



Scan Engineering Telecom SPb

Демонстрационный проект для MicroBlaze на базе FreeRTOS

Руководство пользователя

Версия 1.0



Код документа: UG-FPGA-00-FRTOS-APP-DEMO
Дата сборки: 28 августа 2015 г.
Листов в документе: 26

© 2015, ООО «Скан Инжиниринг Телеком - СПб»
<http://www.setdsp.ru>

Содержание

Перечень рисунков	3
Перечень листингов	3
Перечень процедур	3
Перечень сокращений и условных обозначений	4
1 Введение	6
2 Описание проекта	7
2.1 Основа для реализации	7
2.2 Организация проекта	7
2.3 Область размещения и условия запуска	7
2.4 Структура каталогов проекта	8
2.5 Этапы работы	9
3 Компиляция проекта	11
3.1 Управление компиляцией	11
3.2 Подготовка к компиляции	15
3.3 Компиляция	20
4 Запуск проекта	22
4.1 Через JTAG интерфейс	22
4.2 Из флеш-памяти	25
Список литературы	26

Перечень рисунков

2-1	Уровни организации исходного кода проекта	8
3-1	Уровни организации исходного кода проекта	12
3-2	Сегменты кода откомпилированного образа	15
3-3	Выбор места расположения рабочего пространства в Xilinx SDK	15
3-4	Выбор пункта меню настройки репозитариев в Xilinx SDK	16
3-5	Настройка местоположения репозитариев в Xilinx SDK	16
3-6	Создание проекта аппаратной платформы в Xilinx SDK	17
3-7	Настройки проекта аппаратной платформы в Xilinx SDK	17
3-8	Создание проекта BSP ОС FreeRTOS в Xilinx SDK	18
3-9	Настройки проекта BSP ОС FreeRTOS в Xilinx SDK	18
3-10	Параметры проекта BSP ОС FreeRTOS в Xilinx SDK	19
3-11	Импорт проекта в Xilinx SDK	19
3-12	Выбор проекта для импорта в Xilinx SDK	20
3-13	Сборка проекта в Xilinx SDK	20
3-14	Содержимое файла «Iscrip.td» в Xilinx SDK	21
3-15	Протокол сборки проекта в Xilinx SDK	21
4-1	Выбор режима программирования FPGA через JTAG интерфейс в Xilinx SDK	22
4-2	Выбор загружаемого в FPGA микросхему двоичного образа «bootloop» в Xilinx SDK	23
4-3	Создание цели запуска в режиме отладки проекта «FreeRTOS_FPGA_Demo» через JTAG интерфейс в Xilinx SDK	23
4-4	Окно отладки двоичного образа проекта «FreeRTOS_FPGA_Demo» в Xilinx SDK	24
4-5	Запуск работы двоичного образа проекта «FreeRTOS_FPGA_Demo» в Xilinx SDK	24
4-6	Терминальный вывод сообщение о работе двоичного образа проекта «FreeRTOS_FPGA_Demo»	25

Перечень листингов

2-1	Общий вид структуры каталогов проекта «FreeRTOS_FPGA_Demo»	8
3-1	Возможные числовые значения символа G_APP_HW_PLATFORM	12
3-2	Включение поддержки в исполняемый бинарный образ различных IP-ядер, submodule, микросхем	13
3-3	Настройки FMC submodule SFM-4A250	13
3-4	Включения поддержки в бинарный образ всех устройств типа uart_lite	14
3-5	Включении поддержки CLI	14
3-6	Включение поддержки технологии SETFabric-Stream™	14

Перечень процедур

3-1	Выбор места расположения рабочего пространства в Xilinx SDK	15
3-2	Выбор места расположения репозитариев	15
3-3	Создание нового проекта для аппаратной платформы в Xilinx SDK	16
3-4	Создание нового проекта BSP поддержки аппаратной платформы на базе FreeRTOS» в Xilinx SDK	18
3-5	Импорт проекта «FreeRTOS_FPGA_Demo» в рабочее пространство	19
3-6	Компиляция рабочей версии	20
4-1	Запуск двоичного образа проекта «FreeRTOS_FPGA_Demo» через JTAG интерфейс	22

Перечень сокращений и условных обозначений

ADC	Analog-to-Digital Converter	10
AMC	Advanced Mezzanine Card	9
AXI	Advanced eXtensible Interface	9, 10
BRAM	Block Random Access Memory	7, 8, 14
BSP	Board Support Package	3, 7, 9, 14, 18, 19
CDMA	Central Direct Memory Access	12
CLI	Command Line Interface	3, 12–14
DAC	Digital-to-Analog Converter	10
DDR	Double Data Rate	8, 10, 14, 23
ELF	Executable and Linkable Format	14, 25
FMC	FPGA Mezzanine Card	3, 9–13
FPGA	Field-Programmable Gate Array	3–12, 22, 23
IP	Intellectual Property	3, 6, 7, 9, 10, 12, 13
JTAG	Joint Test Action Group	2, 3, 11, 22, 23
MBL	MicroBlaze Boot Loader	8, 20, 22, 25
PCIe	PCI Express	10, 12
PCI	Peripheral Component Interconnect	4
SDK	Software Development Kit	3, 7–9, 11, 12, 15–24
SDRAM	Synchronous Dynamic Random Access Memory	8, 10, 14, 23
SPI	Serial Peripheral Interface	8
SREC	Motorola S-record format	25
SRIO	Serial RapidIO	10, 12
TCL	Tool Command Language	7
USB	Universal Serial Bus	22
VHDL	VHSIC Hardware Description Language	6
VHSIC	Very High Speed Integrated Circuits	4
ЗАО	Закрытое Акционерное Общество	5–7, 9, 11, 14
ОС	Операционная Система	3, 7, 18, 19
ОСПВ	Операционная Система Реального Времени	7

Общие сведения

Данный документ описывает программный проект, который предназначен для демонстрации функциональных возможностей различных аппаратных платформ, созданных разработчиками ЗАО «Скан Инжиниринг Телеком» для FPGA модулей собственного производства. Программный проект носит название «FreeRTOS_FPGA_Demo».

1 Введение

Использование готовых FPGA модулей производства ЗАО «Скан Инжиниринг Телеком» в качестве аппаратной основы для реализации системных решений пользователя и предполагает разработку для этих модулей различных структур, загружаемых в FPGA микросхемы. Такие структуры формируются пользователем на языках «Verilog» и «VHDL» путем объединения создаваемых функциональных блоков. Получаемые в результате объединения функциональных блоков структуры в терминологии Xilinx носят название аппаратные платформы.

Разработчики ЗАО «Скан Инжиниринг Телеком» предлагают пользователям FPGA модулей брать за основу для организации собственных аппаратных платформ универсальную структуру, называемую «Микропроцессорная система на MicroBlaze». Описание которой приведено в документе [1].

«Микропроцессорная система на MicroBlaze» представляет собой структуру, описывающую архитектуру микропроцессорной системы, построенной на основе микропроцессорного IP-ядра MicroBlaze. Полноценное функционирование этой микропроцессорной системы возможно только при наличии двоичного образа приложения пользователя, исполняемого ее процессором. В качестве приложения пользователя разработчиками ЗАО «Скан Инжиниринг Телеком» предлагается программный проект «FreeRTOS_FPGA_Demo».

2 Описание проекта

Проект «FreeRTOS_FPGA_Demo» реализуется в среде разработки Xilinx SDK. Исходный текст проекта написан на языке «Си». Получаемый в результате компиляции проекта двоичный образ предназначен для исполнения на микропроцессорном ядре MicroBlaze.

Проект «FreeRTOS_FPGA_Demo» является демонстрационным проектом и может входить в состав различных демонстрационных пакетов BSP, предназначенных для различных FPGA модулей производства ЗАО «Скан Инжиниринг Телеком». Проект поставляется с текстами программ в открытом исходном виде.

Разработчики компании ЗАО «Скан Инжиниринг Телеком» рекомендуют использовать заложенные в проект «FreeRTOS_FPGA_Demo» функциональные решения в качестве основы для построения собственных приложений пользователя, ориентированных на исполнение микропроцессорным ядром MicroBlaze для тех аппаратных платформ, которые строятся на основе «Микропроцессорной системы на MicroBlaze».

2.1 Основа для реализации

В основе реализации проекта «FreeRTOS_FPGA_Demo» лежат принципы построения приложения пользователя для встраиваемых систем на базе OCPB.

В качестве OCPB для проекта «FreeRTOS_FPGA_Demo» используется «FreeRTOS». Это широко применяемая в мире встраиваемых систем OCPB, имеющая очень компактное ядро ОС, поддерживающее вытесняющую многозадачность, а также имеет минимальный, но функционально достаточный, набор объектов синхронизации, используемых в написании кода приложений пользователя.

Реализация проекта «FreeRTOS_FPGA_Demo» на базе OCPB «FreeRTOS» позволяет сформировать среду исполнения различных демонстрационных приложений, выполняемых в виде задач «FreeRTOS».

Основной функциональной задачей, решаемой демонстрационными приложениями в виде задач «FreeRTOS», является реализация модели организации управления потоками данных между IP-ядрами устройств, входящими в состав аппаратных платформ.

2.2 Организация проекта

Организацию исходного кода всего проекта «FreeRTOS_FPGA_Demo» можно представить в условных уровнях. Графическое представление такой организации показано на рисунке 2-1.

Уровни отражают расположения составных частей исходного кода проекта. Получаемые в процессе компиляции кода составных частей объектные файлы включаются в состав исполняемого двоичного образа проекта «FreeRTOS_FPGA_Demo» компоновщиком среды разработки Xilinx SDK.

Первый уровень является основой для компиляции любого исходного кода приложения пользователя на языке «Си» в среде разработки Xilinx SDK. Исходный код и подключаемые при компоновке исполняемого двоичного образа пользователя системные библиотеки этого уровня располагаются в каталогах среды разработки Xilinx SDK.

Второй уровень является принадлежностью пакета программной поддержки аппаратной платформы. Исходный код этого уровня на языках «Си» и «ASM» автоматически формируется из набора заранее подготовленных разработчиками ЗАО «Скан Инжиниринг Телеком» шаблонов и TCL скриптов. TCL скрипты предназначены для исполнения кода ядром среды разработки Xilinx SDK, на стадии формирования пакета программной поддержки аппаратной платформы, в рабочем пространстве пользователя. Для каждой ревизии аппаратной платформы, загружаемой в FPGA микросхему, необходимо формировать свой пакет программной поддержки. Место расположения шаблонов и TCL скриптов определяется настройкой репозитариев среды разработки Xilinx SDK.

Третий уровень представляет собой исходный код самого проекта «FreeRTOS_FPGA_Demo».

2.3 Область размещения и условия запуска

Описанная в документе [1] инфраструктура IP-ядер «Микропроцессорная система на MicroBlaze» содержит в себе два блока памяти, в которых можно размещать инструкции, исполняемые микропроцессорным ядром MicroBlaze.

Первый блок памяти BRAM подключен непосредственно к портам «IL» и «DL» микропроцессорного ядра MicroBlaze. В адресном пространстве микропроцессорного ядра MicroBlaze этот блок памяти занимает место с нулевого адреса (0x00000000), его объем 32 Кбайта для FPGA Virtex-6 или 128 Кбайт для Virtex-7.

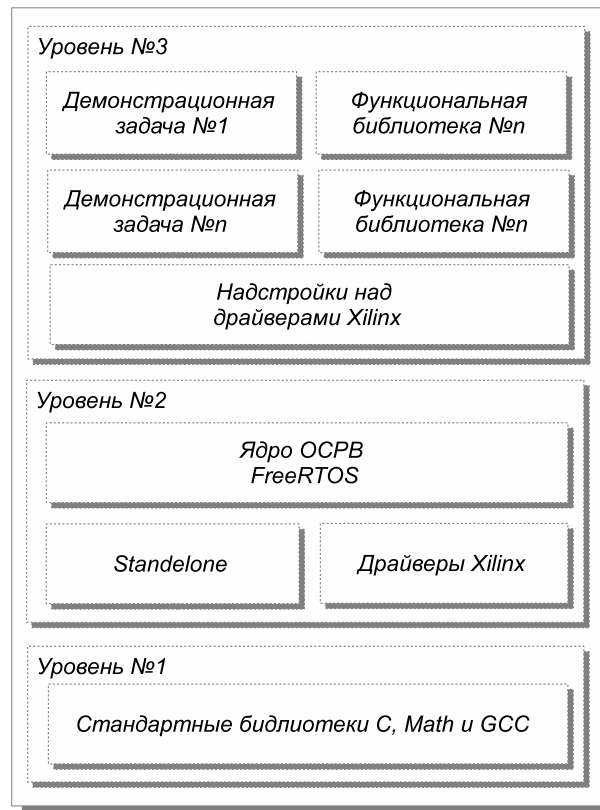


Рисунок 2-1: Уровни организации исходного кода проекта

Второй блок памяти SDRAM DDR подключен непосредственно к портам «IC» и «DC» микропроцессорного ядра MicroBlaze. В адресном пространстве микропроцессорного ядра MicroBlaze этот блок памяти занимает место с адреса 0xC0000000, его объём зависит от типа FPGA модуля и может составлять 128/256/512 Мбайт.

В процессе компиляции проекта «FreeRTOS_FPGA_Demo» средой Xilinx SDK формируется двоичный образ, в котором содержатся сегменты с исполняемыми инструкциями, данными и необходимыми, для определения точек запуска на микропроцессорном ядре MicroBlaze, значениями таблицы векторов прерываний.

Исполняемые инструкции и данные двоичного образа проекта «FreeRTOS_FPGA_Demo» размещаются во втором блоке памяти SDRAM DDR. Размещение начинается с адреса 0xC0000000.

Значения таблицы векторов прерываний двоичного образа проекта «FreeRTOS_FPGA_Demo» размещаются в первом блоке памяти BRAM. Размещение начинается с адреса 0x00000000.

Получаемый в результате компиляции двоичный образ проекта «FreeRTOS_FPGA_Demo» может сохраняться в «Platform Flash» или специальной SPI флеш-памяти для длительного хранения. Для осуществления загрузки и запуска двоичного образа на аппаратной платформе, после подачи питания на FPGA микросхему, используется загрузчик MBL. Более подробно с работой загрузчика MBL и его использованием вы можете ознакомиться в документе [2] «Загрузчик MBL».

2.4 Структура каталогов проекта

Проект «FreeRTOS_FPGA_Demo» размещается на файловой системе в каталоге с одноименным названием. В общем виде структура каталогов проекта приведена в листинге 2-1.

Листинг 2-1: Общий вид структуры каталогов проекта «FreeRTOS_FPGA_Demo»

```

\---FreeRTOS_FPGA_Demo
  +---projects
    |   +---HW_S6T_SP605
    |   +---HW_SAMC_713
    |   +---HW_SVP_713
    |   +---HW_SVP_726
    |   \---HW_V7T_VC709
    |
  \---src

```



```

+---App_Libs_Units
|   +---BaseIPC_Lib
|   +---CLI_Lib
|   \---SFBS_Test_Lib
+---App_Units
+---HW_Units
|   +---boards
|   \---periphery
\---OS_Units

```

В каталоге «FreeRTOS_FPGA_Demo» располагаются каталоги верхнего уровня «projects», «src» и «doc».

Каталог «doc» содержит документацию, поставляемую с проектом «FreeRTOS_FPGA_Demo».

Каталог «projects» содержит несколько подкаталогов, в которых размещаются файлы настройки проектов для среды разработки Xilinx SDK. Имена подкаталогов формируются по шаблону «HW_[имя модуля FPGA]», например, для АМС модуля SAMC-713 имя подкаталога будет «HW_SAMC_713». При работе со средой разработки Xilinx SDK пользователь импортирует из данных подкаталогов в свое рабочее пространство тот вариант проекта, который соответствует FPGA модулю, на котором он собирается запускать на исполнение двоичный образ, получаемый в процессе компиляции проекта «FreeRTOS_FPGA_Demo».

Каталог «src» содержит файлы с исходными текстами проекта, написанными на языке «Си». Файлы размещаются в различных подкаталогах. Каждый подкаталог имеет свое назначение. Обязательными подкаталогами для проекта являются «App_Units», «HW_Units» и «OS_Units». В этих подкаталогах содержатся файлы, в которых реализован основной код проекта. В подкаталоге «App_Libs_Units» размещается код, реализующий дополнительный функционал.

Содержимое различных подкаталогов в каталогах «projects» и «src» определяется конкретным типом FPGA модуля и используемым (или не используемым) совместно с ним FMC (FPGA Mezzanine Card) submodule. Содержимое подкаталогов формируется разработчиками ЗАО «Скан Инжиниринг Телеком» на стадии производства пакета BSP, отражающего специфику конкретной комбинации используемых FPGA модулей, FMC submodule, а также различных по функциональному составу ревизий аппаратных платформ для FPGA микросхем.

В подкаталоге «OS_Units» располагаются файлы, в которых реализуется следующий функционал:

- низкоуровневая инициализация устройств аппаратной платформы в виде IP-ядер, имеющих AXI (Advanced eXtensible Interface) интерфейс;
- инициализация программных таймеров и задач FreeRTOS из состава проекта;
- переопределение функции «printf» стандартной библиотеки языка «Си», с целью обеспечения возможности ее работы с портами ввода/вывода аппаратной платформы (IP-ядра «uartlite» или «uart_16550»). Функции поддерживают безопасный режим работы, в котором возможно одновременное обращение к функции из нескольких задач FreeRTOS.

В подкаталоге «HW_Units» располагаются файлы, реализующие функции для работы с низкоуровневыми драйверами IP-ядер устройств, имеющих AXI интерфейс, входящих в состав аппаратной платформы. Реализуемые функции в файлах подкаталога «HW_Units» позволяют работать с некоторыми функциями драйверов устройств в безопасном режиме для одновременной работы из нескольких задач FreeRTOS. В этот каталог пользователь может включать свои файлы с исходным кодом функций, позволяющих работать с низкоуровневыми драйверами устройств, которые пользователь разрабатывает самостоятельно и с которыми он хочет работать внутри проекта «FreeRTOS_FPGA_Demo».

В подкаталоге «App_Units» располагаются файлы с исходным кодом, реализующим программные таймеры и задачи FreeRTOS. Данный код реализует основные демонстрационные функции проекта «FreeRTOS_FPGA_Demo». В этот каталог пользователь может включать свои файлы с исходным кодом, реализующим необходимые пользователю дополнительные программные таймеры и задачи FreeRTOS.

В подкаталоге «App_Libs_Units» располагаются другие подкаталоги с файлами дополнительных библиотек, в которых реализуются различные функции. Обращение к данным функциям может осуществляться из следующих источников: из обработчиков прерываний программных таймеров, из основных функций задач FreeRTOS и из обработчиков прерываний различных устройств аппаратной платформы. Как правило, подкаталог «App_Libs_Units» является местом размещения файлов функциональных библиотек с исходным кодом, который пользователь может самостоятельно включить в демонстрационный проект «FreeRTOS_FPGA_Demo».

2.5 Этапы работы

Работу полученного в процессе компиляции исходного кода проекта «FreeRTOS_FPGA_Demo» двоичного образа с момента его запуска на исполнение микропроцессорным ядром MicroBlaze условно можно разделить на три этапа:

- инициализация аппаратной платформы;

- инициализация и запуск FreeRTOS;
- работа планировщика задач ядра FreeRTOS.

Первый этап, инициализации аппаратной платформы, включает в себя несколько шагов. Количество шагов зависит от следующих характеристик:

- конфигурации аппаратной платформы;
- типа FPGA модуля и его аппаратной ревизии;
- наличия субмодуля(-ей) и их аппаратной ревизии.

Исходный код, работающий на этапе инициализации аппаратной платформы находится в подкаталогах «HW_Units», «HW_Units/boards» и «HW_Units/periphery». Точкой начала процесса инициализации аппаратной платформы является функция «HwDevicesInit()», находящаяся в файле «HW_Platform.c».

Инициализация аппаратной платформы включает в себя следующие шаги:

- активацию кеша для работы микропроцессорного ядра MicroBlaze с памятью SDRAM DDR;
- настройку опорных задающих генераторов, установленных на FPGA модуль;
- настройку опорных задающих генераторов, установленных на FMC субмодулях;
- настройку параметров работы различных микросхем FMC субмодулей (например, ADC/DAC, делителей частот опорных задающих генераторов);
- настройку параметров работы интерфейсных контроллеров различных шин (PCIe, SRIO (Serial RapidIO), Ethernet, Aurora и т. д.);
- настройку параметров работы устройств в виде IP-ядер пользователя, включаемых в состав аппаратной платформы;
- регистрацию физических устройств в виде IP-ядер с AXI интерфейсом в качестве доступных для работы ресурсов;
- регистрация обработчиков прерываний от устройств в виде IP-ядер.

Второй этап, инициализации и запуска FreeRTOS, включает в себя следующие шаги:

- инициализация аппаратного таймера, являющегося основой для генерации системных тиков FreeRTOS;
- переключение потоков ввода/вывода системной консоли в безопасный режим работы (возможность одновременного вызова функции «printf» из нескольких задач FreeRTOS и ее корректная работа);
- инициализация программных таймеров;
- инициализация функциональных задач, заключенных в задачи FreeRTOS;
- запуск планировщика задач ядра FreeRTOS;

Исходный код, работающий на втором этапе, распределен по файлам, находящимся в подкаталогах «OS_Units» и «App_Units». Точкой начала процесса инициализации FreeRTOS является функция «os_init()», находящаяся в файле «Os_Platform.c». Точкой запуска процесса работы планировщика ядра FreeRTOS является функция «os_start()», так же находящаяся в файле «Os_Platform.c».

Третий этап, работа планировщика задач ядра FreeRTOS, является основным этапом работы проекта «FreeRTOS - FPGA_Demo». На этом этапе происходит исполнение всех демонстрационных задач, реализуемых в проекте. Планировщик задач ядра FreeRTOS приступает к выполнению всех задач, инициализация которых была проведена на втором этапе, выделяя каждой задаче для исполнения определенный интервал времени микропроцессорного ядра MicroBlaze, равный одному системному тикю.

Исходный код всех демонстрационных задач, работающий на третьем этапе, в основном распределен по файлам подкаталогов «App_Units» и «App_Libs_Units». При исполнении демонстрационных задач, вызов функций поддержки работы устройств в виде IP-ядер осуществляется из кода, распределенного по файлам подкаталога «HW_Units».

3 Компиляция проекта

Компиляция проекта осуществляется в среде разработки Xilinx SDK.

Изначально в концепцию реализации проекта «FreeRTOS_FPGA_Demo» заложена возможность его использования для работы с разными конфигурациями аппаратных платформ, предназначенными для различных FPGA модулей производства ЗАО «Скан Инжиниринг Телеком». Это позволяет использовать проект «FreeRTOS_FPGA_Demo» для работы с разными семействами FPGA микросхем Xilinx.

В зависимости от семейства FPGA микросхем, компания Xilinx предлагает разные версии среды разработки Xilinx SDK. Данное руководство охватывает работу с:

- Xilinx SDK Release Version: 14.6;
- Xilinx SDK Release Version: 2014.2.

3.1 Управление компиляцией

При осуществлении процесса компиляции проекта «FreeRTOS_FPGA_Demo» пользователю предоставляются следующие возможности по управлению компиляцией:

- выбор режима компиляции;
- настройка аппаратной платформы, для которой будет осуществлена компиляция.

3.1.1 Режимы компиляции

Компиляция проекта «FreeRTOS_FPGA_Demo» может осуществляться в двух режимах:

- Debug;
- Release.

Режим «Debug» используется на стадии разработки, когда требуется отладка исходного кода проекта «FreeRTOS_FPGA_Demo». Процесс отладки производится с использованием внешнего JTAG устройства. Получаемый в процессе компиляции в этом режиме двоичный образ проекта «FreeRTOS_FPGA_Demo» характеризуется:

- относительно медленным исполнением микропроцессорным ядром MicroBlaze, так как при компиляции проекта используются флаги компилятора уровня оптимизации `-O0` и включения в образ символов отладки `-g3`;
- большим объемом, в следствии включения в состав образа дополнительного объема выводимых в консоль текстовых сообщений, используемых с целью более подробного информирования о внутреннем состоянии протекающих процессов в исполняемом образе.

Режим «Release» используется для формирования окончательного рабочего облика двоичного образа проекта «FreeRTOS_FPGA_Demo». Получаемый в процессе компиляции в этом режиме двоичный образ характеризуется:

- максимально возможной скоростью исполнения микропроцессорным ядром MicroBlaze, так как при компиляции проекта используются флаги компилятора уровня оптимизации `-O3` и отсутствуют включения в образ символов отладки;
- минимальным объемом, в связи с отсутствием в составе образа дополнительного количества выводимых в консоль текстовых сообщений, используемых с целью более подробного информирования о внутреннем состоянии протекающих процессов в исполняемом образе.

3.1.2 Настройка аппаратной платформы

Для получения исполняемого двоичного образа проекта «FreeRTOS_FPGA_Demo», пригодного для работы с конкретной аппаратной платформой, применяются макросы оператора `##define` препроцессора языка «Си» и настройки символов компиляции проекта в свойствах проекта среды разработки Xilinx SDK.

Определение макросов оператора `##define` препроцессора языка «Си» распределены по разным заголовочным «h» файлам. Основными файлами являются файлы «App.h» и «Hw_Platform.h». Макросы определяют:

- набор поддерживаемых FPGA модулей и выбор конкретного модуля для которого будет компилироваться двоичный образ проекта;
- набор поддерживаемых FMC субмодулей, если они используются с FPGA модулем и выбор конкретного субмодуля, поддержка которого войдет в компилируемый двоичный образ проекта;
- выбор поддержки подключения к FPGA микросхеме, устанавливаемой на FPGA модулях и FMC субмодулях;

- включение поддержки различных устройств в виде IP-ядер, включенных в состав аппаратной платформы, например, контроллеры: CDMA, PCIe, SRIO, Ethernet и т. д.
- включение поддержки и настройка различных функциональных задач, реализуемых в задачах FreeRTOS;
- включение поддержки и настройка функциональных элементов, например, поддержка консоли ввода/вывода команд управления CLI или дополнительных технологий, например, SETFabric-Stream™ PE.

Выбор конкретного FPGA модуля, для которого компилируется двоичный образ проекта «FreeRTOS_FPGA_Demo», осуществляется через опцию компилятора `-D`, путем определения для символа «G_APP_HW_PLATFORM» числового значения в настройках свойства проекта среды разработки Xilinx SDK. Настройка значения «G_APP_HW_PLATFORM» показана на рисунке 3-1. Числовое значение для «G_APP_HW_PLATFORM» выбирается из predeterminedных значений для поддерживаемых модулей FPGA, определяемых в файле «App.h», как показано в листинге 3-1.

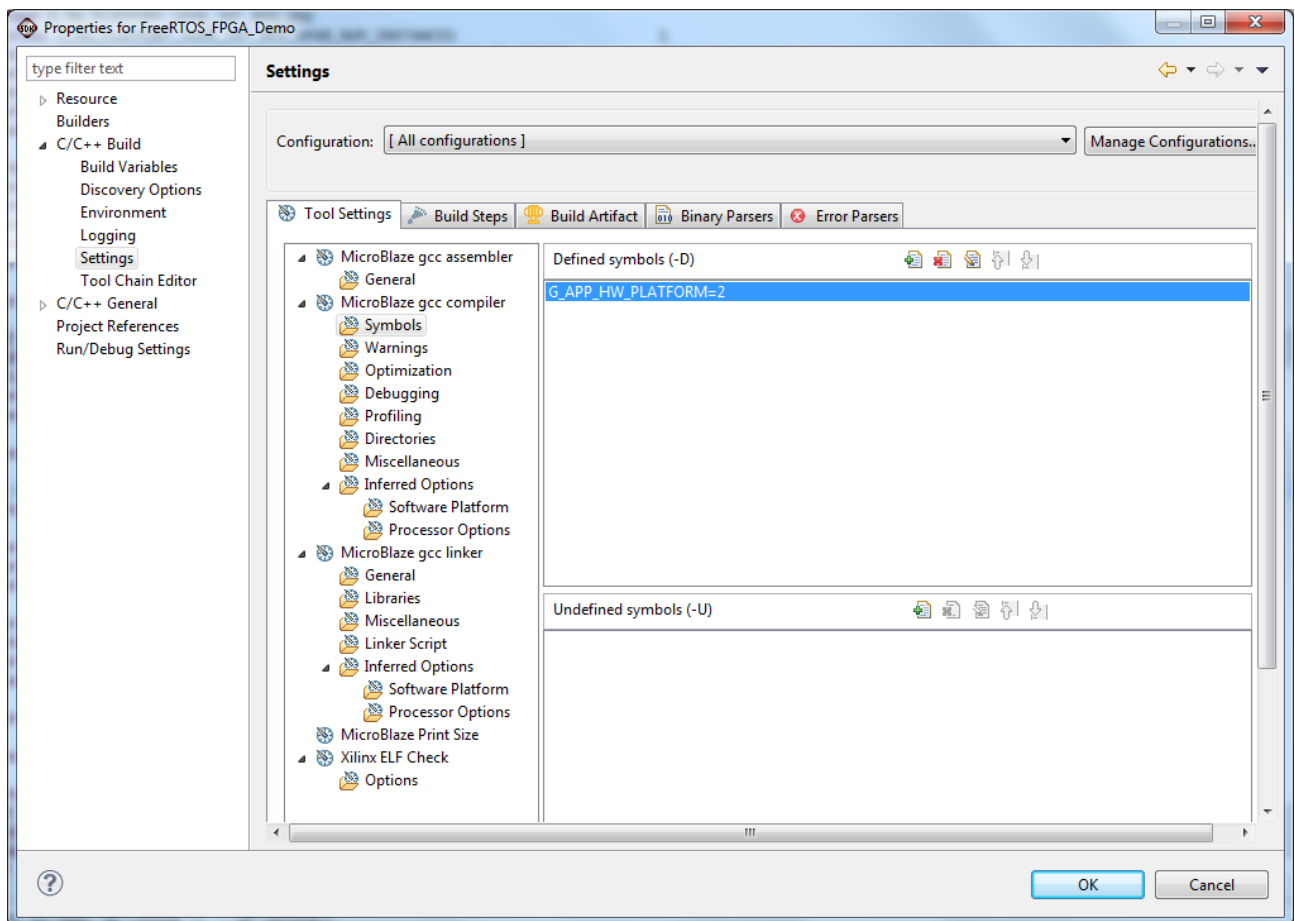


Рисунок 3-1: Уровни организации исходного кода проекта

Листинг 3-1: Возможные числовые значения символа G_APP_HW_PLATFORM

```

...

#define G_APP_HW_PLATFORM_S6T_SP605_REV_D      1
#define G_APP_HW_PLATFORM_SVP_713_REV_1       2
#define G_APP_HW_PLATFORM_SAMC_713_REV_1      3
#define G_APP_HW_PLATFORM_V7T_VC_709_REV_1    4
#define G_APP_HW_PLATFORM_SVP_726_REV_1      5

...

```

В файле «Hw_Platform.h» определяется набор поддерживаемых устройств в виде IP-ядер, входящих в состав аппаратной платформы и FMC субмодулей, используемых с FPGA модулем. Например, для FPGA модуля SVP-713 с установленным субмодулем SFM-4A4250 секция настройки может выглядеть так, как показано в листинге 3-2, секция настроек FMC субмодуля может выглядеть так, как показано в листинге 3-3.

Листинг 3-2: Включение поддержки в исполняемый бинарный образ различных IP-ядер, submodule, микросхем

```

...

#if (G_APP_HW_PLATFORM == G_APP_HW_PLATFORM_SVP_713_REV_1)

/* Enable support for submodules. */
#define G_HW_PLATFORM_FMC_SFM_4A250_USE

/* Enable support for IP cores. */
#define G_HW_PLATFORM_UARTLITE_USE
#define G_HW_PLATFORM_PCIE_USE
#define G_HW_PLATFORM_CDMA_USE
#define G_HW_PLATFORM_ADC_DATAMOVER_USE
#define G_HW_PLATFORM_USER_INT_REG_USE
#define G_HW_PLATFORM_GPIO_LEDS_USE
#define G_HW_PLATFORM_SYSTEM_MONITOR_USE
#define G_HW_PLATFORM_USER_FREQ_GENERATOR_SI570_USE

#endif

...

```

Листинг 3-3: Настройки FMC submodule SFM-4A250

```

...

/* Settings for SFM-4A250 module. */
#ifdef G_HW_PLATFORM_FMC_SFM_4A250_USE

#define G_HW_PLATFORM_FMC_SFM_4A250_MIN_SI570_FREQ 1000000 /* HZ */
#define G_HW_PLATFORM_FMC_SFM_4A250_MAX_SI570_FREQ 25000000 /* HZ */
#define G_HW_PLATFORM_FMC_SFM_4A250_STARTUP_FREQ_HZ 25000000 /* HZ */

#define G_HW_PLATFORM_FMC_SFM_4A250_LMK_USE
#ifdef G_HW_PLATFORM_FMC_SFM_4A250_LMK_USE
#define G_HW_PLATFORM_FMC_SFM_4A250_LMK_STARTUP_DIVIDER 1
#endif

...

#endif

...

```

С целью осуществления возможности компиляции проекта «FreeRTOS_FPGA_Demo» для разных аппаратных платформ в файле «Hw_Platform.h» осуществляется привязка символов из файла «xparameters.h», генерируемого автоматически в процессе формирования аппаратной платформы, к абстрактному слою символов, используемому в части кода проекта, отвечающего за реализацию функций работы с низкоуровневыми драйверами устройств в виде IP-ядер. Использование такой абстрактной привязки позволяет за счет макросов, расположенных в файле «Hw_MiscUtils.h», включить автоматически, на стадии работы препроцессора компилятора, поддержку всех однотипных устройств, определяемых в файле «xparameters.h», в компилируемый двоичный образ проекта «FreeRTOS_FPGA_Demo». Например, для включения поддержки в двоичный образ всех устройств типа «uart_lite» в файле «Hw_Platform.h» существует секция, показанная в листинге 3-4. Остальные устройства тоже имеют свои секции для подключения.

Выбор поддержки и настройки задач FreeRTOS и функциональных элементов дополнительных технологий, включаемых в состав компилируемого двоичного образа, определяется в файле «App.h». Например, для подключения CLI используется определение символа «G_APP_CONSOLE_USE», как показано в листинге 3-5. В случае использования дополнительных технологий, например, SETFabric-Stream™ PE, ее подключение будет таким, как показано в листинге 3-6.

Листинг 3-4: Включения поддержки в бинарный образ всех устройств типа `uart_lite`

```
...

/* Enable support for devices such as milestones "uartlite" */
#ifdef G_HW_PLATFORM_UARTLITE_USE

#define G_HW_PLATFORM_UARTLITE_XPAR_NUM_INSTANCES           XPAR_XUARTLITE_NUM_INSTANCES
#define G_HW_PLATFORM_UARTLITE_XPAR_INTERRUPT_VECTOR_ID_PREFIX XPAR_INTC_0_UARTLITE_
#define G_HW_PLATFORM_UARTLITE_XPAR_INTERRUPT_VECTOR_ID_POSTFIX _VEC_ID
#define G_HW_PLATFORM_UARTLITE_XPAR_DEVICE_ID_PREFIX        XPAR_UARTLITE_
#define G_HW_PLATFORM_UARTLITE_XPAR_DEVICE_ID_POSTFIX       _DEVICE_ID

#endif

...
```

Листинг 3-5: Включения поддержки CLI

```
...

/* Defined use the Console task in Debug mode. */
#ifdef DEBUG
#define G_APP_CONSOLE_USE
#endif

...
```

Листинг 3-6: Включение поддержки технологии SETFabric-Stream™

```
...

/* Defined use the SETFabric-Stream™ */
#define G_APP_SET_FABRIC_STREAM_USE

#ifdef G_APP_SET_FABRIC_STREAM_USE

#define G_APP_SFBS_SERV_USE

...

#endif

...
```

3.1.3 Настройка области размещения

Получаемый в процессе компиляции проекта «FreeRTOS_FPGA_Demo» исполняемый двоичный образ в формате ELF делится на сегменты, которые перед запуском на исполнение размещаются в разных типах памяти, BRAM или SDRAM DDR, разрешённые для хранения исполняемых инструкций микропроцессорного ядра MicroBlaze. В общем случае, сегменты двоичного образа выглядят так, как показано на рисунке 2-1.

Управление процессом размещения сегментов двоичного образа проекта «FreeRTOS_FPGA_Demo» осуществляется путем настроек в файле «lscript.ld», включаемого в состав проекта.

Изменяя содержимое файла «lscript.ld» пользователь может изменить размер «Стека» и «Кучи». Конкретные значения этих параметров зависят от объёма исходного кода проекта «FreeRTOS_FPGA_Demo» и устанавливаются разработчиками ЗАО «Скан Инжиниринг Телеком» при формировании конкретного пакета BSP, в состав которого входит проект «FreeRTOS_FPGA_Demo».



Рисунок 3-2: Сегменты кода откомпилированного образа

3.2 Подготовка к компиляции

3.2.1 Выбор рабочего пространства

Процедура 3-1. Выбор места расположения рабочего пространства в Xilinx SDK

1. Запустите среду разработки Xilinx SDK и выберите место расположения рабочего пространства «Workspace». По умолчанию место расположения рабочего пространства «Workspace» находится в папке: «C:\set_spb\dev\[имя модуля]\[имя сборки аппаратной платформы]\SDK\SDK_Export».

Например, для модуля «SAMC-713», на котором запускается аппаратная платформа с именем сборки «Core 0», путь к «Workspace» выглядит так, как показано на рисунке 3-3.

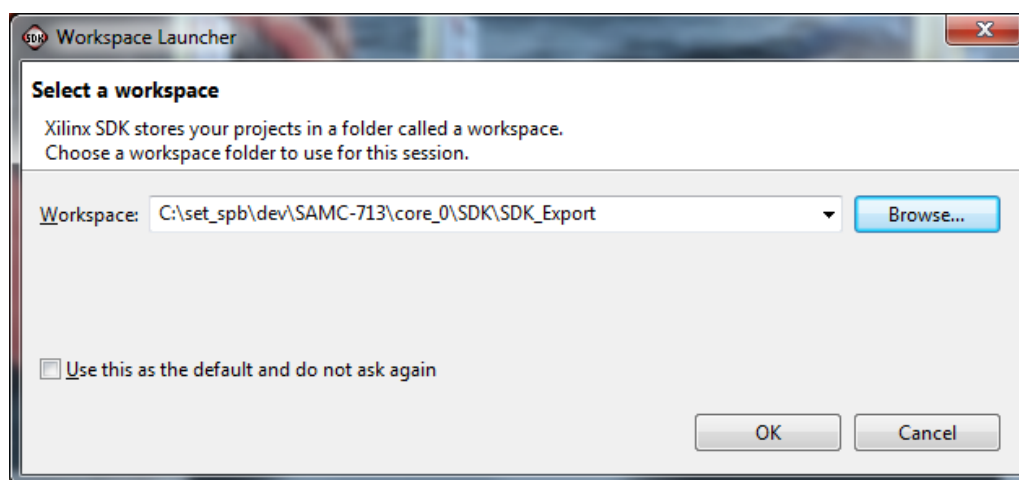


Рисунок 3-3: Выбор места расположения рабочего пространства в Xilinx SDK

3.2.2 Настройка репозитариев

Процедура 3-2. Выбор места расположения репозитариев

1. Выберите пункт меню «Xilinx Tools > Repositories», как показано на рисунке 3-4.

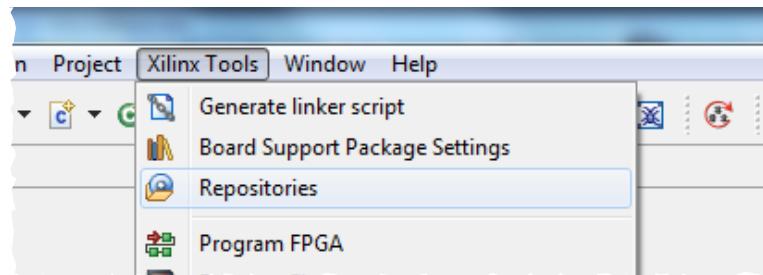


Рисунок 3-4: Выбор пункта меню настройки репозитариев в Xilinx SDK

2. В открывшемся окне, нажав кнопку «New», выберите место расположение репозитариев:

- «C:\set_spb\dev\[имя модуля][имя сборки аппаратной платформы]»;
- «C:\set_spb\dev\[имя модуля]\free_rtos\KernelAwareBSPRepository».

Например, для модуля «SAMC-713», на котором запускается аппаратная платформа с именем сборки «Core 0», пути к репозитариям выглядят так, как показано на рисунке 3-5.

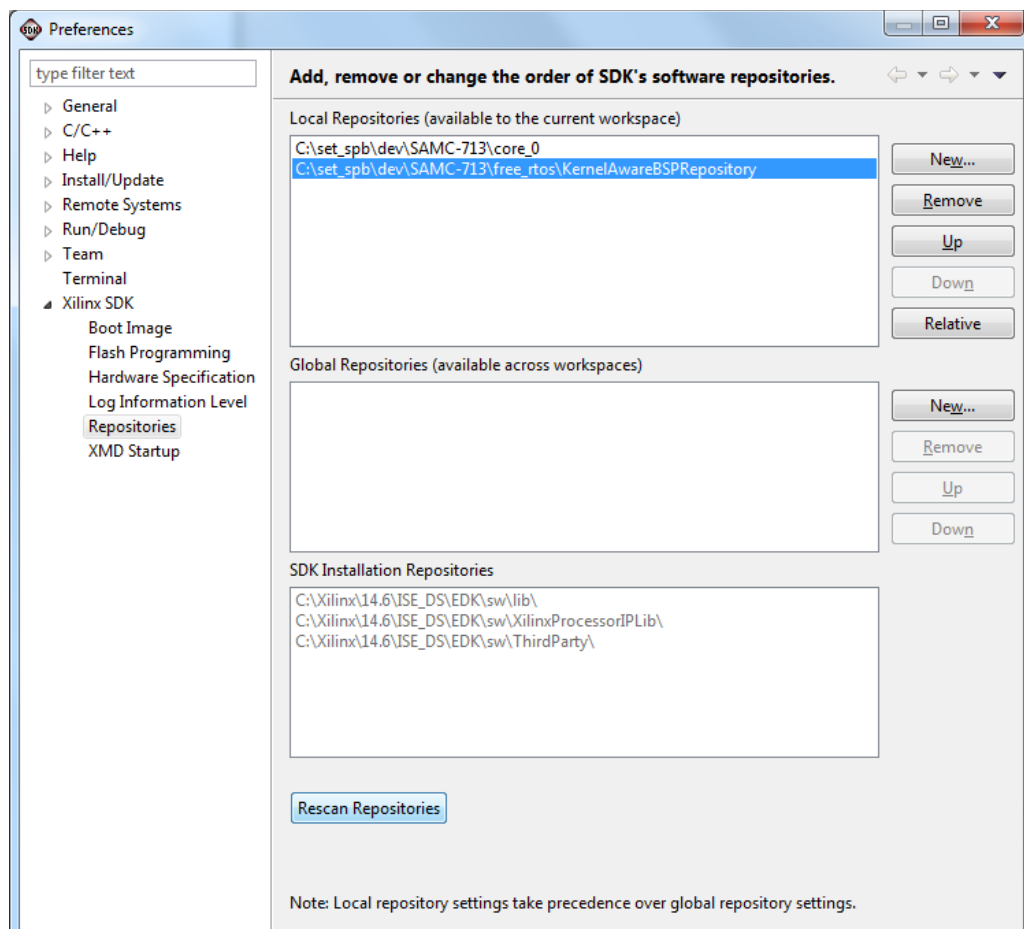


Рисунок 3-5: Настройка местоположения репозитариев в Xilinx SDK

3. Нажмите кнопку «Rescan Repositories», чтобы обновить информацию о репозитариях и выйдите из открытого окна, нажав кнопку «ОК».

3.2.3 Создание проекта аппаратной платформы

Процедура 3-3. Создание нового проекта для аппаратной платформы в Xilinx SDK

1. Выберите пункт меню «File > New > Project».
2. В открывшемся окне выберите в списке проектов тип «Hardware Platform Specification», как показано на рисунке 3-6 и нажмите кнопку «Next».

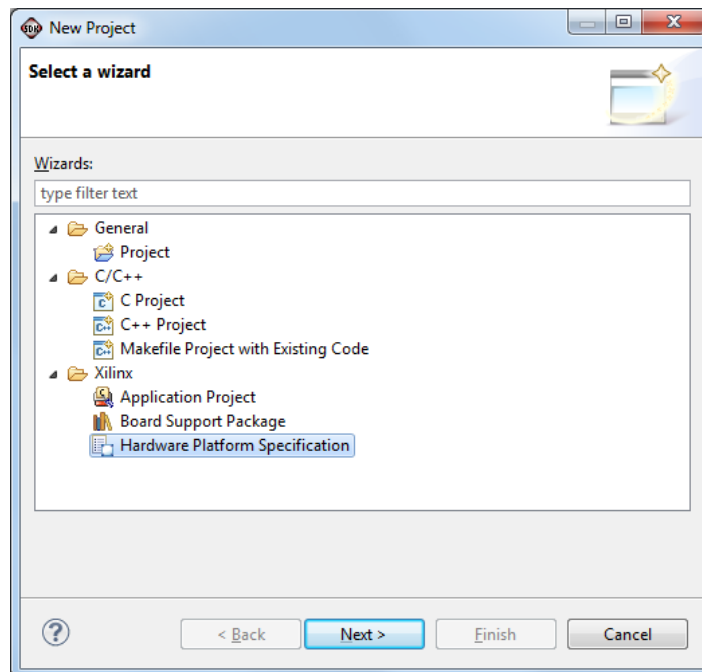


Рисунок 3-6: Создание проекта аппаратной платформы в Xilinx SDK

3. В открывшемся окне, как показано на рисунке 3-7, выберите место расположения файла спецификации аппаратной платформы, нажав кнопку «Browse». Файл располагается в следующей папке: «C:\set_spb\dev\[имя модуля]\[имя сборки аппаратной платформы]\SDK\SDK_Export\hw».

Для среды разработки Xilinx SDK Release Version: 14.6 имя файла спецификации аппаратной платформы носит название «system.xml».

Для среды разработки Xilinx SDK Release Version: 2014.2 имя файла спецификации аппаратной платформы носит название «system.hdf»

Например, для модуля «SAMC-713», на котором запускается аппаратная платформа с именем сборки «Core 0», путь к файлу спецификации аппаратной платформы выглядит следующим образом: «C:\set_spb\dev\SAMC-713\core_0\SDK\SDK_Export\hw\system.xml».

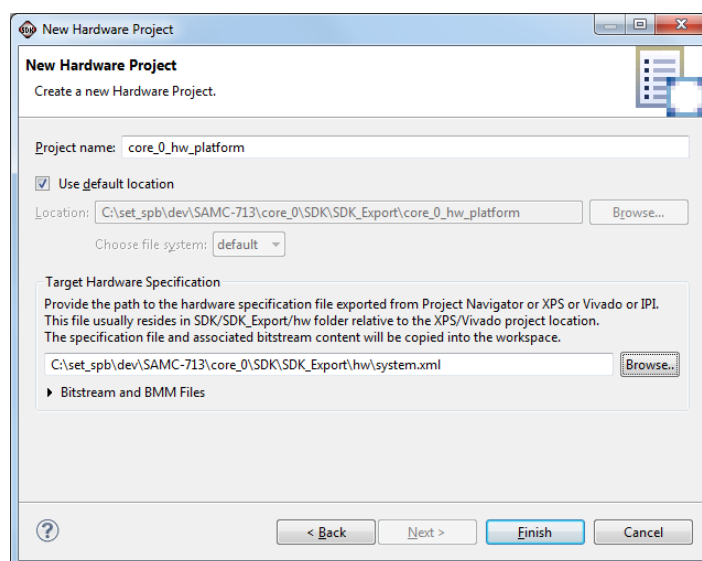


Рисунок 3-7: Настройки проекта аппаратной платформы в Xilinx SDK

4. Закройте окно, нажав кнопку «Finish».

3.2.4 Создание проекта поддержки аппаратной платформы BSP

Процедура 3-4. Создание нового проекта BSP поддержки аппаратной платформы на базе FreeRTOS» в Xilinx SDK

1. Выберите пункт меню «File > New > Project».
2. В открывшемся окне выберите в списке проектов тип «Board Support Package», как показано на рисунке 3-8 и нажмите кнопку «Next».

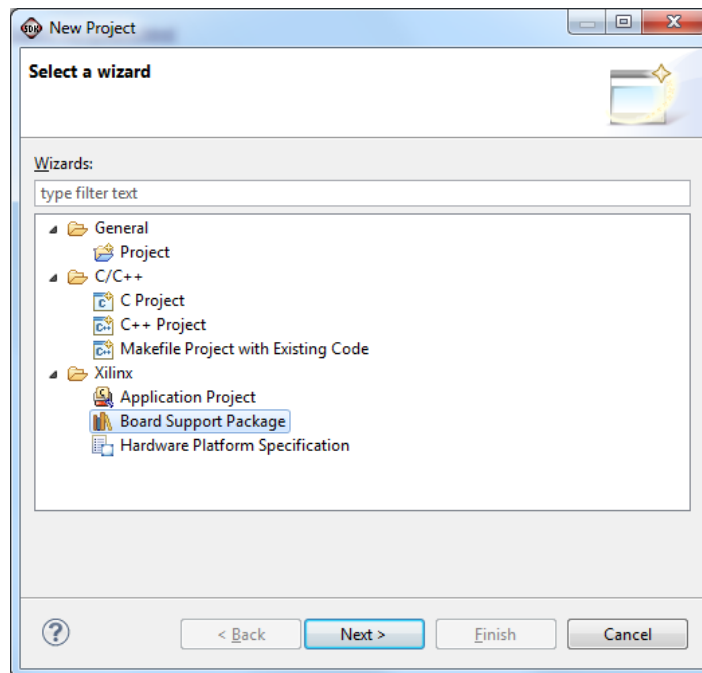


Рисунок 3-8: Создание проекта BSP ОС FreeRTOS в Xilinx SDK

3. В открывшемся окне, как показано на рисунке 3-9, выберите тип проекта «freertos» и нажмите кнопку «Finish».

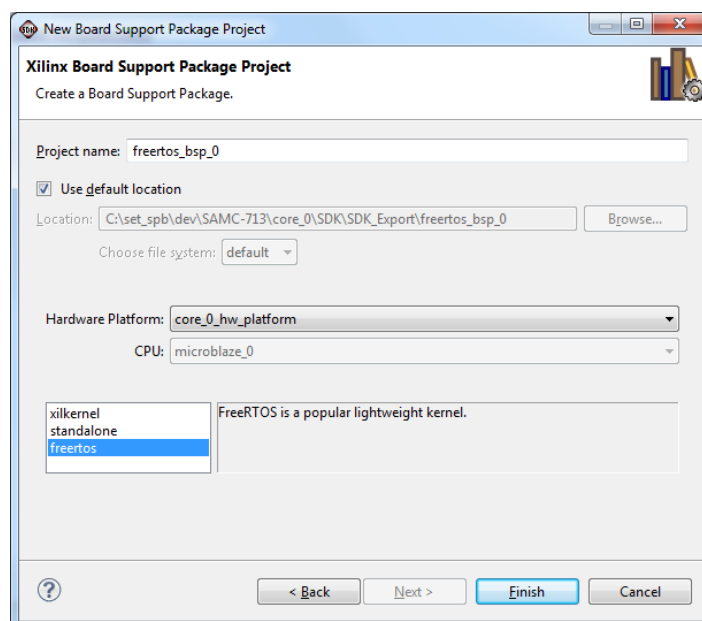


Рисунок 3-9: Настройки проекта BSP ОС FreeRTOS в Xilinx SDK

4. В открывшемся окне (рисунок 3-10) измените значения полей и нажмите кнопку «OK»:

- значение поля «max_priorities» на 11;
- значение поля «minimal_stack_size» на 4096;
- значение поля «queue_registry_size» на 50.

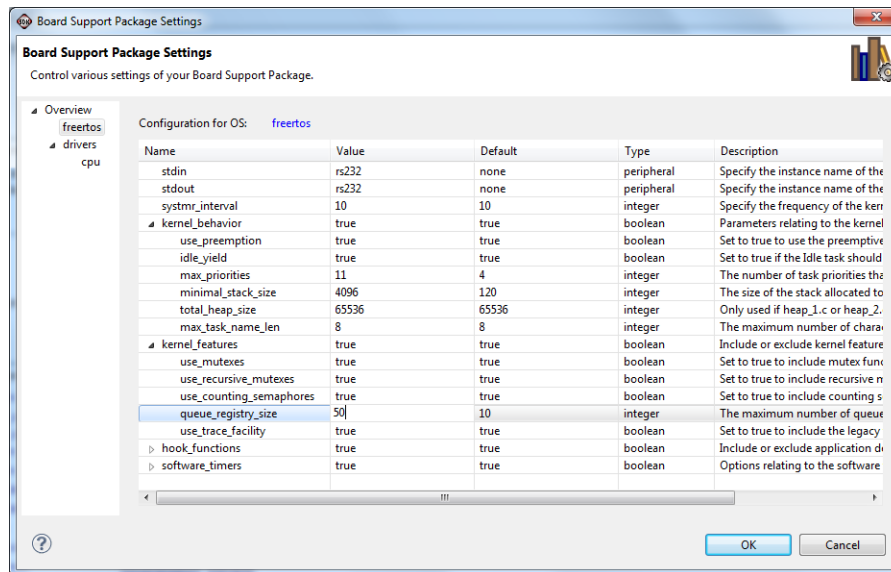


Рисунок 3-10: Параметры проекта BSP ОС FreeRTOS в Xilinx SDK

3.2.5 Импорт проекта

Процедура 3-5. Импорт проекта «FreeRTOS_FPGA_Demo» в рабочее пространство

1. Выберите пункт меню «File > Import».
2. В открывшемся окне (рисунок 3-11) выберите в списке проектов тип «Existing Projects into Workspace» и нажмите кнопку «Next».

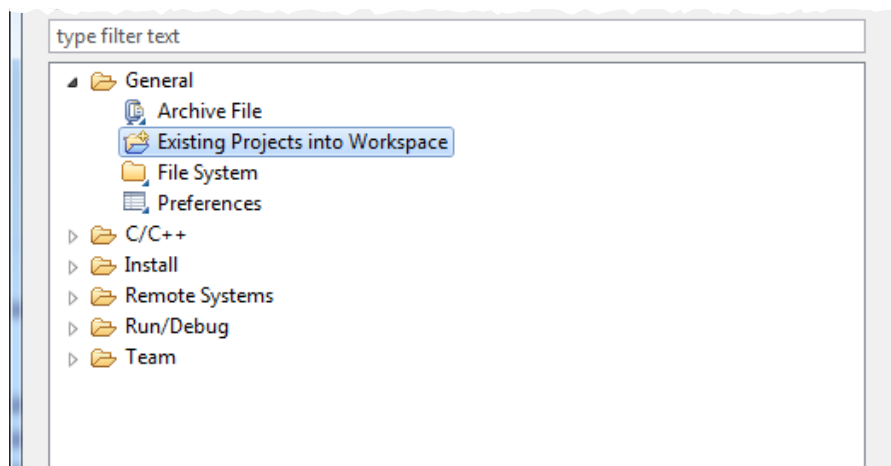


Рисунок 3-11: Импорт проекта в Xilinx SDK

3. В открывшемся окне (рисунок 3-12) выберите место расположения импортируемого проекта «C:\set_spb\dev\[имя_модуля]\FreeRTOS_FPGA_Demo», и нажмите кнопку «Finish».

Например, для модуля «SAMC-713», на котором запускается аппаратная платформа с именем сборки «Core 0», путь к файлу спецификации аппаратной платформы выглядит следующим образом: «C:\set_spb\dev\SAMC-713\free_rtos\FreeRTOS_FPGA_Demo\projects».

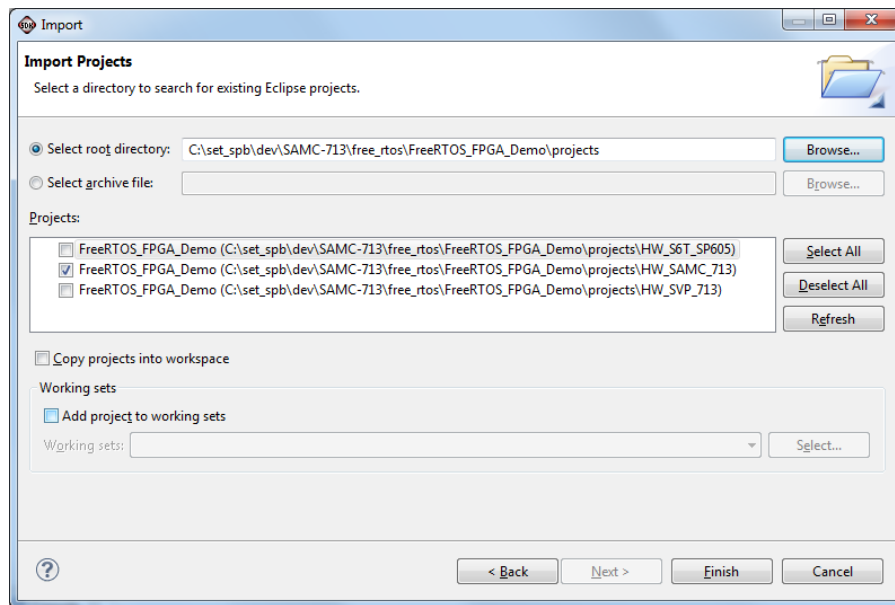


Рисунок 3-12: Выбор проекта для импорта в Xilinx SDK

3.3 Компиляция

Процедура 3-6. Компиляция рабочей версии

1. Выберите в окне «Project Explorer» проект «FreeRTOS_FPGA_Demo» (см. рисунок 3-13), режим компиляции «Release».

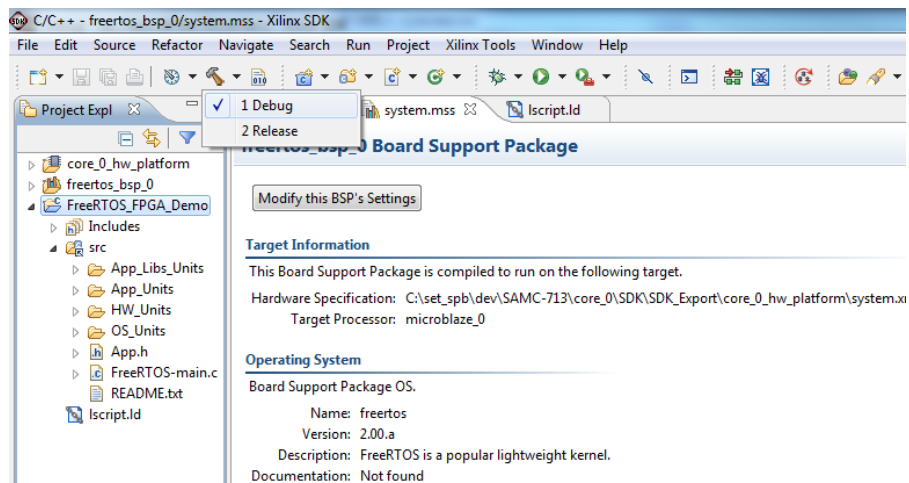


Рисунок 3-13: Сборка проекта в Xilinx SDK

2. Проверьте содержимое файла «Iscript.Id» на предмет правильного размещения сегментов двоичного образа проекта «FreeRTOS_FPGA_Demo» (рисунок 3-14).

Например, для модуля «SAMC-713», на котором запускается аппаратная платформа с именем сборки «Core 0», для загрузчика MBL в файле «Iscript.Id» должно быть:

- в поле «Stack Size» значение «0x100000»;
- в поле «Heap Size» значение «0x100000»;
- в полях столбца «Memory Region» таблицы «Section to Memory Region Mapping» должно быть значение «ddr3_sdram4_S_AXI_BASEADDR».

4 Запуск проекта

Запуск на исполнение полученного в процессе компиляции двоичного образа проекта «FreeRTOS_FPGA_Demo» осуществляется на аппаратной платформе, предварительно загруженной в FPGA микросхему.

Загрузка аппаратной платформы и двоичного образа проекта может быть осуществлена двумя следующими путями:

- непосредственной загрузки в FPGA микросхему аппаратной платформы и исполняемого двоичного образа из среды разработки Xilinx SDK;
- предварительной записи аппаратной платформы и исполняемого двоичного образа в флеш-память, подключенной к FPGA микросхеме.

Первый путь, непосредственной загрузки в FPGA микросхему, предполагает, что к FPGA модулю подключен JTAG отладчик фирмы Xilinx, работающий со средой разработки Xilinx SDK. Загрузка осуществляется непосредственно из этой среды разработки. Загружаемые в FPGA аппаратная платформа и исполняемый двоичный образ способны работать до момента времени, пока подается питание на FPGA модуль. После отключения питания и повторной его подачи, загрузку аппаратной платформы и исполняемого бинарного образа нужно будет выполнить повторно. Данный вариант загрузки и работы, как правило, пригоден для этапов разработки и функционального тестирования.

Второй путь, предварительной записи во флеш-память, подключенной к FPGA микросхеме, используется для подготовки FPGA модуля к постоянной эксплуатации. Эксплуатация FPGA модуля предполагает периодическую подачу и отключение питания модулю. При использовании данного пути, непосредственно сам запуск двоичного образа проекта «FreeRTOS_FPGA_Demo» осуществляется автоматически, загрузчиком MBL, после подачи питания FPGA микросхеме, когда в нее загружается аппаратная платформа и загрузчик MBL из «Platform Flash». Прибегать к использованию данного пути загрузки и запуска на этапе разработки нужно как можно реже, так как ресурс работоспособности флеш-памяти ограничен циклами перезаписи.

Перед началом запуска полученного в процессе компиляции двоичного образа проекта «FreeRTOS_FPGA_Demo» подключите устройство JTAG отладчика фирмы Xilinx к FPGA модулю и персональному компьютеру, на котором установлена среда Xilinx SDK.

Также необходимо подключить к последовательному порту ввода/вывода (консольному порту) FPGA модуля и персональному компьютеру USB-кабель, и запустить на персональном компьютере терминальную программу, чтобы видеть в окне терминальной программы текстовые сообщения, формируемые работающим двоичным образом проекта «FreeRTOS_FPGA_Demo». Настройки последовательного порта терминальной программы: 115200-8-N1.

4.1 Через JTAG интерфейс

Процедура 4-1. Запуск двоичного образа проекта «FreeRTOS_FPGA_Demo» через JTAG интерфейс

1. После окончания процесса компиляции двоичного образа проекта «FreeRTOS_FPGA_Demo» в среде Xilinx SDK выберите пункт меню «Xilinx Tools > Program FPGA», как показано на рисунке 4-1.

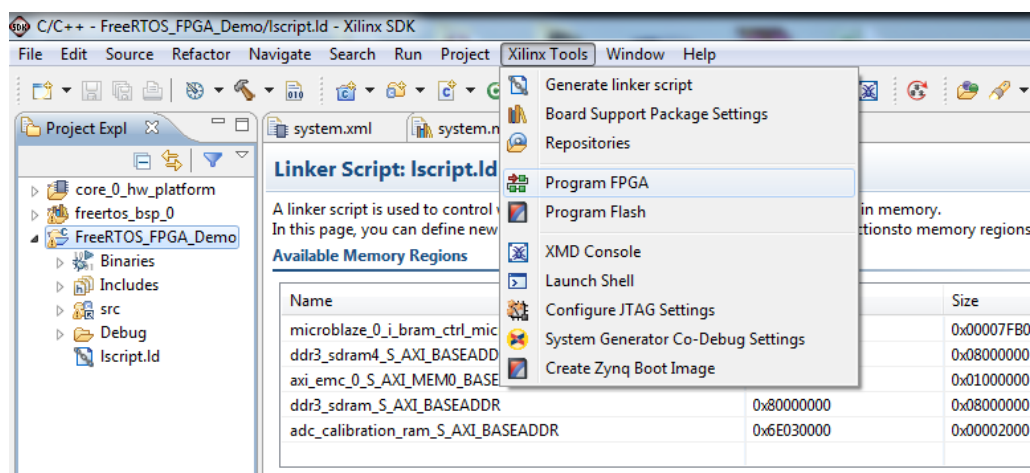


Рисунок 4-1: Выбор режима программирования FPGA через JTAG интерфейс в Xilinx SDK

- В открывшемся окне (рисунок 4-2) выберите в таблице «Software Configuration» загружаемый в FPGA микросхему двоичный образ «bootloop» и запустите процесс загрузки в FPGA микросхему аппаратной платформы и двоичного образа «bootloop», нажав кнопку «Program».

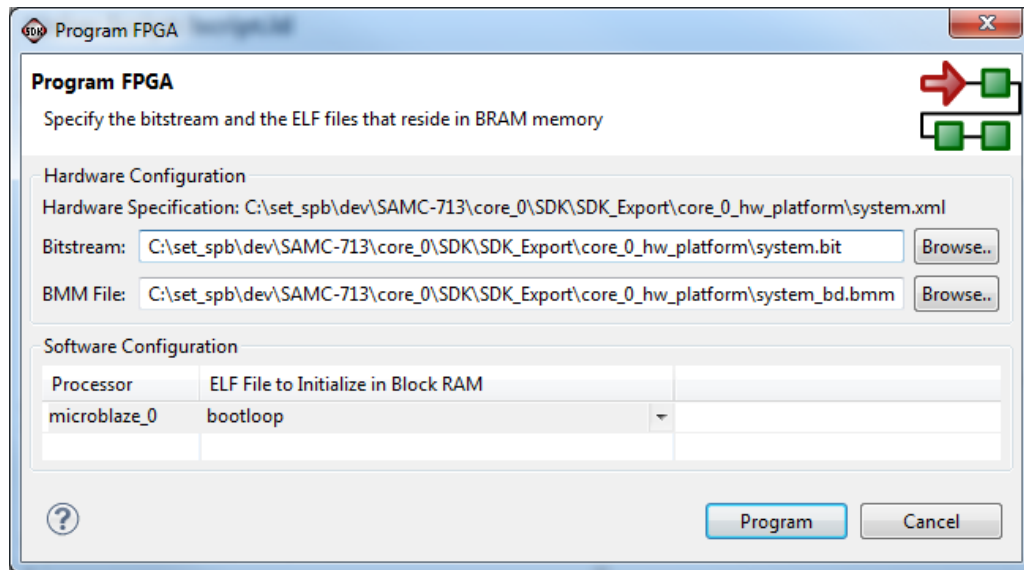


Рисунок 4-2: Выбор загружаемого в FPGA микросхему двоичного образа «bootloop» в Xilinx SDK

- После окончания процесса программирования выбранной FPGA микросхемы создайте цель запуска в режиме отладки двоичного образа проекта «FreeRTOS_FPGA_Demo». Для этого выберите пункт меню «Run > Debug As > Launch on Hardware (GDB)», как показано на рисунке 4-3.

После выбора пункта меню «Launch on Hardware (GDB)» произойдет автоматическое формирование цели запуска и осуществится загрузка двоичного образа проекта «FreeRTOS_FPGA_Demo» в SDRAM DDR модуля FPGA.

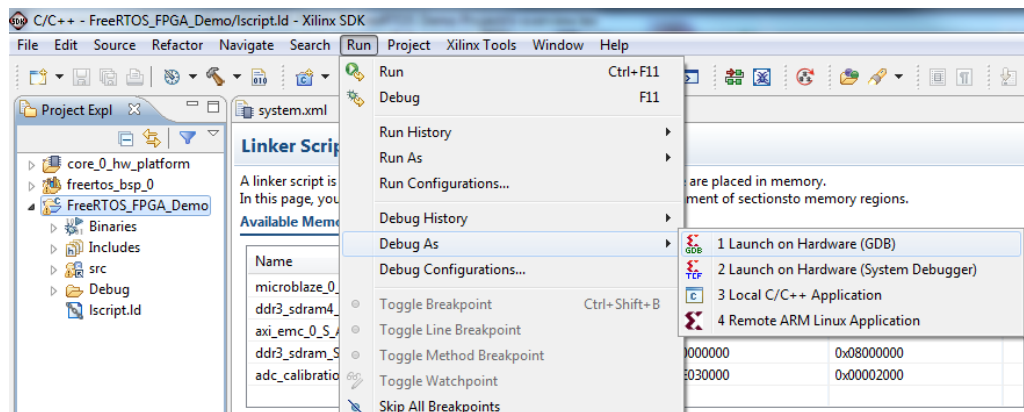


Рисунок 4-3: Создание цели запуска в режиме отладки проекта «FreeRTOS_FPGA_Demo» через JTAG интерфейс в Xilinx SDK

- Успешным входом в процесс отладки двоичного образа проекта «FreeRTOS_FPGA_Demo» будет содержание главного окна среды разработки Xilinx SDK, которое соответствует изображению рисунка 4-4.

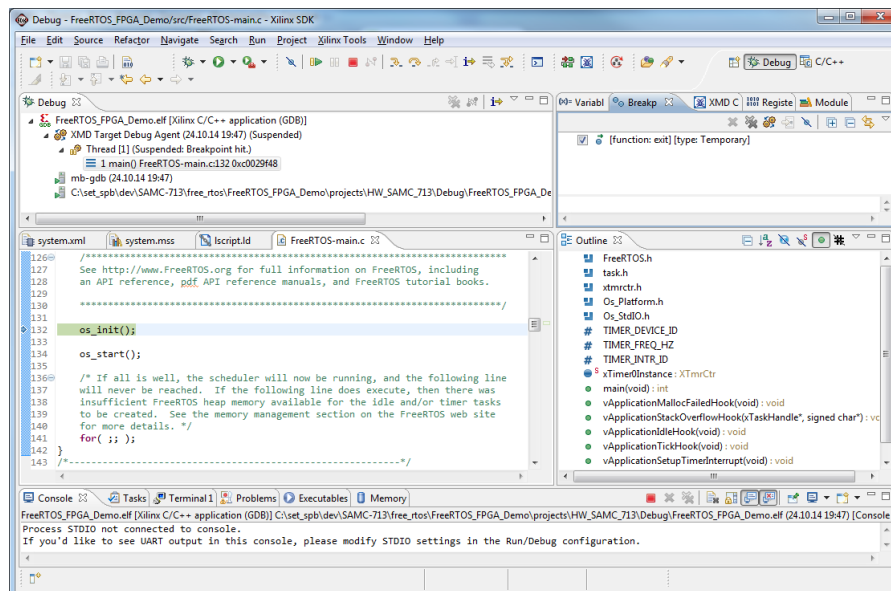


Рисунок 4-4: Окно отладки двоичного образа проекта «FreeRTOS_FPGA_Demo» в Xilinx SDK

- Запустите работу двоичного образа проекта «FreeRTOS_FPGA_Demo», нажав кнопку «Resume», показанную на рисунке 4-5.

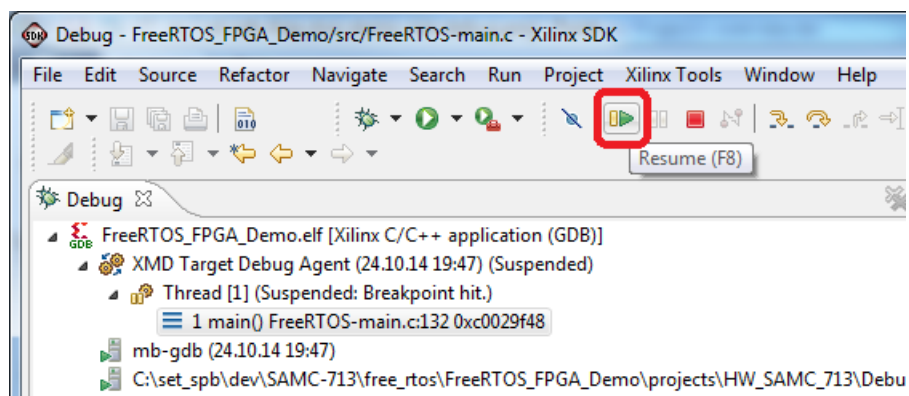


Рисунок 4-5: Запуск работы двоичного образа проекта «FreeRTOS_FPGA_Demo» в Xilinx SDK

Об успешной работе запущенного двоичного образа проекта «FreeRTOS_FPGA_Demo» будет свидетельствовать наличие выводимых им текстовых сообщений в запущенную на персональном компьютере терминальную программу. Содержимое окна терминальной программы приведено на рисунке 4-6.


```

COM8 - PuTTY
.....
channel 2 calibration complete

delay windows:
||0||0|0|0|0|0|0|0|0|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|
||1||0|0|0|0|0|0|0|0|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|
||2||0|0|0|0|0|0|0|0|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|
||3||0|0|0|0|0|0|0|0|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|
||4||0|0|0|0|0|0|0|0|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|
||5||0|0|0|0|0|0|0|0|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|
||6||0|0|0|0|0|0|0|0|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|1|

Calibration completed

***** INIT STEP NUM 5 FINISHED *****

Finished SAMC-713 board initialization.

FreeRTOS V7.5.2 is started.

```

Рисунок 4-6: Терминальный вывод сообщение о работе двоичного образа проекта «FreeRTOS_FPGA_Demo»

4.2 Из флеш-памяти

Запуск полученного в процессе компиляции проекта «FreeRTOS_FPGA_Demo» двоичного образа из флеш-памяти на исполнение осуществляется загрузчиком MBL.

Для осуществления записи двоичного образа во флеш-память, необходимо выполнить процедуры по конвертации этого образа из файла формата ELF в файл формата SREC и выполнить запись полученного SREC файла во флеш-память. Более подробную информацию о том как правильно выполнить эти процедуры можно узнать в документе [2]. Этот документ описывает работу с загрузчиком MBL.

Применительно к описанию работы загрузчика MBL, проект «FreeRTOS_FPGA_Demo» носит характер демонстрационного приложения пользователя, которое упоминается и описывается в документе [2]. Все приведенные в документе [2] действия, связанные с подготовкой и записью демонстрационного приложения пользователя во флеш-память, необходимо применить по отношению к проекту «FreeRTOS_FPGA_Demo».

Список литературы

1. Инфраструктура IP-ядер «Микропроцессорная система на MicroBlaze». Руководство пользователя. [UG-IP-IS-MBS](#) (цит. на с. 6, 7).
2. Загрузчик MBL. Руководство пользователя. [UG-FPGA-00-MBL](#) (цит. на с. 8, 25).