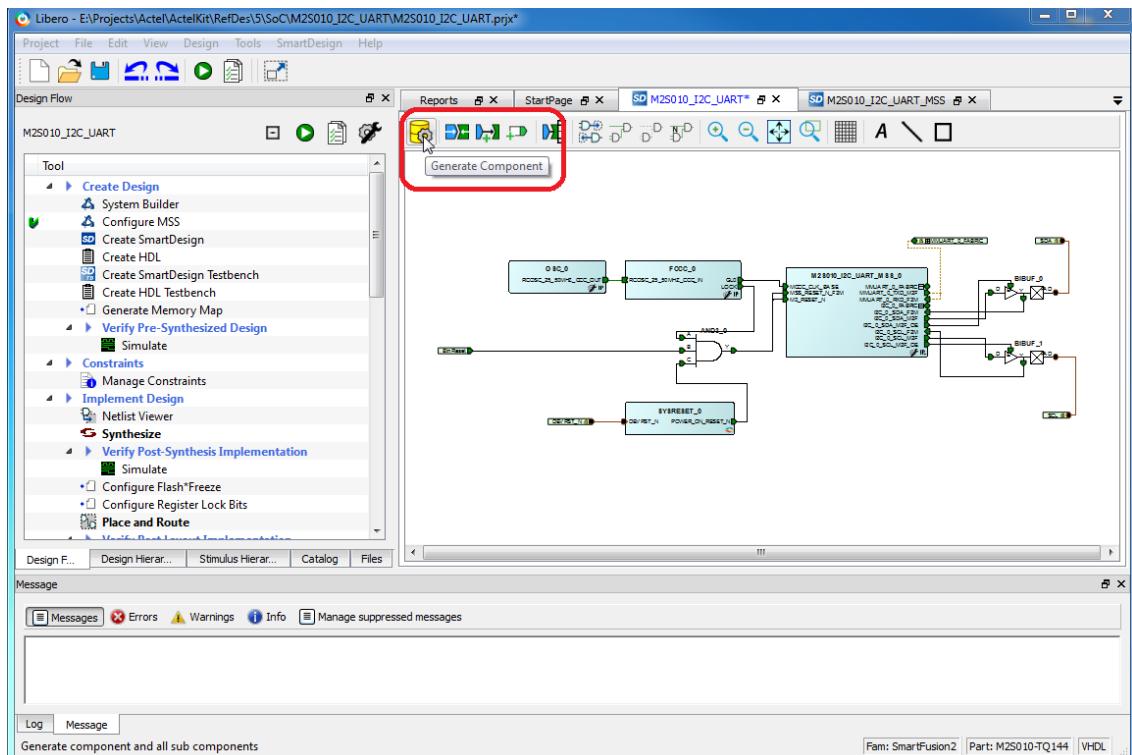


Рис. 25.

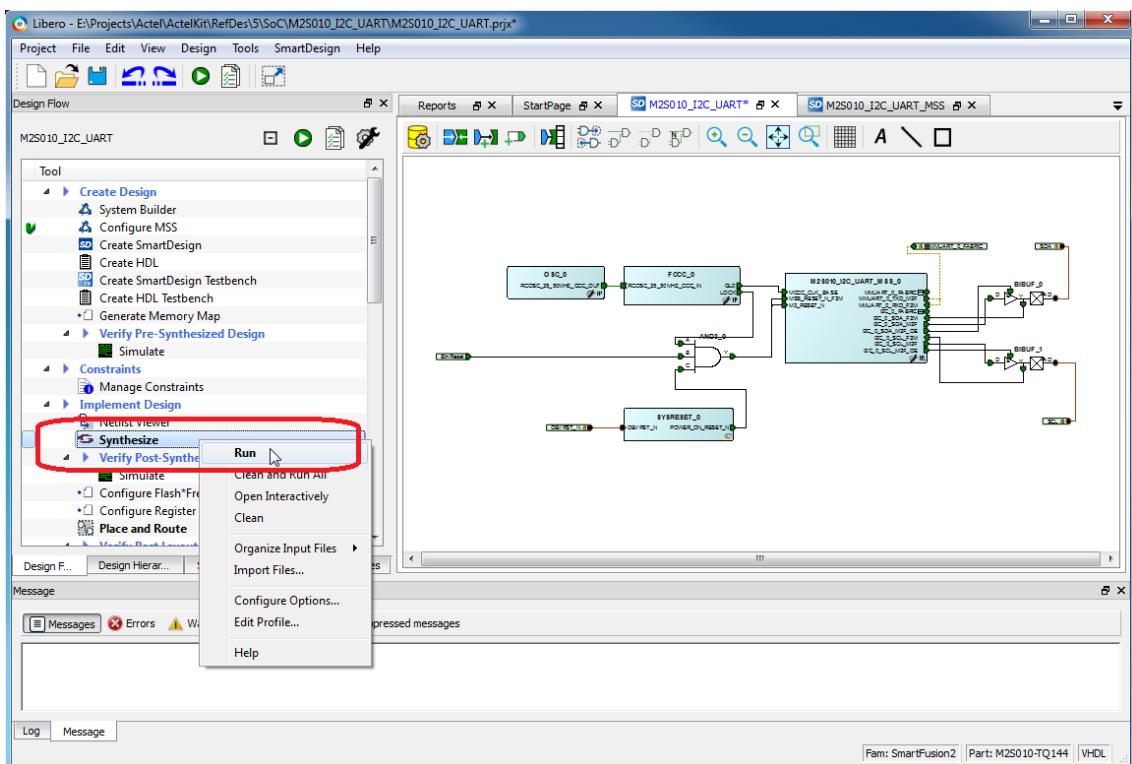


Рис. 26.

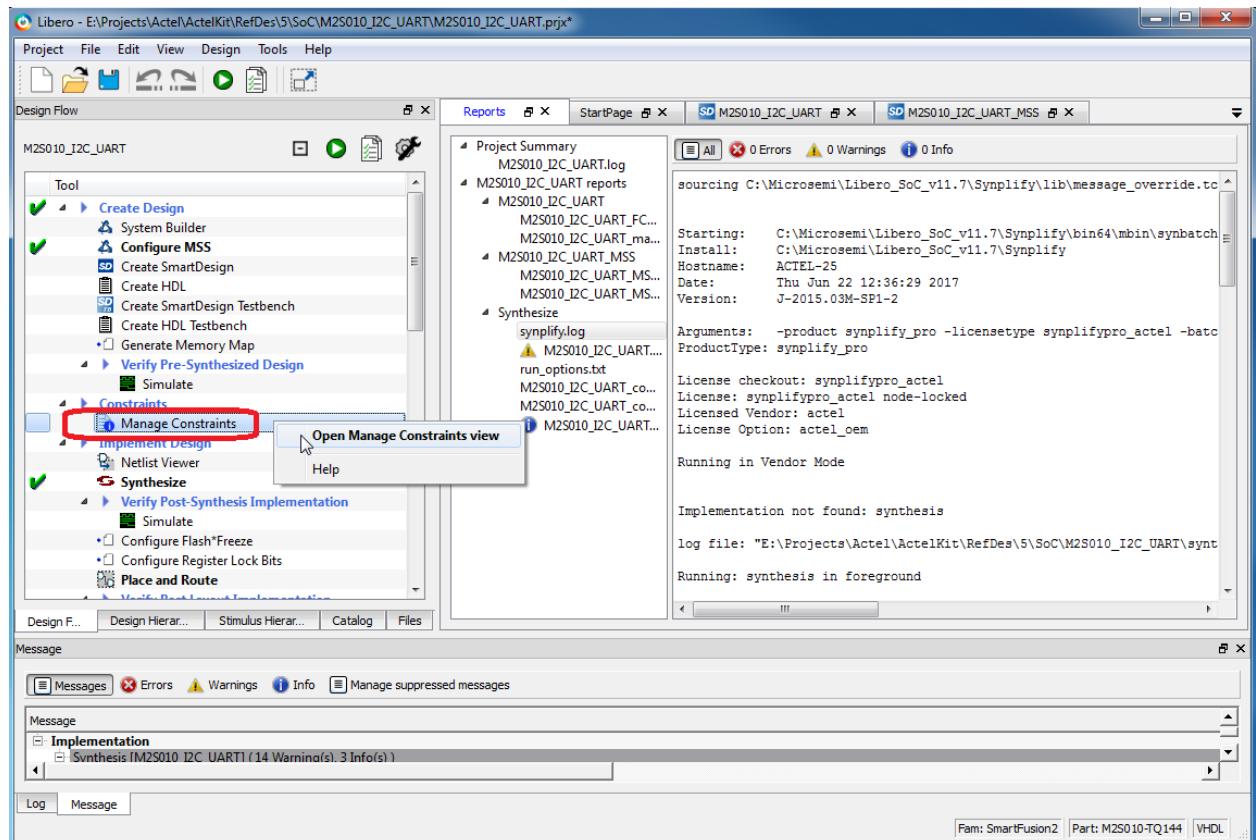


Рис. 27.

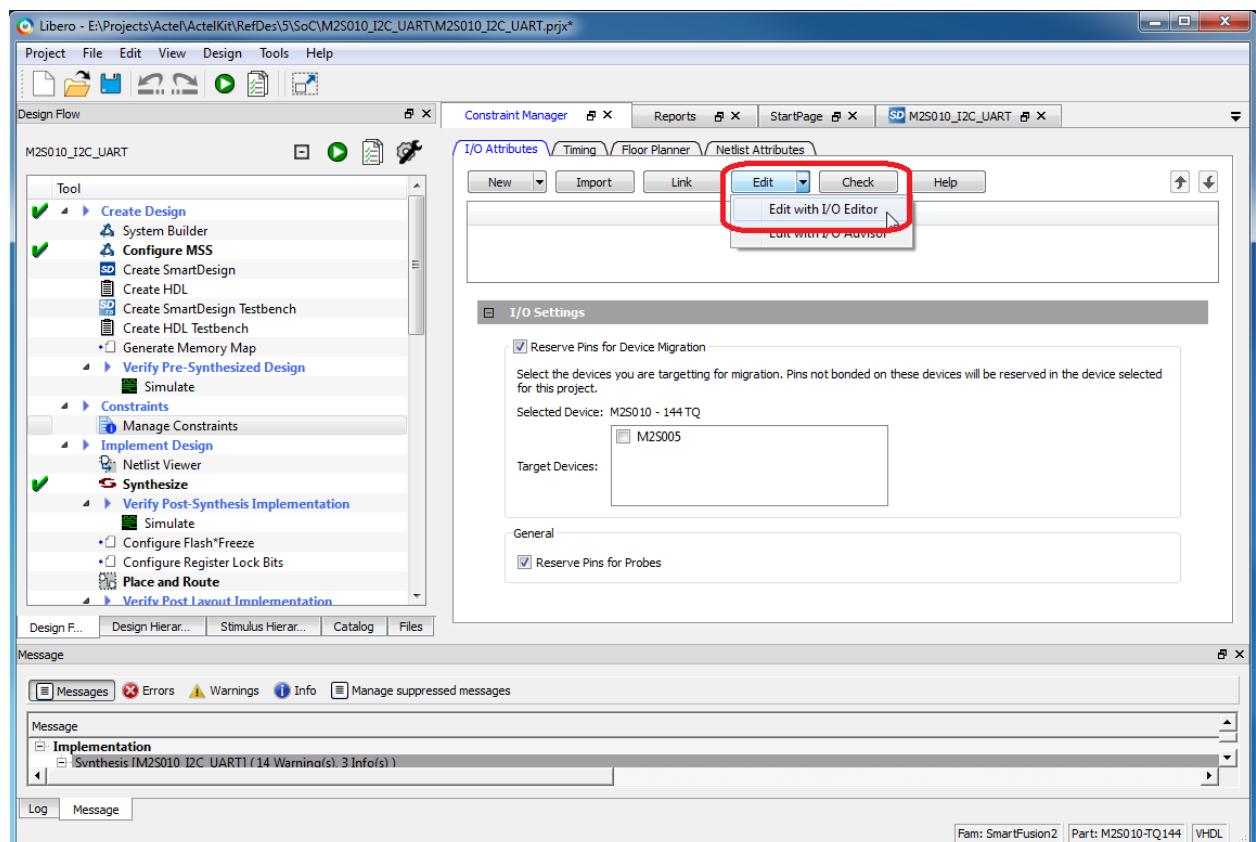


Рис. 28.

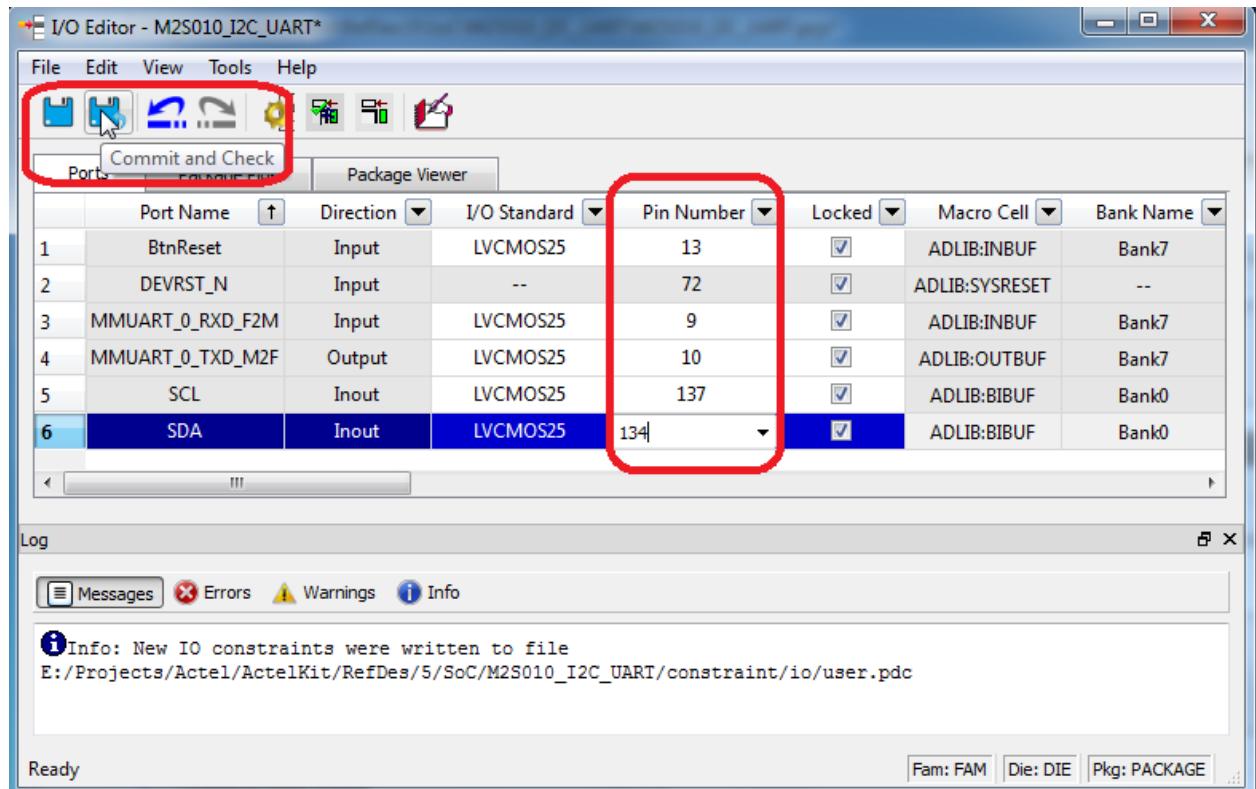


Рис. 29.

Выполним команду **Configure Firmware Cores** во вкладке **Design Flow** (рис. 30).

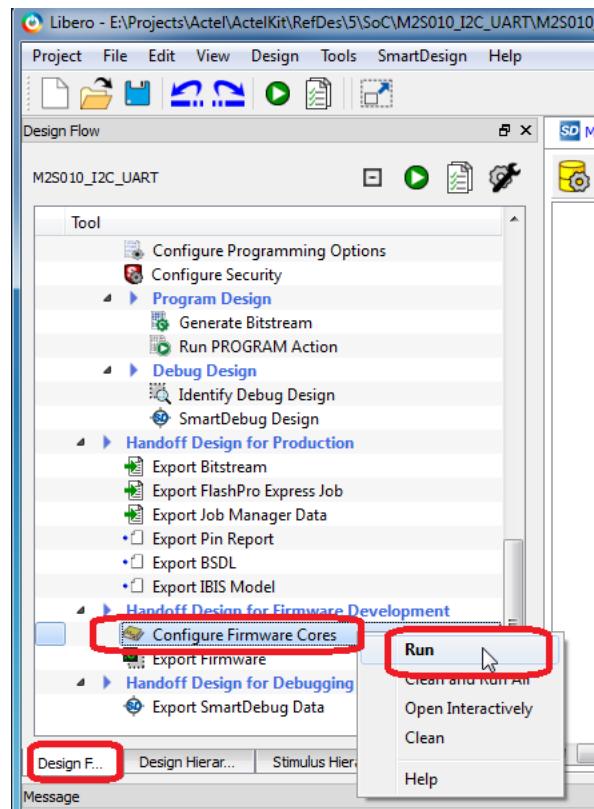


Рис. 30.

В случае если во вкладке **DESIGN\_FIRMWARE** один или несколько драйверов отображены курсивом, необходимо нажать кнопку **Download All Firmware** (рис. 31) для загрузки отсутствующих драйверов по сети Internet с сайта производителя.

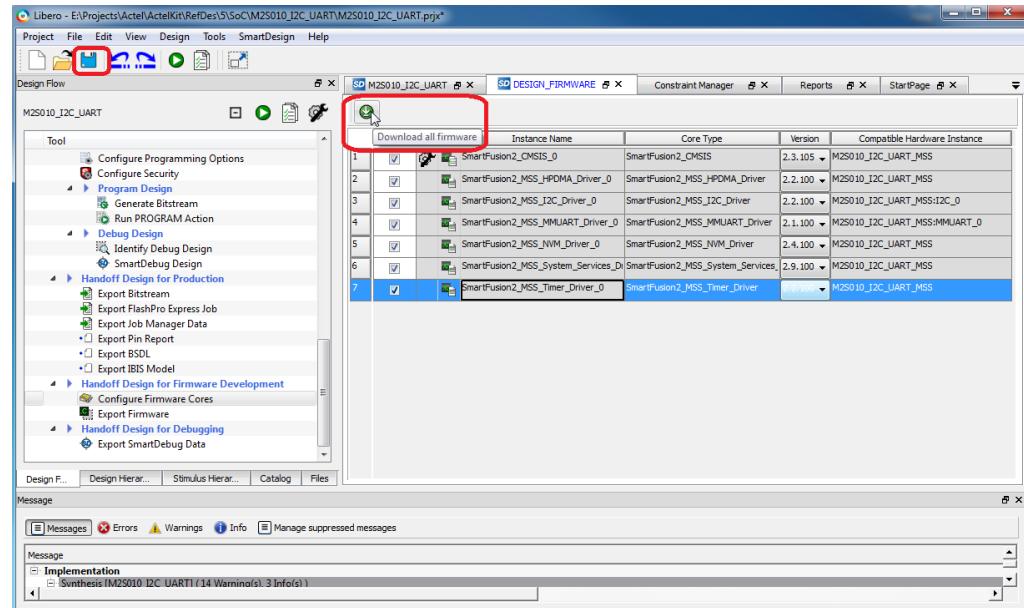


Рис. 31.

Теперь создадим проект встроенного программного обеспечения (ВПО) процессора ARM Cortex-M3 выполнив команду **Export Firmware**. В появившемся окне выберем среду разработки **SoftConsole 3.4** и опцию **Create software project including hardware configuration and firmware drivers** (рис. 32).

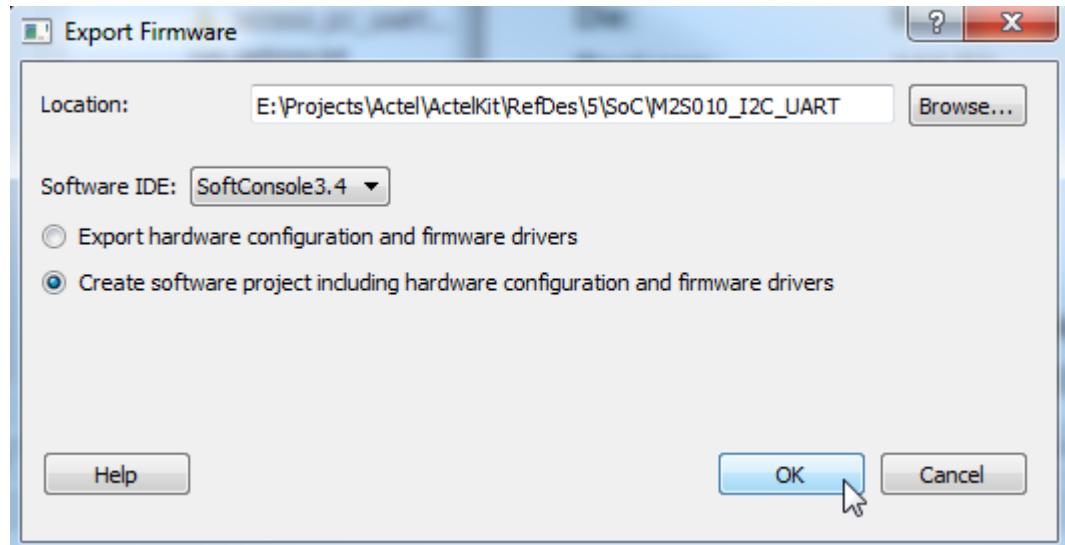


Рис. 32.

## Разработка встроенного программного обеспечения

Переходим к разработке кода встраиваемого приложения. Кратко процедура опроса датчиков, выполняемая микропроцессором, должна учесть все особенности аппаратной части, описанные выше и выполнить ряд шагов, то есть:

- 1) Перевести сенсоры из спящего режима в рабочий;
- 2) Включить шунтирующий мультиплексор для соединения основной и дополнительной шин I2C;
- 3) Считать данные сенсоров;
- 4) Скомпоновать массив данных полученных от датчиков в единый массив;
- 5) Выдать скомпонованный массив данных через интерфейс UART в ПК для обработки приложением на персональном компьютере.
- 6) Сформировать необходимый интервал временной задержки между чтением данных.
- 7) Перейти к выполнению п. 3.

Откроем созданный проект ВПО в среде SoftConsole v.3.4 (рис. 33). Заменим содержимое файла main.c и исходным кодом, содержащимся в одноименном файле из каталога /Source архива с файлами проекта, прилагаемом к данному руководству (можно заменить файл в проекте файлом из архива) (рис. 34).

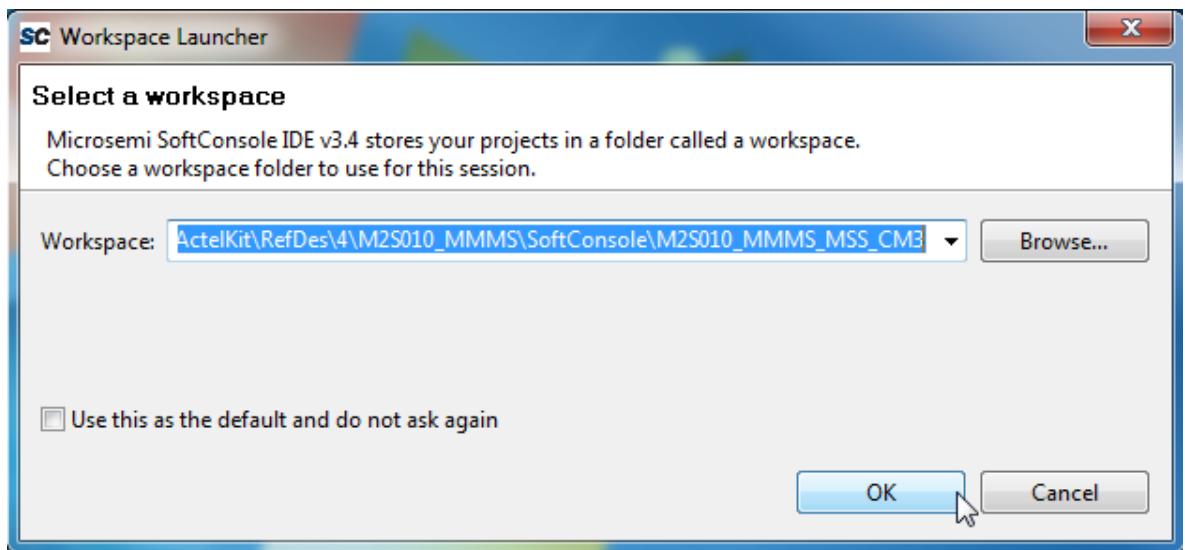


Рис. 33.

```

46 int main(void) {
47     uint8_t RegAdd[2];
48     static uint8_t stateWorking;
49
50     // инициализация UART
51     MSS_UART_init(gp_my_uart, MSS_UART_115200_BAUD, MSS_UART_DATA_8_BITS
52                   | MSS_UART_NO_PARITY | MSS_UART_ONE_STOP_BIT);
53
54     // инициализация I2C
55     MSS_I2C_init(MSS_I2C_MASTER, MSS_MASTER_SER_ADDR, clock_speed);
56
57     // включаем режим bypass дополнительного интерфейса Auxiliary I2C (ES_CI)
58     slave_address = 0x68;
59     mss_instance = I2C_writeU(0x0A, 0x06); // многократный
60     stateWorking = 0;
61
62     while (1) {
63         if (stateWorking == 0) {
64             UART_Polled_Rx(gp_my_uart, uart_rx_buf, 1);
65             if (uart_rx_buf[0] == 119) {
66                 ...
67             }
68         }
69     }

```

Рис. 34.

Установим активной конфигурацию Release (рис. 35).

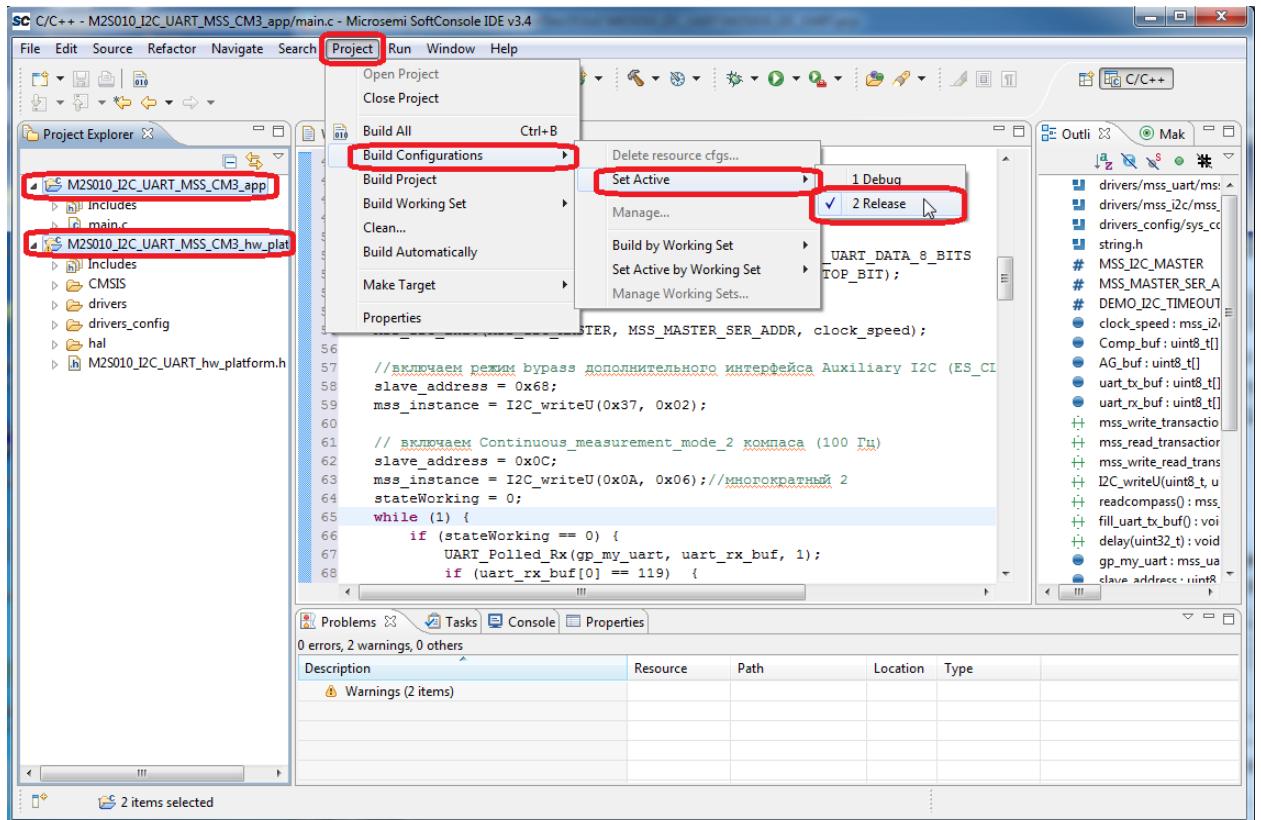


Рис. 35.

Выделив проект M2S010\_I2C\_UART\_MSS\_CM3\_app, отредактируем его свойства (рис. 36). Во вкладке C/C++ Build > Settings перейдем в окно Tool Settings и в настройках GNU C Linker > Miscellaneous (рис. 37) изменим значение параметра Linker flags на

`-T${workspace_loc:/M2S010_I2C_UART_MSS_CM3_hw_platform/CMSIS/startup_gcc/production-smartfusion2-execute-in-place.ld}`

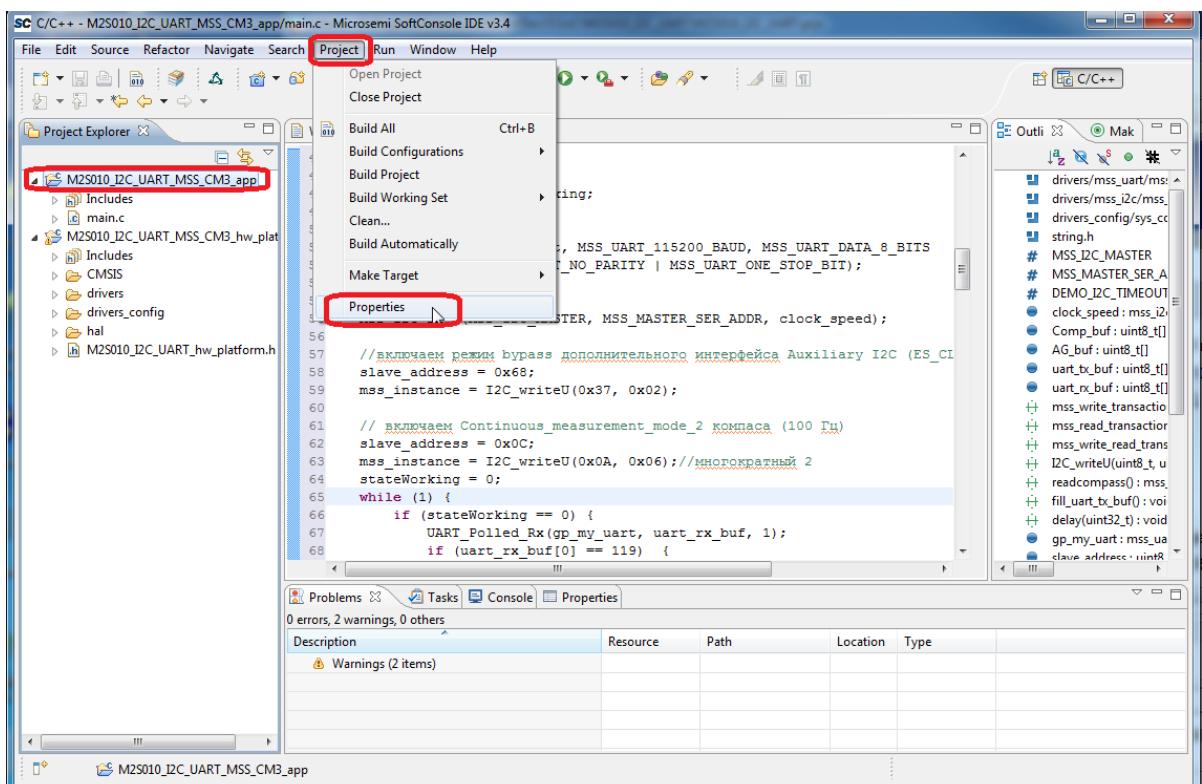


Рис. 36.

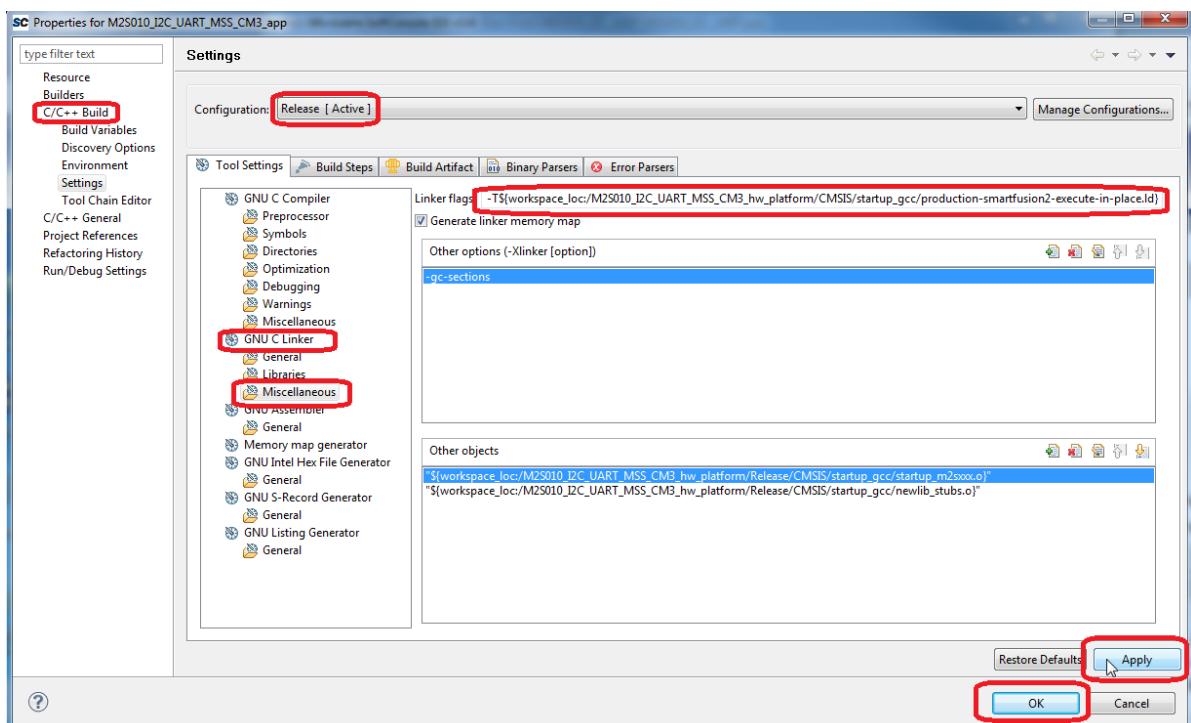


Рис. 37.

Выполним команду основного меню **Project > Clean** и получим файл с исполняемым кодом нашего приложения в паке Release проекта ВПО (рис. 38).

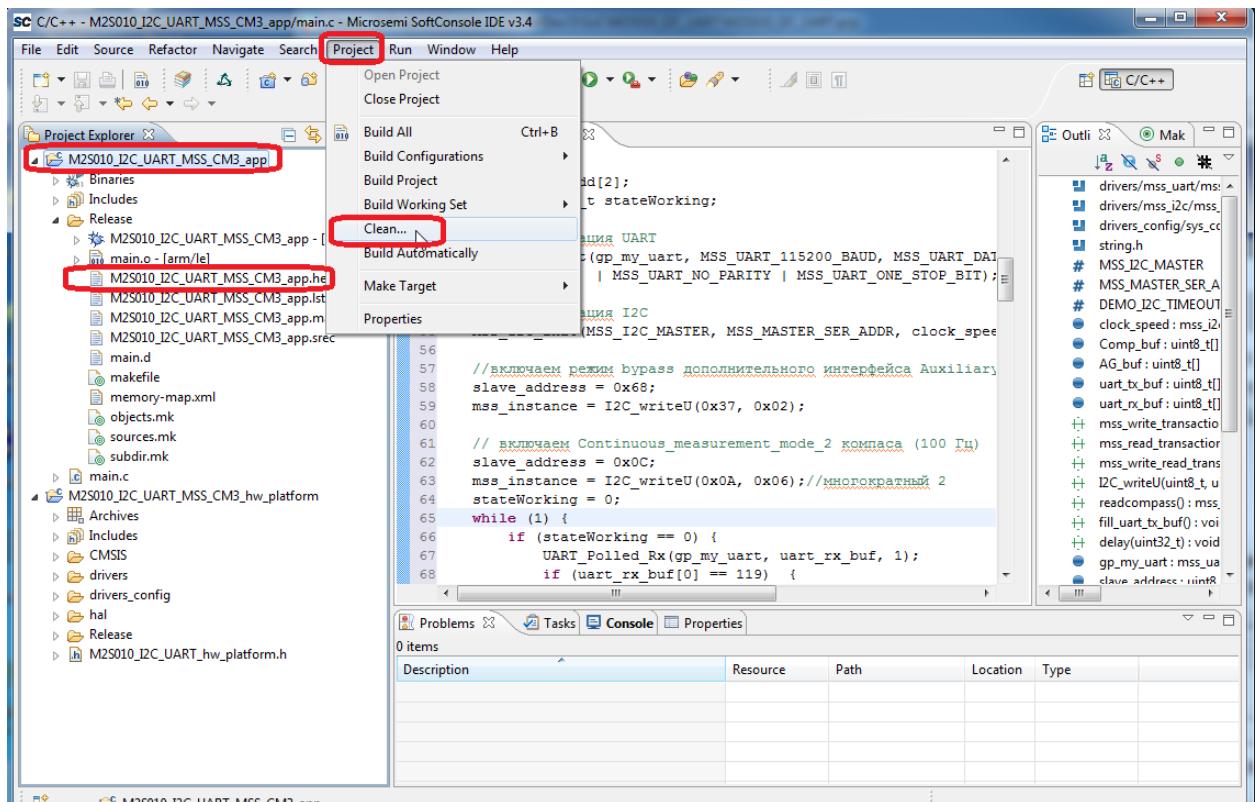


Рис. 38.

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 битстрим FPGA Fabric и ВПО.

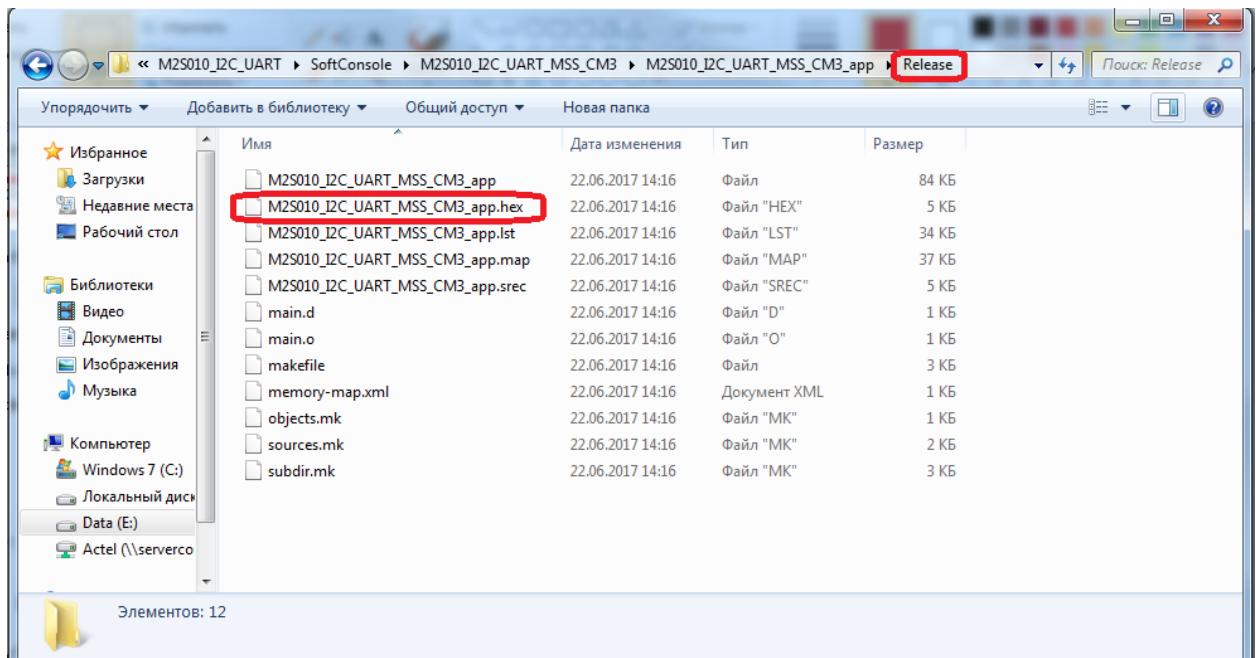


Рис. 39.

## 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. 40).

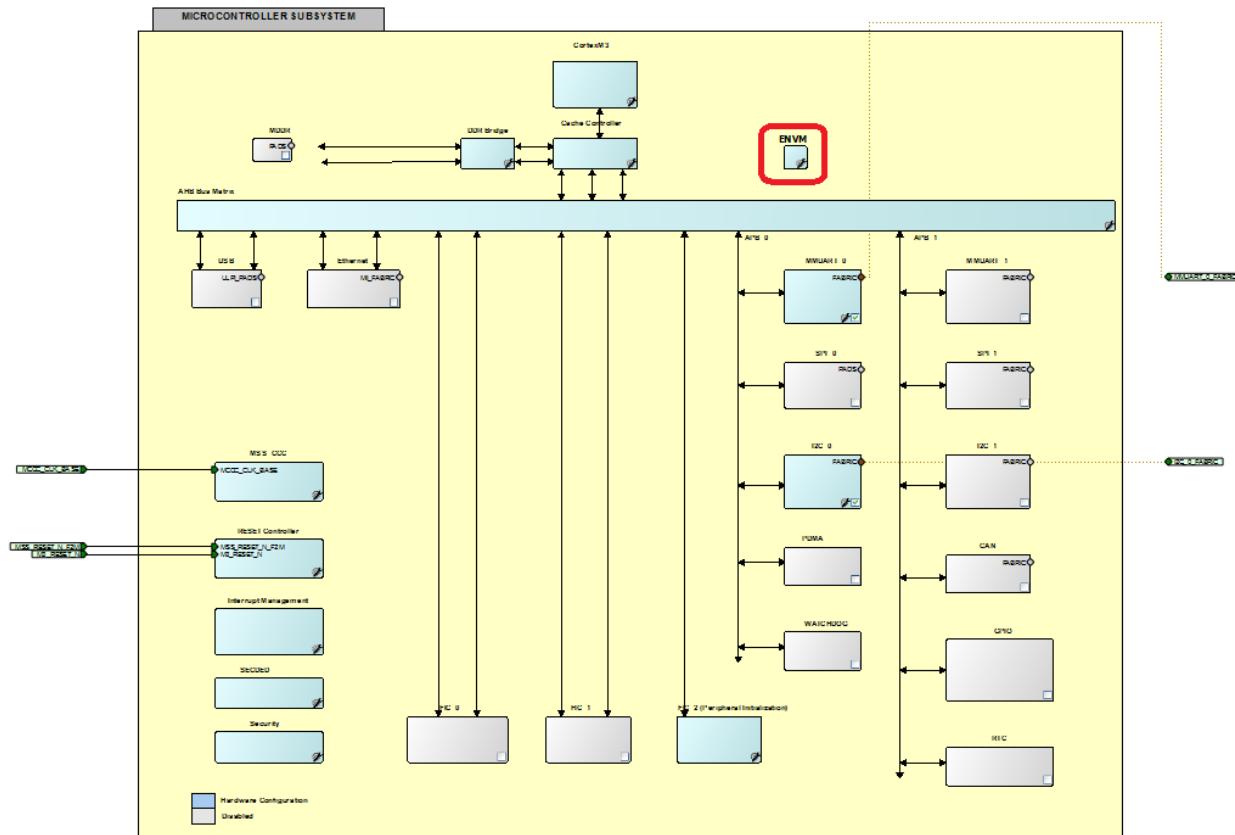


Рис. 40.

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 прошивки ВПО M2S010\_I2C\_UART\_app.hex (рис. 41).

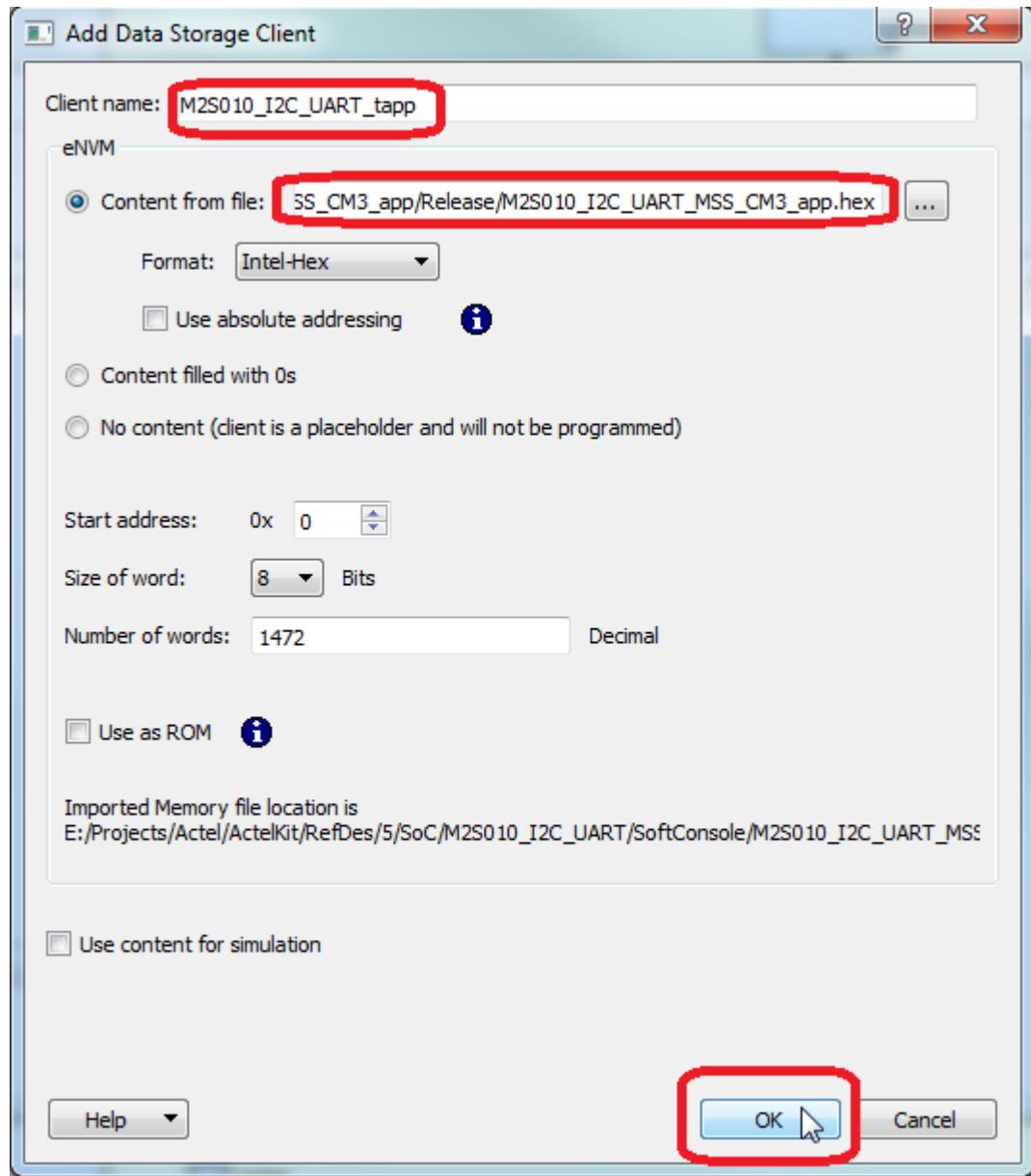


Рис. 41.

После чего окно **eNVM Configurator** примет вид, представленный на рис. 42. В окне отобразится строка с основными характеристиками загруженного файла.

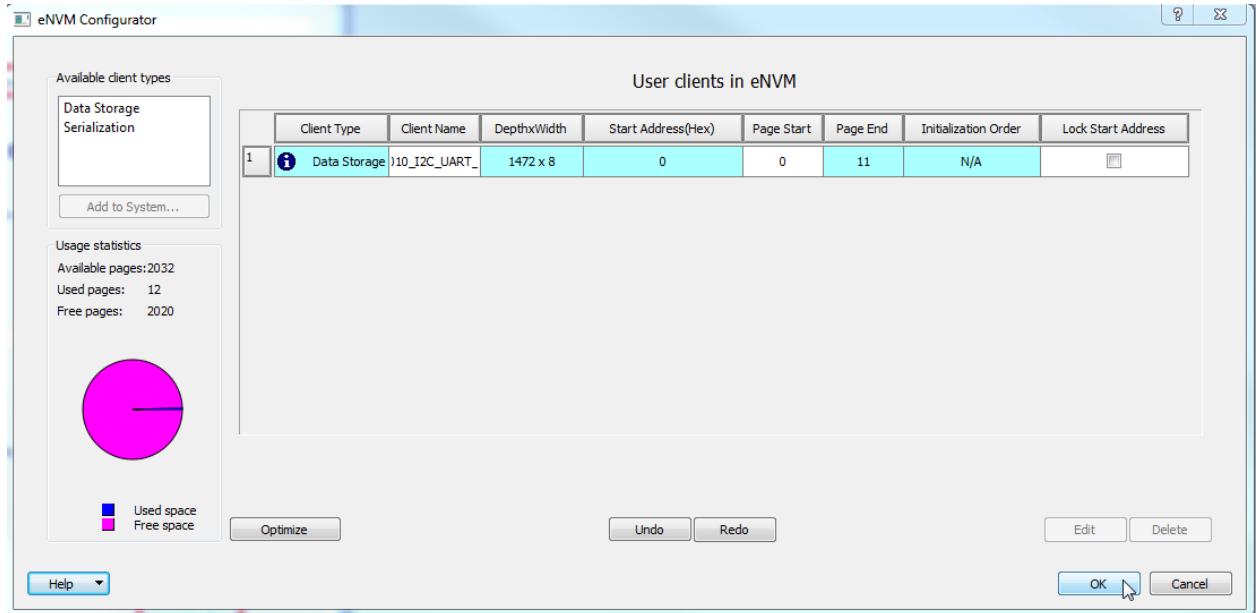


Рис. 42.

Сохраним изменения, выполним процедуру генерации компонента (рис. 43).

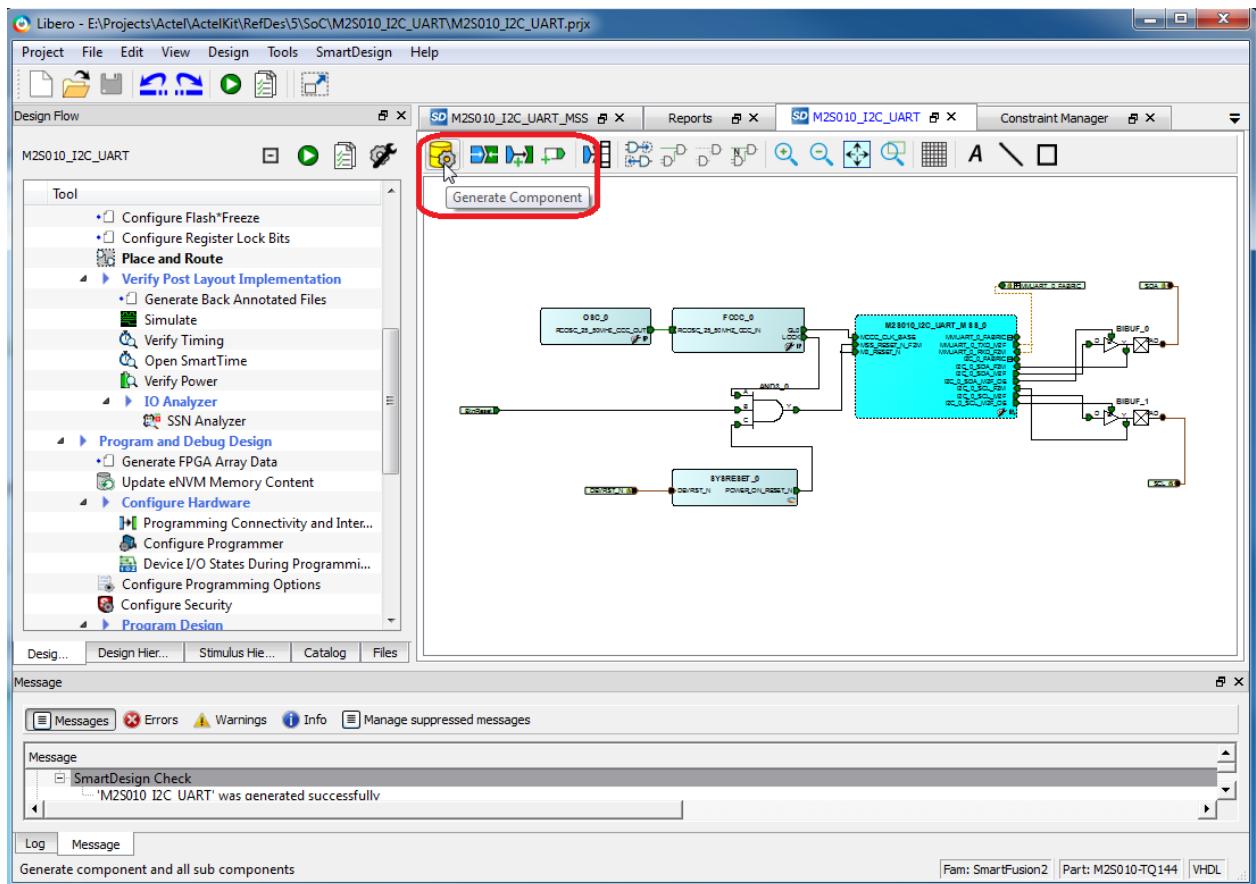


Рис. 43.

Подключим программатор к отладочному набору и персональному компьютеру, запустим процесс генерации файла прошивки и программирования (рис. 44).

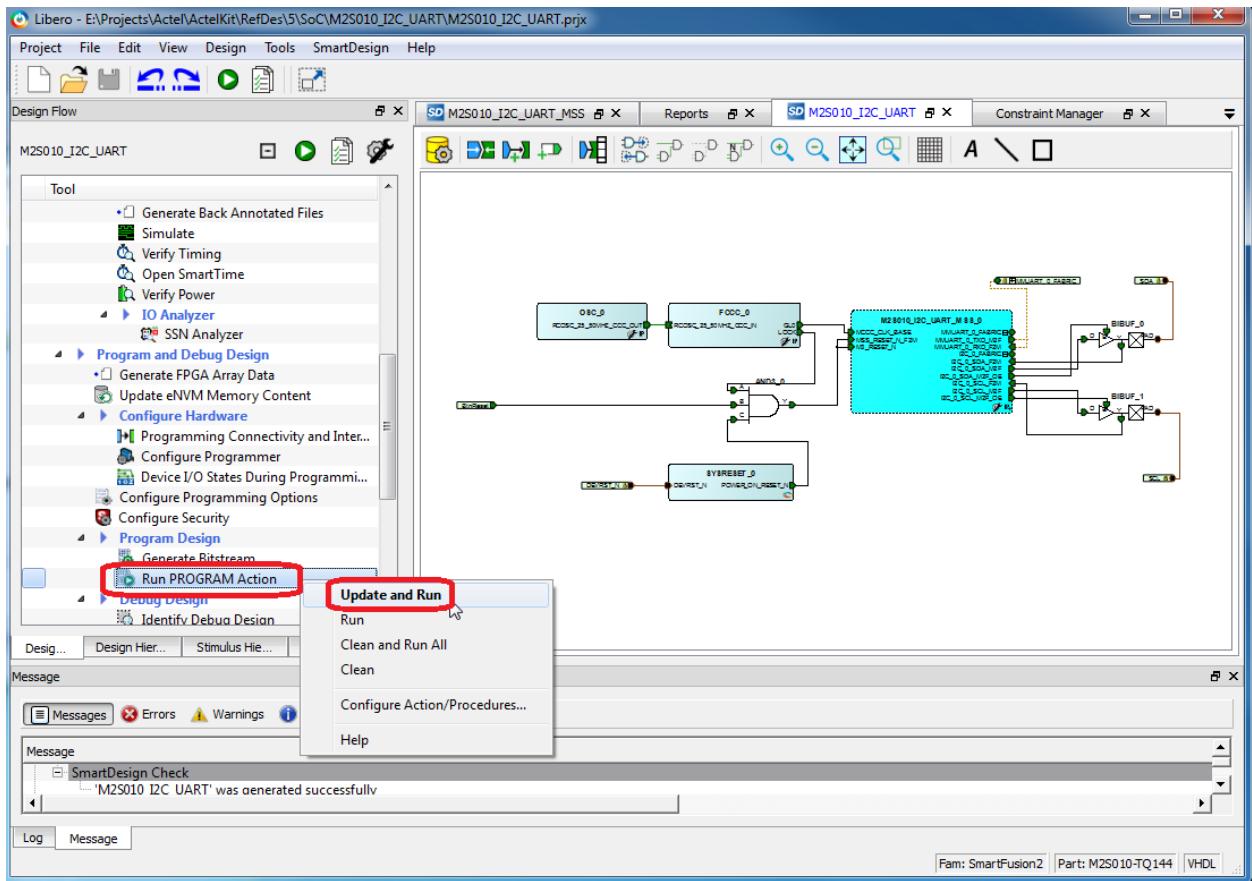


Рис. 44.

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 отладочного набора. В данном проекте контакты микросхемы M2S010-TQ144 назначены таким образом, чтобы вывести все входящие и выходящие сигналы проекта (SCL, SDA, UART\_Rx, UART\_Tx) на указанные разъемы GPIO. В соответствии с таблицей 2 необходимо подключить сигналы SCL и SDA к одноименным контактам модуля GY-91. Выполнить соединения можно с помощью соединительных проводов, входящих в состав набора.

Таблица 2. Подключение перемычек к разъему X9 отладочного набора SF2-Junior-KIT

№ контакта разъема X9 SF2-Junior-KIT	Номер контакта M2S010-TQ144	Название цепи проекта СнК	Название контакта модуля GY-91
11	134	SDA	SDA
6	137	SCL	SCL
3	-	-	3V3
2	-	-	GND

Согласно спецификации рабочий диапазон напряжений MPU-9255 2,4 ... 3,6В. Подключим питающее напряжение +2,5В контакта 3 разъема X9 на контакт 3V3 модуля GY-91, а нулевой потенциал (GND) со второго контакта X9 к одноименному контакту GY-91.

В состав набора SF2-Junior-KIT входят приемопередатчики USB – Bluetooth dongle и модуль UART- Bluetooth HC-06. Для связи отладочного набора с персональным компьютером

(ПК) необходимо в USB разъем ПК включить USB – Bluetooth dongle, а к разъему X10 отладочного набора в соответствии с таблицей 3 подключить модуль HC-06.

Таблица 3. 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

Выполняем подключения, описанные в таблицах 2, 3 (рис. 45). Прошиваем СнК конфигурационной последовательностью нашего проекта.

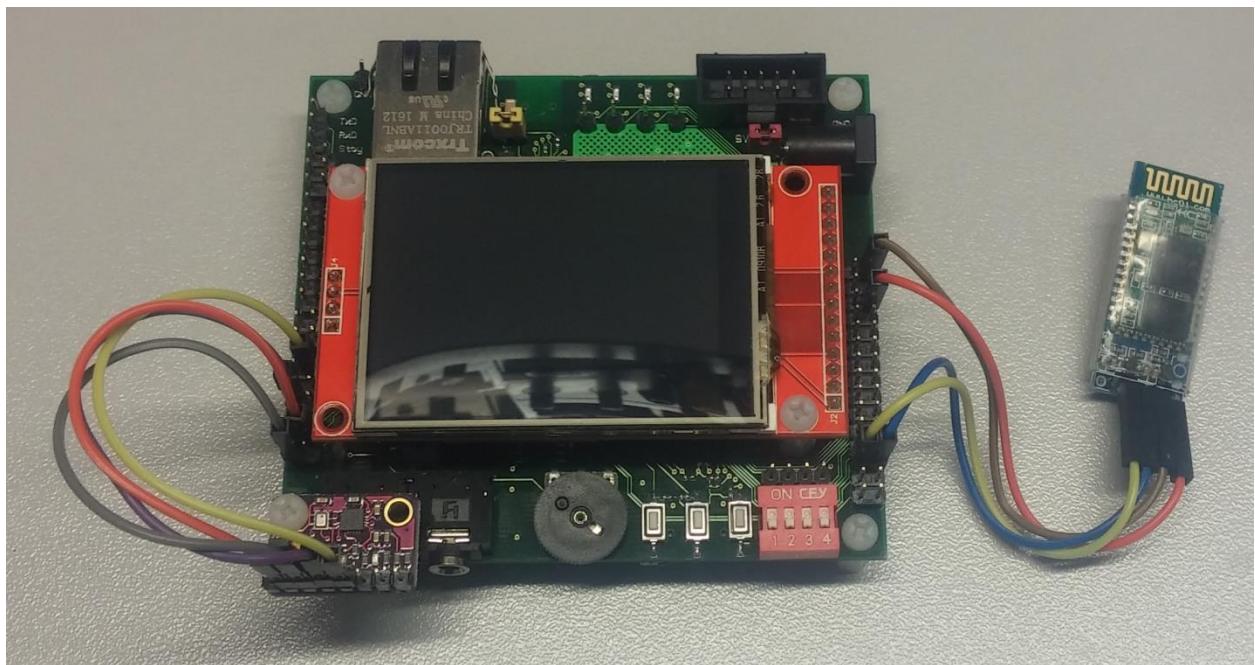


Рис. 45.

Для отображения результатов измерений на экране ПК воспользуемся программой визуализатором трехмерных сцен 3D-Viewer.exe. В данной программе самолет принимает положение в виртуальном пространстве в соответствии с данными получаемыми от гироскопа. Кроме того, данные всех подключенных датчиков отображаются на экране (рис. 46).

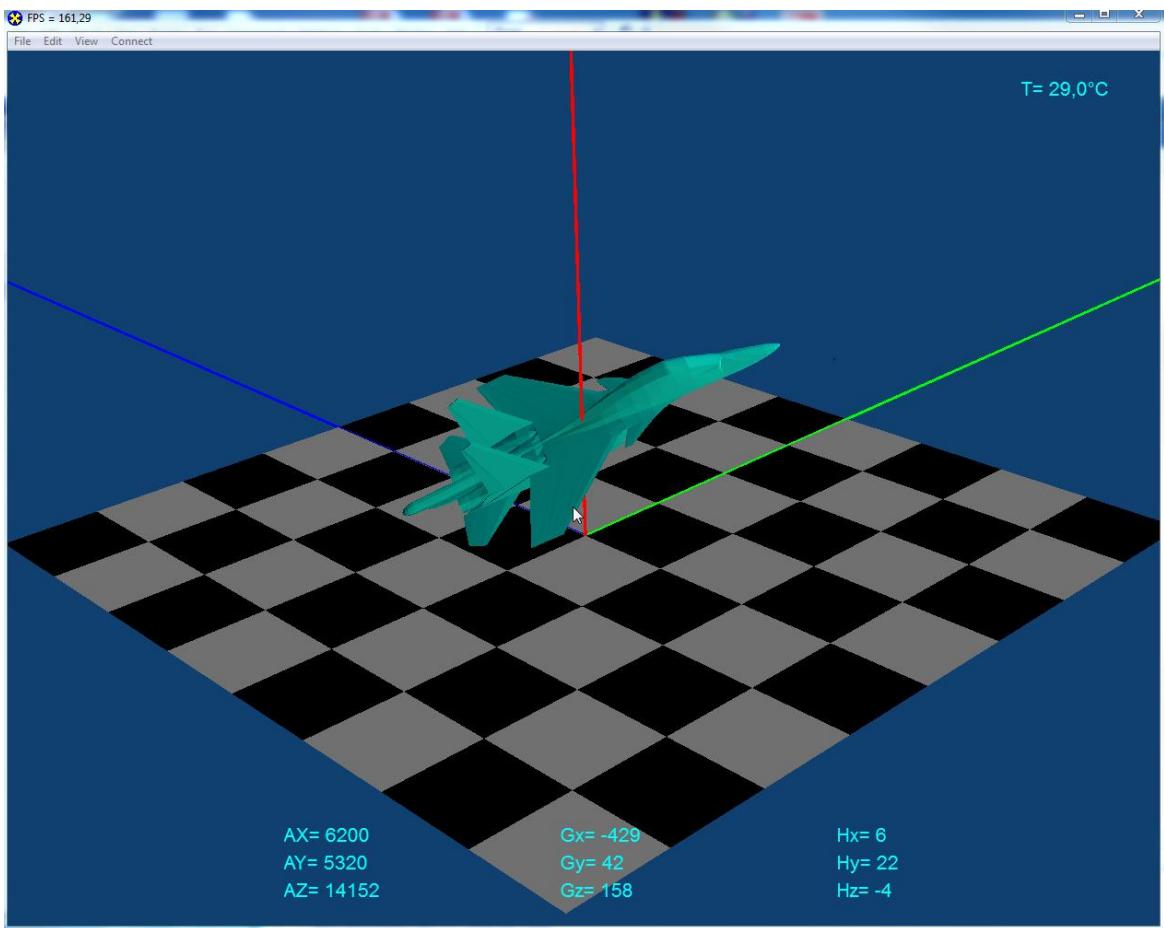


Рис. 46.

Проекты Снк и ВПО, разработка которых описана в данном руководстве, файлы с исходным кодом встроенного программного обеспечения, а также утилитой для визуализации работы приложения на экране ПК 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**.