

ПРОГРАММНО-АППАРАТНАЯ ПЛАТФОРМА RENESAS SYNERGY. ОСНОВЫ

Richard Oed



RENESAS synergy™
Accelerate. Innovate. Differentiate.

2017.07

BIG IDEAS
FOR EVERY SPACE

ОСНОВЫ ПЛАТФОРМЫ RENESAS SYNERGY™

Ричард Уд

РУКОВОДСТВО ПО ИСПОЛЬЗОВАНИЮ КНИГИ «ОСНОВЫ ПЛАТФОРМЫ RENESAS SYNERGY™»

Книга об основах платформы Synergy предназначена для использования как в качестве вводного руководства для инженеров, делающих первые шаги с Synergy, так и в качестве удобного справочника для тех, кто уже знаком с платформой. Информировем, что эта версия книги ссылается на программный пакет Synergy (SynergySoftware Package (SSP)) версии 1.2.

Поскольку платформа постоянно развивается, становится неизбежным тот факт, что к тому времени, когда книга будет печататься, появятся разделы, которые больше не актуальны. Чтобы книга оставалась полезной для вышеуказанных целей, существует поддерживаемая онлайн-экосистема, которую вы можете найти на сайте www.renesas.com/synergy-book. В книге мы выделяем три разных типа информации: информация, которая не будет изменяться (например, вводные главы), информация, которая будет поддерживаться (например, примеры программного обеспечения), и информация, которая публикуется только в Интернете (чтобы избежать повторений и неоправданного использования места в книге).

Copyright: © 2017 Renesas Electronics Europe GmbH

Отказ от ответственности:

Это издание предоставляется исключительно в информативных целях без каких-либо гарантий на правильность и полноту.

Содержание не предназначено для того, чтобы полагаться на него как на справочное руководство по проектированию и за любые последствия, вытекающие из использования этой книги издательство ответственности не несёт.

CONTENTS

	FOREWORD	07
1	INTRODUCTION TO THE RENESAS SYNERGY™ PLATFORM	09
1.1	Challenges in Today's Embedded System Design	11
1.2	Synergy Software Package	11
1.3	The Synergy Microcontroller Family	12
1.4	The Synergy Tools and Kits	14
1.5	The Synergy Solutions	15
1.6	The Synergy Gallery	15
2	DETAILS OF THE RENESAS SYNERGY™ SOFTWARE	17
2.1	Introduction to the Synergy Software Package (SSP)	18
2.2	Introduction to the Board Support Package (BSP)	19
2.3	Introduction to the HAL Drivers	20
2.4	Introduction to the Application Frameworks	22
2.5	Introduction to the Functional Libraries	23
2.6	Included Middleware from Express Logic (X-Ware™)	24
2.6.1	FileX®	24
2.6.2	GUIX™	25
2.6.3	USBX™	26
2.6.4	NetX™/NetX Duo™	27
2.7	The RTOS of Choice: ThreadX®	27
2.7.1	Why use an RTOS?	28
2.7.2	The Main Features of ThreadX®	29
3	AN INTRODUCTION TO THE APIs OF THE SYNERGY™ SOFTWARE	30
3.1	API Overview	30
3.2	API Syntax	33
3.3	API Constants, Variables and Other Topics	34
3.4	API Usages	35
4	GETTING THE RENESAS SYNERGY™ PLATFORM TOOLCHAIN UP AND RUNNING	38
4.1	Introduction to the Synergy Gallery	38
4.2	Downloading and Installing e² studio and the Synergy Software Package	39
4.2.1	Download of e² studio and the SSP	39
4.2.2	Installing the Tool	40
4.2.3	Starting for the First Time	43
4.2.4	Keeping Your Installation Up-To-Date	44
4.3	Installing the IAR Embedded Workbench® for Renesas Synergy™	45
5	WORKING WITH THE DEVELOPMENT ENVIRONMENTS FOR RENESAS SYNERGY™ PLATFORM	46
5.1	The Eclipse™ Based e² studio	46
5.1.1	Short Introduction to the Philosophy of Eclipse	46
5.1.2	Configurators: A Short Introduction	49
5.1.3	Importing and Exporting Synergy Projects	51
5.1.4	Team Collaboration and File Handling	53
5.2	IAR Embedded Workbench®	53

6	RENESAS SYNERGY™ KITS	54
6.1	The Different Types of Hardware Kits	54
6.2	The SK-S7G2 Starter Kit	54
6.3	Synergy Development Kits	56
6.3.1	The DK-S7G2 Development Kit	57
6.3.2	The DK-S3A7 Development Kit	58
6.3.3	The DK-S124 Development Kit	59
6.4	The PK-S5D9 Promotion Kit	60
7	STARTING THE RENESAS SYNERGY™ STARTER KIT SK-S7G2 FOR THE FIRST TIME	61
7.1	Connection and Out-Of-The-Box Demo	61
7.2	Downloading and Testing an Example	62
8	HELLO WORLD! – HELLO BLINKY!	64
8.1	Your First Project Using e ² studio	65
8.1.1	Creating a Project with The Project Configurator	65
8.1.2	Setting Up the Runtime Environment with The Synergy Configurator	67
8.1.3	Writing the First Lines of Code	68
8.1.4	Compiling the First Project	70
8.1.5	Downloading and Debugging the First Project	71
9	INCLUDING A REAL-TIME OPERATING SYSTEM	74
9.1	Threads, Semaphores and Queues	74
9.2	Adding a Thread to ThreadX® using e ² studio	75
10	SENDING DATA THROUGH USB USING A QUEUE	81
10.1	Setting Up an USB Port in Synergy Platform	81
10.2	Sending Messages	83
10.3	Setting Up a Receiver On the Host Side	84
11	EVENT ANALYSIS WITH TRACEX®	86
11.1	An Introduction to TraceX®	86
11.2	Using TraceX® with e ² studio	87
11.2.1	Installation	87
11.2.2	Setting Up TraceX®	88
11.2.3	Built-in Views and How to Use Them	89
11.2.4	Viewing and Interpreting the Data	90
11.3	Using TraceX® With IAR Workbench® for Renesas Synergy™	92
12	WHERE TO GO FROM HERE	93
12.1	Product Examples	93
12.2	Application Examples	94
12.3	More Details on the Synergy Gallery	95
12.3.1	Qualified Software Add-On	95
12.3.2	Verified Software Add On	95
12.3.3	Synergy Partner Showcases	95
12.3.4	Additional Software from Renesas	96
13	SUMMARY AND ACKNOWLEDGEMENTS	98
	INDEX	100

ПРЕДИСЛОВИЕ

Если Вы действительно счастливчик, то однажды произойдет что-то такое, что изменит Вашу жизнь или жизнь тех, кто вокруг Вас. Я был настолько удачлив, что испытал это дважды. Может быть, даже и три раза! Я имею в виду не выигрыш в лотерею или какой-то другой личный опыт, а событие в своей карьере.

Первый раз это случилось, когда я присоединился к компании “Hitachi” в апреле 1989 года. Hitachi в тот момент приступила к снятию с производства микроконтроллеров серии 63xx, которые производились по лицензии компании Motorola до тех пор, пока по неизвестным мне причинам лицензия была отозвана. Когда стало ясно, что нам придётся доставлять клиентам неудобства, остановив производство популярной линейки микроконтроллеров, Hitachi приняли решение нанять большую команду инженеров и поручить им разработку абсолютно нового микроконтроллера следующего поколения. То, что последовало далее сохранило главное преимущество семейства 63xx, которое они заменили, а именно: электрическую совместимость между устройствами с OTP (one time programmable) и масочной памятью. Это сделало прототипирование гораздо проще, чем раньше, когда приходилось ждать рабочие образцы с производства прежде, чем можно было приступить к оценке прототипов. И, в то время, когда инженеры отходили от использования сложного языка ассемблера в пользу более удобных высокоуровневых языков программирования, была введена архитектура, основанная на регистрах, что позволило применить более эффективную обработку прерывания и, следовательно, эффективный код и быстрое выполнение структурного программирования. Затем в начале 1990-х годов мир был взорван технологией flash памяти, с которой у нас (Renesas) нет ни одного изветного сбоя до сих пор.

Микроконтроллер H8 предоставил так много преимуществ, что стал микроконтроллером, который выбирали для множества различных решений. В середине и в конце 90-х годов мы поставляли более 1 миллиона микроконтроллеров в месяц для использования в управлении двигателями в приводах в ПК. H8 выбрали для одного из самых ранних интеллектуальных счетчиков, только для одних этих устройств мы поставили более 30 миллионов микроконтроллеров. Было и много других потребителей. Причина такого широкого признания продукта заключалась в подтверждении преимуществ, которые он предоставлял для инженерных сообществ. Изменило ли это их жизнь? Возможно, это слишком сильное заявление. Но это, безусловно, позволило разработчикам делать то, что ранее было невозможно.

Хорошо. Я упомянул еще две другие инновации, меняющие жизнь. Следующим было изобретение цветных жидкокристаллических дисплеев с активной матрицей (TFTLCD). Я до сих пор помню, как звонил клиентам и предлагал «демонстрацию нового цветного дисплея, который обеспечивал непревзойденный высокий контраст, невероятные углы обзора и ... полный цвет!» Ни один клиент не отказывался от посещения! Мне повезло, что мой первый заказчик был из Европы, с ним я совершил первую рабочую командировку в Японию, чтобы посмотреть, как производятся дисплеи TFT. Вот что может произойти, когда вы предоставляете технологии, продукты и решения, меняющие жизнь.

В последующие годы Hitachi объединилась с компанией Mitsubishi и, позднее, с NEC, обе из которых обладали превосходными линейками микроконтроллеров. И снова у нас были невероятные результаты... самая большая в мире компания, производящая микропроцессоры долгое долгое время, собственная технология флэш-памяти, предлагающая наивысшую производительность (доступ в режиме ожидания с нулевым ожиданием на 120 МГц), 40-нм процесс, обеспечивающий исключительную интеграцию (до 8 МВ на чипе) и наши собственные 300-миллиметровые производственные линии, обеспечивающие высочайшее качество.

Примерно в то же время многие другие поставщики кремния прекратили инвестировать в собственные разработки своих процессоров и начали лицензировать технологию ARM® для универсальных микроконтроллеров. Постепенно клиенты

начали выбирать то, что они воспринимали как «основанное на стандартах» ядро ARM®, а не фирменные ядра, независимо от преимуществ, которые они могли бы предложить. Мы могли бы написать много текста о том, действительно ли микроконтроллеры на базе ARM® основаны на стандартах, но давайте оставим это для другой книги. Дело в том, что все больше и больше клиентов просили их. Некоторые даже сказали бы, что ARM® коммерциализирует рынок, приводя цены к историческим минимумам. Отлично подходит для клиентов, но не настолько хорош для компании, где основой бизнеса являются микроконтроллеры. Пришло время для переосмысления. Мы говорили с клиентами о лучших фирменных ядрах, но большинство потребителей не требовали более высокой производительности. Мы говорили о лучшей интеграции, но большинство систем имели достаточно памяти и периферийных опций, доступных на чипе. Возможно, это была первая подсказка нам о том, что будущее не лежит исключительно в аппаратных средствах. Но мы не поняли её в то время.

В начале 2013 года я был назначен в небольшую команду в компании Renesas Electronics Europe для обсуждения нашей будущей стратегии по развитию микроконтроллеров. Было ясно, что что-то должно измениться. Включение в производство ядер ARM® было практически неизбежным из-за растущего спроса со стороны клиентов. Но мы знали, что это ничего не изменит само по себе... на рынке уже было более десяти других поставщиков микропроцессоров ARM® (число которых недавно существенно сократилось из-за ряда слияний и поглощений). Мы должны были найти что-то совершенно новаторское.

Во время наших последующих встреч мы консультировались с рядом ключевых заказчиков о ежедневных проблемах их проектных групп. Стало ясно, что микроконтроллер не мог быть единственным центром в будущем. Задача разработчиков теперь заключалась в программном обеспечении, тестировании и интеграции, связи, безопасности... На горизонте маячил призрак Интернета Вещей.

Так начался наш проект. И какой проект! Не только проектирование совместимых семейств микроконтроллеров на базе ARM®, но и массивная программная платформа, состоящая из более чем 1 миллиона строк кода и представляющая примерно 200 человеко-лет развития, по-настоящему удобный набор инструментов и многие другие нововведения. Мы называем это Renesas Synergy™ Platform. Это смена правил игры. Вы можете начать программировать на API, сократить время выхода на рынок, сэкономить деньги на этом пути и тратить время на то, что вы делаете лучше всего - инновации!

Вот о чем эта книга... помочь Вам сделать первые шаги на Renesas Synergy™ Platform. Это может изменить вашу жизнь!

Энди Хардинг

Директор по маркетингу
Renesas Electronics Europe

1 ВВЕДЕНИЕ В RENESAS SYNERGY™ PLATFORM

Когда меня попросили написать книгу, чтобы помочь инженерам сделать первые шаги с Renesas Synergy™ Platform, я был сначала польщён и взволнован, но через некоторое время, совершенно обескуражен. С чего начать? Проще говоря, Renesas добился революционного нового уровня абстракции в процессе проектирования, что позволило разработчику начать разработку в интерфейсе прикладного программирования (API), что в свою очередь сокращает время выхода на рынок, снижает общую стоимость продукта и устраняет барьеры для начала производства. Ниже уровня API находится массивная программная платформа, которая содержит более 200 человеко-лет работы разработчиков и более миллиона строк кода. И Renesas готова поддерживать, обеспечивать гарантией и сохранять платформу как продукт в соответствии с высокими стандартами качества и надёжности.

Так что же такое SynergyPlatform? На рис. 1-1 показаны пять основных элементов этой системы:

- **Synergy Software.** Программное обеспечение Synergy: полная программная платформа с общими API-интерфейсами.
- **Synergy Microcontrollers.** Микроконтроллеры Synergy: семейство масштабируемых ARM® Cortex®-M микроконтроллеров.
- **Synergy Gallery.** Галерея Synergy: веб-доступ к специальному программному обеспечению Synergy, инструментам и т. д.
- **Synergy Tools and Kits.** Инструменты и наборы Synergy: интуитивно понятные средства и наборы для разработки.
- **Synergy Solutions.** Решения Synergy: специальные решения для продуктов и приложений.

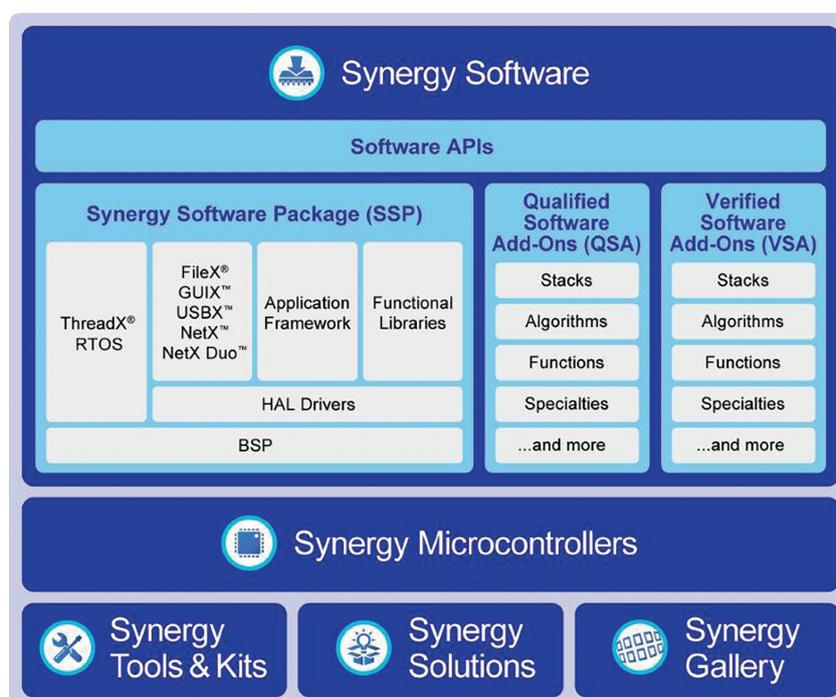


Figure 1-1: The five elements of the Synergy Platform

Уникальность системы SynergyPlatform заключается в том, что она имеет гарантию и поддержку как продукт, произведённый компанией Renesas. Она также является мечтой разработчика, потому что делает их жизнь очень простой, позволяя им вести разработку из API и иметь в качестве преимущества единственный заслуживающий доверия источник поддержки - Renesas. Также мы не забыли и о покупателях, поэтому разработка может начаться без каких-либо предварительных затрат, а также гарантировать конкурентную цену устройства при запуске производства. Просто повторите - без предварительных затрат - вы заходите в Галерею Synergy, загружаете пакет программного обеспечения Synergy (SSP) и набор инструментов, и начинаете разработку НЕМЕДЛЕННО!

Далее следует более технически ориентированное описание платформы, которое стоит прочитать, если вы инженер, который хочет знать всё. Но, с другой стороны, если вам не терпится начать свою собственную разработку на уровне API, тогда вы можете пропустить следующие главы и перейти непосредственно к главе 4 «Начало работы с набором инструментов Synergy Platform» (поясняет установку набора инструментов) или в главу 8 «Hello World — Hi Blinky» (описывает первое приложение в системе SynergyPlatform).

Ключевые части платформы были оптимизированы для совместной работы, создавая новые блоки по мере необходимости, а также повторно используя проверенный IP. Для разработчика это является гарантией, что все работает без сбоев и что разработка программного обеспечения может начинаться сразу, не беспокоясь о базовых уровнях, инициализации периферийных устройств или установке операционной системы реального времени (RTOS). Это освобождает разработчиков от решения этих базовых задач и помогает сократить время выпуска продукта на рынок.

Микроконтроллеры на базе ARM Cortex-M были разработаны с учетом программного обеспечения и отлично масштабируются на любом уровне, как высоком, так и низком. Это возможно благодаря тому, что регистры сохраняют свои адреса и битовые позиции в разных сериях. В разных сериях существует общность периферийных IP-блоков. Это означает, что переход от одной серии микроконтроллеров к другой в большинстве случаев просто потребует перекомпиляции проекта с новыми настройками, а совместимость выводов между сериями позволяет очень просто расположить устройство на печатной плате.

И последнее, но не менее важное – Галерея Synergy доступна в сети предоставляет очень удобную единую точку доступа ко всему, что необходимо для разработки приложения Synergy: компиляторы, среда разработки, инструменты, документация и поддержка не только от Renesas, но и от третьих лиц. Таким образом, с системой SynergyPlatform, у вас под рукой будет полная экосистема для разработки!

1.1 Проблемы в современной разработке встраиваемых систем

За последние десятилетия встраиваемые системы изменились. До начала 21 века в них использовалось несколько простых интерфейсов, таких как кнопки ввода или символьный ЖК-дисплей для вывода, и использовалась одна функция, в основном реализованная как простой цикл внутри `main ()` с прерываниями для обработки ограниченного количества задач. Для этой системы было бы достаточно микроконтроллера с быстродействием в несколько MIPS (миллион инструкций в секунду), пары килобайт памяти и базовой последовательной связи.

Современные встраиваемые системы имеют высокие коммуникационные способности, имеют широкий спектр интерфейсов, таких как Ethernet, беспроводные или графические пользовательские интерфейсы, которые должны быть сконфигурированы / обработаны и которые обмениваются данными и сообщениями как друг с другом, так и с внешним миром для формирования полноценного приложения. Для этого может потребоваться микроконтроллер с тактовой частотой 100 МГц или более, несколько мегабайт флэш-памяти и, возможно, 128 КБ ОЗУ. Операционная система реального времени (ОСРВ, англ. real-time operating system - RTOS) может быть полезна, или даже необходима, так как разным задачам требуется назначить приоритет и выполнять их одновременно. Разработка таких систем уже невозможна таким же способом, как разрабатывались системы предыдущего поколения, поскольку повышенные требования к подключению делают эти системы менее аппаратно-ориентированными и более нацеленными на программное обеспечение.

Кроме того, циклы разработки становятся короче, и всё чаще появляются запросы новых функций. Все это не только налагает большую нагрузку на разработчика, которому приходится решать все новые задачи, но и является огромным капиталовложением, которое, однако, может быть видно не с самого начала.

Предварительно интегрированная платформа предлагает низкий барьер для начала разработки и сводит к минимуму совокупную стоимость продукта, поскольку затраты и необходимая работа очевидна, что помогает ускорить процесс разработки, сокращает время выхода на рынок и сохраняет бюджет проекта. Все это позволяет интегрировать больше функций в доступное время и с одинаковыми затратами и, следовательно, создавать сильно дифференцированные продукты.

Стоит также упомянуть, что Synergy Platform не требует предварительных расходов. Приобретая только один микроконтроллер Synergy, вы можете получить доступ к коммерческому программному обеспечению со стоимостью в несколько сотен тысяч долларов, без ограничения количества мест разработки, количества конечных продуктов или общего количества микроконтроллеров.

1.2 Программный пакет Synergy

В основе полной экосистемы Synergy Platform лежит пакет программного обеспечения Synergy (Synergy Software Package - SSP), специально оптимизированный для архитектуры микроконтроллеров Synergy. Он объединяет фреймворки приложений для служб на системном уровне, функциональные библиотеки, содержащие специальное программное обеспечение, например, для обработки цифровых сигналов, очень эффективные драйверы слоя аппаратных абстракций (Hardware Abstraction Layer — HAL) для всех периферийных устройств и Пакет Поддержки Платформы (Board Support Package - BSP), включая код запуска для плат и микроконтроллеров.

Помимо программного обеспечения, упомянутого выше, SSP также включает в себя известные и проверенные пакеты Express Logic, такие как ThreadX® RTOS и коммуникационные стеки и графическое связующее ПОX-Ware™. SSP охватывает все основные функции, такие как инициализация микроконтроллера и его периферийных устройств, или настройка операционной системы реального времени. Многие инженеры проводят месяцы, если не годы, разрабатывая такого рода низкоуровневое программное обеспечение, вместо того, чтобы сосредоточить свое время на разработку приложения - это и есть сила Synergy Platform!

Если вам интересно узнать подробности о SSP, будьте терпеливы. Мы рассмотрим их в главах 2 и 3.

1.3 Семейство микроконтроллеров Synergy

Если посмотреть на семейство микроконтроллеров Synergy, изначально существует четыре серии - S1, S3, S5 и S7, они используются в конечных продуктах, от небольших, работающих от батарей датчиков до высокопроизводительных систем с интенсивной обработкой. Встроенные периферийные устройства для подключения, графические дисплеи, системы безопасности и другие дополнения делают их хорошо подходящими для быстро растущего рынка Интернета вещей, но отнюдь не ограничены только этим.

Все микроконтроллеры Synergy основаны на 32-разрядных ядрах ARM Cortex-M (см. Рис. 1-2 для исходных серий). Серии S1 основаны на ядре M0+, в то время как S3, S5 и S7 основаны на M4. Все они включают в себя стандартные периферийные устройства от ARM, такие как векторный контроллер вложенных прерываний (Nested Vectored Interrupt Controller - NVIC) или ARM Модуль управления защитой памяти (Memory Protection Unit). Кроме того, компания Renesas добавила свой собственный IP-блоки, где у ARM отсутствует решение, и требуется дополнительная производительность или функции, где это необходимо. Этот дополнительный IP основан на проверенной технологии Renesas, адаптированной к требованиям совместимости и масштабируемости системы Synergy Platform.



Figure 1-2: The initial series of the Synergy MCUs

В первоначально выпущенной серии микроконтроллеров представлены следующие функции:

- **Серия S1 — сверхнизкое потребление тока:** частота ядра до 32 МГц. Флэш-память до 256 Кб. Сверхнизкое потребление тока в активном режиме 70,7 мкА на 1 МГц. Диапазон напряжений от 1,6 В до 5,5 В. Периферийные устройства включают в себя аналого-цифровые преобразователи, таймеры, емкостные сенсорные датчики, последовательные интерфейсы и модули безопасности.
- **Серия S3 — высокая эффективность:** включает блок с плавающей точкой (FPU). Частота ядра от 32 МГц до 100 МГц. Флэш-память объемом до 1 Мб. Диапазон напряжений от 1,6 В до 5,5 В. Периферийные устройства включают в себя аналого-цифровые преобразователи, таймеры, сегментный жидкокристаллический контроллер, USB и последовательные интерфейсы, модули безопасности.
- **Серия S5 — высокая интеграция:** включает блок с плавающей точкой (FPU). Частота ядра между 100 МГц и 200 МГц. До 2 Мб флэш-памяти. Диапазон напряжений от 2,7 до 3,6 В. Периферийные устройства включают в себя аналого-цифровые преобразователи, таймеры, графический жидкокристаллический контроллер, емкостный сенсорный блок, Ethernet MAC, USB, CAN, последовательные интерфейсы, модули безопасности.
- **Серия S7 — высокая производительность:** включает FPU. Частота ядра между 200 МГц и 300 МГц. До 4 Мб флэш-памяти. Диапазон напряжений от 2,7 до 3,6 В. Периферийные устройства включают в себя аналого-цифровые преобразователи, таймеры, графический жидкокристаллический контроллер и двумерный графический движок, Ethernet MAC, USB, CAN, последовательные интерфейсы, модули безопасности.

Каждая серия будет постепенно расширяться по мере появления новых устройств, и Renesas уже планирует дополнительные серии с еще большим количеством инноваций!

Все микроконтроллеры в каждой серии совместимы по характеристикам и контактам. Это позволяет легко масштабировать и повторно использовать код с одного устройства на другом. Разработчики извлекают из этого выгоду, так как нет необходимости выбирать конечное устройство в самом начале, потому что переход на другой микроконтроллер в дальнейшем будет простым. Совместимость поддерживается не только в каждой серии, но и между всеми четырьмя сериями платформы. Это относится не только к периферийным устройствам, но и к распиновке. Например, устройства серии S3 и устройства серии S7 в корпусе LQFP-100 имеют одинаковую разводку и, следовательно, заменяют друг друга. Более того, можно даже создать плату с посадочными местами под разные корпуса микроконтроллеров, вписанными друг в друга, что позволит гибко конфигурировать конечный продукт.

На рис. 1-3 показаны основные функции и периферийные устройства серии S7 микроконтроллеров Synergy, представляющих множество устройств входящих в семейство микроконтроллеров Synergy, имеющих самую большую встроенную память и самый полный набор периферийных устройств. По этой причине наши примеры и проекты в этой книге основаны на комплекте Synergy Starter Kit серии S7.

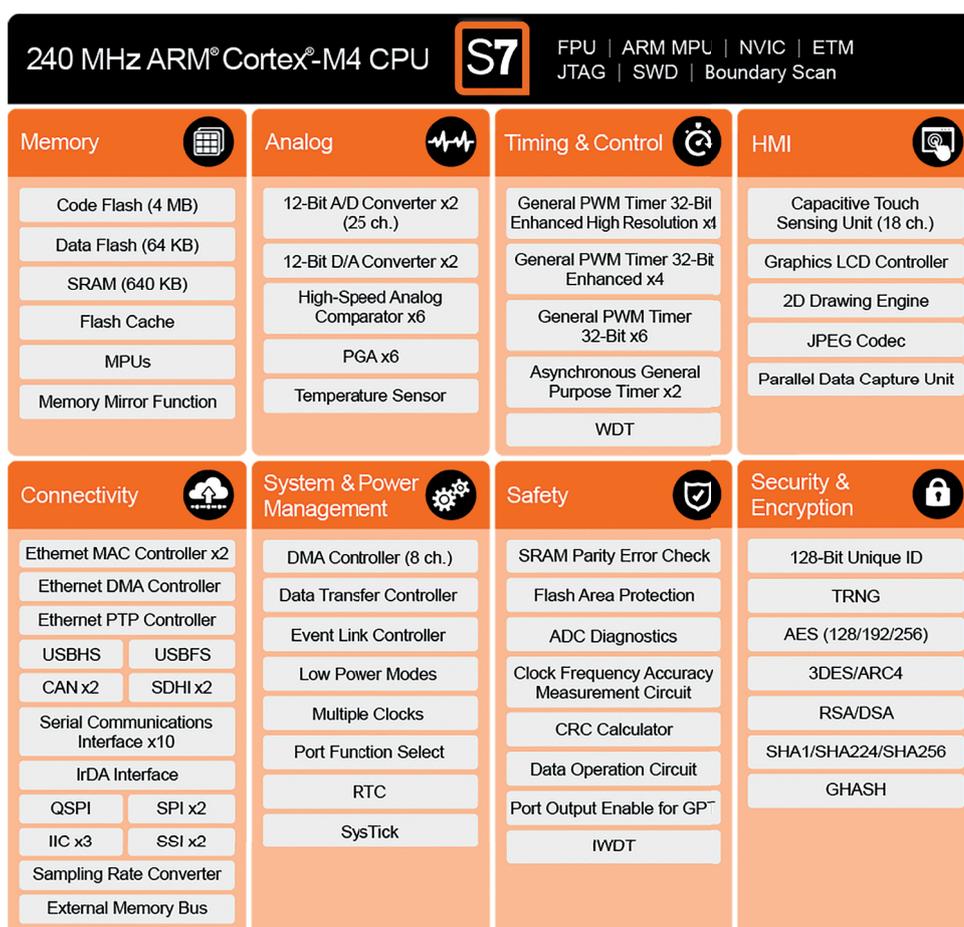


Figure 1-3: Block diagram of the S7 Series of Synergy MCUs

Периферийные устройства в младших микроконтроллерах в основном являются подмножеством тех, которые находятся в старших устройствах. Например, контроллер реального времени (RTC), появившийся в серии S1, является подмножеством RTC, размещённым в микроконтроллерах Synergy серии S7. Он не имеет функции сохранения времени захвата и обнаружения несанкционированного доступа и имеет уменьшенную функциональность таймера, контрольные регистры не имеют расширений, поскольку они сокращены до минимального набора функций. Кроме того, смещение адресов контрольных регистров постоянны, даже когда функции удалены. Это означает,

что программное обеспечение, записанное для RTC устройства серии S1, будет работать на устройстве серии S7 без изменений. Переход от S7 к устройству S1 Series означает, что необходимо изменить только те строки кода, которые принадлежат к функции, недоступной на меньшем устройстве, что значительно облегчает жизнь разработчикам. Это ясно показывает, что при разработке кристаллов было заложено много мысли.

1.4 Инструменты и наборы Synergy

Компания Renesas предприняла огромные шаги для создания полезных программных и аппаратных средств разработки, которые могут быть использованы для изучения технических возможностей системы и которые выведут пользователя за пределы стадии оценки, вплоть до производства.

Доступные инструменты разработки программного обеспечения включают следующие компоненты:

- **e² studio**: интегрированная среда разработки решений (Integrated Solution Development Environment — ISDE) на базе Eclipse от Renesas, включает специальные конфигураторы для платформы Synergy. Использует инструменты генерации кода GCC ARM[®] Embedded.
- **IAR Embedded Workbench[®] для Renesas Synergy™**: Интегрированная среда разработки от IAR Systems[®] для платформы Synergy с использованием собственного компилятора ARM Cortex[®]-M IAR. Работает с автономным конфигуратором Renesas Synergy™ (Synergy Standalone Configurator — SSC).
- **TraceX[®]**: средство визуализации событий и состояний операционной системы реального времени от Express Logic.
- **GUIX Studio™**: программа на базе ПК от Express Logic для проектирования графических пользовательских интерфейсов.

Среда “e² studio” содержит все инструменты, необходимые для создания, компиляции и отладки проектов для платформы Synergy. Он называется интегрированной средой разработки решений, поскольку добавлены дополнительные ориентированные на решение компоненты и плагины, что делает его более мощным. Это особенно актуально для конфигураторов, которые обеспечивают простой графический доступ к различным аппаратным функциям, таким как модуль тактового генератора или конфигурация портов ввода-вывода, без необходимости глубокого изучения руководства пользователя. Эти конфигураторы автоматически создают все необходимые настройки и код инициализации и включают функцию проверки ошибок для обнаружения проблемных настроек уже во время разработки.

Во время разработки для запуска первых тестов будет необходимо аппаратное обеспечение. В большинстве случаев это будет необходимо задолго до того, как будет готова собственная плата пользователя. С этой целью Synergy Platform предлагает два разных типа комплектов, и оба имеют встроенный интерфейс отладки и программирования J-Link[®]:

- **Synergy Starter Kit (SK)**: для общей оценки возможностей системы. Использует микроконтроллеры S7G2 Group, так как это микроконтроллер из старшего семейства, дающее хорошее представление о платформе Synergy. Вы можете оценить всю экосистему с помощью этого набора, а затем выбрать любое устройство из разных семейств для собственного проекта, поскольку все они также применимы к меньшим братьям и сестрам. Около 80% контактов микроконтроллера доступны через разъемы, поэтому подключение вашего собственного оборудования не будет являться проблемой, это действительно важная функция для оценки возможностей.
- **Synergy Development Kit (DK)**: для полного цикла проектирования с доступом ко всем контактам через несколько портов расширения и до четырех разъемов Pmod™. Они оснащены Bluetooth[®] с низким энергопотреблением для беспроводного подключения к мобильному устройству и многими другими специализированными платами расширения

1.5 Решения Synergy

Для специальных потребностей, таких как человеко-машинные интерфейсы (human-machine interfaces — HMI), интеллектуальная сенсорная сеть или промышленная сеть, Решения Synergy предоставляет специализированные наборы, которые выходят за рамки типичного встроенного оборудования для разработки продуктов. Существует два вида решения:

- **Пример продукта Synergy (Synergy Product Example — PE):** даёт представление того, как будет разработан конечный продукт. Поставляется с необходимым аппаратным обеспечением, программным обеспечением и полной документацией, включая схемы, макеты, спецификации, файлы Gerber и специальные «путевые документы для проектирования», описывающие методологию того как и почему был совершён тот или иной выбор в разработке и обоснование подбора компонентов во время процесса проектирования.
- **Пример применения Synergy (Synergy Application Example — AE):** Примеры построения технологии, которые могут быть использованы для создания приложения. Они включают в себя несколько компонентов, создающих демонстрацию того, как несколько технологий могут быть использованы для создания продукта. Например, AE подключенный к облаку системы продемонстрировал бы использование различных беспроводных сетей, человеко-машинных интерфейсов и облачных соединений и сервисов.

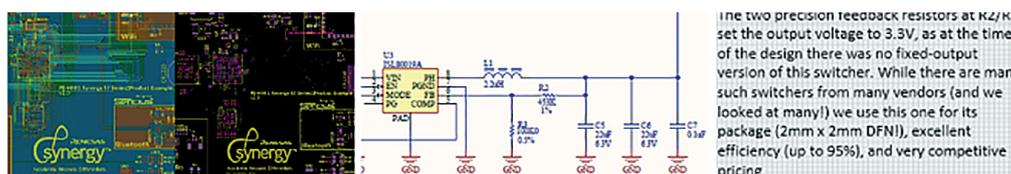


Figure 1-4: Product examples include schematics, PCB layout files, Designer's Notebook and much more

Решения Synergy основывается на программном пакете Synergy, а также на VSA (Verified Software Add-Ons) и QSA (Qualified Software Add-Ons) и партнерских проектах (Partner Projects - PP) от третьих сторон. Они дополняют программные опции, доступные в SSP, обеспечивая ещё больше удобства в разработке и версии для их оценки можно загрузить из галереи Synergy. Более подробная информация о VSA, QSA и PP представлена в [главе 12](#).

1.6 Галерея Synergy

Галерея Synergy — это ваша единственная точка доступа ко всему, что связано с программным обеспечением Synergy, инструментами Synergy и лицензированием. Вдабавок доступ к ПО и сервисам сторонних поставщиков, которые участвуют в экосистеме Synergy Platform. Это требует однократной регистрации, после которой предоставляется доступ ко всему программному обеспечению и документации, необходимой для разработки. Это также место для запроса лицензии для использования в фактической разработке и для массового производства.

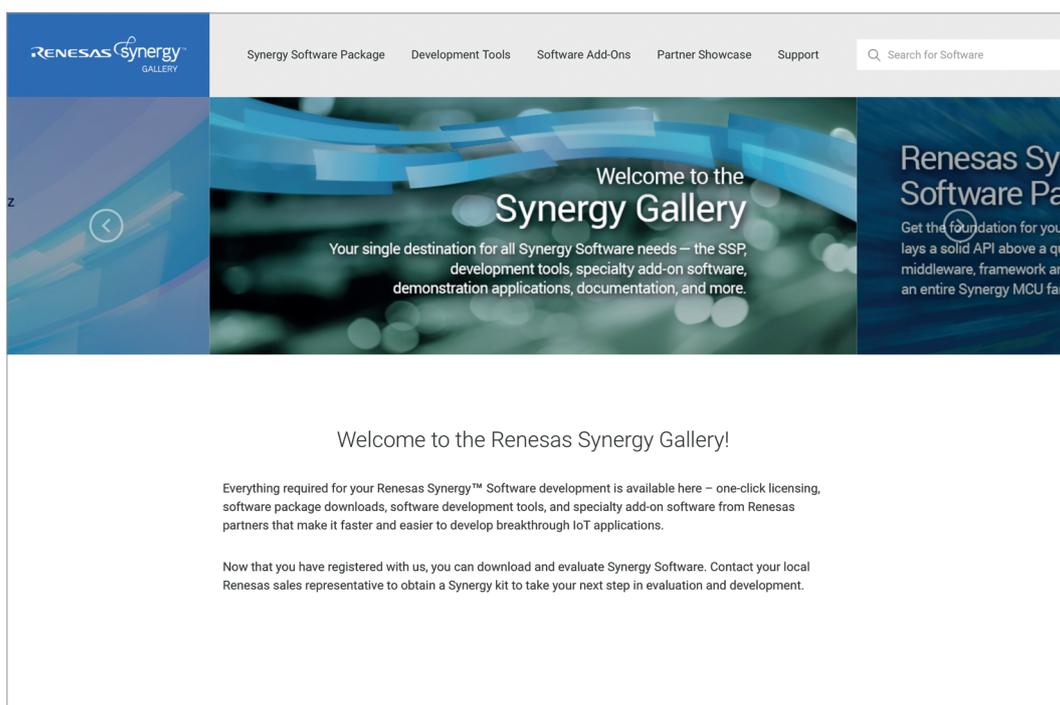


Figure 1-5: The welcome screen of the Synergy Gallery is the gateway for the access to all tools, software and support

И если что-то пойдет не так во время разработки, всегда доступна помощь в галерее Renesas. Просто выберите «Поддержка» в верхней части страницы, чтобы перейти на главную страницу поддержки. Для начала ознакомьтесь с базой знаний (<https://en-eu.knowledgebase.renesas.com/>) или форумом Renesas Rulz (<http://renesasrulz.com/synergy/>). Здесь вы можете задать свой вопрос и получить ответ от инженера службы поддержки Renesas.

В будущем платформа Synergy будет включать полную и безопасную облачную инфраструктуру для ваших конечных продуктов.

Вопросы, которые следует вынести из этой главы:

- Платформа Renesas Synergy™ состоит из пяти элементов:
 - Программное обеспечение Synergy
 - Микроконтроллеры Synergy
 - Галерея Synergy
 - Инструменты и наборы Synergy
 - Решения Synergy

2 ПОДРОБНЕЕ О ПРОГРАММНОМ ОБЕСПЕЧЕНИИ SYNERGY™

Что вы узнаете в этой главе:

- Что представляют собой различные компоненты программного пакета Synergy (SSP) и какие уровни они имеют.
- Подробная информация о различных уровнях их совместной работе.
- Особенности промежуточного ПО X-Ware™ и операционной системы реального времени ThreadX®.

Лучший микропроцессор не может раскрыть свой полный потенциал, либо если работающее на нем программное обеспечение не в состоянии выполнить задачу, либо если экосистема программного обеспечения слишком сложна, чтобы быть полезной для разработчика. Хорошо продуманная концепция, такая как Synergy Platform, помогает создавать приложения, которые легко разрабатываются, просты в обслуживании и которые используют максимум из производительности и возможностей микроконтроллера.

Простота, заложенная в платформе Synergy Platform, была достигнута благодаря тому, что все основные компоненты, микроконтроллеры, программное обеспечение, инструменты, комплекты и прикладные решения отлично работают вместе.

Программный пакет Synergy (SSP) был специально оптимизирован для архитектуры микропроцессоров Synergy, которая, в свою очередь, была разработана с учетом программного обеспечения. SSP объединяет фреймворки приложений, функциональные библиотеки, драйверы слоя аппаратных абстракций (HAL) и, как основу для всего, Пакет Поддержки Платформы (BSP) вместе с широко используемой операционной системой реального времени ThreadX® и промежуточным программным обеспечением X-Ware™ от компании Express Logic. ThreadX и X-Ware лицензированы для неограниченной разработки и использования в продукции, если используются вместе с SSP. Это дает единый и простой в использовании программный пакет, в котором все функции могут быть доступны с помощью простого и надежного API.

Полный пакет SSP и даже части, созданные Express Logic, полностью поддерживаются Renesas. Это означает, что у вас, как у разработчика, будет только один контактный адрес, если вам нужна поддержка. Не требуется иметь дело с разными каналами поддержки, для всего внутри Synergy Platform есть только одно: Renesas! Много работая с различными программными пакетами от разных поставщиков в множестве проектах, я знаю, что это означает: жизнь намного проще, так как нет необходимости искать специалистов и говорить с разными людьми о проблеме и обеспечивает интерфейс между ними.

В любой среде разработки Synergy разработчики обычно имеют полную экранную видимость всего исходного кода SSP во время разработки и отладки. Однако некоторые программные модули SSP защищены, что означает, что в момент, когда их можно просматривать на экране и при использовании источника для компиляции, они не могут быть изменены, распечатаны или сохранены в файл вне среды разработки (см. Рис. 2-1). В случае, если вам нужно продвинуться дальше, лицензия исходного кода SSP, доступная от Renesas, позволяет преобразовывать защищенные программные модули в текстовые файлы, которые можно редактировать, печатать и сохранять в файл. Лицензии исходного кода SSP можно приобрести для отдельных или всех программных модулей SSP.

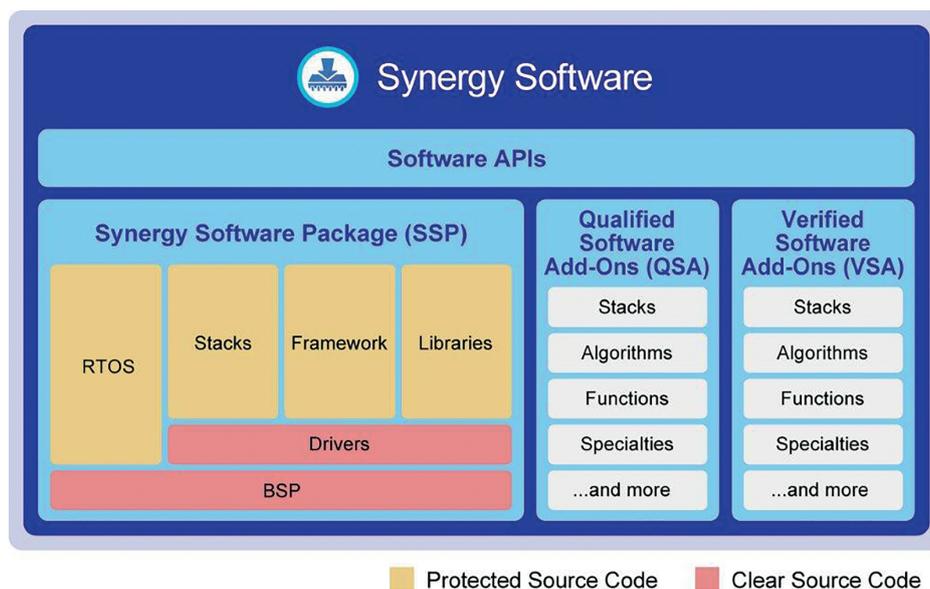


Figure 2-1: Parts of the source code is protected from editing, but can be unlocked by obtaining a source code license

2.1 Знакомство с пакетом программного обеспечения Synergy (SSP)

Как упоминалось ранее, пакет программного обеспечения Synergy (SSP) представляет собой всеобъемлющий программный продукт, охватывающий все аспекты разработки программного обеспечения для встроенных систем. Он состоит из следующих частей:

- **Пакет Поддержки Платформы (BSP)** настраивается для каждого комплекта аппаратуры Synergy и микроконтроллера. Он включает в себя загрузочный код для всех поддерживаемых блоков. Разработчики, использующие настраиваемые аппаратные средства, могут воспользоваться преимуществами BSP, поскольку они могут быть адаптированы для конечных продуктов и вашей собственной платы с помощью утилиты пользовательского создания пакетов поддержки платформы (Custom Board Support Package Creator), который можно загрузить из Галереи Synergy.
- **Драйверы слоя** аппаратных абстракций независимы от операционной системы реального времени (RTOS-независимые HAL-драйверы) — достаточно эффективные драйверы для всех периферийных устройств и системных сервисов. Они устраняют необходимость долгого глубокого изучения встроенной периферии микроконтроллера, поскольку они абстрагируют пользователя от настройки битов и регистров.
- **Фреймворки приложений** содержат службы системного уровня, связывающие RTOS с уровнем абстракции оборудования (HAL) для межпроцессного обмена сообщениями, работы служб безопасности, последовательной связи, воспроизведения аудио, сенсорного управления, Bluetooth® с низким энергопотреблением и многого другого. Полнота этих фреймворков уменьшает количество ошибок и экономит время при разработке приложения.
- **Функциональные библиотеки** содержат, например, специализированное программное обеспечение для обработки цифровых сигналов или функции безопасности и шифрования, также сокращают время разработки и повышают стабильность конечного приложения.
- **Операционная система реального времени (RTOS) ThreadX®** предоставляет многозадачное ядро реального времени с приоритетным планированием и небольшим объемом требуемой памяти. ThreadX® был развернут в более чем 5,5 миллиардов встроенных устройств в различных областях.
- **Стеки и связующее ПО X-Ware™** включает в себя файловые системы (FileX®), графические пользовательские интерфейсы (GUIX™), стеки USB и TCP/IP (USBX™, NetX™ и NetX Duo™). Все компоненты полностью оптимизированы для системы Synergy Platform и интегрированы в неё.

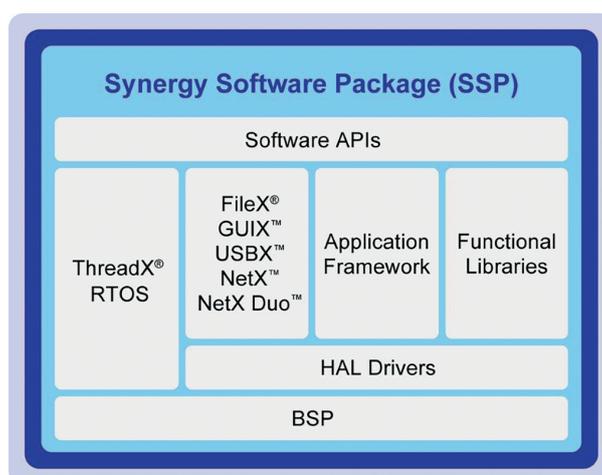


Figure 2-2: Everything in the Synergy software ecosystem is built in layers

Рисунок 2-2 показывает архитектуру различных уровней. К каждому из них можно получить доступ с помощью API-вызовов напрямую или в стеке. Например, аудиоприложение может вызывать аудиораздел прикладных фреймворков, который затем использует часть I²S HAL для обеспечения доступа для чтения/записи к внешнему аудиоконвертеру с использованием последовательного порта. Конечно, конечное приложение может также напрямую обращаться к HAL или BSP, используя вызовы API, но использовать фреймворк гораздо удобнее.

Все модули SSP поддерживают приложения, написанные на C++, кроме того C++ приложения могут быть сконфигурированы и использованы в e² studio и IAR Embedded Workbench[®] для системы Synergy™.

SSP — это программное обеспечение класса “готового к использованию в серийном производстве”, разработанное с использованием передовых отраслевых практик и совместимое со стандартом MISRA C:2012, процессом жизненного цикла программного обеспечения ISO/IEC/IEEE 12207 и вторым изданием стандарта CERT[®]. Оно поставляется с полным комплектом программного обеспечения, в котором указаны эталонные тесты, размер кода, время выполнения и условия тестирования, при которых были получены результаты. Renesas предоставляет более подробную информацию об обширной гарантии качества для Synergy Platform в Галерее Synergy.

Обновления и исправления ошибок для SSP будут происходить регулярно, с запланированными релизами по плану и с каждой крупной или небольшой версией, представляющей новые функции. Даже при длительном использовании Renesas пользуется хорошей репутацией в поддержке своих продуктов, превышая сроки, при которых другие производители прекратили бы обслуживание. В SSP исправления ошибок и добавление улучшений применяются к предыдущей основной версии, в то время как исправления ошибок продолжают внедряться во всех выпусках вплоть до окончания производства микроконтроллера плюс один год. Это пожизненное обслуживание, гарантирующее, что разработчики программного обеспечения могут продолжить работу с их текущим базовым уровнем, не беспокоясь о том, что им придется немедленно переключиться на новую версию из-за ошибок, не зафиксированных в их версии. Долгое время поддерживая клиентов, работающих над крупными (и длинными) проектами, я считаю, что это отличная возможность, так как во время разработки вам не нужно менять версии набора инструментов. Вы не оставлены один на один с программным обеспечением, распространяемым по принципу «как есть».

2.2 Знакомство с Пакетом Поддержки Платформы

Поэтому этот пакет является особенным для комбинации платы и микроконтроллера, который выбирается во время проектирования в e² studio с использованием различных конфигураторов интегрированной среды разработки Synergy (Integrated Synergy Development Environment — ISDE). Каждая предоставляемая Renesas платформа для разработки поддерживается своим BSP. Конфигураторы внутри e² studio будут извлекать необходимые файлы из SSP и настраивать их на основе установок, сделанных в пользовательском интерфейсе. BSP содержит в основном данные, а не исполняемый код, пакет состоит из файлов конфигурации, файлов заголовков и API.

Ядро самого BSP совместимо со стандартным интерфейсом программного обеспечения для микроконтроллеров ARM — CMSIS (ARM[®] Cortex[®] Microcontroller Software Interface Standard), в соответствии с требованиями и соглашениями об именовании этого стандарта.

BSP предоставляет публичные функции, доступные для любого проекта с его использованием, которые обеспечивают доступ к функциональности, которая является общей для всех микроконтроллеров и плат, поддерживаемых пакетом. К ним относятся функции блокировки/разборки аппаратного и программного обеспечения, обработчиков прерываний, регистрации подпрограмм обратного вызова, очистки флагов, генерации программной задержки и функции защиты регистров. Для легкого отличия от других частей SSP имена этих функций начинаются с R_BSP_, а имена ассоциированных макросов с BSP_. Единственным исключением являются функции, предоставляющие функциональность, описанную в CMSIS-Core. Кроме того, BSP включает функцию, которая возвращает информацию внутри структуры (число и назначенные порты ввода/вывода) о светодиодах, используемых на плате.

Новые платы и устройства будут добавлены в BSP, когда они будут доступны, обеспечивая прочную долгосрочную основу для текущих и новых проектов. Поддержка на уровне платформ для настраиваемых плат может быть легко создана с помощью утилиты пользовательского создания BSP, которую можно загрузить с SSP-страницы Галереи Synergy, вместе с приложенной инструкцией с объяснением процедуры.

2.3 Знакомство с драйверами слоя аппаратных абстракций (HAL — HardwareAbstractionLayer)

В верхней части Пакета Поддержки Платформы (BSP) находится слой аппаратных абстракций (HAL), который предоставляет драйверы для периферийных устройств и который взаимодействует с регистрами специальных функций микроконтроллеров для реализации простых в использовании интерфейсов, изолируя программиста от аппаратного обеспечения. Это набор модулей, и каждый из них является драйвером для периферийных устройств, доступных в микроконтроллере Synergy, таких как SPI (Serial Peripheral Interface - последовательный периферийный интерфейс) или ADC (Analog-to-Digital Converter — аналого-цифровой преобразователь). Их имена начинаются с r_, что позволяет легко отличать их от других частей программного пакета Synergy. Эти модули по своей сути независимы от операционной системы реального времени и состоят из двух компонентов: низкоуровневого драйвера (low-level driver — LLD), который напрямую управляет регистрами периферии и использует разные версии одного и того же периферийного устройства, и высокоуровневого драйвера (high-level driver — HLD), код которого специально предназначен для реализации уникальных режимов периферийного оборудования Renesas, но не имеет прямого доступа к регистрам. HLD предоставляет программный интерфейс приложения (API) для фреймворков или пользовательской программы и использует LLD для взаимодействия с микроконтроллером. Преимущество этой архитектуры заключается в том, что LLD обеспечивает очень быстрое написание кода и что HLD делает интерфейсы API переносимыми через разные серии микроконтроллеров Synergy.

Интерфейс для абстрактного аппаратного обеспечения согласован во всех модулях HAL и может быть расширен. Некоторые периферийные устройства поддерживают несколько интерфейсов, а некоторые интерфейсы поддерживаются несколькими периферийными устройствами. Преимуществом этого является гибкость, позволяющая менять требования на более высоком уровне. Если, например, код был первоначально написан для выделенного контроллера I²C, но позже потребуется использовать функциональные возможности интерфейса последовательной связи (SCI), также позволяющего работать с I²C интерфейсом, достаточно изменить информацию о конфигурации. Сам код приложения остается неизменным. Хотя функциональность API можно получить непосредственно через интерфейсы HAL, большинство функций также можно получить через различные ф, доступные в программном пакете Synergy.

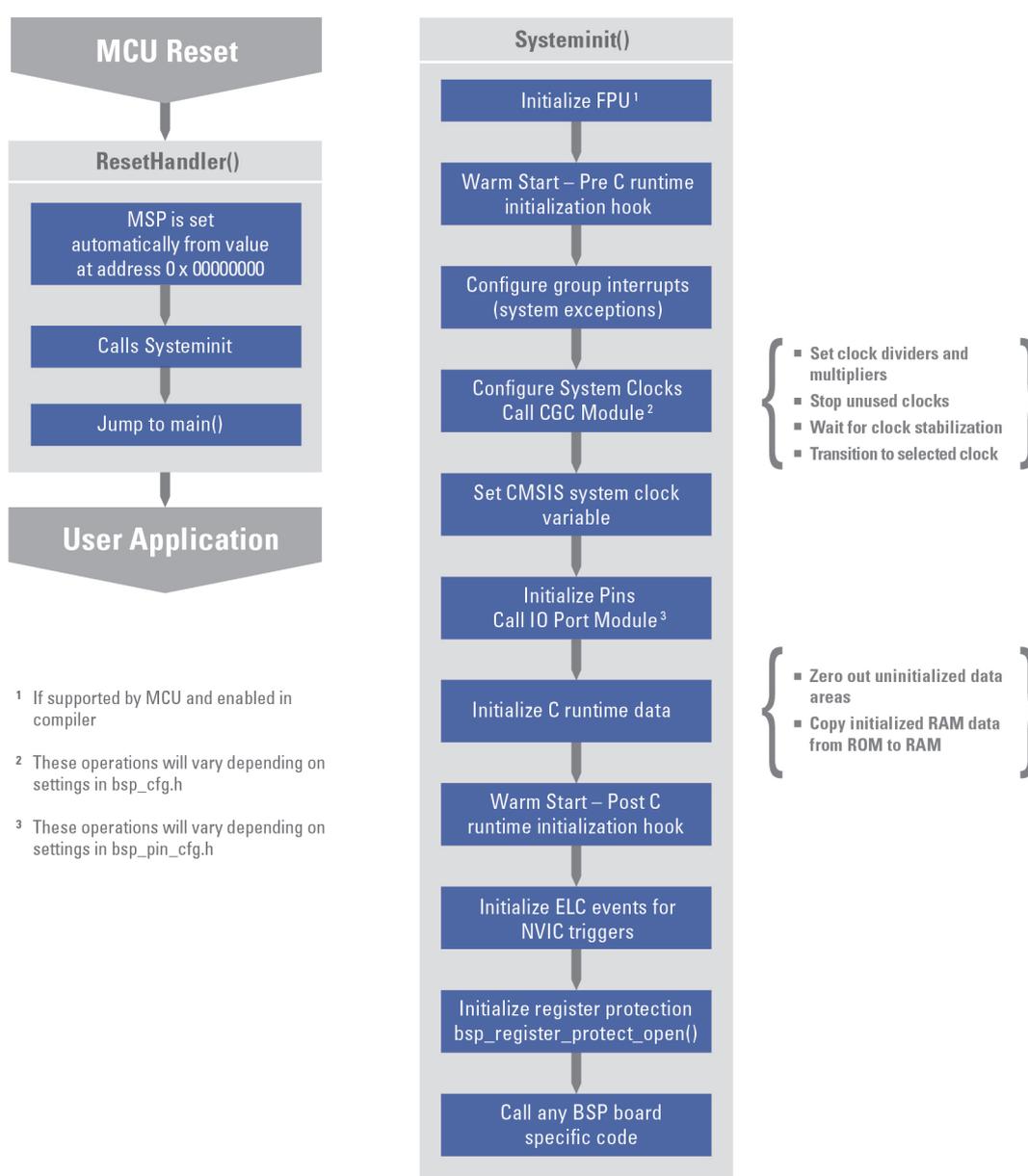


Figure 2-3: The flowchart for the BSP showing the move from reset to main()

2.4 Знакомство с фреймворками приложений

Располагаясь на вершине слоя аппаратных абстракций, фреймворки приложений предоставляют разработчикам еще более высокий уровень абстракции, что позволяет лучше использовать код и упростить программирование, что приводит к сокращению времени разработки. Эти фреймворки обеспечивают абстрагирование различных системных и технологических сервисов, что позволяет использовать богатые функциональные возможности с помощью простых API. Различные фреймворки интегрированы с функциями ОС реального времени ThreadX® для управления конфликтами ресурсов и синхронизации между несколькими пользовательскими потоками (задачами).

Фреймворки приложений доступны, например, для периферийных устройств, таких как SPI (Serial Peripheral Interface — последовательный периферийный интерфейс), ADC (Analog-to-Digital Converter — аналого-цифровой преобразователь) или IIC (Inter-Integrated Circuit), а также для общих служб, таких как аудио, емкостной сенсорный датчик или JPEG. Многие из фреймворков используют службы друг друга, а также объединяют несколько вызовов HAL и BSP. Например, для функций синхронизации фреймворк аналого-цифрового преобразователя использует службы из интерфейса таймера GPT и функции из интерфейсовDMA или DTC для эффективной передачи данных через общий интерфейс.

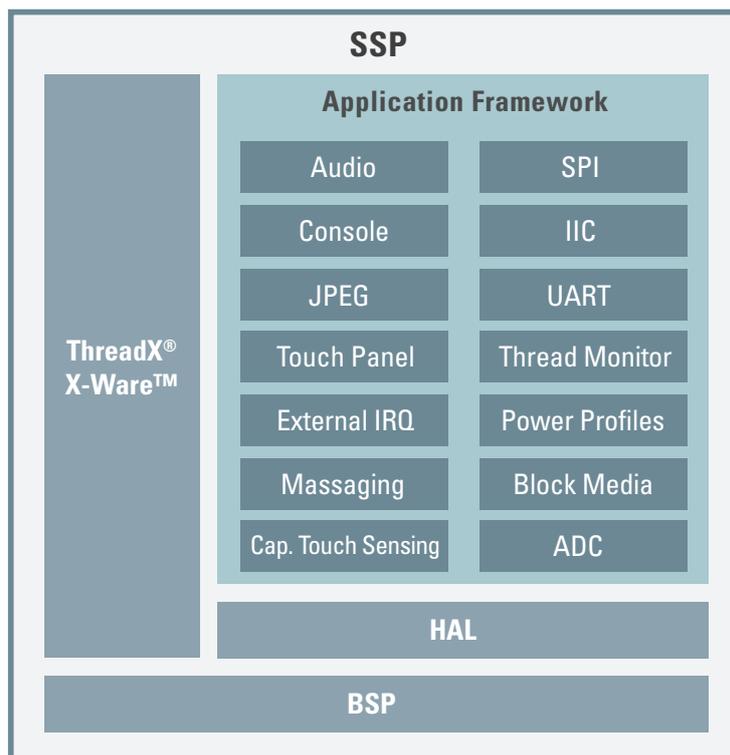


Figure 2-4: The Synergy Application Frameworks provide abstraction to various system-level and technology specific services

Разработчики, использующие фреймворки приложений, выигрывают от более высокого уровня абстракции, повторного использования кода между процессорами и, следовательно, потенциальными продуктами, от более согласованного кода и освобождаются от необходимости переосмысления часто используемых задач в приложениях. В то же время, фреймворки приложений Synergy уменьшают необходимость дальнейшего непосредственного взаимодействия с оборудованием, доступным в микроконтроллере. Все имена модулей начинаются с sf_, что позволяет легко отличать их от других частей программного пакета Synergy.

2.5 Знакомство с функциональными библиотеками

Функциональные библиотеки добавляют дополнительные возможности для платформы Synergy и напрямую взаимодействуют с HAL. Они включают в себя чрезвычайно полезные и тщательно оптимизированные функции и могут быть легко доступны через API. К ним относятся, например, библиотека аппаратной ускоренной цифровой обработки сигналов (Digital Signal Processing - DSP) с использованием стандартного интерфейса программного обеспечения микроконтроллера ARM® Cortex® (CMSIS), в том числе более 60 функций для различных типов данных, а также библиотеки аппаратно-ускоренного шифрования и безопасности с богатыми набором криптографических алгоритмов.

Функциональные библиотеки принадлежат Renesas и другим вендорам, они прошли предварительную проверку, и поддерживаются Renesas, гарантируя максимально возможное качество кода.

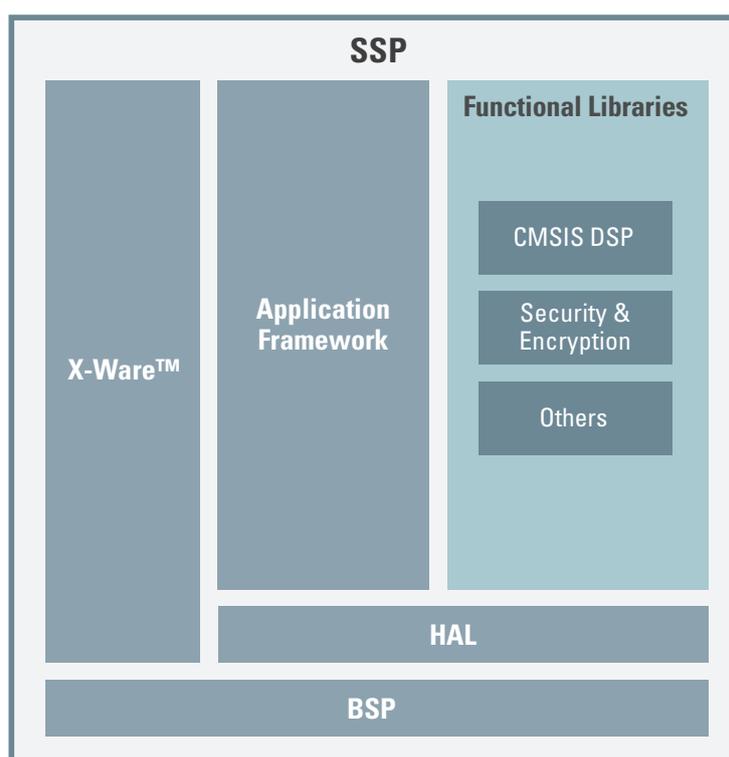


Figure 2-5: The Synergy Functional Libraries add additional capabilities to the Synergy Platform

Дополнительные библиотеки будут добавлены в новые версии программного пакета Synergy (SSP) с течением времени, интегрируя еще больше функций. Это уменьшит время обучения новых программистов микроконтроллеров и ещё сильнее сократит время выхода продукта на рынок.

2.6 Связующее ПО X-Ware™ от ExpressLogic

В дополнение ко всему программному обеспечению, предоставленному Renesas, программный пакет SSP также включает в себя дополнительные библиотеки промежуточного программного обеспечения, созданные Express Logic и поддерживаемые Renesas для обработки файлов (FileX®), графических пользовательских интерфейсов (GUIX™), USB-подключения в режиме хоста и устройства (USBX™) и сетевой связи (NetX™ и NetX Duo™). Эти библиотеки, а также операционная система реального времени ThreadX® предоставляют последовательный и структурный API и используют такой стиль написанного кода, который позволяет сохранить простоту использования для разработчика. Эта беспрецедентная последовательность делает пакет программного обеспечения Synergy хорошо оснащенным для ускорения разработки новых продуктов для быстрого выхода на рынок.

2.6.1 FileX®

Express Logic FileX® — это высокопроизводительная файловая система FAT (File Allocation Table) для SSP, разработанная специально для быстрой реализации, быстрой логики и небольшого размера. Она полностью интегрирована с SSP и ThreadX® и доступна для всех поддерживаемых микроконтроллеров Synergy.

Поддерживаются все форматы FAT, включая FAT12, FAT16, FAT32 и exFAT (расширенный FAT). FileX® построен специально для встроенных систем реального времени, поддерживая внутренний кэш входа FAT. Есть также и некоторые другие функции:

- Смежное распределение файлов
- Последовательный сектор и кластерное чтение/запись
- Длинные имена файлов и соглашения об именах 8.3
- Неограниченное создание объектов FileX®, таких как носители, каталоги и файлы
- Дополнительная отказоустойчивость

FileX® поддерживает запоминающее устройство USB, карты SD/eMMC, SPIFlash, QSPIFlash и встроенную флеш-память, а также оперативную память в качестве носителя.

Высокая надежность достигается благодаря обнаружению и восстановлению ошибок, различным вариантам отказоустойчивости и встроенному сбору статистики производительности.

Библиотека взаимодействует с другими частями SSP через блок Media Interface (fx_io). Абстрактный интерфейс, используя указатели функций вместо прямых вызовов функций, выступает в качестве уровня адаптации для драйверов у мультимедийных устройств. Функции вызываются между файловыми системами и медиадрайверами блоков Synergy Platform в фреймворках приложений, таких как SD/MMC и SPI Flash, которые, в свою очередь, зависят от SDMMC, Flash и SPI-интерфейсов HAL.

Интерфейс остается неизменным для любого медиа-драйвера. Все драйверы носителей функционально идентичны на уровне ввода-вывода файлов и могут быть взаимозаменяемы друг с другом без изменения кода приложения.

2.6.2 GUIX™

GUIX™ от Express Logic — это высокопроизводительная платформа графического интерфейса пользователя SSP, оптимизированная для архитектуры микроконтроллеров Synergy и позволяющая создавать высококачественные пользовательские интерфейсы. Она полностью интегрирована с OCPB ThreadX® и имеет небольшой размер. Она реализована как чистая библиотека на языке C и добавляет только те функции, которые используются приложением в окончательной сборке, используемые приложением в окончательное изображение. Отклик на запрос пользователя становится довольно быстрым благодаря минимальному распределению функций вызова и оптимизации функций прорисовки графических объектов, функции обработки событий и поддержки аппаратного графического ускорителя Renesas.

Исполняемая библиотека GUIX™ поддерживает несколько экранов и несколько языков с кодировкой UTF-8. Она включает функции для горизонтальных, вертикальных и выпадающих списков, просмотр или ввод одно- или многострочного текста, флажки, кнопки, ползунки и полосы прокрутки. Обеспечивающий работу фреймворк позволяет использовать очереди событий и сигналы, управление окнами и экранами просмотра, а также рисование.

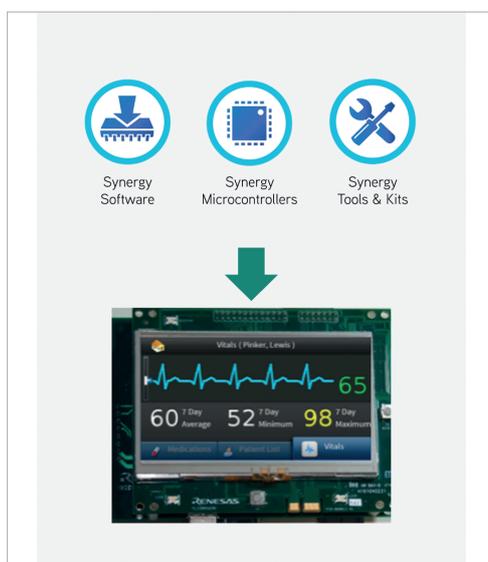


Figure 2-6: GUIX™ allows the easy creation of compelling GUIs

GUIX™ имеет специальное приложение для рабочего стола под названием GUIX Studio™, которое предоставляет полную среду проектирования экрана WYSIWYG (What You See Is What You Get— что видишь, то и получишь), где разработчики могут создавать графические элементы методом перетаскивания (drag-and-drop) для создания привлекательных пользовательских интерфейсов, которые могут запустить их исполнения на рабочей станции Windows® вместе в GUIX Studio™, что позволяет быстро и легко продемонстрировать и оценить концепции пользовательского интерфейса.

Как только пользовательский интерфейс будет завершен, GUIX Studio™ сгенерирует код на C, совместимый с библиотекой GUIX™, готовый к компиляции для микроконтроллеров. Разработчики могут использовать в приложении предварительно созданные шрифты, а шрифты могут быть сгенерированы в монохромных или сглаженных форматах, используя встроенный инструмент создания шрифтов, который поддерживает любой набор символов, включая символы Unicode для многоязычных приложений. В GUIX Studio™ можно по умолчанию настраивать цвета и стили рисования виджетов GUIX™, что позволяет разработчикам настраивать внешний вид своего приложения под свои нужды.

Как всегда, функции библиотеки вызывается через интуитивно понятные, читаемые и высокофункциональные вызовы API. Настраивать приложения на основе GUIX™ легко: нужно включить файл `gx_api.h` в исходный код приложения и связать объекты с библиотекой GUIX.

2.6.3 USBX™

USBX™ — это компактный и высокопроизводительный стек от Express Logic для USB контроллера в режиме хоста или устройства, для приложений SSP, позволяющий микроконтроллерам Synergy взаимодействовать с USB-устройствами или подключаться к рабочим станциям Windows®, Apple® и Linux через USB. Он полностью интегрирован с ThreadX® и доступен для всех поддерживаемых микроконтроллерах Synergy.

Стек включает поддержку режимов передачи данных с низкой скоростью (1,5 Мбит/с), полной скоростью (12 Мбит/с) и высокой скоростью (480 Мбит/с), а также соответствует USB 1.1 и USB 2.0. Он может действовать в двух режимах: хост и устройство.

Режим хоста USBX™ используется, когда приложение требует связи с некоторыми USB-устройствами, такими как USB-клавиатура, USB-принтер или USB-флеш-диск, это означает, что периферийные устройства действуют как подчиненные. В этом режиме есть два основных компонента: основной USB-стеки USB-контроллер.

Основной стек USBX™ отвечает за обнаружение подключения и извлечения устройств, а также всех протоколов, доступных для USB (control, bulk, interrupt and isochronous). Основной стек обеспечивает правильное обнаружение и настройку устройства, а его механизм plug-and-play (включи и работай) ищет драйвер класса USB, отвечающий за это устройство.

USB-контроллер поддерживает основные стандарты USB, такие как OHCI и EHCI. Это дает возможность для создания многопользовательского хост контроллера. Драйвер(ы) класса USB отвечает за управление каждым USB-устройством после его определения. Поддерживаются несколько стандартных классов USB, таких как CDC, HID, HUB, STORAGE, AUDIO или PRINTER, но также можно использовать собственные классы. Режим хоста USBX™ может поддерживать каскадные хабы, несколько устройств конфигурации и составные устройства.

В режиме устройства USBX™ используется, когда приложение должно взаимодействовать с рабочим столом Windows®, Apple® или Linux. В этом случае микроконтроллер Synergy считается устройством USB или вспомогательным устройством. Архитектура для стороны устройства аналогична стороне хоста, хотя и упрощена. Основной стек устройства USBX™ отвечает за обработку запросов на эnumерацию со стороны хоста USB. Класс драйвера USB-устройства отвечает за управление USB-устройством, как только эnumерация была проведена хостом. В режиме устройства USBX™ может поддерживать сложную инфраструктуру устройства, включая несколько конфигураций, несколько интерфейсов и несколько альтернативных настроек. Как и в режиме USB-хоста, в библиотеку включена поддержка нескольких стандартных классов USB, таких как CDC, HID, STORAGE, PIMA или RNDIS.

Работа с библиотекой USBX™ очень проста, и первая рабочая версия может быть получена за очень короткий промежуток времени. Все, что нужно сделать, это добавить и настроить задачи (поток), очередь и четыре модуля в Synergy Configurator в e² studio. Поток — набор действий, выполняющий определенные задачи. В главе 10 будет предложено упражнение, которое даст почувствовать вам эту простоту самостоятельно.

2.6.4 NetX™/NetX Duo™

В NetX™ и NetX Duo™ от Express Logic реализовано приложение протокола TCP/IP для SSP. NetX™ обеспечивает оптимизированный стек TCP/IP, совместимый с IPv4, в свою очередь NetX Duo™ обеспечивает применение двух стеков IPv4 и IPv6, что является перспективным решением.

NetX™ и NetX Duo™ имеют уникальную архитектуру Piconet™, в которую включены только те сервисы и протоколы, которые фактически используются. В сочетании с zero copy API-интерфейсом, который исключает циклы процессора, осуществляющие обычно перемещение данных в буферы пользователей и из них, они идеально подходят для приложений, требующих сетевого подключения, поскольку приложения могут использовать освобожденные циклы процессора для более полезных задач. NetX Duo™ соответствует стандартам RFC, проверенным набором тестов отраслевым стандартом IxIA IxANVL™, и получил сертификат соответствия IPv6-Ready Logo, как доказательство того, что он прошел тесты на совместимость, которые администрируются и проверяются на Форуме IPv6.

В дополнение к zero copy API NetX™ также предоставляет API совместимый со стандартом BSD для приложений с устаревшим BSD кодом. Приложение может создавать любое количество пулов пакетов в любом количестве областей памяти. Эти zero copy пакеты могут быть связаны с пакетами из одного и того же пула или даже с другим пулом для обеспечения большей пропускной способности больших полезных нагрузок.

Базовые пакеты UDP проходят через NetX™ и NetX Duo™ без какого-либо переключения или копирования или системного контекста. Они доставляются непосредственно в ожидающие потоки. Эта технология называется UDP Fast Path™ Technology.

NetX™ и NetX Duo™ включают протоколы сетевой конфигурации, такие как DHCP (клиент/сервер) и SNMP (протокол сетевого времени), службы доменных имен, такие как DNS, mDNS, DNS-SD и NAT и передачу электронной почты с помощью POP3 и SMTP. Также доступно HTTP web сервер и сетевое управление, и протоколы подключения PPP, Telnet, FTP и TFTP. NetX Duo™ получил сертификаты безопасности в соответствии с IEC 61508 SIL 4, IEC 62304 Class C, ISO 26262 ASIL D и EN 50128 SW-SIL4.

Для NetX Duo™ доступны дополнительные компоненты, такие как HTTPv6, FTPv6, DNSv6, Telnetv6 и DHCPv6 (только клиент). Решения высокого уровня, такие как SSL / SSH / TLC, доступны в качестве компонентов расширенного программного обеспечения. Информацию о них можно найти в Галерее Synergy в Интернете.

2.7 Операционная система реального времени ThreadX®

ThreadX®, созданная ExpressLogic, представляет собой операционную систему реального времени (OSPB) для семейства микроконтроллеров Synergy. Она была разработана специально как для высокопроизводительных, так и для графических приложений, а также для встроенных систем с ограниченной памятью и специальными требованиями с точки зрения детерминизма. Она требует небольшой размер флэш-памяти (менее 2 КБ на микроконтроллерах серии S3, S5 или S7), небольшой объем ОЗУ (минимум <1 КБ для ОЗУ ядра) и короткое время переключения контекста (0,7 мкс на микроконтроллере SynergyR7FS7G2).

В качестве многозадачной OSPB ThreadX® использует несколько передовых алгоритмов планирования, обеспечивает трассировку событий в режиме реального времени, передачу сообщения и прерывание управления, а также обеспечивает множество других сервисов и является полностью детерминированной. Она имеет проверенную репутацию с более чем 5,5 миллиардами развертываний на рынках потребителей, медицинской электроники и промышленной автоматизации и отвечает многочисленным стандартам безопасности и качества.

Среди других дополнительных функций можно назвать архитектуру picokernel™, планирование Preemption-Threshold™, Event-chaining™ и богатый набор системных сервисов. Она также поддерживает промежуточное ПО X-Ware™ с сервисами, которые требуются различным компонентам.

2.7.1 Почему используется ОСРВ?

Не так давно жизнь программиста-разработчика встроенной системы была сравнительно простой. Большинство систем состояло из простого основного цикла, в котором одна бесконечная фоновая задача выполнялась в бесконечном цикле. Он выполнял такие задачи, как считывание входов с выводов ввода/вывода (например, состояние кнопки), выполнение определенных вычислений, таких как умножение измеряемого напряжения и тока на потребление энергии, обновление дискретных выходов и т. д.

Несколько периферийных устройств, подключенных к микроконтроллеру, выдавали прерывание, чтобы уведомить центральный процессор о том, что, возможно, новый результат преобразования из преобразователя данных доступен и должен быть прочитан, или что порт RS-232 завершил передачу последнего сообщения и ожидает новых данных. Эти асинхронные события традиционно обрабатываются внутри процедур обслуживания прерываний. И если прерывание не происходило, программное обеспечение находилось на холостом ходу в фоновом цикле, ожидая его появления.

Поскольку встроенные системы становятся все более сложными, имея более высокую возможность подключения к внешнему миру или больше пользовательских интерфейсов для обслуживания и выполнения большего количества задач, описанную выше настройку трудно поддерживать, а операционная система становится не только чем-то предпочтительным, но и необходимым. Зачем? Время отклика фонового цикла было трудно предсказать и определенно не было детерминированным, поскольку на него влияло дерево решений внутри кода и даже изменялось после внесения новых дополнений в программное обеспечение. Другим недостатком было то, что все задачи (или «потоки») в фоновом цикле имели одинаковый приоритет, так как код выполнялся последовательно, что означает, что считывание кнопки и вычисление мощности в приведенном выше примере всегда выполнялись один за другим, даже если кнопка была нажата несколько раз. Другими словами, некоторые из этих событий могли быть пропущены. Система должна быть более отзывчивой и более предсказуемой. Она должна быть в режиме реального времени.

Увидев явную потребность в операционной системе, появляется следующий вопрос: почему бы не использовать готовые ОС, такие как Windows® или Linux? Они являются частью нашей повседневной жизни и, похоже, хорошо выполняют свою работу. Есть несколько причин не использовать их. Прежде всего, они предлагают слишком много функций, не требуемых во встроенной системе, и они недостаточно гибко настраиваются. Во-вторых, им нужно больше ресурсов и пространства памяти, больше, чем доступно в встроенном приложении с ограниченными ресурсами. И последнее, но не менее важное: их временная неопределенность еще слишком велика, чтобы соответствовать требованиям систем реального времени.

Все это означает, что существуют особые требования к ОС, работающим на наших встроенных системах. Первым и самым важным требованием является предсказуемость временного поведения. ОСРВ должна быть детерминированной, что означает, что верхние пределы времени блокировки, такие как время, в течение которого прерывания отключены, должны быть известны и доступны разработчику.

Второе требование заключается в том, что ОСРВ должна управлять диспетчером задач. Она должна знать о временных ограничениях и предельных сроках, и она должна предоставлять услугу контроля задержек с высоким разрешением. И, наконец, операционная система должна быть быстрой, небольшой и настраиваемой.

Еще одной важной функцией, предоставляемой ОСРВ, является управление потоками, позволяющее выполнять квазипараллельные задачи на микроконтроллере с помощью потоков (поток можно понимать как легкий процесс), заботясь о состоянии потоков, о очередности процессов, и позволяя применять превентивные потоки и обработку прерываний. Также есть и другие сервисы, предоставляемые приложению, это планирование, синхронизация потоков, межпоточная связь, совместное использование ресурсов и часы реального времени в качестве ссылки на время.

В то время как многие разработчики программного обеспечения отказываются от использования ОСРВ, потому что они боятся сложности, операционные системы реального времени предлагают множество преимуществ, таких как повышенная

чувствительность к событиям в реальном времени, приоритезация и легкое добавление потоков, сокращение времени разработки сразу, как только первые шаги будут освоены, а сервисы добавлены в приложение.

Как и во всех хороших вещах в жизни, есть, конечно, некоторые недостатки в использовании OCPB: для этого требуется дополнительная оперативная память и флэш-память, и каждому потоку нужен дополнительный стек, что еще больше увеличивает потребность в памяти. Стоимость OCPB - еще один фактор, но это не нужно в нашем особом случае: при использовании платформы Synergy все включено. И простая в использовании API ThreadX® как часть SSP значительно снижает ресурсы, затраченные на освоение!

2.7.2 Основные характеристики ThreadX®

Рассмотрев преимущества использования OCPB, настало время взглянуть на основные функции ThreadX®. Она мала по размеру кода, с очень небольшим размером памяти RAM и оптимизирована для очень быстрой работы и низких программных накладных расходов, в основном для ресурсов микроконтроллера.

Расширенные алгоритмы диспетчеризации задач в реальном времени и эффективные многозадачные подпрограммы обеспечивают циклическую диспетчеризацию и временную привязку, а также приоритетную и Preemption-Threshold™ диспетчеризацию. Для приоритезации потоков доступно до 1024 уровней приоритета, причем число приоритетов по умолчанию равно 32.

Ресурсы во встроенной системе в основном ограничены ресурсами времени и памяти. ThreadX® предоставляет несколько мощных инструментов для управления ими. Память должна распределяться быстро и детерминированным образом, поэтому SSP обеспечивает возможность создания и управления любым количеством пулов, состоящих из блоков фиксированного размера. Поскольку блоки в пуле имеют фиксированный размер, фрагментация памяти не является проблемой, но размер блока памяти должен быть таким же или больше, чем самый большой запрос на память из приложения, иначе распределение не будет выполнено. Но случай, когда размер блока был выбран достаточно большой для такого сценария, может привести к растрачиванию памяти, если запросы поступают с разными размерами блоков. Обходной путь для этого - создание нескольких пулов блоков памяти, содержащих блоки памяти различного размера. Этот недостаток компенсируется быстрым распределением и уничтожением этих блоков, поскольку эти распределения выполняются по простому списку доступных блоков. Это обеспечивает самую быструю связанную обработку списка ссылок и может сохранять фактический блок памяти в кеше.

ThreadX® позволяет использовать не только пулы блоков с фиксированными размерами, но также создавать пулы с несколькими байтами, которые ведут себя аналогично стандартной куче C. Память распределяется только с нужным размером в байтах, основанная на принципе первого подходящего (first-fit). Недостатком этого является то, что, подобно кучам в пулах байтов, они легко фрагментируются, создавая несколько недетерминированное поведение.

Другими ресурсами, управляемыми OCPB, являются таймеры приложений, позволяющие создавать неограниченное количество таймеров для задач пользователя. Эти таймеры приложений могут быть запущены в трех режимах: одноразовый, периодический и относительный. Одноразовый таймер будет вызывать пользовательскую функцию только один раз после истечения таймера, в то время как периодический таймер вызывает функцию пользователя повторно через определенный интервал. Относительный таймер основан на непрерывно увеличивающемся 32-разрядном тактовом счетчике. Все задержки таймера выражаются в тактах (tick), 1 такт равен 10мс, но можно и задать другие настройки. SSP управляет активацией и окончанием таймера без линейного поиска, что уменьшает программные накладные расходы в приложениях, использующих таймер, таких как связи и межзадачное управление.

Синхронизация потоков и обмен данными между задачами - еще одна важная тема во встроенных системах. ThreadX® предоставляет несколько механизмов для этого и делает эту задачу очень простой. Семафоры и взаимные исключения (мьютексы) помогают предотвратить инверсию приоритета и позволяют создавать неограниченное количество объектов.

Также доступны сложные функции обратного вызова и Event-chaining™, а также дополнительное приоритетное наследование. Кроме того, поток ThreadX® может приостанавливаться либо по принципу FIFO, либо в порядке приоритета, избегая проблем с недостатком времени для обработки в потоках.

Флаги событий - еще одна функция синхронизации потоков. Группа флагов событий состоит из 32 битов, где каждый бит представляет собой другое логическое событие, и потоки могут ждать подмножество этих битов. Флаги событий также поддерживают Eventchaining™. Как и в случае семафоров, можно создать неограниченное количество объектов и получить информацию о производительности флагов во время выполнения.

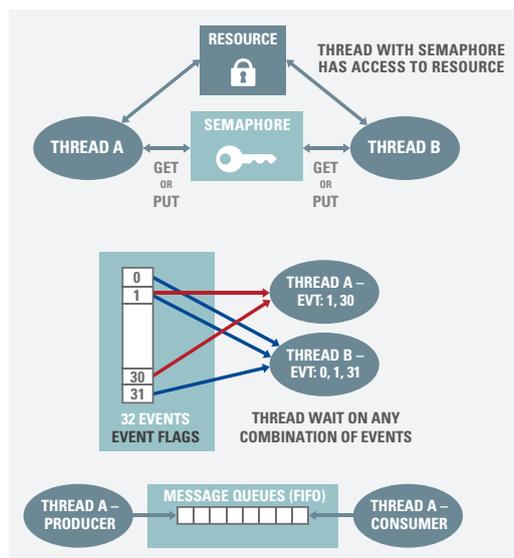


Figure 2-7: ThreadX® synchronization and communication features

Последняя функция синхронизации, которая будет упомянута здесь, — очереди сообщений. Используя модель производитель-потребитель для межзадачной коммуникации, ThreadX® поддерживает размеры сообщений от 1 до 16 четырехбайтовых слов, причем почтовые ящики являются особым случаем очереди сообщений с размером 1. Потоки приложений могут регистрировать функцию обратного вызова для дальнейших уведомлений. Как и в случае с флами событий, во время выполнения доступна информация о производительности.

Для целей отладки реализована очень важная функция: встроенная функция отслеживания событий, поддерживаемая визуализатором TraceX® на рабочей станции, который позволяет просматривать последовательность различных событий изнутри e²studio.

И, наконец, ThreadX® совместим с Misra-C: 2004, Misra-C2012 и пре-сертифицирован по следующим стандартам:

- | | |
|--------------------|----------------------|
| ‡ IEC 61508 | ‡ IEC 60730-1 H |
| ‡ IEC 62304 | ‡ UL 60335-1 R |
| ‡ UL 60730-1 H | ‡ IEC 60335-1 R |
| ‡ CSA E60730-1 H | ‡ UL 1998 |
| ‡ ISO 26262 ASIL D | ‡ EN 50 128 SW-SIL 4 |

Вопросы, которые следует вынести из этой главы:

- Программный пакет Synergy (SSP) построен в виде слов, и к ним можно получить доступ с помощью простых API-вызовов.
- Фреймворки приложений устраняют необходимость писать базовый код.
- ThreadX® и промежуточное ПО X-Ware™ являются хорошо зарекомендовавшими себя компонентами, что позволяет создавать полнофункциональные приложения.

3 ЗНАКОМСТВО С API, ВСТРОЕННЫМ В ПРОГРАММНЫЙ ПАКЕТ SYNERGY

Что вы узнаете в этой главе:

- Какие различные уровни имеет пакет программного обеспечения Synergy и как они могут быть доступны.
- Подробности о программных интерфейсах приложениях их использовании.

Во время работы над программным пакетом Synergy (Synergy Software Package -SSP) разработчики ориентировались на простоту дальнейшего использования. SSP предлагает потрясающие функциональные возможности, и одновременно является довольно простым в использовании, поскольку разработка программных интерфейсов приложений (Application Programming Interface - API) очень простая и всесторонняя. API инкапсулирует сложность SSP, но все же предоставляя программисту полный контроль над функциональностью. Даже программирование сложной задачи, такой как передача данных по USB, сводится к нескольким строкам кода и может быть достигнуто без необходимости читать тома руководств или изучать каждую деталь набора регистров заданного периферийного устройства микроконтроллера. Это существенное упрощение задач разработчиков, так как они могут сосредоточиться на наборе функций своего приложения, вместо того, чтобы писать низкоуровневый код, который не добавляет никакой ценности в разработку, но занимает много времени при написании и тестировании. Давайте посмотрим на некоторые детали API!

3.1 Обзор API

Приложения могут получить доступ ко всем функциям в SSP посредством интуитивно понятных и всеобъемлющих вызовов API, независимо от того, в каком из слоев находится необходимая функциональность. Это позволяет легко и просто написать код, который легко понять, он прост в обслуживании и безболезнен для портов случае, например, если для определенной задачи необходимо использовать другое периферийное устройство микроконтроллера, что сейчас во время разработки происходит всё чаще.

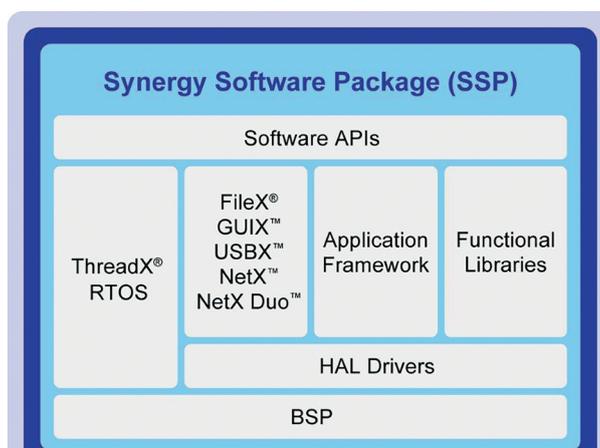


Figure 3-1: The different layers of the Synergy Software Package

Всё в том же примере, для осуществления перехода от встроенной SPI-периферии к SPI-функциям встроенного SCI-порта потребуется только одно простое изменение, если используется SSP: в Synergy Configurator во вкладке «Threads(Потоки)» драйвер SPI Driver `r_spi` необходимо заменить на драйвер SPI Driver `r_sci_spi`. Поскольку оба драйвера будут использовать тот же экземпляр с именем `g_spi0`, нет необходимости что-либо менять в коде приложения.

Другая возможность - использовать устройство SPI на sf_spi. В этом случае только драйвер g_spi0 драйвера r_spi внутри фреймворка должен быть заменен на драйвер r_sci_spi внутри конфигулятора Synergy. Все вызовы API фреймворков g_sf_spi_device0 остаются неизменными. Нет необходимости изменять код. Все, кто, как и я, уже имел в прошлом задачу изменить периферийные устройства в существующем коде, действительно оценят эту простоту, предлагаемую платформой Synergy. Это означает, что внесение изменений в дизайн после запуска проекта не создает больших проблем и их очень легко реализовать, также это сокращает время тестирования, необходимое для минимального минимума.

Архитектура различных слоев показана на Рисунке 3-1: внизу находится микроконтроллер Synergy, на котором находится Пакет Поддержки Платформы(BSP), который отвечает за то, чтобы микроконтроллер был сброшен в основное приложение и обеспечил другие сервисы для программного обеспечения на уровне выше. Следующий уровень — это слой аппаратных абстракций(HAL), который изолирует разработчика от непосредственного взаимодействия с регистром микроконтроллера и делает программное обеспечение на уровне выше HAL более переносимым для всей платформы Synergy.

В над HAL находятся фреймворки приложений, функциональные библиотеки и промежуточное программное обеспечение X-Ware™, а также OCPB. В то время как фреймворки приложений выполняют часто используемые системные службы, такие как обмен данными, воспроизведение аудио или емкостное сенсорное считывание, функциональные библиотеки содержат специализированное программное обеспечение, например, для обработки цифровых сигналов или функций, связанных с безопасностью. В самой верхней части находится пользовательская программа, которая вызывает слои ниже через API. Подробная информация о слоях и частях SSP доступна в главе 2.

Каждый из разных уровней осуществляется через вызовы API напрямую или в виде стеков. Например, звуковое приложение может вызывать структуру воспроизведения аудио в приложениях, в которой используется драйвер I²S (Inter-IC Sound) и два контроллера Data Transfer Controller (DTC) для обеспечения доступа к внешнему аудио преобразователю для чтения/записи, преобразователь подключен к микроконтроллеру через интерфейс I²S и общий таймер generalPWM. Без условно, конечное приложение может также напрямую обращаться к драйверам HAL и BSP, но использование фреймворков приложений намного проще, поскольку в этом случае не требуется подробное знание вышеописанных блоков.

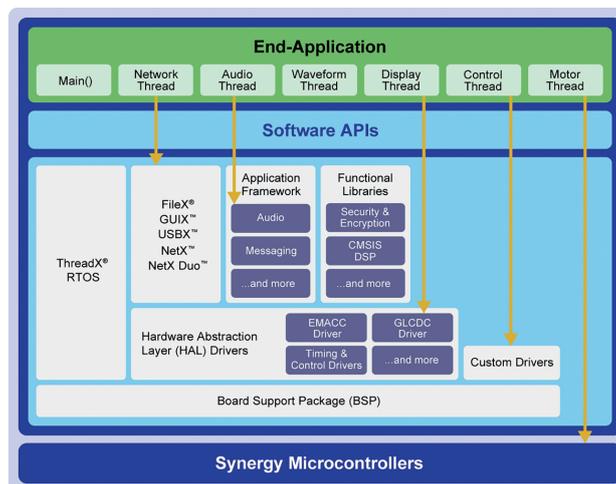


Figure 3-2: User programs can have direct access to the different layers of the SSP, but using the Application Frameworks or the Functional Libraries is much easier

3.2 Синтаксис API

Прежде чем глубоко погрузиться в детали API, мы должны кратко рассмотреть соглашения об именах различных API и файлов. Это необходимо, чтобы не только программирование приложения стало легкой задачей, но и для того чтобы облегчить понимание любого кода от другого программиста или вашего собственного кода через несколько месяцев, поэтому четкая структура, предоставляемая SSP будет полезна.

- **R_BSP_xxx**: префикс для общей функции BSP, например. `R_BSP_VersionGet ()`.
- **BSP_xxx**: префикс для определений BSP, например. `BSP_CODE_VERSION_MAJOR`.
- **SSP_xxx**: префикс для общих определений SSP, в основном коды ошибок, например. `SSP_SUCCESS`.
- **g_<interface>_on_<instance>**: имя постоянной глобальной структуры экземпляра, реализующего API интерфейса. Например `g_spi_on_rspi` для функции HAL Layer или `g_sf_spi_on_sf_spi` для функции Application Frameworks.
- **r_<интерфейс>_api.h**: имя заголовочного файла интерфейсного модуля, например. `r_spi_api.h`.
- **r_<ИНТЕРФЕЙС>_<Function>**: имя API-интерфейса HAL, например. `R_RSPI_Open()`.
- **sf_<framework>_<function>**: модуль уровня фреймворка приложения.
- **sF<FRAMEWORK>_xxx**: API-интерфейс уровня фреймворка приложения, например `SF_SPI_Open()` или `SF_AUDIO_PLAYBACK_Open()`.

Соглашения об именовании промежуточного программного обеспечения X-Ware™ схожи. Они построены следующим образом: <ID>_<существительное>_<глагол>. ID — это модуль, существительное - это объект (таймер, семафор и т. д.), а глагол — действие, которое необходимо выполнить (создание, закрытие, получение, ...). Например `tx_queue_create()`, который создает очередь сообщений в ThreadX®.

Ниже приведен краткий обзор нескольких ID промежуточного программного обеспечения X-Ware™ и OCPB ThreadX®:

- **tx**: функции, связанные с ThreadX®, например, `tx_thread_create()`.
- **nx**: функции, связанные с NetX™, например, `nx_packet_copy()`.
- **nxd**: функции, связанные с NetXDuo™, например, `nxd_ipv6_enable()`.
- **gx**: функции, связанные с GUIX™, например, `gx_display_create()`.
- **fx**: функции, связанные с FileX®, например, `fx_directory_create()`.
- **ux**: функции, связанные с USBX™, например, `ux_device_stack_initialize()`.

Помимо понимания формального синтаксиса API, очень полезно согласовать пару определений. Зачастую, каждый из нас имеет несколько иное понимание некоторых терминов, что вызывает путаницу. Итак, вот список терминов, используемых в книге:

- **Модули**: модули могут быть периферийными драйверами, обычным исполняемым кодом или чем-то другим, и являются строительными блоками SSP. Модули обычно являются независимыми блоками, но они могут зависеть от других модулей. Приложения могут быть созданы путем объединения нескольких модулей, чтобы предоставить пользователю необходимые функции.
- **Экземпляр модуля**: одиночная и независимая конфигурация модуля.
- **Интерфейсы**: интерфейсы — это способ, при помощи которого модули предоставляют общие функции. Через них модули, которые придерживаются одного и того же интерфейса, могут использоваться взаимозаменяемо. Представьте интерфейс как контракт между двумя модулями, где модули соглашаются работать вместе, используя информацию, согласованную в контракте.
- **Экземпляры SSP**: в то время как интерфейсы определяют предоставляемые функции, экземпляры фактически реализуют их. Каждый экземпляр привязан к определенному интерфейсу при помощи перечисления, структуры данных и API прототипов интерфейса. Это позволяет приложению заменять экземпляры, когда это необходимо.
- **Драйверы**: драйвер — это особый тип модуля, который напрямую изменяет регистры в микроконтроллере Synergy.

3.3 Константы, переменные и другие инструменты API

Как уже упоминалось, модули являются строительными блоками SSP. Они могут выполнять разные задачи, но все модули разделяют основную концепцию предоставления функциональности на уровнях выше и требуют функциональности из более низкоуровневых слоёв, как показано на рисунке 3-3. Итак, самое простое приложение, использующее SSP, состоит из одного модуля кодом пользователя на вершине.

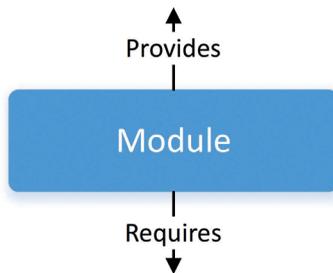


Figure 3-3: Modules are built in a way that they provide functionality to the caller and require functionality from the level below

Модули могут быть наложены друг на друга, создавая стек SSP. Процесс наложения выполняется путем сопоставления того, что один модуль предоставляет с тем, что требуется другому модулю. Например, для модуля фреймворка воспроизведения звука среди прочих требуется интерфейс передачи данных, который может быть предоставлен модулем драйвера контроллера передачи данных (DTC). Его код не включён в модуль воспроизведения звука намеренно, для того, чтобы модуль DTC повторно использовался другими модулями, например, драйверами UART или SD-карты.

Возможность наложения модулей дает много преимуществ, так как обеспечивает гибкость, необходимую для уровня приложения. Если бы они были зависимы напрямую от других модулей, переносимость различных пользовательских разработок была бы сложной задачей. Архитектура SSP, в том виде в котором она была разработана, обеспечивает гибкость для замены модулей входов и выходов (например, замена драйвера UART на драйвер SPI) с использованием интерфейсов SSP. Модули, поддерживающие один и тот же интерфейс, могут использоваться взаимозаменяемо. Помните контракт, упомянутый выше? Все модули соглашаются работать вместе, используя информацию, которая была согласована в контракте.

В микроконтроллерах Synergy иногда наблюдается перекрытие функций между различными периферийными устройствами. Например, связь I²C может быть достигнута за счет использования периферийного устройства I²C или использования периферийного устройства SCI в его простом режиме I²C, причем оба периферийных устройства предоставляют разный набор функций. Интерфейсы построены таким образом, что они обеспечивают общие функции, которые ожидало бы большинство пользователей, опустив некоторые из более продвинутых функций, которые, однако, в большинстве случаев все еще доступны через расширения интерфейса.

По меньшей мере три структуры данных включены в каждый интерфейс SSP: структура управления, зависящая от модуля с именем `<interface>_ctrl_t*`, которая используется как уникальный идентификатор для использования модуля; структура конфигурации с именем `*<interface>_cfg_t*` в качестве входных данных для модуля для начальной конфигурации во время вызова `open()`; и структуры экземпляра, состоящий из указателя на структуру управления, указателя на структуру конфигурации и указателя на структуру API. Название этой структуры стандартизировано как `g_<interface>_on_<instance>`, например `g_spi_on_spi`. Сама структура доступна для использования посредством объявления `extern` в файле заголовков экземпляров, например, `r_spi.h`. Несмотря на то, что в случае простого драйвера эти структуры могли быть объединены, их создание именно таким образом расширяет функциональность интерфейсов.

Все интерфейсы включают структуру API, содержащую указатели на функции всех поддерживаемых функций. Например, структура для цифро-аналогового преобразователя (DAC) содержит указатели на такие функции, как open, close, write, start, stop и versionGet. Структура API - это тот инструмент, который позволяет заменять модули на другие модули, которые являются экземплярами того же интерфейса.

Как только событие произошло, то через функции обратного вызова модули подают об этом сигнал в пользовательское приложение. Примером такого события может быть байт, полученный по каналу UART. Обратные вызовы также необходимы, чтобы пользовательское приложение могло реагировать на прерывания. Они должны быть как можно короче, поскольку они вызываются изнутри процедуры обслуживания прерываний. В противном случае они могут оказать негативное влияние на производительность системы в режиме реального времени.

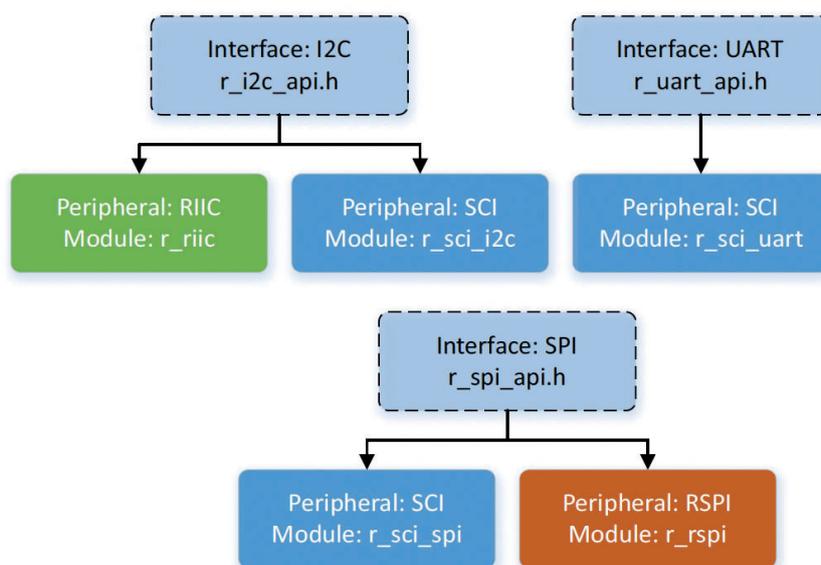


Figure 3-4: On Synergy MCUs some peripherals will have a one-to-one mapping between the interface and instance, while others will have a one-to-many mapping

В то время как интерфейсы определяют предоставляемые функции, экземпляры фактически реализуют их. Каждый экземпляр привязан к определенному интерфейсу и использует перечисления, структуры данных и прототипы API из интерфейса. Это позволяет приложениям, которые используют интерфейс для замены экземпляра, когда это необходимо, сэкономить много времени после изменения кода или используемых периферийных устройств. На микроконтроллерах Synergy некоторые периферийные устройства, например, I2C, будут иметь взаимно однозначное сопоставление (оно только сопоставляется с интерфейсом I2C), в то время как другие, такие как SCI, будут иметь сопоставление один-ко-многим (он реализует три интерфейса: I2C, UART, SPI), см. Рисунок 3-4 для графического представления соответствия.

3.4 Использование API

После теоретической справки наконец настало время взглянуть на то, как легко работать с пакетом программного обеспечения Synergy. Для этого в качестве примера мы будем использовать простой UART и объясним, как использовать API и где найти информацию о разных элементах. Если вы хотите параллельно выполнять эту работу на своем ПК, вы можете вернуться к этому разделу после главы 5.

Первым шагом в использовании модуля SSP всегда является выбор правильного интерфейса для требуемых функций. В нашем случае мы хотим общаться через интерфейсный блок UART в интерфейсе последовательного соединения (SCI), который доступен на всех сериях микроконтроллеров Synergy, поэтому внутри e² studio мы выбираем драйвер UART

на `r_sci_uart` в Synergy Configurator. Это делается, сначала выделив HAL/Common thread на левой стороне страницы Threads, а затем выбрав New Stack → Driver → Connectivity → UART Driver on `r_sci_uart` с правой стороны под HAL/Common Stacks. После добавления дополнительные требуемые записи будут добавляться автоматически до уровня, на котором пользователю необходимо сделать выбор или принять решение о функциональности. В случае UART будет добавлен экземпляр каждого из `g_transfer0` и `g_transfer1` Transfer Drivers on `r_dtc` Event SCIO TXI (RXI). После добавления драйвер UART (`g_uart0` UART Driver on `r_sci_uart`) будет отображаться красным цветом, указывая на отсутствие настроек.

Наведение курсора на такой элемент покажет требуемую функциональность. В случае вышеприведенного примера, прерывание приема, передачи и ошибки необходимо включить в разделе модуля `g_uart0` UART Driver on `r_sci_uart` в окне свойств драйвера. При необходимости экземпляр модуля, который по умолчанию называется `g_uart0` (число зависит от выбранного физического интерфейса), может быть назван пользовательским именем (например, `my_uart`). Это добавляет еще один уровень абстракции. Если название разработчиком программного обеспечения выбрано правильно, это может сделать код более переносимым и лучше поддерживаемым, а не говоря уже о том, что он станет более читаемым.

Окно «Свойства» также позволяет настраивать параметры, такие как скорость передачи, `parity` и т. д. последовательного порта. Конфигурация выполняется в графическом интерфейсе пользователя, и нет необходимости инициализировать порт с правильными значениями вручную. Все это делается с помощью Synergy Configurator. После того, как прерывания включены, на вкладке «Threads» больше ничего не нужно делать, и вы заметите, что драйвер `UARTg_uart0` on `r_sci_uart` больше не выделяется красным цветом, так как мы выполнили все необходимые настройки.

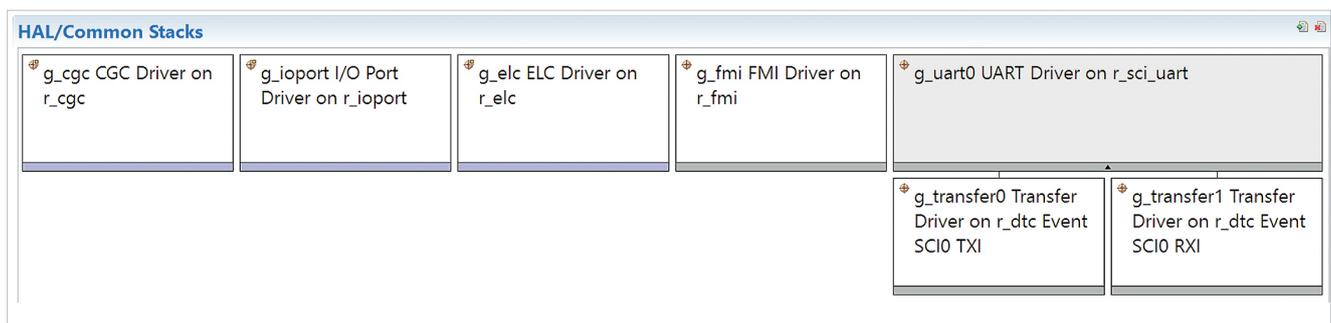


Figure 3-5: The UART stack displayed in e² studio

В настоящее время нет необходимости заботиться о других элементах окна HAL/Common Stacks. Это основные драйверы, необходимые для Пакета Поддержки Платформы для портов ввода/вывода, контроллера связи событий (ELC), контроллера тактирования (CGC) и заводской информации о микроконтроллере (FMI). Драйвер FMI включает в себя общий API для чтения записей из таблицы Factory MCU Information Flash Table, содержащий информацию о функциях и периферийных устройствах, реализованных на используемом микроконтроллере. Все эти драйверы автоматически добавляются конфигуратором.

Чтобы передавать и получать данные через периферийное устройство, остаётся только правильно настроить порты ввода/вывода для SCI UART во вкладке «Pins» в Synergy Configurator. Расширение списка периферийных устройств в левой части представления будет отображать доступные интерфейсы. Дальнейшее расширение Connectivity: дерево SCI позволяет выбрать запись SCIO и отобразится конфигурацию контактов (Pin Configuration) для этого порта. Выбор асинхронного UART в рабочем режиме (Operation Mode) обеспечит правильную настройку контактов.

Сохранение изменений и щелчок по значку «Создать контент проекта» (Generate Project Content) создадут необходимые исходные файлы и будут заполнять структуры управления и конфигурации с именем `uart_ctrl_t*` и `*uart_cfg_t *` соответственно, они будут помещены в файлы заголовков приложения, которые будут включены в код пользователя, откуда они могут быть доступны с помощью указателей, предоставляемых интерфейсом `g_uart0`.

С завершением установки и созданием SSP-файлов взаимодействие с интерфейсами упрощается. В случае примера с периферийным устройством UART, оно будет открыто и настроено, при вызове следующей функции:

```
err = g_uart0.p_api->open(g_uart0.p_ctrl, g_uart0.p_cfg);
```

где `g_uart0` является именем данного экземпляра при его настройке в окне свойств; `p_api`, `p_ctrl` и `p_cfg`, являются указателями на структуры, сгенерированные конфигуратором. Аналогично, для записи в UART потребуется вызов функции ниже:

```
err = g_uart0.p_api->write(g_uart0.p_ctrl, p_src, numBytes);
```

где `p_src`, является указателем на пользовательский массив, содержащий данные типа `uint8_t`, которые должны быть отправлены по UART, а `numBytes` — количество байтов, которое должно быть записано как 32-разрядное целое число без знака. С точным синтаксисом можно ознакомиться в справочной главе по интерфейсам HAL в Руководстве пользователя SSP или с помощью функции интеллектуальной документации `e2studio`. Этот короткий пример — очень хорошая демонстрация возможностей программного пакета Synergy. Для настройки порта необходимо написать только две строки кода и передать через него массив данных. Все остальное уже создано и выполняется конфигураторами и драйверами SSP. Не нужно читать руководство по микроконтроллеру и узнавать о различных регистрах, и я считаю, что это экономит много времени при разработке приложения.

Вопросы, которые следует вынести из этой главы:

- SSP состоит из слоев.
- Каждый слой может быть доступен непосредственно через интерфейсы прикладного программирования (API), но пользователям рекомендуется использовать API фреймворков приложений и функциональные библиотеки, а не напрямую обращаться к нижним уровням.
- Использование фреймворков приложений и функциональных библиотек делает код более переносимым и поддерживаемым.
- Синтаксис API прост и интуитивно понятен.

4 НАЧАЛО РАБОТЫ С НАБОРОМ ИНСТРУМЕНТОВ RENESAS SYNERGY™ PLATFORM

Что вы узнаете в этой главе:

- Где взять и как установить набор инструментов для платформы Synergy.
- Шаги, которые необходимо выполнить во время первого запуска интегрированной среды разработки решений (ISDE).

В этой главе мы загрузим и установим два инструмента, необходимых для работы над проектами программного обеспечения для платформы Synergy: ISDE e²studio и пакет программного обеспечения Synergy (SSP).

В следующих параграфах будут описаны все необходимые шаги, и вы обнаружите, что установка обоих инструментов не требует никаких хлопот, поскольку установщики позаботятся обо всем, что необходимо, как только вы сделаете свой выбор, что и где устанавливать. После завершения установки мы проведем короткую проверку работоспособности, чтобы убедиться, что установка выполнена верно.

В том случае, если вы предпочитаете работать с IAR Embedded Workbench® для Renesas Synergy™ (IAR EW for Synergy), адаптированную версию этой главы вы сможете найти в Интернете.

4.1 Введение в Галерею Synergy

Первым шагом процесса установки является переход в галерею Synergy в Интернете на страницу <https://synergycastle.renesas.com>, которая является удобной единой точкой доступа ко всему, что необходимо для разработки приложения Synergy: среда разработки, компиляторы, инструменты, документация и поддержка. У вас под рукой есть полная экосистема. И не только от Renesas, но и от Express Logic, IAR Systems® или от создателей, проверенных или квалифицированных дополнений программного обеспечения (QSAs).

Чтобы получить доступ, необходимо зарегистрироваться в качестве пользователя. Регистрация проста, содержит только несколько вопросов, на которые необходимо ответить прежде чем будет предоставлен доступ к сайту. В процессе регистрации у вас также будет возможность зарегистрировать свою компанию или организацию. Это даст вам дополнительные преимущества, поскольку первый пользователь, зарегистрировавший компанию, становится Суперпользователем. Суперпользователь компании может создать любое количество лицензий на программное обеспечение для разработки и/или производства. Если ваша компания уже зарегистрирована, попросите Суперпользователя вашей компании добавить вас в базу данных пользователей. Если вы не знаете, какую регистрацию использовать, не беспокойтесь, регистрация компании может быть выполнена позже. Независимо от того, зарегистрированы ли вы как отдельное лицо или как компания, вы всегда будете иметь полный доступ к сайту.

После регистрации достаточно просто войти в систему, чтобы просмотреть сайт и немедленно загрузить любой пакет программного обеспечения. И все доступно с начальной страницы, поэтому переключение между разными страницами не требуется. Если вам понадобится помощь или у вас есть вопрос, на который веб-сайт не отвечает, есть много вариантов, чтобы связаться с Renesas для получения справки.

Самый простой способ получить почти мгновенную поддержку — использовать функцию Live Chat на странице поддержки. Специалисты Synergy из Renesas обеспечат поддержку 24 часа в сутки 5 дней в неделю. И это особенно замечательно, так как вы не будете разговаривать с колл-центром где-то на этой планете, а с реальным экспертом

по платформе Synergy, знаящим её наизусть. Другие возможности — найти базу знаний или задать более сложные вопросы на форуме Renesas Rulz, обсуждая это с другими разработчиками Synergy. На странице поддержки вы также можете получить доступ к документации, Synergy Explorer или просмотреть Synergy Wiki

4.2 Загрузка и установка интегрированной среды разработки e² studio и пакета программного обеспечения Synergy

Интегрированная среда разработки e² studio содержит все инструменты, необходимые для создания, компиляции и отладки проектов для платформы Synergy. Он называется ISDE (Integrated Solutions Development Environment), поскольку имеет дополнительные компоненты, ориентированные на решение, а также различные плагины, что делает её более мощной. Это особенно актуально для конфигураторов, которые обеспечивают легкий графический доступ к различным аппаратным функциям без необходимости глубокого изучения руководства пользователя, таким как функции блока управления прерываниями (ICU) или модуля системы тактирования. Эти конфигураторы автоматически создадут все необходимые настройки и код инициализации, кроме того, они включают функцию проверки ошибок для обнаружения проблемных настроек уже во время разработки, экономя много часов, потраченных на написание и/или отладку кода, не добавляя полезности к самому приложению. По крайней мере, я был бы более чем счастлив, если бы такой инструмент был доступен, когда я впервые начал писать программное обеспечение для встроенных процессоров. С платформой Synergy намного проще!

В следующих параграфах вы ознакомитесь с различными этапами загрузки и установки e² studio и SSP на рабочей станции Windows®.

4.2.1 Скачивание e² studio и SSP

Первым шагом будет загрузка двух инсталляторов для e² studio и SSP из галереи Synergy:

На домашней странице галереи Synergy сначала выберите «Инструменты разработки» (Development Tools) и e² studio ISDE из меню. Появится домашняя страница e² studio, нажмите на синюю кнопку «Загрузить» (Download) в нижней правой части изображения, что приведет вас прямо к странице «Сведения о загрузке» (Download Details). Но воздержитесь от немедленного начала загрузки. Сначала проверьте минимальные требования к ПК, чтобы убедиться, что инструмент будет работать на вашем компьютере, а затем запустите загрузку, нажав на синюю кнопку загрузки справа от заголовка последней версии. Прочтите и примите появившееся лицензионное соглашение, и начнется загрузка заархивированного инсталлятора. Потребуется некоторое время пока загружается программа установки e² studio. В это время вы можете просмотреть инструкции по установке, расположенные на вкладке «Документация» (Documentation). Они будут полезны позже при установке ISDE.

Загрузка SSP происходит по таким же основным шагам: на любой странице галереи Synergy вверху выберите SSP, чтобы вы попали на домашнюю страницу пакета программного обеспечения. Опять же, выберите кнопку «Загрузить», которая приведет вас к соответствующей странице «Сведения о загрузке». Еще раз проверьте минимальные требования к ПК, чтобы они соответствовали вашей системе. Еще раз, синяя кнопка загрузки справа от заголовка последней версии начнет процесс загрузки. Как и при загрузке e² studio, найдите время, чтобы просмотреть инструкции по установке во вкладке «Документация», и загрузите примечания к релизу и руководство пользователя для пакета программного обеспечения SSP.

4.2.2 Установка инструментов

Как только загрузка будет завершена, можно приступить к процессу установки. В папке загрузки должно быть два файла: Один называется `setup_e2_studio_5_3_0_023.zip` (номер версии может отличаться), содержащий установщик для `e2 studio`, а другой - `SSP_Distribution_1.2.0.zip` (опять же номер версии может отличаться), содержащий установщик пакета программного обеспечения SSP.

Очень важна последовательность установки: сначала нужно установить студию ISDE `e2 studio`, чтобы позже включить правильную интеграцию SSP. Чтобы запустить установку ISDE, сначала необходимо распаковать скачанный архив. Это создаст файл с именем `setup_e2_studio_5_3_0_023.exe`. Дважды щелкните по нему, чтобы начать настройку. Если вы используете Windows 10, в зависимости от параметров безопасности вашей рабочей станции, установка может прекратиться с предупреждением операционной системы, что «Windows защищает ваш компьютер». В этом случае нажмите «Дополнительная информация» и «Выполнить в любом случае», и он будет продолжен. Windows 7 и 8 могут вызвать аналогичное предупреждение, просто нажмите «Да», и установка продолжится.

Следуйте диалоговым окнам установщика. Если ваша рабочая станция находится подключена к прокси-серверу, вы можете настроить ее через `WelcomeScreen` (нижняя правая ссылка), чтобы убедиться, что установка будет работать должным образом. На следующем экране должен быть указан путь установки. Используйте значение по умолчанию или измените его в соответствии с вашими потребностями, но запомните свой выбор. Позже вам понадобится установить SSP в ту же папку. На следующем экране можно выбрать семейства устройств. Обязательно выберите `Renesas Synergy™`.

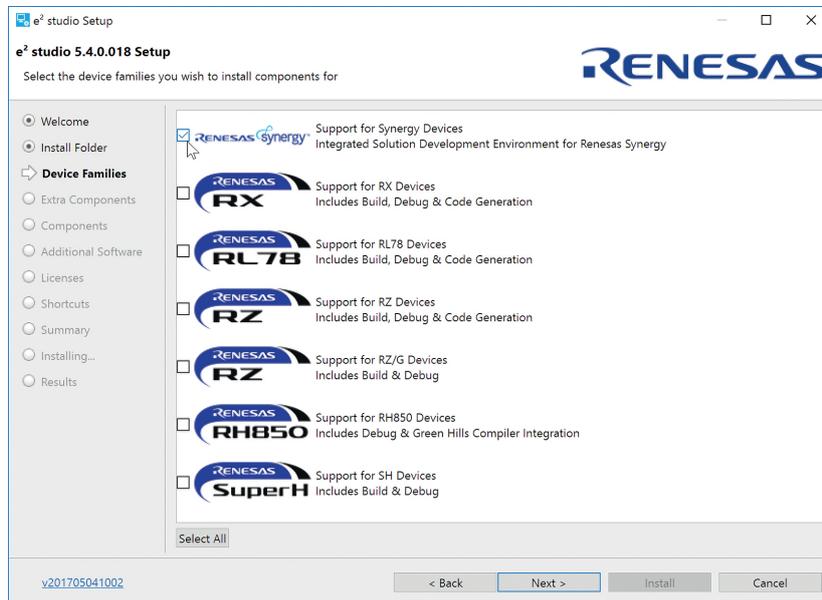


Figure 4-1: Make sure to select the support for Synergy devices

Нажмите «Далее» (Next), и вам будет предложено несколько вариантов дополнительных компонентов для установки. Сделай свой выбор. Хотя большинство вариантов не особенно полезны для `Renesas Synergy`, вы можете установить поддержку SVN и GIT. Если вы еще не определились, не волнуйтесь: вы можете повторно запустить установщик в любое время позже, чтобы добавить или удалить функции, не переходя снова в процесс установки.

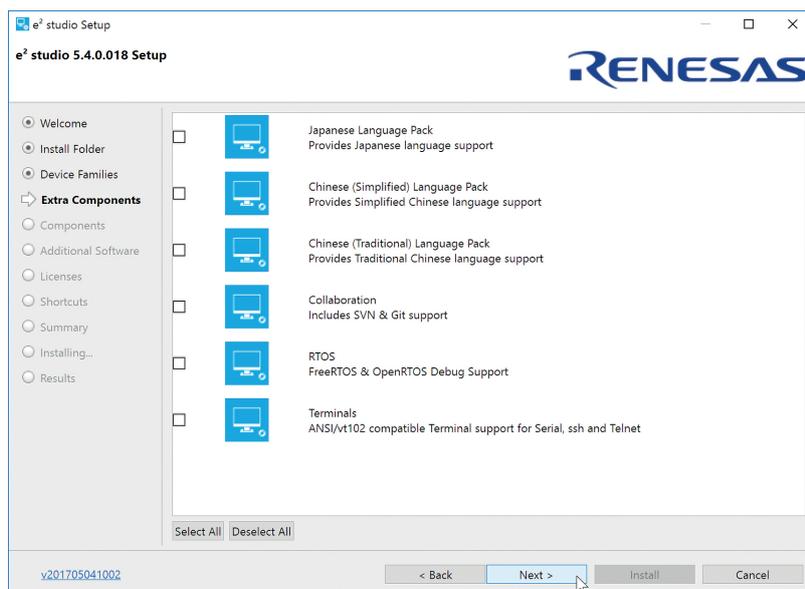


Figure 4-2: Nothing here is essential at the beginning and can be added later

На следующих экранах компонентов (см. Рис. 4-3) всё, что отмечено галочкой выбора потребуется для работы, но вы можете настроить компоненты. Но старайтесь не отменить поддержку Renesas Synergy RZ Debug. Если вы это сделаете, вы не сможете создать конфигурацию отладки и подключиться к целевой плате.

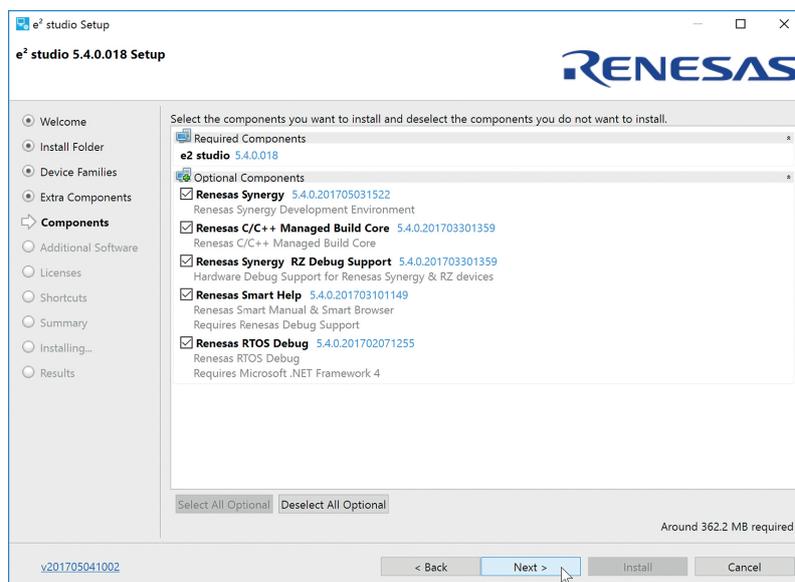


Figure 4-3: You can leave everything selected on this screen

Следующий экран позволит вам выбрать дополнительное программное обеспечение и компилятор (см. Рис. 4-4). Установите флажок рядом с GCC ARM® Embedded 4.9 2015q3 (или более поздней версии), чтобы установить инструменты генерации кода. Следуйте указаниям на следующих экранах, примите условия соглашений об использовании программного обеспечения и начните настройку. Будьте терпеливы, это займет несколько минут. Разрешите установку драйверов устройств Renesas после запроса.

Позже в процессе установки набора инструментов GCC запустится автоматически. Следуйте указаниям установщика и дождитесь завершения установки. При запросе пути установки настоятельно рекомендуется оставить его по умолчанию или записать точное местоположение. Интеграция автоматической привязки будет работать только в том случае, если она была установлена по умолчанию. Если вы изменили путь, вам придется вручную интегрировать его позже.

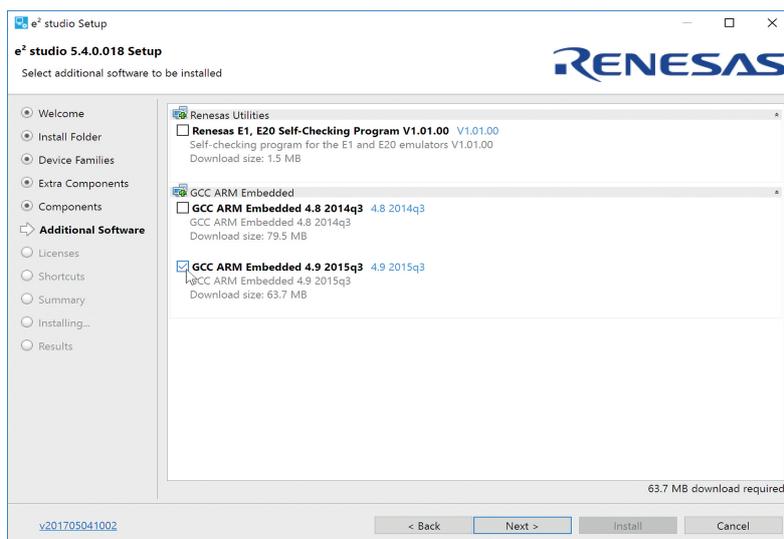


Figure 4-4: It's best to always select the latest compiler version

Последний экран, требующий некоторых заключительных шагов. Установите флажок около пункта Добавить путь в переменную среды (Add path to environment variable), как показано на рисунке 4-5. Это добавит путь набора инструментов к среде Windows и обязательно обеспечит легкую регистрацию набора инструментов e² studio. На этом этапе нет необходимости запускать gccvar.bat, поэтому вы можете снять этот флажок. Нажмите «Готово», а затем закройте инсталлятор e² studio, как только закончите.

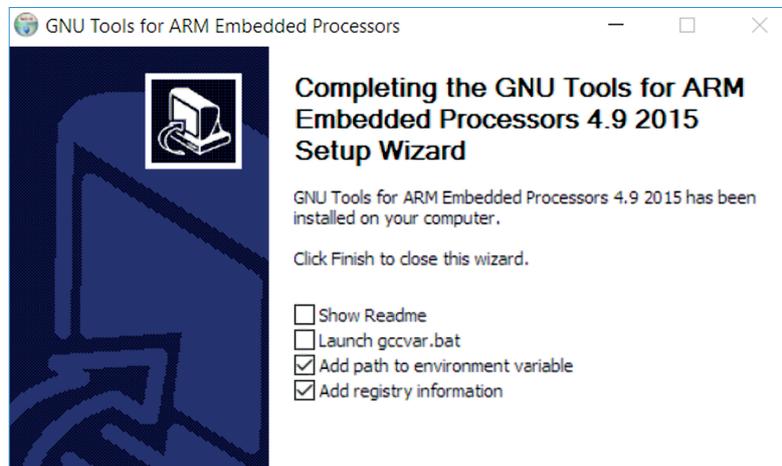


Figure 4-5: It is important to add the path to the environment variable

Следующим шагом будет установка SSP. Для этого найдите загруженный zip-файл SSP Distribution_1.2.0.zip и извлеките и запустите файл SSP_Distribution_1.2.0.exe. Следуйте инструкциям установщика, пока не достигнете экрана «Выбор компонентов» (Choose Components). Убедитесь, что выбран пакет SSP Release Pack и SSP Documentation. На следующем экране проверьте путь, чтобы подтвердить, что вы устанавливаете SSP в корневую папку ранее установленной e² studio. По умолчанию ISDE был установлен на C:\Renesas\e2_studio. Если вы изменили это во время установки e² studio, вы должны ввести здесь правильный путь. Нажмите «Установить», и установка начнется. Во время процесса вам будет предложено установить документацию. Примите значение по умолчанию, которое указывает на C:\Renesas\Synergy или введите свой собственный путь. Это установит следующие документы: HTML-версия Руководства пользователя SSP, файл readme с подробными обновлениями и PDF-файл с примерами разработчиков SSP.

Как только эта установка завершится, можно запустить ISDE в первый раз.

4.2.3 Первый запуск

Запустите e² studio из меню «Пуск» на ПК. Он попросит вас указать место для рабочего пространства. Вы можете принять по умолчанию или выбрать своё собственное. Поскольку это первый раз, когда вы запускаете ISDE, он будет извлекать и обновлять дополнительные файлы поддержки и после этого перезагрузится. Во время процесса запуска если вы установили привязку GCC в папку по умолчанию вы должны увидеть небольшое окно под названием Toolchain Integration. Обязательно выберите версию инструментальной панели GCC ARM® Embedded, которую вы установили ранее, и нажмите «Зарегистрироваться». В противном случае будет запрещено компилировать ваши проекты позже, и вам придется вручную регистрировать инструменты изнутри ISDE.

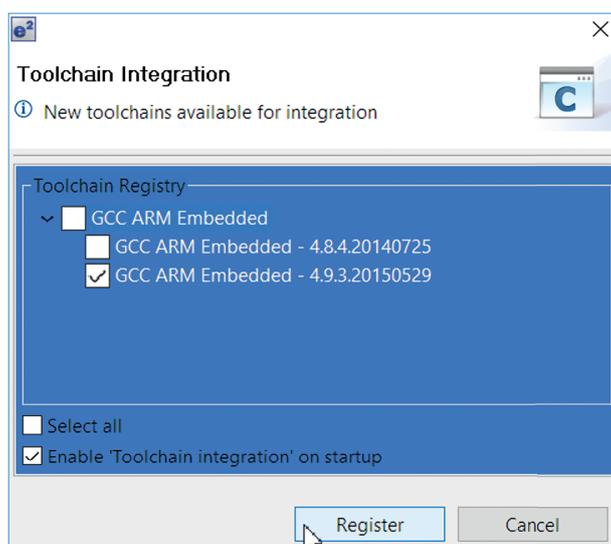


Figure 4-6: Failing to select the installed tool chain here will result in build problems

Когда вы попадёте на экран приветствия, нажмите «Workbench». Следующий шаг - создать первый простой проект, чтобы убедиться что всё работает. Для этого перейдите в меню File → New → Synergy C Project в строке меню, в котором будет отображаться конфигуратор проекта. Введите название для этого проекта, например, «MySynergyProject», и выберите правую часть инструментальной панели GCC ARM® Embedded. Следующий шаг — указать e² studio на действительный файл лицензии. Во время установки SSP лицензия на пробное использование была помещена в следующую папку: <e2 studio installation path>\internal\projectgen\arm\Licenses. Если во время установки вы решили принять путь по умолчанию, это будет путь C:\Renesas\e2_studio\internal\projectgen\arm\Licenses, и селектор файлов доставит вас прямо в это место. Выберите там (только) файл с именем SSP_License_Example_EvalLicense_20160205.xml (или аналогичный). После загрузки он отобразит информацию о лицензии, поддерживаемые компоненты и соответствующие разрешения в окне ниже.

Лицензия на пробное использование, которую вы только что установили, позволяет использовать все части инструментов Synergy. Единственное ограничение заключается в том, что просмотр или редактирование исходного кода для компонентов ThreadX®, X-Ware™, библиотек, стеков и кода API приложений будет ограниченным. Если вы хотите просмотреть или отредактировать этот код, вам понадобится лицензия на разработку и производство, которую можно запросить в галерее Synergy.

Нажмите «Далее», чтобы перейти к экрану выбора устройства и инструментов. Выберите плату. Поскольку мы хотим протестировать среду разработки, для этого первого теста просто выберите S7G2 SK и установите соответствующее устройство R7FS7G27H3A01CFC, которое автоматически не будет указано. Посмотрите на панель инструментов. Она должна считать GCC ARM Embedded с версией 4.9.3.20150529 или более поздней.

Если вы установили GNU-инструменты, но они не перечислены здесь, вам необходимо их интегрировать вручную. Для этого перейдите в меню «Help» (Справка) e² studio и выберите запись «Add Renesas Toolchains» (Добавить инструменты Renesas). Обратите внимание, что опция «Сканирование» работает только в тех случаях, когда установочные файлы были установлены по умолчанию. Если вы не установили инструменты GNU в своем местоположении по умолчанию, либо введите путь к вашей пользовательской папке в поле ввода, либо перейдите к местоположению, используя браузер файлов.

Следующий шаг - выбрать шаблон проекта. Шаблон проекта включает в себя правильную библиотеку поддержки платформ (BSP), зависящую от устройства и платы, и, если опция была выбрана, включена поддержка ThreadX[®] (независимо от того, какой шаблон выбран, ThreadX[®] можно включить на вкладке «Threads» в конфигураторе Synergy Configurator. Кроме того, Некоторые из шаблонов включают в себя также полные примеры проектов. Для нашего теста достаточно просто использовать базовый пакет, поэтому выберите «Blinky». Нажмите «Finish» (Готово). Конфигуратор извлечет все необходимые файлы и спросит вас, хотите ли вы открыть перспективу Synergy Configuration. При согласии «Yes» конфигуратор Synergy Configuration будет открыт. В конфигураторе вы можете настроить SSP и большую часть встроенного оборудования процессора. Для шаблона, который мы использовали для этого примера, все уже установлено, поэтому нажмите на маленькую зеленую стрелку, обозначающую «Generate Project Content» (Создание содержимого проекта) в правом верхнем углу окна, и конфигуратор создаст файлы конфигурации и добавит их в проект.

Наконец нажмите на маленький символ «Hammer» (Молоток) возле левого угла панели инструментов наверху ISDE, и проект будет компилироваться и компоноваться. По завершении он должен показать статус 0 ошибок, 0 предупреждений после завершения.

Теперь все настроено и работает, и вы готовы начать свой собственный проект. Но на мгновение остановитесь и задумайтесь на несколько секунд, что мы только что сделали. Я бы подумал, что для создания этой огромной программной среды для полного семейства микроконтроллеров процесс очень прост. Помните, что мы установили пакет программного обеспечения с более чем 200 человеко-годами усилий по разработке и полным ISDE менее чем за 10 минут. Я считаю это очень легким процессом!

И если во время установки произошло что-то неожиданное, вы знаете, что помощь всегда доступна, и вы можете начать общаться с экспертом Renesas Synergy™ Platform даже прямо из e² studio: зайдите в Help → Synergy Helpdesk (Справка → Служба поддержки Synergy), и вы будете направлены на Онлайн-чат в Интернете. Другая возможность — щелкнуть значок Live Chat на вкладке «Summary» (Сводка) в перспективе Synergy Configuration. Это откроет окно чата непосредственно в e² studio.

4.2.4 Постоянное обновление вашей версии

Установив инструменты, вы наверняка хотите сохранить их в хорошей форме, а это значит, что вы хотите сохранить их в актуальном состоянии. Или вы так взволнованы всеми возможностями, предлагаемыми ISDE на базе Eclipse, что хотите установить дополнительное программное обеспечение. Сохранение потока актуальных версий достигается путем загрузки последних версий из галереи Synergy. Использование функции «Проверка обновлений» внутри e² studio не рекомендуется, так как обновления там не добавляют ничего полезного для платформы Synergy.

Если вы хотите установить дополнительное программное обеспечение, перейдите в раздел Help → Install New Software (Справка → Установить новое программное обеспечение). На следующем экране выберите «All Available Sites» (Все доступные сайты) для поиска по всем сайтам, и вам будет представлен список программного обеспечения, доступного для загрузки. И, конечно же, еще один способ получить обновления — это перейти на ваш любимый веб-сайт Галерея Synergy, где вы можете найти обновления и дополнительное программное обеспечение. Для обновлений для SSP галерея Synergy — лучшее место.

Теперь, когда мы знаем, что установленная версия работает, следующие шаги состоят в том, чтобы проверить инструменты разработчика и написать собственный код.

4.3 Установка IAR Embedded Workbench® для Renesas Synergy™

Установка IAR Embedded Workbench® для Renesas Synergy™ осуществляется в основном по тем же самым шагам. Для справки, полная глава с описанием доступна в Интернете на домашней странице книги в виде PDF-файла.

Вопросы, которые следует вынести из этой главы:

- Единственной точкой доступа к загрузке инструментов является галерея Synergy.
- Установка проста, но следует позаботиться о установке в правильном порядке.
- Если вы не установите GCC-инструменты в свой каталог по умолчанию, вам придется регистрировать их вручную внутри ISDE.

5 РАБОТА СО СРЕДОЙ РАЗРАБОТКИ ПЛАТФОРМЫ RENESAS SYNERGY™

Что вы узнаете в этой главе:

- Разница между перспективами, представлениями и редакторами.
- Для чего предназначены разные конфигурации и как они используются.
- Как импортировать и экспортировать проекты.

Теперь, когда мы знаем детали программного пакета Renesas Synergy™ (SSP) и как использовать его API, пришло время изучить различные среды разработки, доступные для платформы Synergy. На момент написания этой книги доступны две среды: основанная на Eclipse e² studio от Renesas, которая будет рассмотрена в разделе 5.1 и IAR Embedded Workbench® для Renesas Synergy™, рассмотренная в разделе 5.2.

5.1 e² studio на базе Eclipse™

e² studio является собственной средой разработки Renesas, она основана на популярной и широко распространенной интегрированной среде разработки (Integrated Development Environment — IDE) Eclipse™, подходит для разных языков программирования и целевых платформ. Её можно легко настроить и расширить, и, следовательно, это среда разработки для тысяч разработчиков по всему миру и де-факто является стандартом. И она будет регулярно обновляться, чтобы использовать новейшие и самые лучшие инструменты EclipseSDK и CDT.

e² studio использует все преимущества Eclipse и включает в себя дополнительные представления и перспективы конфигуратора для поддержки всех функций платформы Synergy. Она содержит все инструменты, необходимые для создания, компиляции и отладки проектов любого размера и сложности и направляет разработчика через три этапа разработки программного обеспечения: подготовка, сборка и отладка.

В следующих частях этой главы рассказывается о рабочей среде Eclipse и о том, как использовать его различные элементы. Хотя мы рассмотрим некоторые вопросы, в этой книге будет объяснено не всё. Более подробную информацию можно найти в руководстве пользователя Рабочей среды (Workbench), перейдя в Help → Help Contents (Справка → Содержание справки) в e² studio.

5.1.1 Краткое введение в философию Eclipse

Главное окно e² studio, называемое workbench (рабочая среда), состоит из нескольких базовых элементов пользовательского интерфейса, таких как перспективы, представления, редакторы, панели меню и инструментов.

Как только e² studio запускается, она открывает перспективу(ы), которая использовалась последним. Перспектива — это набор представлений, редакторов и панелей инструментов, а также их расположение внутри рабочей среды. Если вы измените порядок окон, панелей инструментов или представлений, эти изменения будут сохранены в текущей перспективе и будут доступны, как только вы снова откроете её в следующий раз.

В e² studio существует несколько заранее определенных перспектив, и вы можете открыть одновременно несколько из них (например, перспективу C/C++ и перспективу ресурсов). Переход с одной перспективы на другую можно осуществить, щелкнув по квадратному значку рядом со списком перспектив в правой части панели инструментов,

в котором появится всплывающее окно со списком всех доступных перспектив или выбрав Window → Perspective (Окно → Перспектива) в главном меню. Вы можете закрыть перспективу, используя те же пункты меню или щелкнув правой кнопкой мыши на перспективе в списке.

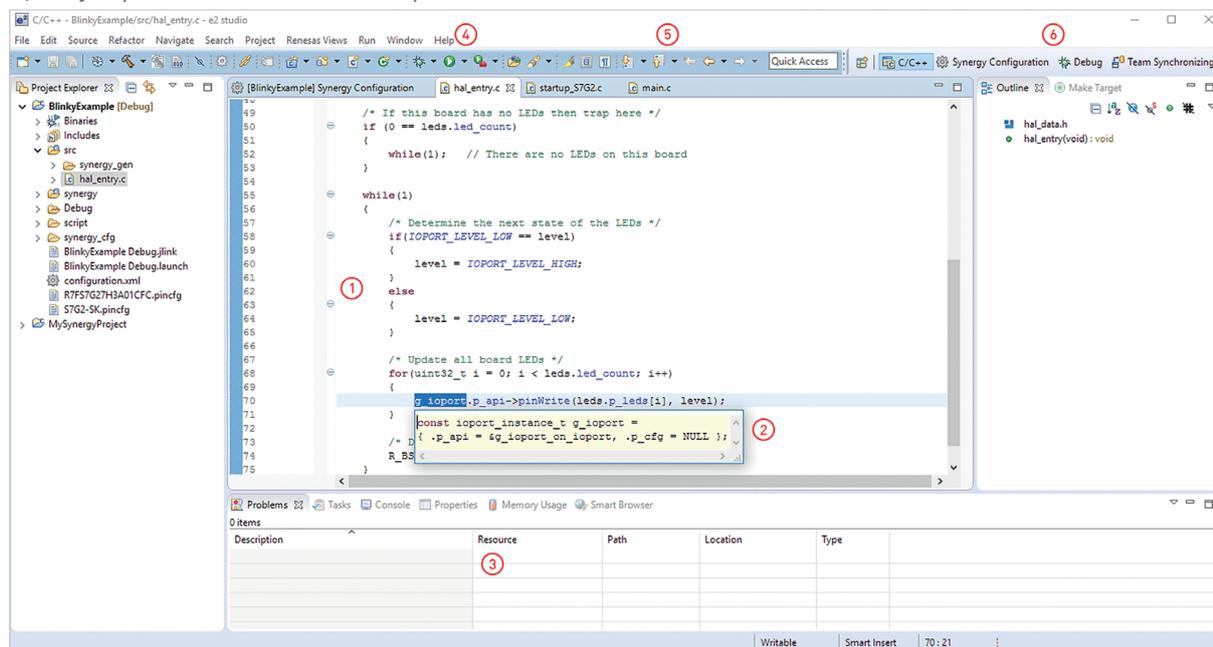


Figure 5-1: The workbench of e² studio consisting of editors (1), Smart Manuals (2), views (3), menu bar (4), tool bars (5) and perspectives (6)

ПЕРСПЕКТИВЫ (PERSPECTIVES) в e² studio

- C/C++:** Это стандартная перспектива e² studio и используется для разработки программ и редактирования исходного кода. По умолчанию она открывает редактор; представление ProjectExplorer (Проводник проекта) слева, где вы можете управлять своими проектами; представление Outline, отображающее все переменные и определения исходного файла, активного в редакторе; а внизу — блокнот с вкладками, где размещены некоторые представления, такие как Problems (Проблемы) (например, любые ошибки синтаксиса или компиляции), Tasks (Задачи) (показаны задачи Doxygen) или Properties(Свойства).
- Debug (Отладка):** эта перспектива используется для запуска программ, диагностики и отладки проблем, возникающих во время выполнения. e² studio открывает по умолчанию следующие представления: представление Debug (да, перспектива Debug включает в себя представление «Debug»); панель инструментов Debug; окно(-а) редактора; представление Outline справа (так же как и в перспективе C/C++); внизу — некоторые многоуровневые представления для анализа и визуализации; а сверху — блокнот с вкладками, где можно просмотреть переменные, точки останова, регистры и т. д.
- Resource (Ресурс):** по умолчанию эта перспектива включает все представления, которые есть в представлении C/C ++, кроме окна внизу, показывающего представление Tasks.
- Synergy Configuration (Конфигурация Synergy):** это одна из перспектив, в которой вы будете проводить много времени в начале. Большая часть рабочей среды занята перспективой Synergy Configuration (помните, что перспектива может включать и другие перспективы), с несколькими представлениями для разных configurаторов Synergy, такими как Clock Configurator (конфигуратор блока тактирования) или Pin Configurator (конфигуратор контактов). Он также включает в себя представление Properties (Свойства), где могут быть сделаны настройки для разных периферийных устройств, потоков и драйверов; и представление Package (Пакет), где вы видите визуальную схему пакета, выбранного для микроконтроллера, где отображается текущая конфигурация контактов, вместе со статусом, показывающим была ли обнаружена проблема конфигурации или нет. При необходимости Synergy Configurator может быть доступен из перспективы C/C++ и Debug.

- **Team Synchronization (Синхронизация команды):** В этой перспективе вы можете объединить изменения, которые сделали вы с изменениями других членов команды. В главе 5.1.4 это будет рассмотрено более подробно.
- **Trace (Трассировка):** Эта перспектива открывает представление статистики и гистограммы, а также консоль ARM® CoreSight™ ITM Live Trace Console, позволяющую отслеживать установленную программу.

Если вы измените свою перспективу и вам не понравятся изменения, вы всегда можете вернуть ее обратно к стандартному виду, щелкнув правой кнопкой мыши на имени перспективы на главной панели инструментов и выбрав Reset (Сброс). Это также помогает очистить беспорядок, который вы создали после интенсивного написания кода или отладки, с большим количеством открытых и отстыкованных представлений и редакторов! Нет ни одного экрана, достаточно большого, чтобы держать все окна в удобном расположении, поэтому, в конце концов, вы окажетесь в очень запутанной рабочей среде.

ПРЕДСТАВЛЕНИЯ (VIEWS)

Представление — это окно, в котором вы можете что-то рассмотреть, например, набор регистров, свойства или список файлов. В представлении также могут быть альтернативные представления, такие как диаграмма работы потоков в реальном времени или шкалы и управляющие элементы, которым можно подключить переменные. В представлениях может быть собственное меню или панель инструментов. Действия, вызванные меню или панелью представления, будут влиять только на элементы в этом представлении, но не на другие представления, даже если они находятся в одной и той же перспективе.

Несколько представлений могут быть размещены вместе в одном окне, которое затем называется блокнот с вкладками. Вы также можете перемещать представления из одного блокнота в другой или полностью отстыковать их. Это одна из приятных функций e² studio, которая позволяет организовать все, что требуется вашему рабочему процессу, и быть уверенным, что последний макет вашей перспективы автоматически будет сохранен для вас.

e² studio предлагает множество разных представлений. Renesas даже создал несколько дополнительных представлений, таких как Conflicts (Конфликт), Fault Status (Состояние сбоя) или RTOS Resources (РесурсыОСРВ), что делает интегрированную среду разработки решений (ISDE) более универсальной. Их можно найти в главном меню в разделе Renesas Views.

РЕДАКТОРЫ (EDITORS)

Это представление (редактор считается особым видом представлений), с которым каждый разработчик программного обеспечения будет проводить большую часть своего времени. Помимо обычных и ожидаемых функций, таких как компиляция кода или выделение ключевых слов, редактор, встроенный в e² studio, также включает специальную функцию Smart Manual (Умное руководство). Это устраняет необходимость изучения 1000 страниц документации, так как вы получаете контекстно-зависимую помощь по SSP и самому микроконтроллеру.

Наведение курсором мыши по любому из тысяч выделенных слов в e² studio приведет к отображению соответствующей информации. Для переменных будет отображаться объявление, для структур и перечислений — все члены и функции, описание, прототип и параметры. Если переменная связана с регистром микроконтроллера, также будет отображаться подробная информация о назначениях бит. В Smart Manual даже предлагает соответствующие примеры по применению и мультимедийные учебные материалы, если они доступны, и отображает их в представлении Smart Browser в нижней части представления C/C ++.

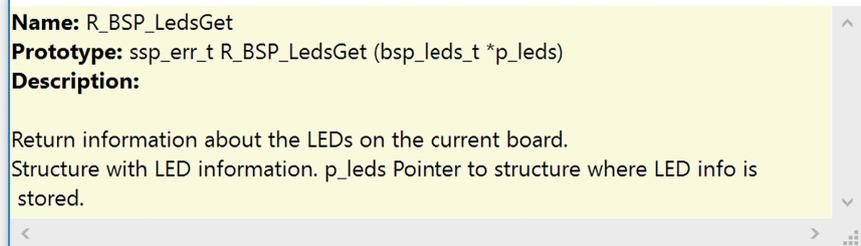
Эта функция очень крутая и экономит много времени, так как вам не придется переключаться между программами SSP, руководством Synergy Microcontroller (MCU) и e² studio. Вы получаете необходимую информацию там, где она вам нужна: прямо в точке вашего курсора мыши.

```

/* LED state variable */
ioport_level_t level = IOPORT_LEVEL_HIGH;

/* Get LED information for this board */
R_BSP_LedsGet (&leds);

```



Name: R_BSP_LedsGet
Prototype: ssp_err_t R_BSP_LedsGet (bsp_leds_t *p_leds)
Description:
Return information about the LEDs on the current board.
Structure with LED information. p_leds Pointer to structure where LED info is stored.

Figure 5-2: The Smart Manual feature of e² studio will display all necessary information directly where it is needed

Функция автозавершения кода- еще одна из тех функций ISDE, которые сохраняют много времени при разработке кода. Нажатие <ctrl>+space после переменной вызовет окно с доступными параметрами, например, членами API-структуры. Нажав на один из них, вы введете его в свой код. Опять же, не нужно заглядывать в руководство, просто нажмите и выберите

5.1.2 Конфигураторы: краткое введение

Конфигураторы ve² studio графически направляют разработчика программного обеспечения через определенные параметры устройства микроконтроллера Synergy и сообщают об их использовании. Примерами этого являются настройка контактов или предупреждение конфликтов. Конфигураторы также генерируют код запуска или размещают компоненты программного обеспечения SSP в проекте.

КОНФИГУРАТОР ПРОЕКТА (PROJECT CONFIGURATOR)

Существует несколько конфигураторов, которые могут помочь разработчику, и первый, с которым сталкивается большинство пользователей — это Project Configurator, который ведет через процесс создания нового проекта Synergy с нуля или из шаблона, предоставленного конфигуратором. Он позволяет выбирать параметры для проекта, такие как используемый набор инструментов, какое использовать устройство и плату или должен ли быть создан проект для примера.

В конце этого процесса проект и все необходимые файлы будут автоматически созданы и добавлены в активный проект в рабочей среде e² studio. Нет необходимости создавать что-либо вручную — не нужно просматривать параметры и настройки make-файла, не нужно исследовать, какие файлы заголовков необходимо включить, как назначить компилятору использование определенной серии и устройства из большого семейства микроконтроллеров Synergy. Все делается за вас, как только вы нажимаете Finish на последнем экране конфигуратора.

По завершении формирования кода Project Configurator переключится на перспективу Synergy Configuration, где могут быть произведены настройки для разных компонентов SSP.

КОНФИГУРАТОР SYNERGY (SYNERGY CONFIGURATOR)

Конфигуратор Synergy сначала отображает сводку текущего проекта и дает объяснение различных доступных вкладок и способов их использования. Следующая вкладка - вкладка BSP (Пакет Поддержки Платформы), позволяющая просматривать и редактировать аспекты настройки платы, например, выбор устройства или платы. В связанном представлении свойств Properties могут быть сделаны дополнительные настройки для BSP, например, определён размер основного стека (стек, используемый вне контекста задач) или период времени сторожевого (watchdog) таймера.

Следующая вкладка с именем Clocks предназначена для установки начальной конфигурации внутреннего тактирования. Отображается графическое представление системы тактирования на кристалле и могут быть внесены изменения в дерево синхронизации. Наведение курсором мыши над элементами приведет к их краткому описанию. Если выполнены несовместимые настройки, соответствующий компонент будет выделен красным цветом, и будет дано объяснение проблемы. Кроме того, сама вкладка изменяется и отображает небольшой восклицательный знак, указывающий на наличие проблемы.

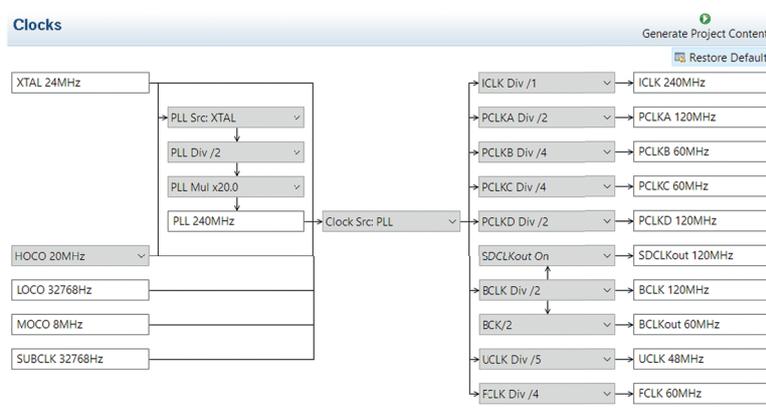


Figure 5-3: The Clocks view of the Synergy Configurator

Вкладка Pins покрывает начальную настройку контактов микроконтроллера Synergy. Контакты могут быть перечислены либо на основе портов, либо на периферийных функциях. Перспектива Package справа от configurатора показывает корпус устройства, выделяющий настроенные контакты и маркирует ошибки, если есть конфликты или неполные настройки. Они также будут перечислены в Problems, также как и в представлении Pin Conflicts.

Следующая вкладка Threads позволяет добавлять и настраивать задачи в проекте Synergy. Различные модули и объекты могут быть добавлены к отдельным потокам, а их свойства могут быть изменены в представлении Properties. В представлении Threads отображается множество разных настроек потоков, что позволяет использовать простую графическую конфигурацию модулей. Новые модули и потоки добавляются легко, при этом все необходимые модули автоматически добавляются до уровня, где требуется вмешательство разработчика. Если достигнут этот уровень, то модуль будет отмечен красным цветом, а описание необходимых настроек или проблем будет дано, как только курсор мыши будет наведен на модуль. Если проблема решена, модуль возвращается к стандартному цвету.

Структура обмена сообщениями настроена на вкладке Messaging, что позволяет создавать классы сообщений данных приложений и определять список потоков, подписавшихся на каждое адресованное сообщение. Далее находится вкладка ICU, для блока управления прерываниями. Эта вкладка ранее использовалась для просмотра и редактирования доступных источников прерываний и связанного с ними уровня приоритета, но устарела (и поэтому оставлена пустой) в последней версии e² studio, так как прерывания теперь настроены через представление Properties выбранного модуля.

Окончательная вкладка в представлении Synergy Configuration называется Components. Она позволяет отображать и выбирать различные компоненты Synergy. Изменения, такие как добавление или удаление модулей из текущего

проекта, не должны выполняться в этом представлении, для этого предназначена вкладка Threads, это представление предназначено для просмотра и выбора версий модулей.

После того, как все настройки в Synergy Configuration будут выполнены, будет сгенерирован или извлечен из SSP и добавлен в проект соответствующий исходный код. Это делается нажатием на символ Generate Project Contents в верхней части перспективы. Если вы забыли нажать на нее, не волнуйтесь! Этот процесс произойдет также, если будут обнаружены изменения в конфигурации или сгенерированных файлах.

Конфигуратор Synergy — отличный инструмент, так как он поможет вам выполнить все шаги, необходимые для подготовки проекта, и произвести для него начальные настройки. Как и в случае с Конфигуратором проектов, нет необходимости просматривать тысячи страниц документации и глубоко изучать их, поскольку конфигуратор предоставит необходимую информацию на абстрактном уровне и обеспечит правильность и правдоподобие всех настроек. Если вы вспомните, сколько времени вы потратили во время своего последнего проекта, не используя Synergy Configurator, просто совершив, например, все настройки контактов и настройки прерывания, имея дело с множеством взаимозависимых аппаратных регистров, вы, несомненно, оцените все усилия, которые Renesas вложил в свои новые инструменты. Это намного быстрее!

5.1.3 Импорт и экспорт проектов Synergy

Время от времени вам придется импортировать или экспортировать проекты. Возможно, вы захотите использовать один из многочисленных примеров проектов с веб-сайта Renesas или захотите поделиться своей последней самой величайшей разработкой с одним из ваших коллег. Независимо от того, хотите ли вы импортировать или экспортировать проект, это удобно сделать из самой e² studio с помощью мастера импорта или мастера экспорта Synergy.

ИМПОРТ ПРОЕКТОВ

Существует два способа импорта проектов. Оба запускаются в меню File → Import. После появления окна мастера импорта разверните раздел General entry (Общая запись), в котором вы можете выбрать либо Existing Projects into Workspace (Существующие проекты в рабочей области), либо Rename & Import Existing C/C++ Project into Workspace (Переименовать и импортировать существующий проект C/C++ в рабочую область).

В любом случае для импорта вам будет предоставлен выбор либо исходного каталога, где выполняется проект, либо архивного файла. Если найден один или несколько проектов, вы можете выбрать их для импорта. Если вы выбрали вариант «Переименовать и импортировать», вам также необходимо задать импортированному проекту новое имя. Нажмите Finish проект(ы) будут импортированы в ваше рабочее пространство.

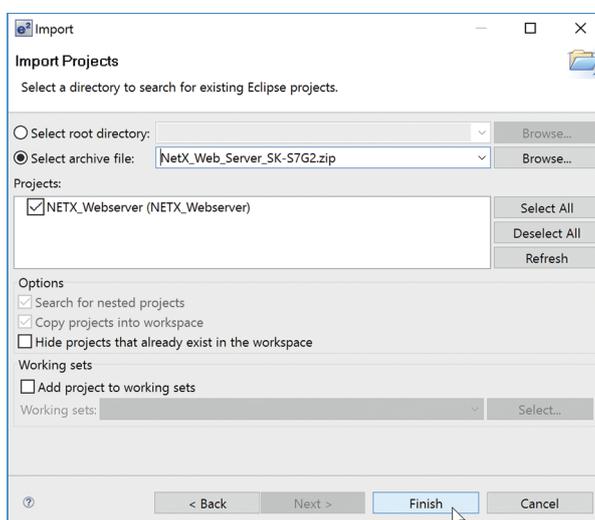


Figure 5-4: The projects found at a given location are listed for import

ЭКСПОРТ ПРОЕКТОВ

Экспорт проектов так же прост, как и их импорт. Опять же, есть два пути для решения этой задачи. Первый из них — перейти в меню e² studio выбрать File → Export. Второй – щелкнуть правой кнопкой мыши по проекту и выбрать Export Synergy Project. При использовании первой версии в открывшемся окне экспорта есть два дополнительных шага: Первый — это расширение Общей записи, а вторая – выбор записи архивного файла Renesas Synergy из появившегося списка.

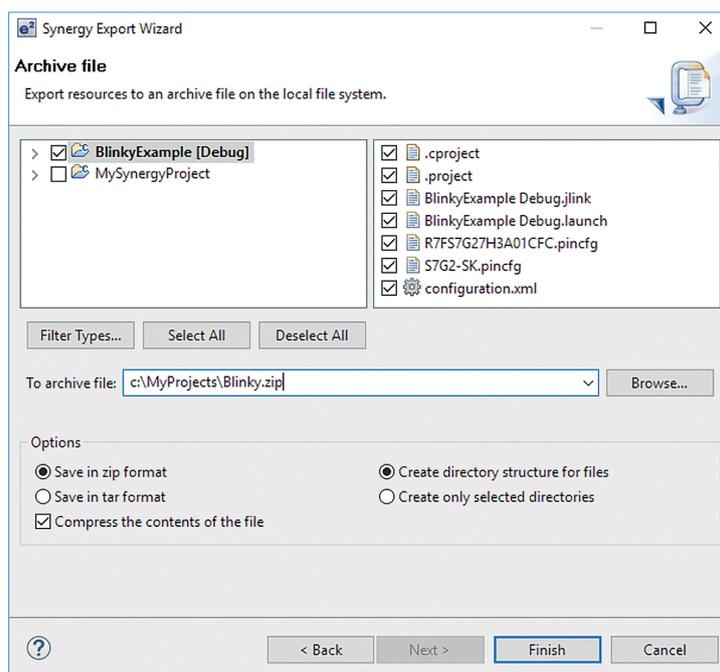


Figure 5-5: The Export Wizard in e² studio

После появления мастера экспорта Synergy можно выбрать проекты и файлы для экспорта, а также формат сжатия и требуемую структуру каталогов. При нажатии на Finish будет завершен процесс экспорта.

Не пытайтесь экспортировать проект, просто скопировав каталог своего рабочего пространства, так как это нарушит лицензионное соглашение пакета программного обеспечения Synergy, что не позволит вам повторно распространять любой контент SSP. Кроме того, нет необходимости включать файлы SSP в ваш экспорт: поскольку файл configuration.xml является частью экспорта, любой контент SSP можно воссоздать, открыв файл (что приведет к запуску Synergy Configurator) и снова нажав на Create Project Content. Это приведет к восстановлению файлов, пропущенных во время экспорта. Прямая копия также вряд ли будет работать из-за многочисленных ссылок на имя и местоположение проекта. Мастер экспорта Synergy удалит их, и мастер импорта восстановит их при импорте.

5.1.4 Совместная работа с командами и обработка файлов

Командная разработка в e² studio осуществима и проста, так как среда обеспечивает интерфейс для контроля версий, поэтому вы можете загружать и выгружать файлы из ISDE. Но какие части проекта нужно синхронизировать? Эмпирическое правило: все, что создается конфигуратором Synergy, синхронизировать не требуется.

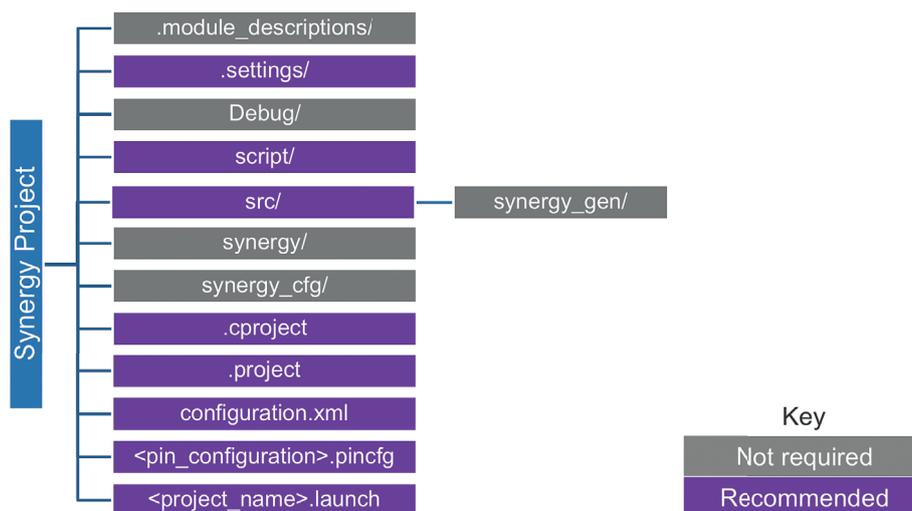


Figure 5-6: There is no need to check-in items in grey

5.2 IAR Embedded Workbench®

Поскольку эта книга в основном охватывает набор инструментов, предоставленную Renesas, эта часть главы, объясняющая использование IAR Embedded Workbench® для Renesas Synergy™ (EW for Synergy), предоставляется онлайн версии.

Вопросы, которые следует вынести из этой главы:

- e² studio предоставляет различные перспективы и представления для группировки функциональности.
- Функция Smart Manual, Project Configurator и Synergy Configurator ускоряют разработку и уменьшают количество ошибок.
- Файлы, созданные Synergy Configurator, не нужно экспортировать и проверять в системе управления версиями

6 ОТЛАДОЧНЫЕ ПЛАТЫ RENESAS SYNERGY™

Что вы узнаете в этой главе:

- Какие различные аппаратные отладочные платы существуют для платформы Synergy, подробная информация о них и варианты использования.

Во время каждого проекта разработки есть момент, когда вам понадобятся аппаратные средства для запуска ваших первых тестов. И как известно многим инженерам, это почти всегда требуется задолго до того, как собственное разрабатываемое оборудование можно назвать готовым к использованию. Или, может быть, разработчики аппаратного обеспечения не хотят передавать один из своих редких прототипов своим любимым коллегам по программному обеспечению, которые могут даже уничтожить плату (не то, что бы это когда-либо со мной случилось!).

Renesas имеет решение для таких случаев: Renesas предлагает два вида простых в использовании плат, которые позволяют разработчикам программного обеспечения легко и без проблем начать работу с платформой, так как они могут сразу проверить свои программы. Разработчик оборудования также получит пользу от этих плат, поскольку он может получить доступ к большинству контактов микроконтроллера, удобно расположенным на разных разъемах отладочных плат.

6.1 Различные типы аппаратных отладочных плат

Какую отладочную плату выбрать, зависит от того, чего хочет достичь разработчик. Начальная отладочная плата Starter Kit SK-S7G2 предлагает легкий доступ к платформе Synergy, поскольку она основана на серии микроконтроллеров Synergy, предлагающей самые большие объемы памяти и самый большой выбор встроенных периферийных устройств. Таким образом, это отличный выбор, если пока не ясно, какой микроконтроллер окончательно будет выбран для проекта или если вы еще не определились, какая отладочная плата подходит для вас.

С другой стороны, отладочные платы типа Development Kit больше предназначены для полного прототипирования проектов, предоставления доступа ко всем контактам и наличием дополнительного оборудования для конкретных приложений, таких как емкостной сенсорный дисплей. Общими для всех плат являются их модульность и расширяемость, а также их чрезвычайно простая и безошибочная конфигурация.

6.2 Отладочная плата Starter Kit SK-S7G2

Отладочная плата Starter Kit SK-S7G2 - это ваш недорогой пропуск ко всей платформе Synergy и является платой, на которой основана эта книга. Это позволяет без проблем изучать платформу Synergy, включая микроконтроллер и все его периферийные устройства. Компактная конструкция обеспечивает доступ к более чем 80% контактов устройства через разъемы Arduino и Pmod™, что позволяет быстро создать прототип вашего приложения.

Встроенный сенсорный дисплей QVGA обеспечивает взаимодействие с платой, особенно если используется вместе с промежуточным программным обеспечением GUIX™ из пакета программного обеспечения Synergy (SSP) и GUIX Studio™ - программы для Windows, которую можно загрузить из галереи Synergy и использовать для создания графических интерфейсов пользователя (GUI). Подключение к внешней среде доступно через USB, Ethernet, RS-232/485 и Bluetooth Low Energy (BLE) 4.1.

Через встроенный отладчик J-Link™ доступна отладка программного обеспечения и программирование устройства. Полный комплект документации можно найти на веб-сайте Renesas: Руководство пользователя начального набора инструментов Starter Kit SK-S7G2, краткое руководство и схемы плат, а также технический паспорт S7G2 и руководство пользователя к нему.

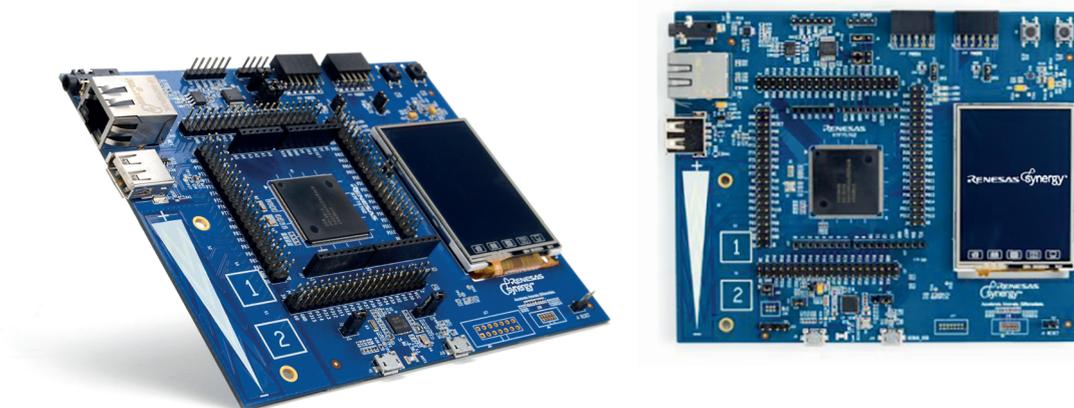


Figure 6-1: The SK-S7G2 Starter Kit

Другие характеристики Starter Kit включают:

- 2,4-дюймовый цветной дисплей TFTLCD QVGA (320 x 240) с сенсорным экраном, который использует внутреннее ОЗУ контроллера S7G2 для буферной памяти дисплея.
- Расширение:
 - Совместимый разъем Arduino UNO Shield
 - 2 разъема Pmod™
 - Штыревые разъемы
- Проводная связь
 - USB 1 x HS Host, 1 x FS Device)
 - Ethernet с поддержкой RMI и IEEE 1588 Precision Time Protocol (PTP)
 - RS232/RS485, доступен на контактах с трансивером
 - CAN, доступен на контактах с трансивером
- Беспроводная связь
 - Bluetooth Low Energy (BLE) 4.1 на плате плюс множество опций через разъемы Arduino и Pmod™
- Память
 - Встроено в микросхему: флэш-память 4 МБ + 640 КБ SRAM
 - Внешняя флэш-память 8-Мбайт QSPI

6.3 Макетные платы Synergy

Если отладочная плата SK-S7G2 была вашим стандартным пропуском в платформу Synergy, то отладочные платы Development Kits (DK) — это ваш VIP-пропуск, так как они могут использоваться для полного прототипирования проекта со 100% набором контактов, доступных через разъемы Pmod™ и через штыревые контакты.

Поскольку большинство контактов на микроконтроллерах Synergy поддерживают различные функции, они могут быть подключены к нескольким разъемам или устройствам на платах. DIP-переключатели на печатной плате обеспечивают легкую, безопасную и безошибочную маршрутизацию различных функций. Каждый DIP-переключатель управляет высокоскоростным буфером, который, в зависимости от настройки переключателя, либо подключает, либо изолирует контакты микроконтроллера от периферийного устройства или разъема. Кроме того, DIP-переключатели могут считываться программным обеспечением через IIC-порт расширителей ввода-вывода и, если коммутатор открыт, программное обеспечение может активировать буферы. Светодиоды указывают, когда соответствующее устройство подключено под программным управлением. Это довольно крутая функция, так как вам не нужно перемещать тонны крошечных переключателей, поэтому маршрутизация сигналов очень проста.

Есть несколько функций, которые доступны на каждой из отладочных плат DK:

- Полный доступ ко всем контактам и функциям микроконтроллера.
- Проводная связь:
 - USB
 - RS232/RS485, доступен на контактах с трансивером
 - CAN, доступен на контактах с трансивером
- Беспроводная связь
 - Bluetooth Low Energy (BLE) 4.1 на плате плюс множество опций через разъемы Arduino и Pmod™
- Несколько датчиков температуры, ускорения, света и т. д.
- Отладка и программирование через встроенный отладчик J-Link®.

Для разных устройств доступно несколько макетных плат для разработки, а на печатной плате будут доступны дополнительные периферийные устройства (в зависимости от набора функций используемого микроконтроллера). Как и для отладочных плат, для всех макетных плат имеется полный комплект документации.

6.3.1 Отладочная плата для разработки Development KitDK-S7G2

В плате DK-S7G2 находится микроконтроллер серии S7 S7G2, который является флагманским устройством в платформе Synergy. Разработанный для простоты в использовании, он имеет все упомянутые уже функции и многое другое. Комплект состоит из нескольких плат: одна плата для разводки, имеет различные интерфейсы, такие как Ethernet 10/100 или разъемы для платы камеры, основная плата с микроконтроллером, несколько разъемов, светодиоды и т. д. а также две платы расширения, одна плата для поддержки WQVGA TFT LCD и одна плата для камеры, с поддержкой сенсорного дисплея CMOS VGA.

Уникальными особенностями платы в дополнение к уже упомянутым являются:

- Цветной дисплей с емкостным сенсором прикосновений 4.3 “WQVGA (480 x 272) TFTLCD
- Датчик изображения VGA (640 x 480) CMOS с поддержкой до 30 FPS
- Расширение через штыревые разъемы и четыре разъема Pmod™
- Проводная связь:
 - Два порта Ethernet с интерфейсом RMII и поддержкой протокола IEEE 1588 (PTP)
- Память:
 - Встроенная флэш-память 4 МБ плюс 640 КБ SRAM
 - Внешняя: 16 МБ SDRAM, плюс 16 МБ QSPI flash плюс 2 Гб eMMC
- Полный интерфейс для SD-карты

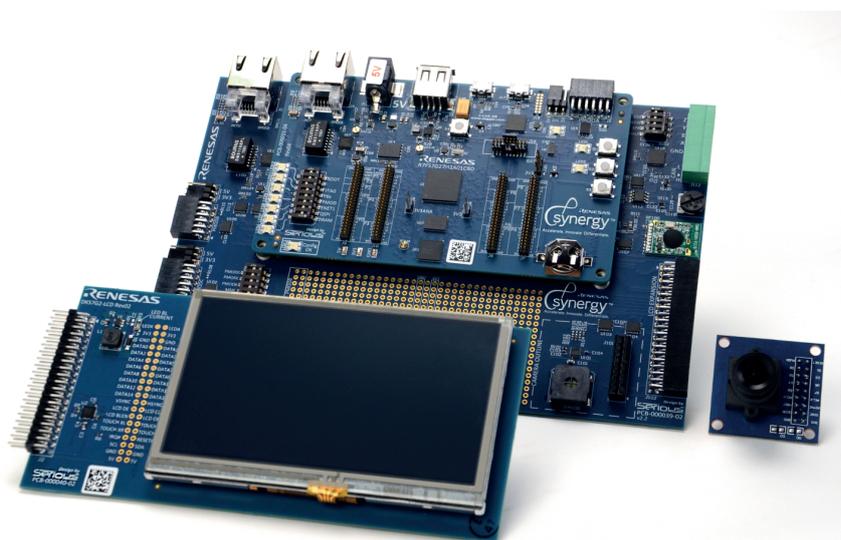


Figure 6-2: The DK-S7G2 Development Kit

6.3.2 Отладочная плата для разработки Development Kit DK-S3A7

DK-S3A7 оснащен микроконтроллером S3A7 серии S3, энергоэффективным микроконтроллером в семействе микроконтроллеров Synergy. Плата была разработана, чтобы обеспечить разработку конечных продуктов при измерении энергоэффективности устройства. Комплект состоит из трех плат: одна основная плата с микроконтроллером и несколькими разъемами и интерфейсами, одна плата с несколькими периферийными устройствами, такими как датчики света и температуры, а также разъемами Pmod™ и одна ЖК-панель с 176-сегментным дисплеем.

Уникальными особенностями платы в дополнение к уже упомянутым являются:

- Съёмная 176-сегментная ЖК-панель T6022A-1PRP0
- Расширение через штыревые разъемы и три разъема Pmod™
- Память:
 - Встроенная: 1 МБ флэш-памяти плюс 160 КБ SRAM
 - Внешняя: 32 МБ QSPI flash

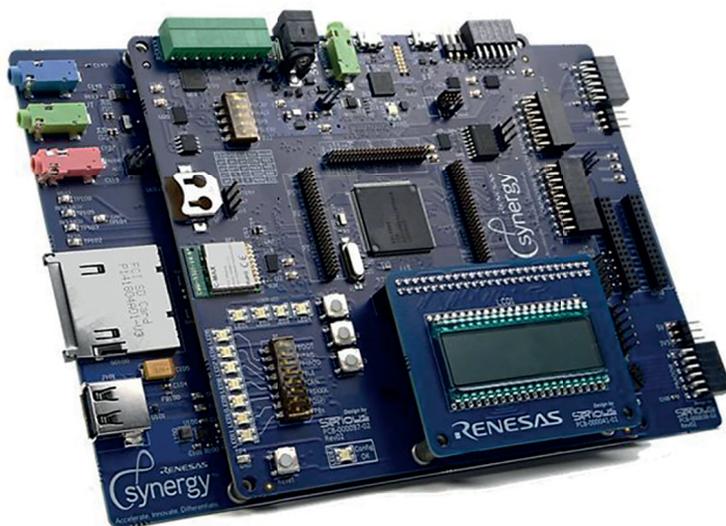


Figure 6-3: The DK-S3A7 Development Kit

6.3.3 Отладочная плата для разработки Development Kit DK-S124

В DK-S124 находится микроконтроллер S124 серии S1, малопотребляющий микроконтроллер с интеллектуальным сочетанием аналоговых и цифровых периферийных устройств. Ее конструкция позволяет делать точные измерения потребления тока микроконтроллера во всех режимах работы, а также детальную оценку производительности аналоговых периферийных устройств. Комплект состоит из двух плат, одна основная плата с микроконтроллером и одна плата дисплея со следующими функциями:

- Небольшая дополнительная плата с графическим дисплеем и интерфейсом RSK Pmod™
- Емкостные сенсорные кнопки, один слайдер
- Расширение через штыревые разъемы и один разъем Pmod™
- Встроенная флеш-память 128 кБ плюс 16 КБ SRAM

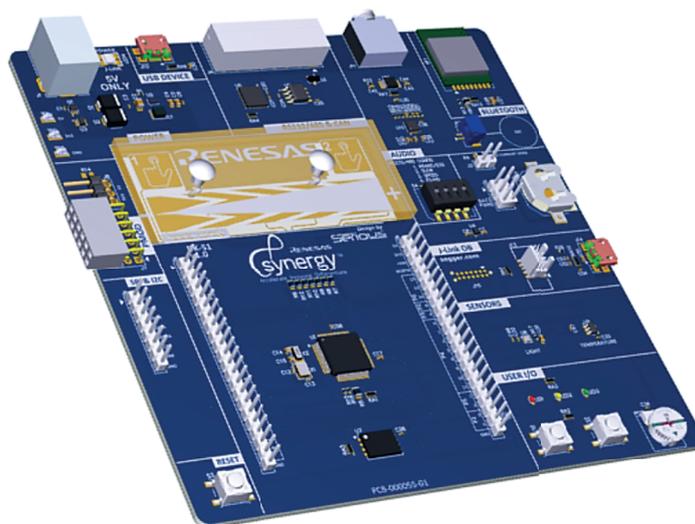


Figure 6-4: The DK-S124 Development Kit

6.4 The PK-S5D9 Promotion Kit

The PK-S5D9 Promotion Kit is your entry card to the world of the S5 series of the Synergy family of microcontrollers. It allows you to evaluate the capabilities of this series within minutes and provides you with easy-to-access interfaces to the S5D9 group of MCUs for application development. Plus, it lets you experience all the ingredients of the Synergy Platform instantly. You can also use it as initial evaluation platform to determine which other kit might be appropriate for further development of your product.

Connectivity to the outside world is available through USB, Ethernet, RS-232/485 and CAN. Four header connectors allow you to easily access most of the pins of the S5D9 microcontroller. Debugging and programming of the device is made easy through the onboard J-Link™ debugger.

An extensive set of documentation is available for this kit from the Renesas website: A Quick Start Guide walking you through the first start of the kit, a User's Manual, giving you all the technical details of the board, as well as the S5D9 data sheet and manual.

Other features of this Promotion Kit include:

- 2.4" TFTLCD QVGA (320 x 240) colour display with touchscreen which uses the internal SRAM of the S5D9 for display frame buffer memory.
- Expansion:
 - Arduino UNO Shield compatible connector
 - 2 x Pmod™ connectors
 - Pin row headers
- Wired connectivity
 - USB (1 x HS Host, 1 x FS Device)
 - Ethernet with RMII and IEEE 1588 Precision Time Protocol (PTP) support
 - PRS232/RS485 available on pins with transceiver
 - CAN available on pins with transceiver
- Wireless connectivity
 - **Bluetooth Low Energy (BLE) 4.1 on board**
- Memory
 - On-chip: 2 MB flash + 640 KB SRAM
 - External: 8-MB QSPI flash
 - PRS232/RS485 available on pins with transceiver
 - CAN available on pins with transceiver
- Other features
 - On-chip: 2 MB flash + 640 KB SRAM
 - Amplified mono audio output on a standard 3.5 mm audio jack
 - Two user buttons and three generic user LEDs

Вопросы, которые следует вынести из этой главы:

- Стартовый комплект SK-S7G2 - это ваш доступ к платформе Synergy, где вы можете проверить большинство своих функций.
- Платы разработки могут использоваться для полного прототипирования проекта, и каждая отладочная плата включает в себя разный набор внешнего оборудования для конкретных целей.

7 ПЕРВЫЙ ЗАПУСК ОТЛАДочНОЙ ПЛАТЫ RENESASSYNERGY™ STARTERKITS7G2

Что вы узнаете в этой главе:

- Как подключить отладочную плату Renesas Synergy™ к вашей рабочей станции.
- Какая существует конфигурация отладки и как ее создать.
- Как загрузить и запустить программу в среде разработки.

В этой главе мы проверим, работает ли Synergy Starter Kit SK-S7G2 и как она общается с рабочей станцией Windows® и отладчиком внутри e² studio. Для этого мы будем использовать проект, сгенерированный нами в разделе 4.2.3. Если вы не сделали этого упражнения, не волнуйтесь, вы можете скачать его с Веб-сайта для этой книги. Но в последнем случае вы должны были уже установить и протестировать набор инструментов Synergy, как описано в главе 4.2.

7.1. Подключение и демонстрация «из коробки»

Если вы этого не сделали, настало время распаковать отладочную плату из коробки. Проверьте содержимое, всё ли на месте: главная плата SK, один USB кабель типа A на микро-B и краткое руководство. Обратите внимание, что основная плата содержит SEGGER J-Link® On-Board (OB), обеспечивающая полные возможности отладки и программирования, поэтому нет необходимости во внешнем эмуляторе при использовании SK, а USBкабеля, поставляемого вместе с комплектом, вполне достаточно.

Затем вставьте малый разъем USBкабеля в разъем под названием DEBUG_USB (J-19), а другой конец - в свободный USB-порт вашей рабочей станции. Зеленый светодиод 4 в верхнем правом углу печатной платы должен загореться, указывая на то, что плата имеет питание.

Как только плата подключена, она включится и выполнит самопроверку, после чего на ЖК-дисплее отобразится заставка. Нажмите на экран, и вы войдете в предварительно запрограммированную демонстрацию термостата. Для этой демонстрации программный пакет Synergy (SSP) использовался для реализации функции считывания внутреннего датчика температуры микроконтроллера S7G2 через его аналого-цифровой преобразователь для отображения информации.

Протестируйте демонстрацию: нажав на одно из полей температуры, вентилятора или системы, вы увидите экран, в котором вы можете изменить состояние кондиционера (включен (on)/выключен (off)), охлаждение (cool), нагрев (heat) или вентилятора (автоматический режим (auto)/включен(on)) или настроить температуру. Нажав на значок настроек, который выглядит как шестерёнка, вы попадаете на экран, где вы можете вносить корректировки в систему, включая единицы измерения, время и дату.

Как только вы закончите тестировать, пришло время двигаться дальше.

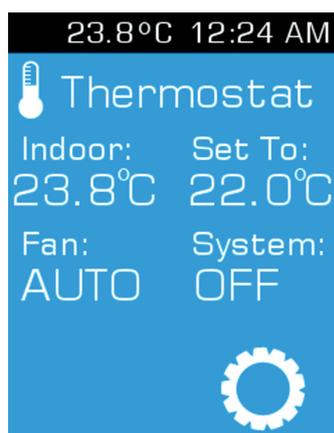


Figure 7-1: Tapping on the different items on the screen will get you to the configuration menus

7.2 Скачивание и тестирование примера

Когда SK-S7G2 подключен к вашей рабочей станции Windows®, запустите ISDE e² studio из меню «Пуск» операционной системы. Если будет предложено указать местоположение рабочей области, используйте тот, который вы ввели во время упражнения в главе 4.2.3 (он уже должен быть указан). Если вы не выполняли упражнение, не волнуйтесь, вы можете загрузить проект с веб-сайта, созданного для этой книги (www.renesas.com/synergy-book)! После загрузки импортируйте его в рабочее пространство в соответствии с инструкциями, изложенными в главе 5.1.3. В этом случае просто используйте расположение рабочего пространства по вашему выбору.

Прежде чем мы сможем загрузить программу в плату и запустить ее, сначала необходимо создать конфигурацию отладки. Нажмите на маленькую стрелку рядом с символом Debug и выберите Debug Configurations из выпадающего списка.

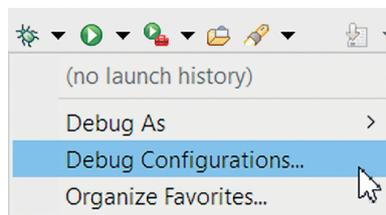


Figure 7-2: To start debugging, select Debug Configurations from the drop-down menu

В появившемся окне выделите MySynergyProject Debug под Renesas GDB Hardware Debugging. Если вы использовали другое название для этого проекта, выберите то, которое вы использовали.

При выборе вашего проекта откроется новый экран для настройки Debug Configuration, в котором будут показаны все его варианты. Не нужно ничего менять для нашей цели тестирования, просто нажмите Debug (Отладка) внизу, и начнется процесс отладки. Если откроется диалоговое окно «Подтверждение перспективы», нажмите «Да».

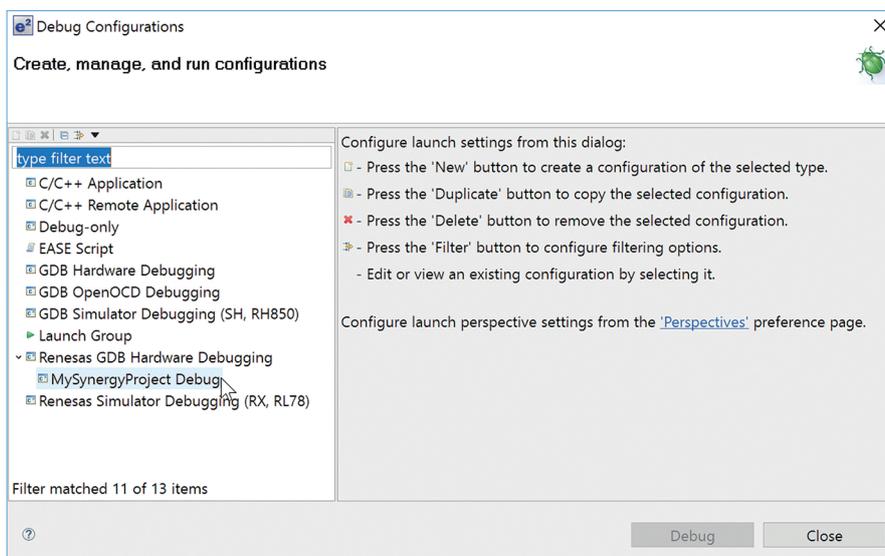


Figure 7-3: Select your project under Renesas GDB Hardware

Когда перспектива отладки открыта, отладчик установит счетчик программ в точку входа программы, обработчик сброса. Нажмите кнопку Resume (Возобновить) , и программа перейдет к следующей остановке внутри функции `main()`, в строке с вызовом `hal_entry()`. Нажмите «Повторить» еще раз, и программа продолжит выполнение, мигая зелеными, красными и оранжевыми светодиодами (светодиоды с 1 по 3) на отладочной плате за один интервал.

Последний шаг — отключить отладчик с платы, нажав кнопку Disconnect (Отключить) , что приведет к остановке выполнения программы.

Теперь, когда вы уверены, что установленная e² studio работает вместе с вашим SK, пришло время написать свою первую программу Synergy. Это будет тема следующей главы.

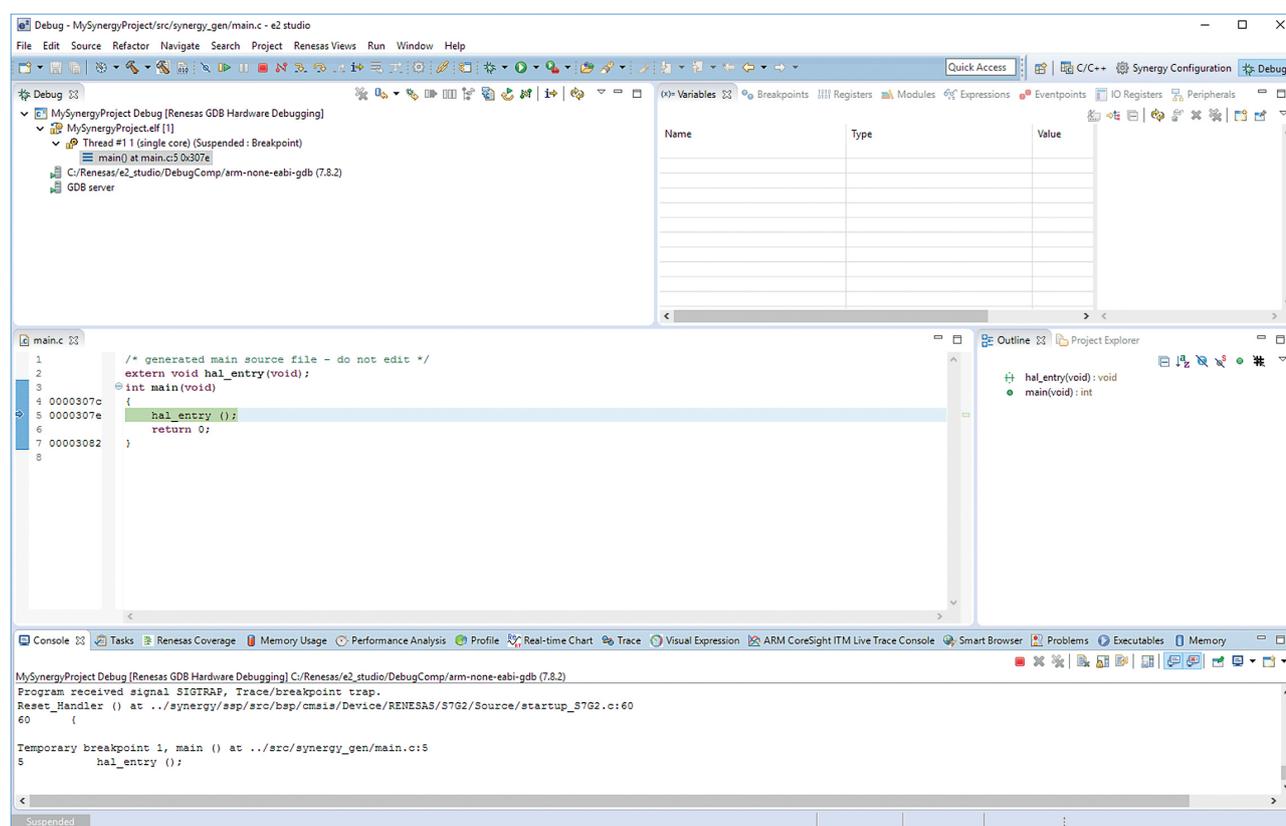


Figure 7-4: The debug perspective of e² studio

Вопросы, которые следует вынести из этой главы:

- Чтобы загрузить и отладить программу на любом оборудовании, сначала необходимо создать конфигурацию отладки.
- После загрузки отладчик установит счетчик программ в точку входа. Следующая остановка будет на `main()`.

8 HELLO WORLD! — HELLO BLINKY!

Что вы узнаете в этой главе:

- Как с нуля создать проект для отладочной платы Renesas Synergy™ SK-S7G2.
- Как изменить настройки программного пакета Synergy (SSP) в конфигураторе Synergy.
- Разработка кода для переключения светодиодов на SK.
- Как загрузить и проверить код на плате.

Самая первая программа, которую большинство новичков писали (и пишут до сих пор) на языке программирования — это та, которая просто помещает строку «Hello World» в стандартное устройство вывода. В моём случае это было набор «Writeln ('Hello World')» в редакторе, так как я начинал с Pascal. С тех пор я писал подобные строки на нескольких других языках, в основном в качестве проверки на работоспособность для установки новой среды разработки.

Когда я перешел к программированию встроенных систем в конце 1980-х, не было экрана, на который могла бы быть отправлена строка. Какие же дать инструкции процессору, чтоб они показали признаки жизни? Светодиоды едва появлялись в этих системах, поэтому выходом стало — обратиться к одному из очень немногочисленных штырьков ввода/вывода и наблюдать за волной с помощью осциллографа. С годами светодиоды стали неотъемлемой частью плат, и мы размещали их достаточно много, а их мигание использовали, как новый «Hello World».

И это является целью главы: Переключать светодиоды на SK, используя все, что вы узнали в предыдущих главах: вы будете писать код (почти) с нуля, создавать новый проект с помощью конфигураторов, использовать API-интерфейсы SSP и, наконец, загружать, отлаживать и запускать код. Это упражнение объединяет все.

В качестве предварительного условия, на рабочем столе Windows® необходимо установить e² studio или IAR Embedded Workbench® для Renesas Synergy™ (EW for Synergy), а также SSP (см. Главу 4), и вы должны убедиться, что ваша настройка работает исправно (как описано в главе 7). И те из вас, кто делал предыдущие практические упражнения: Пожалуйста, следуйте за автором, поскольку он решил, ради тех из вас, кто перешел непосредственно от предисловия к этой главе, снова рассмотреть некоторые из тем, которые уже обсуждались.

Печатная версия книги охватывает только e² studio. Если вы хотите использовать IDE из IAR Systems®, вы можете загрузить полную версию этой главы с веб-сайта книги (www.renesas.com/synergy-book), который включает в себя такое же пошаговое руководство для IAR EW для Synergy.

Для этого упражнения мы снова будем использовать SK-S7G2, который очень хорошо подходит для таких задач, поскольку он позволяет мгновенно исследовать микроконтроллер Synergy и все его периферийные устройства. Внешнее оборудование можно легко подключить, так как более 80% контактов доступны через разъемы. И с микроконтроллером Synergy группы S7G2, являющейся надмножеством, каждая функция семейства микроконтроллеров Synergy может быть оценена, а результаты позже будут применены к младшим членам семейства. На рисунке 8-1 показана блок-схема платы.

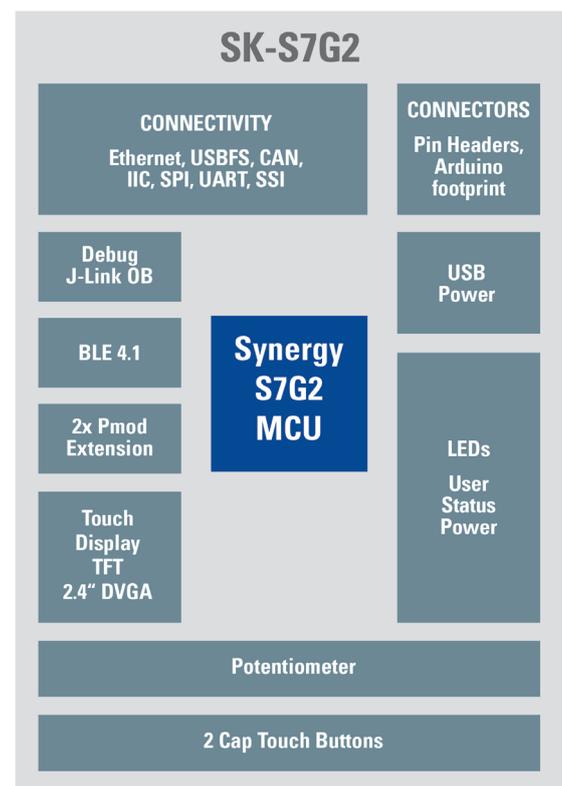


Figure 8-1: Block diagram of the SK-S7G2

8.1 Первый проект в e² studio

8.1.1 Создание проекта при помощи конфигулятора проектов

Запустите e² studio из меню «Пуск» рабочей станции Windows®, если это еще не сделано. Когда ISDE запущена, отпустите экран приветствия, если он появится, так как он заблокирует просмотр других окон.

Для написания новой программы для микроконтроллера в e² studio всегда требуется сначала создать новый проект, так что это первый шаг, который вам нужно проделать.

Для этого перейдите либо в File → New → Synergy C Project, либо щелкните правой кнопкой мыши в Project Explorer и выберите New → Synergy Project. Оба способа откроют конфигулятор проекта Project Configurator.

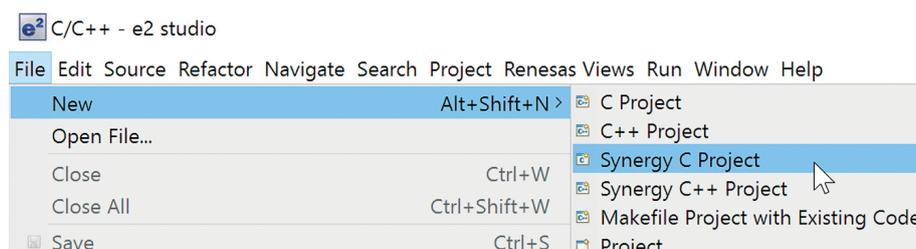


Figure 8-2: First step is to call the Project Configurator

Дайте проекту имя, например, MyBlinkyProject, и проверьте, выделен ли GCC ARM® Embedded под Toolchains. Затем убедитесь, что установлена лицензия для SSP. Если поле с информацией о лицензии пустое, нажмите «Изменить файл лицензии», и появившееся диалоговое окно направит вас к каталогу, в котором была установлена пробная лицензия во время установки (если вы установили e² studio и пакет программного обеспечения Synergy в место по умолчанию, путь будет C:\Renesas\e2_studio\internal\projectgen\arm\Licenses). Выберите файл с именем SSP_License_Example_EvalLicense_20160205.xml (или аналогичным). Если вы уже запросили и получили лицензию на разработку и производство либо от Renesas, либо от суперпользователя вашей компании, укажите в каталоге файлов на её путь. После загрузки вы можете проверить разрешения для разных компонентов SSP в окне сведений о лицензии License Details. После этого нажмите кнопку Next, чтобы перейти к экрану выбора платы.

В разделе выбора устройства Device Selection обратите внимание на поле под названием SSP version: оно должно показать ту же версию пакета программного обеспечения Synergy, который вы загрузили ранее. В меню выбора платы Board выберите S7G2 SK, так как это аппаратное обеспечение, которое мы хотим использовать для нашей небольшой программы Hello World. Убедитесь, что R7FS7G27H3A01CFC отображается рядом с устройством, оно должно быть автоматически выбрано. Если нет, перейдите в раскрывающийся список, пока не найдете его. В окне Select Tools убедитесь, что набор инструментов Toolchain, Toolchainversion и Debugger читают GCC ARM® Embedded, 4.9.3.20150529 (или более поздней версии) и J-Link ARM®. Эти поля должны быть предварительно заполнены за вас. Если нет, измените их, чтобы они соответствовали приведенным выше значениям.

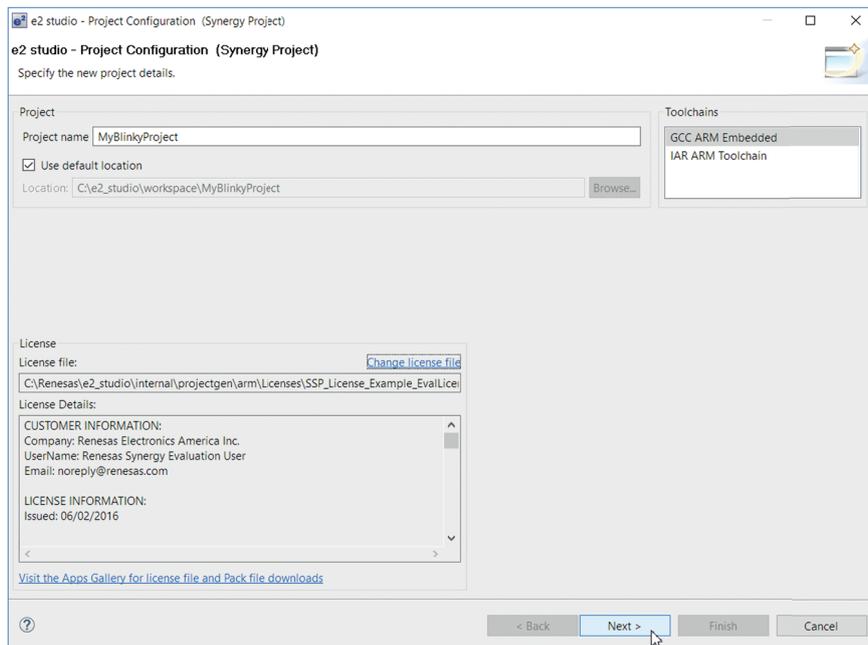


Figure 8-3: The first screen of the Project Configurator mainly asks for the project's name

После этого нажмите Next, чтобы открыть экран выбора шаблона проекта Project Template Selection. Шаблон проекта может включать несколько элементов, по крайней мере, он включает в себя правильный Пакет Поддержки Платформы (BSP) для выбранной комбинации платы/устройства. Некоторые шаблоны даже включают полный демонстрационный проект. В нашем случае выберите запись BSP, которая загрузит пакет поддержки платы для стартового набора. Нажмите Finish.

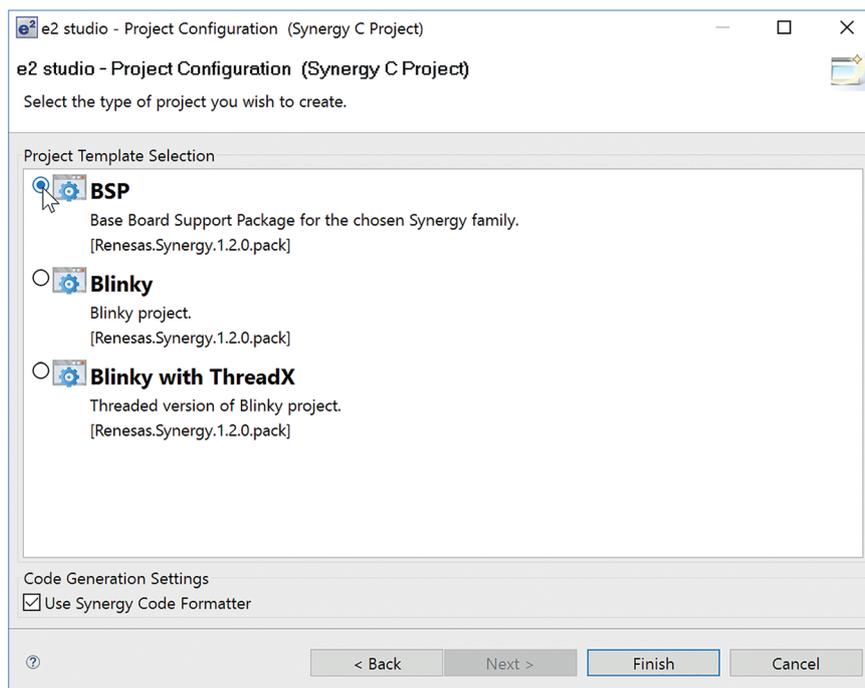


Figure 8-4: Selecting the BSP entry will load the correct Board Support Package

Конфигуратор проекта закрывается и на последнем шаге создаст все необходимые файлы для проекта. Как только эта пост-обработка будет завершена, вас спросят, хотите ли вы открыть перспективу Synergy Configuration. Выберите «Да».

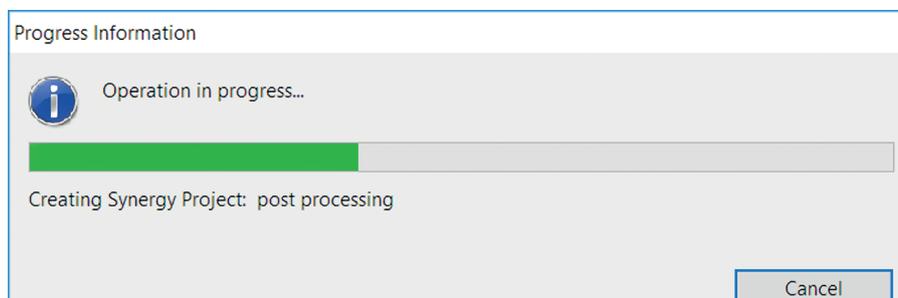


Figure 8-5: The post-processing step of the Project Configurator will create the settings and files for our project

8.1.2 Настройка среды выполнения с помощью конфигуратора Synergy

После запуска Synergy Configurator вам будет представлено краткое описание проекта и краткий обзор выбранного микроконтроллера Synergy. Там же расположен удобный ярлык к онлайн-чату, который позволяет напрямую связаться с одним из специалистов Renesas Synergy. Следующая вкладка под названием BSP позволяет просматривать и редактировать аспекты настройки платы, а на вкладке времени Clocks можно настроить начальную конфигурацию схем тактирования для проекта. Любая потенциальная проблема будет выделена красным цветом и при наведении указателя мыши, отображается объяснение. Третья вкладка в конфигураторе, называемая Pins, покрывает начальную настройку контактов устройства Synergy в проекте. Выводы могут быть перечислены либо на основе портов, либо на периферии. Пакетное представление, опять же с пометкой ошибок, доступно только с правой стороны конфигуратора, уменьшая возможные ошибки до минимума. Вкладка Threads позволяет добавлять и настраивать различные компоненты. Поскольку мы не используем OSPV в этом проекте, есть только запись HAL/Common, показывающая необходимые базовые модули. Далее находится вкладка ICU, для блока управления прерываниями. Эта вкладка ранее использовалась для просмотра и редактирования доступных источников прерываний и связанного с ними уровня приоритета, но устарела (и поэтому оставлена пустой) в последней версии e² studio, поскольку теперь прерывания настраиваются через представление свойств выбранного модуля Properties. Последняя вкладка Components отображает элементы, доступные в SSP, и какие из них в настоящее время включены в проект. Она также позволяет переключаться между различными версиями одного и того же модуля, если у вас установлено несколько версий SSP. Изменения, такие как добавление или удаление модулей, не должны выполняться в этом представлении, для этого предназначена вкладка Threads, это представление предназначено для настройки модулей.

Для нашего проекта нет необходимости изменять что-либо в этом конфигураторе, так как все необходимые настройки уже были сделаны за нас с помощью конфигуратора проектов. В качестве последнего шага в Synergy Configuration необходимо создать дополнительный исходный код, основанный на текущей конфигурации. Нажмите кнопку создания контента проекта Generate Project Content в верхней правой части конфигуратора Synergy. При этом необходимые файлы будут извлечены из SSP, скорректированы до настроек, заданных в конфигураторе и добавленных в проект.

8.1.3 Написание первых строк кода

Когда все автоматически созданные файлы теперь находятся на месте, пришло время взглянуть на то, что было создано. Проводник проекта в левой части ISDE перечисляет все, что в настоящее время включено. Папка src содержит подпапку под названием synergy_gen, содержащую настройки, установленные в конфигураторе в предыдущем шаге. В папке src также есть файл с именем hal_entry.c. Его впоследствии вам будет необходимо отредактировать. Обратите внимание, что, хотя в папке synergy_gen есть файл main.c, ваш код пользователя должен быть реализован в hal_entry.c. В противном случае вы потеряете свои изменения, если вы внесете изменения в конфигуратор Synergy и заново создаете содержимое проекта, так как этот файл будет перезаписываться каждый раз, когда вы нажимаете «Generate Project Content».

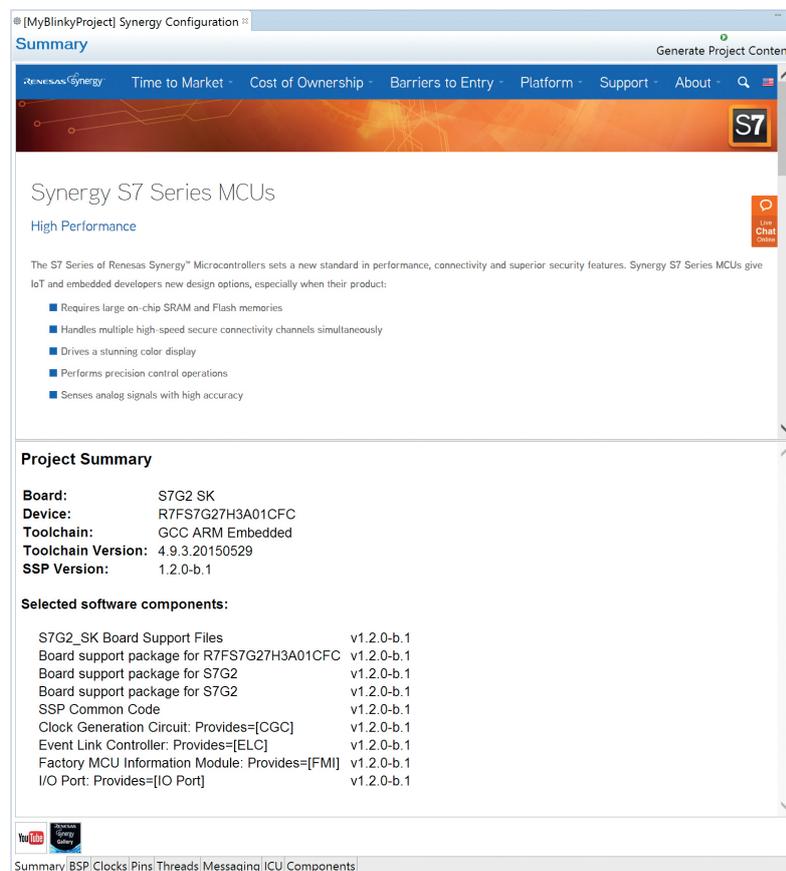


Figure 8-6: The summary tab of the Synergy Configurator

Проект также содержит несколько каталогов со словом «synergy» в названии, они содержат исходные, включенные и конфигурационные файлы для SSP. Общим правилом является то, что содержимое этих папок (и подпапок) не должно изменяться. Они содержат файлы, сгенерированные конфигуратором, и любые внесенные изменения будут потеряны при следующем создании или обновлении содержимого проекта. Исходные файлы, которые может редактировать пользователь - это те, которые находятся непосредственно в корне папки \src или в любой другой папке, добавленной вами.

Теперь для вас пришло время написать первый настоящий исходный код платформы Synergy. План состоит в том, чтобы поочередно переключаться между зеленым светодиодом 1 и оранжевым светодиодом 3 на S7G2 Starter Kit, поэтому вам нужно будет добавить код для включения и выключения и для цикла задержки. Как это сделать?

На самом деле есть два варианта: один использует API уровня аппаратных абстракций (HAL) напрямую, а второй использует HAL вместе с BSP. Какой, по вашему мнению, лучший? Вы можете посмотреть [главу 2](#), если вы не уверены в ответе.

Рассматривая код в файле `\src\synergy_gen\hal_data.c`, мы обнаруживаем следующее определение для экземпляра драйвера порта ввода-вывода `g_ioport`:

```
const ioport_instance_t g_ioport =
{ .p_api = &g_ioport_on_ioport, .p_cfg = NULL };
```

`g_ioport_on_ioport` — это структура, которая объявляет о возможных действиях для портов, указатель на нее присваивается API-указателю экземпляра `g_ioport`. Содержимое структуры можно легко просмотреть, наведя на нее мышью, это покажет, что один из ее членов, `.pinWrite`, является указателем на функцию записи на контакте.

Итак, для включения светодиодов, вы можете написать:

```
g_ioport.p_api->pinWrite(ioport_port_pin_t pin,
                        IOPORT_LEVEL_LOW);
```

Но это означает, что вам действительно нужно знать, к каким портам ввода/вывода LED1 и LED3 подключены и сколько светодиодов доступно для использования! Для этого мы могли бы либо прочитать документацию к плате, либо внимательно рассмотреть схему, чтобы найти правильный порт. Или просто используйте API BSP, который для этой цели предоставляет структуру `bsp_leds_`. Вызов функции BSP:

```
R_BSP_LedsGet(bsp_leds_t * p_leds);
```

Которая заполняет пользовательскую переменную типа `bsp_leds_t` (например, `Leds`) необходимыми значениями.

Вы можете включить LED1, написав:

```
g_ioport.p_api->pinWrite(Leds.p_leds[BSP_LED_LED1],
                        IOPORT_LEVEL_LOW);
```

Вам понадобится вторая строка, включающая, выключающее светодиод 3, установив его уровень пин-сигнала на высокий (`high`).

Наконец, нам нужно обеспечить задержку, чтобы переключение светодиодов было наглядным. Для этого мы снова вызываем BSP API:

```
R_BSP_SoftwareDelay(BSP_DELAY_UNITS_SECONDS, 1);
```

Это создаст один цикл задержки. Макрос `BSP_DELAY_UNITS_SECONDS` определяет единицы задержки, либо в секундах, миллисекундах, либо в микросекундах, в то время как 1 обозначает количество единиц для задержки. Более подробную информацию о каждом вызове функции или переменной в коде можно просмотреть с помощью функции SmartManual в e² studio.

Все, что нужно, чтобы закончить этот пример, нужно скопировать/вставить три строки кода и изменить уровни пинов светодиодов во втором наборе на противоположные. И, наконец, поскольку мы хотим запустить программу на постоянное исполнение, необходимо создать внутри цикла `while(1)`.

Таким образом должны быть набраны следующие строки кода в файл `hal_entry.c`:

```
void hal_entry(void)
{
    bsp_leds_t Leds;
    R_BSP_LedsGet(&Leds);
    while (1)
    {
        g_ioport.p_api->pinWrite(Leds.p_leds[BSP_LED_LED1],
            IOPORT_LEVEL_HIGH);
        g_ioport.p_api->pinWrite(Leds.p_leds[BSP_LED_LED3],
            IOPORT_LEVEL_LOW);
        R_BSP_SoftwareDelay(BSP_DELAY_UNITS_SECONDS, 1);

        g_ioport.p_api->pinWrite(Leds.p_leds[BSP_LED_LED1],
            IOPORT_LEVEL_LOW);
        g_ioport.p_api->pinWrite(Leds.p_leds[BSP_LED_LED3],
            IOPORT_LEVEL_HIGH);

        R_BSP_SoftwareDelay(BSP_DELAY_UNITS_SECONDS, 1);
    }
}
```

При написании кода в `e2 studio` вы всегда можете использовать функцию автозаполнения. Просто нажмите `<ctrl> + <space>` и появится окно, отображающее возможные окончания для структуры или функции. Если вы нажмете на запись, она будет автоматически вставлена в код.

Теперь ваша очередь: Пожалуйста, введите строки кода, представленные выше в файл `hal_entry.c` в вашем проекте. Для этого разверните папку `src` вашего проекта и дважды щелкните по файлу. Это откроет его в редакторе. Если вы не хотите вводить всё, вы можете скачать полный проект с веб-сайта этой книги (www.renesas.com/synergy-book).

8.1.4. Компиляция первого проекта

Когда вы набрали весь код, программа готова к созданию. Существуют две различные конфигурации для сборки: `Debug` и `Release`. Конфигурация `Debug` будет включать всю информацию, необходимую для отладки программы, например, имена переменных и функций, а также отключать определенные оптимизации компилятора, например, разворачивание цикла. Это облегчает отладку, но создаст более крупный и медленный код. Конфигурация `Release` будет удалять всю эту информацию из выходного файла и включать полную оптимизацию, тем самым создавая меньший и более быстрый код, но вы, например, больше не сможете просматривать переменные, если не будете знать их адрес в памяти.

Для вашего первого теста вам понадобится конфигурация Debug, которая также является стандартом по умолчанию. Чтобы создать проект, нажмите на кнопку создания  в строке главного меню, и процесс начнется. Если вы все сделали правильно, компиляция завершится с 0 ошибок и 0 предупреждений. Если есть ошибки компиляции, вам нужно вернуться к вашему коду и дважды проверить, правильно ли вы всё ввели. Если нет, измените свой код соответствующим образом.

После завершения сборки будет создан выходной файл MyBlinkyProject.elf, который необходимо загрузить в процессор, прежде чем мы сможем его запустить и отладить.

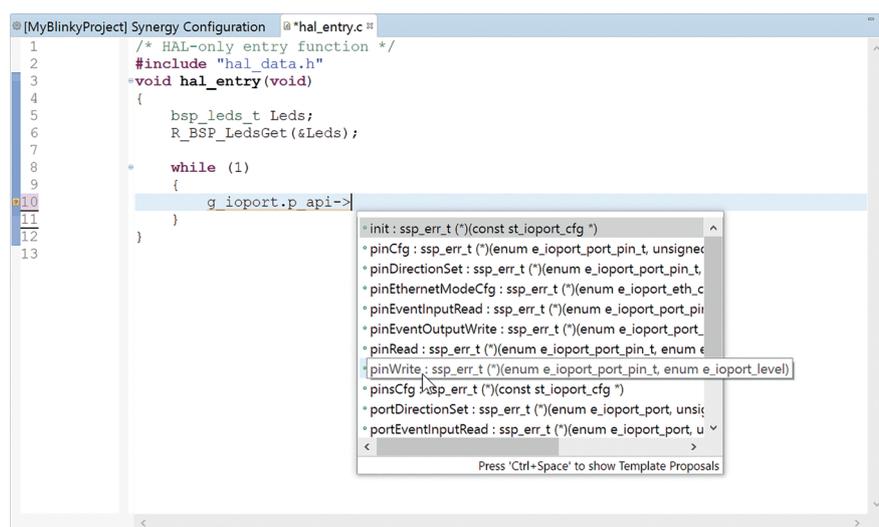


Figure 8-7: Pressing <ctrl>-<space> at a variable or function will activate the code completion feature of the editor

8.1.5 Загрузка и отладка первого проекта

Следующим шагом будет фактически запустить программу на отладочную плату. И вот сейчас подходящий момент для её подключения к рабочей станции Windows®. Вставьте меньший разъем USB-кабеля, поставляемого с платой, в разъем под названием DEBUG_USB, а другой конец - на ПК. Должен загореться зеленый светодиод в верхнем правом углу печатной платы, указывая на то, что плата имеет питание. Если вы только что достали плату из коробки, будет запущена запрограммированная демоверсия, сигнализирующая, что все работает должным образом. Windows® может отображать диалоговое окно, в котором указывается установка J-Link® OB Debugger, который должен быть завершён автоматически.

ЗАГРУЗКА

Чтобы загрузить нашу программу, нам сначала нужно создать конфигурацию отладки. Нажмите на маленькую стрелку рядом с символом Debug  и выберите Debug Configurations из выпадающего списка.

В появившемся окне под Renesas GDB Hardware Debugging выделите MyBlinkyProject Debug. Наконец, нажмите «Debug» в правом нижнем углу окна. Это запустит отладчик, передаст код на устройство Synergy и спросит вас, хотите ли вы перейти на перспективу Debug. Ответьте «Yes». Откроется перспектива Debug, а счетчик программ будет установлен в точку входа - программы, обработчик сброса. Эта конфигурация отладчика должна выполняться только один раз. В следующий раз вы можете запустить отладчик, нажав только на значок Debug.

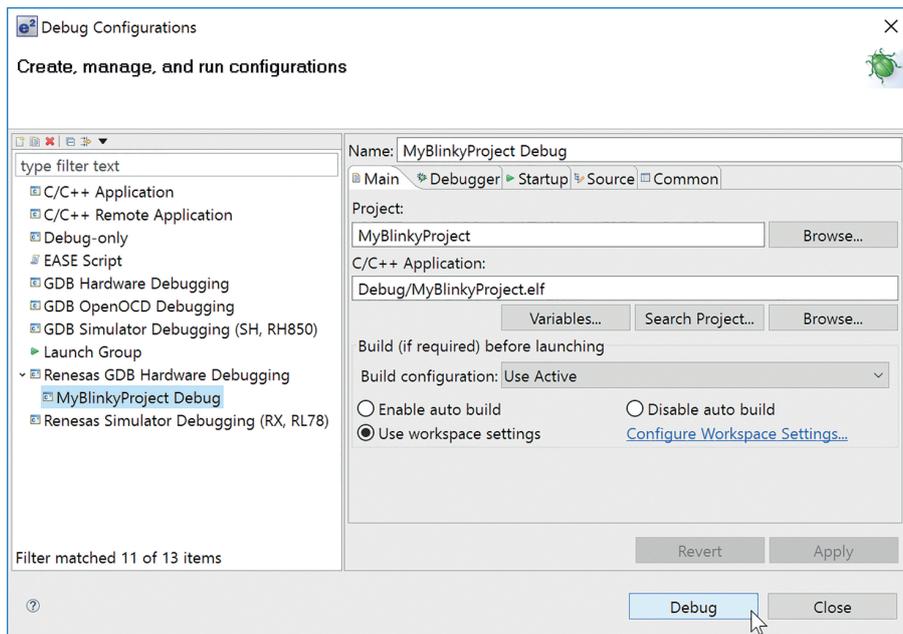


Figure 8-8: Once MyBlinkyProject is selected, no change needs to be made on the different tabs

ВЫПОЛНЕНИЕ

Нажмите кнопку «Resume» , и следующая остановка будет на `main()`, при вызове `hal_entry()`. Нажмите еще раз, и программа в Starter Kit продолжит выполнение, переключая зеленые и оранжевые светодиоды за один интервал времени, как запрограммировано.

НАБЛЮДЕНИЕ ЗА РЕЗУЛЬТАТАМИ

Если все работает должным образом, нажмите кнопку «Suspend» (Приостановить)  в строке главного меню. Это остановит выполнение программы, не завершая ее. В окне редактора активируйте вкладку с файлом `hal_entry.c` и щелкните правой кнопкой мыши в одной из строк с записью в порты. В появившемся меню выберите «Runtoline» (Выполнить до строки). Выполнение возобновится, и программа остановится в строке, на которую вы нажали. Теперь взгляните на представление со вкладкой переменных справа. Вы увидите структуру `Leds`. Разверните её, просмотрите и проанализируйте разные поля. Это представление пригодится при отладке более крупного проекта.

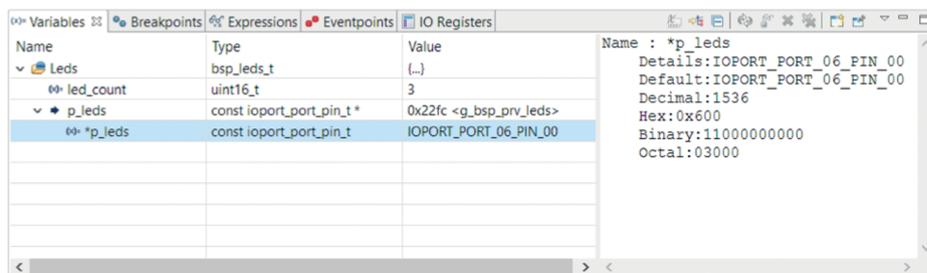


Figure 8-9: Variables and their values can be observed in the variables tab

Последний шаг - отключить отладчик от платы, нажав на кнопку разъединения , остановив выполнение программы.

ПОЗДРАВЛЯЕМ!

Вы освоили свою первую программу для платформы Synergy!

Вопросы, которые следует вынести из этой главы:

- Конфигуратор проекта (Project Configurator) создает все файлы и настройки, необходимые для нового проекта.
- Конфигуратор Synergy (Synergy Configurator) позволяет легко настраивать SSP и среду выполнения на основе графического интерфейса пользователя.
- Конфигурация отладки (Debug Configuration) необходима для отладки проекта, но она создается автоматически и должна быть активирована.
- Для реализации фактического кода требуется очень мало строк.

9 ИСПОЛЬЗОВАНИЕ ОПЕРАЦИОННОЙ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ

Что вы узнаете в этой главе:

- Какие существуют потоки, семафоры и очереди, и как их использовать.
- Как добавить потоки и семафоры в e² studio.
- Как переключить светодиод с помощью кнопки под управлением ОСПВ.

В упражнении в предыдущей главе уже использовалась часть программного пакета Renesas Synergy™ (SSP). В этой главе вы создадите небольшое приложение, использующее ОСПВ ThreadX®, используя задачку для светодиода и семафор для синхронизации с кнопкой, и вы поймете, что для этого на самом деле требуется всего несколько шагов.

Вы создадите полный проект с нуля, так что не беспокойтесь, если вы не сделали предыдущие задания.

9.1 Потоки(задачи), семафоры и очереди

Прежде чем погрузиться в это упражнение, давайте определим некоторые термины, которые мы будем использовать в этой и в следующей главе, чтобы убедиться, что у нас общее понимание.

Сначала нам нужно определить термин «Thread» (Поток). Если вы больше привыкли к термину «Task» (Задача), просто подумайте о том, что поток является своего рода задачей. Некоторые даже используют оба термина поочередно. При использовании ОСПВ приложение, работающее в микроконтроллере, будет разбито на несколько небольших полунезависимых фрагментов кода, причем каждый из них, как правило, контролирует один его аспект. И эти маленькие кусочки называются потоками(или в других источниках — задачами). В одном приложении могут существовать несколько потоков, но только один может быть активным в любой момент времени, поскольку микроконтроллеры Synergy — это одноядерные устройства. Каждый из них имеет собственное пространство стека, назначенный приоритет в отношении других потоков в приложении и может находиться в разных состояниях, таких как готовый, завершенный или спящий. В ThreadX® блок управления потоком содержит элемент, называемый `tx_thread_state`, откуда приложение может считывать текущее состояние. Межпоточная сигнализация, синхронизация или связь выполняются с помощью семафоров, очередей, мьютексов или флагов событий.

Семафор (semaphore) — это ресурс ОСПВ, который может использоваться для сигнализации событий и для синхронизации потоков (в форме поставщик-получатель). Использование семафора позволяет приложению приостанавливать поток до тех пор, пока не произойдет событие и не будет освобожден семафор. Без ОСПВ необходимо было бы постоянно проверять переменную флага или создать код для выполнения определенного действия внутри процедуры обслуживания прерываний (ISR), блокируя другие прерывания в течение некоторого времени. Использование семафоров позволяет быстро выйти из ISR и отложить исполнение кода в связанном потоке.

ThreadX® обеспечивает 32-битные счетные семафоры, и каждый семафор имеет две основные операции: `tx_semaphore_get`, которая уменьшит семафор на единицу и `tx_semaphore_set`, что увеличивает семафор на единицу. Семафоры в ThreadX® являются общими ресурсами. Стоит отметить, что семафоры, как и все объекты ThreadX®, не может вызывать приостановку вне потока, например, внутри процедуры обслуживания прерываний. Поэтому все связанные функции должны вызываться с помощью `TX_NO_WAIT` при вызове из ISR.

Последний термин, о котором нам нужно поговорить, даже если мы не используем его в этом упражнении, это очередь (queue). Но мы сделаем это в следующей главе. Очереди сообщений являются основным методом межпоточной связи. Одно или несколько сообщений могут находиться внутри очереди сообщений. Существует специальный случай: очередь с одним сообщением называется почтовым ящиком. Разрешенные размеры сообщений — от 1 до 16 32-битных слов. Большие сообщения должны передаваться указателем, что уже реализовано в SSP Messaging Framework (sf_message), и размер должен быть указан при создании очереди. Ограничений по количеству очередей в ThreadX® нет. Сообщения помещаются в очередь с помощью функции tx_queue_send() и считываются из очереди с помощью tx_queue_receive(). Новые сообщения помещаются в конце очереди, а полученные сообщения удаляются с начала очереди.

9.2 Добавление потока в ThreadX® используя e² studio

Это упражнение снова основано на комплекте Synergy Starter Kit SK-S7G2. На этот раз мы будем использовать кнопку SW4 в правом верхнем углу платы, чтобы сигнализировать о событии для приложения, которое будет переключать зеленый светодиод 1 в ответ на него. Для реализации мы будем использовать ThreadX®, обработка события будет происходить внутри потока, и уведомление об этом событии будет выполняться с помощью семафора.

Как обычно, первым шагом является создание нового проекта с использованием Project Configurator, что вы уже выполняли в главах 4 и 8. Чтобы начать работу, перейдите в меню File → New → Synergy C Project, введите название проекта, например MyRtosProject, Убедитесь, что настройки в порядке и выберите S7G2 SK на следующем экране. На экране выбора шаблона проекта выберите BSP.

Нажмите «Finish», и после того, как проект был сгенерирован конфигуратором, e² studio переключится на перспективу Synergy Configuration. Перейдите непосредственно на вкладку «Threads». На этой вкладке отобразится одна запись для HAL/Common на панели Threads, содержащая драйверы для контроллера привязки событий (Event Link Controller — ELC), драйверы порта ввода-вывода и контроллера тактирования (Clock Generation Controller — CGC) и заводской информации о микроконтроллере (Factory MCU Information — FMI). Драйвер FMI включает в себя общий API для чтения записей из таблицы Factory MCU Information Flash Table, содержащий информацию о функциях и периферийных устройствах, реализованных на используемом микроконтроллере. Нажмите на значок «New Thread» в верхней части панели, что добавит новый поток.

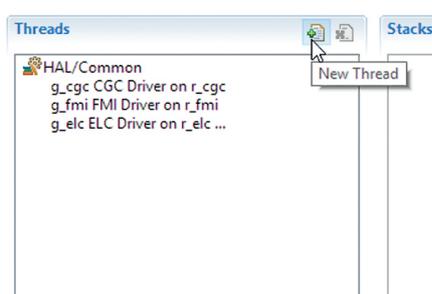


Figure 9-1: After the Synergy Configurator starts, only one thread will be shown. Clicking on the “Add Thread” symbol will add another one

Теперь измените свойства нового потока: Переименуйте символ в led_thread и его название в LED Thread в окне свойства (Properties). Оставьте остальные свойства по умолчанию. На панели «LED Thread Stacks» нажмите значок «New» и выберите Driver → Input → External IRQ Driver on r_icu (см. Рис. 9-2).

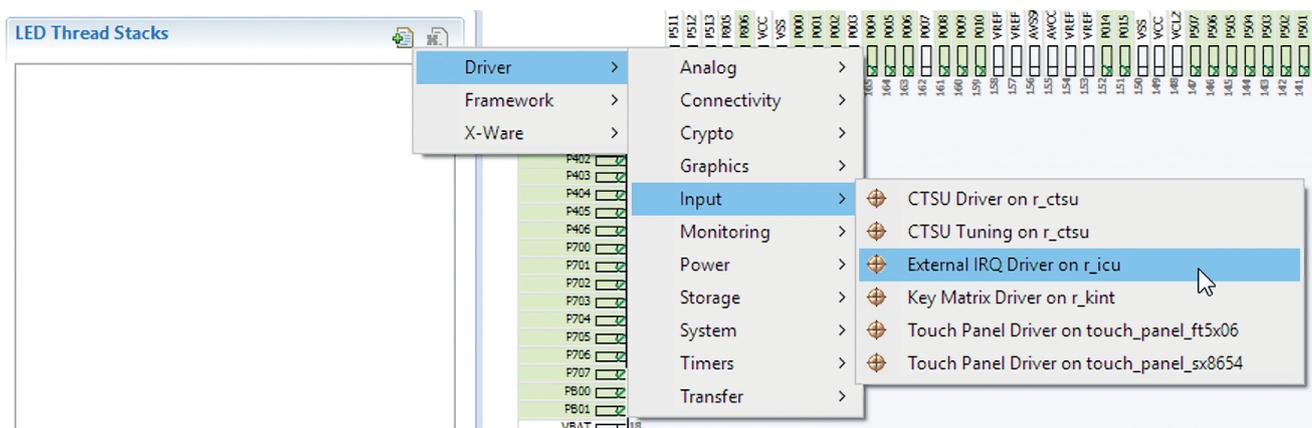


Figure 9-2: Adding a new driver takes only a few mouseclicks

Это добавит драйвер для внешнего прерывания, но он будет отображаться красным цветом, указывая на наличие дополнительных изменений. Наведите указатель мыши на него и посмотрите, что нужно сделать (рисунок 9-3).

Прерывание может быть разрешено путем изменения приоритета прерывания (InterruptPriority) от «Disabled» (Отключено) до приоритета 8 в окне «Properties» (Свойства). На самом деле, это может быть любой другой приоритет между 0 и 14, но 8 – хорошо для начала, так как вы редко сталкиваетесь с конфликтами приоритета прерывания даже в более крупных системах. Обратите внимание, что приоритет 15 зарезервирован для системного таймера (systick) и, следовательно, не может использоваться другими периферийными устройствами.

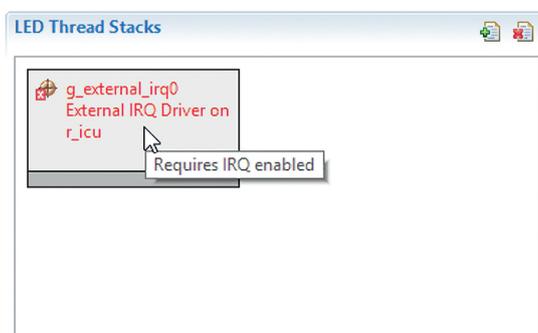


Figure 9-3: Hovering with the mouse will bring up an explanation of which property needs to be changed

Пока вы находитесь в свойствах для `g_external_irq0`, внесите некоторые другие изменения. Сначала измените канал с 0 на 11, так как SW4 подключен к IRQ11. По той же причине измените имя на `g_external_irq11` или на другое на ваш вкус.

Измените «Trigger» от Rising to Falling, чтобы поймать нажатие кнопки и Digital Filtering от Disabled до Enabled. Это поможет подавить дребезг контактов кнопки. И, наконец, измените обратный вызов с NULL на `external_irq11_callback`. Этот шаг создаст функцию, которая вызывается после нажатия SW4. В дальнейшем мы добавим код для функции обратного вызова. На рисунке 9-4 приведено краткое описание необходимых настроек.

Property	Value
▼ Common	
Parameter Checking	Default (BSP)
▼ Module g_external_irq11 External IRQ Driver on r_icu	
Name	g_external_irq11
Channel	11
Trigger	Falling
Digital Filtering	Enabled
Digital Filtering Sample Clock (Only valid when Di	PCLK / 64
Interrupt enabled after initialization	True
Callback	external_irq11_callback
Interrupt Priority	Priority 8 (CM4: valid, CM0+: invalid)

Figure 9-4: The properties of the IRQ driver needed for our application

Теперь вам нужно выполнить только несколько дополнительных шагов, до того момента, когда вы сможете скомпилировать и загрузить свою программу. Следующий шаг - добавить семафор.

Для этого нажмите кнопку «New Object» на панели объектов потока «LEDthread». Если вы не видите эту область, но видите панель HAL/Common Objects, выделите LED Thread на панели Threads, и она станет видимой. Добавьте Семафор, нам понадобится один, чтобы уведомить LED Thread о нажатии кнопки. Измените свойство Name семафора на SW4 Semaphore и свойство Symbol на g_sw4_semaphore.

Оставьте счетчик равным нулю, так как мы будем увеличивать его каждый раз при нажатии кнопки SW4. Теперь вкладка «Threads» в конфигураторе Synergy должна выглядеть так, как показано на рисунке 9-5.

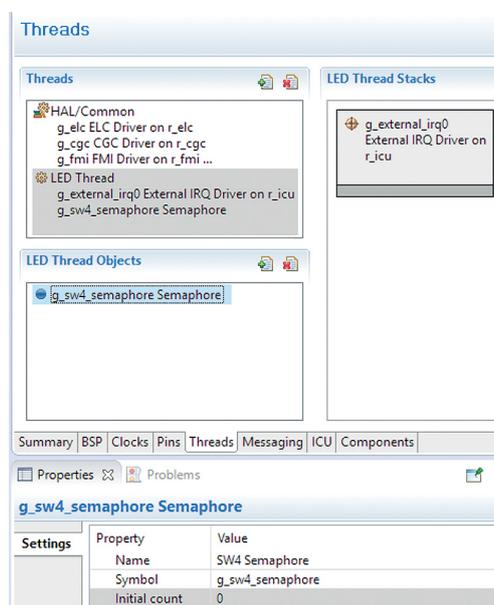


Figure 9-5: This is what the Threads tab should look like after the LED Thread and the semaphore has been added

Последним шагом в конфигураторе Synergy является настройка контакта ввода вывода, к которому SW4 подключен как вход IRQ11. Для этого активируйте вкладку «Pins» внутри конфигулятора и разверните Ports → P0 и выделите P006. На стартовом комплекте S7G2 подключается контакт SW4. На панели «Pin Configuration» справа введите в поле «SymbolicName» SW4, измените режим на режим ввода и замените IRQ на IRQ11_DS (если он еще не установлен), а Chip input/output на GPIO. Обратите внимание, что в окне просмотра пакетов справа выделится контакт 163/P006. Полная конфигурация показана на рисунке 9-6.

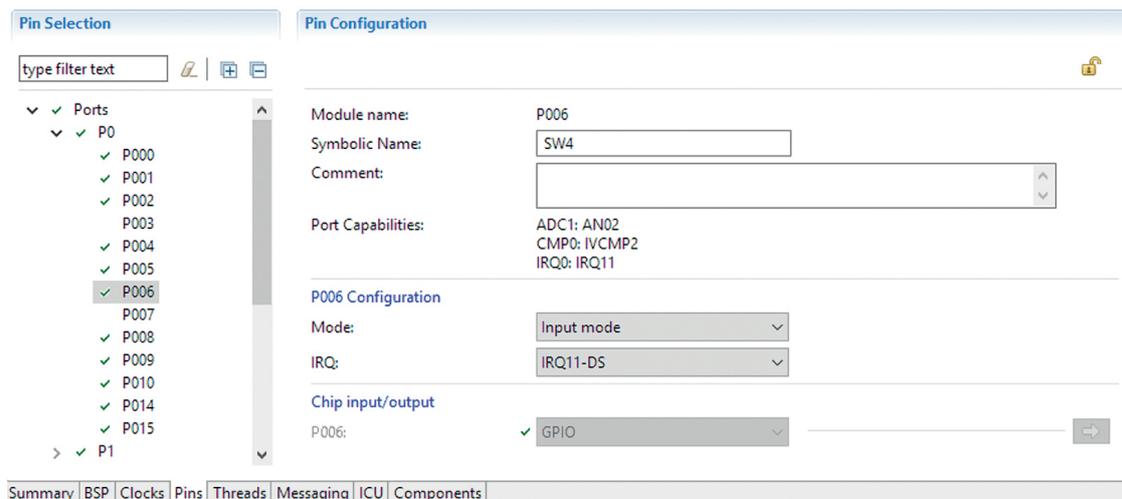


Figure 9-6: Port P006 needs to be configured as input to IRQ11

После этого настройки в конфигураторе завершены. Сохраните изменения и нажмите на значок «Generate Project Content» вверху, чтобы сгенерировать необходимые файлы, папки и настройки.

Конечная задача, которую вам нужно выполнить - добавить код для заполнения структуры Leds, аналогичный тому, что вы уже делали в упражнении в главе 8. Необходимо написать пару строк для переключения светодиода и чтения семафора и создать функцию обратного вызова. Полный код для этого представлен ниже.

Поскольку мы используем поток для управления кнопкой и переключением светодиода, на этот раз нам нужно добавить связанный код в файл `led_thread_entry.c`. Дважды щелкните файл в проводнике проекта, чтобы открыть его в редакторе. Если он не отображается, разверните папку проекта, а затем каталог `src`. Как и в упражнении в главе 8, вам нужно будет добавить структуру для светодиодов и инициализировать ее. Далее определите еще одну переменную для сохранения текущего уровня ввода-вывода, к которому подключен светодиод 1. Назовите его `led_level`. Он должен быть типа `ioport_level_t` и должен быть инициализирован до `IOPORT_LEVEL_HIGH` (уровень «high» соответствует выключенному светодиоду на SK-S7G2).

Следующим шагом будет открытие и настройка внешнего контакта IRQ, подключенного к SW4 на плате. Для этого используйте открытую функцию драйвера IRQ HAL. При этом завершаются необходимые шаги инициализации.

Внутри цикла `while(1)` вам нужно добавить пару операторов и удалить инструкцию `tx_thread_sleep(1)`. Начните с вызова функции, чтобы записать значение уровня `led_level` в выходной регистр контакта ввода/вывода для светодиода 1, за которым следуют инструкции по переключению уровня контакта. Существует несколько способов сделать это. Внесите свои собственные или посмотрите код в конце главы. Не забывайте функцию `Smart Manual e2 studio`, это очень помогает!

Конечным оператором внутри `while(1)` является вызов `tx_semaphore_get()` с адресом семафора и константой `TX_WAIT_FOREVER` в качестве параметров. Это поможет ОСПВбесконечно приостанавливать поток до тех пор, пока семафор не будет установлен из функции обратного вызова в рамках процедуры обслуживания прерываний IRQ 11.

И последнее, что нужно сделать, это добавить процедуру обратного вызова, вызываемую IRQ 11 ISR. Этот код должен быть как можно короче, поскольку он будет выполнен в контексте процедуры обслуживания прерываний. Вы можете найти прототип этой функции в коде HAL/Thread, сгенерированном в папке `synergy_gen`. В нашем случае его можно скопировать из находящегося там файла `led_thread.h`:

```
void external_irq11_callback(external_irq_callback_args_t *p_args);
```

Внутри функции обратного вызова добавьте следующую строку кода для установки семафора после нажатия кнопки SW4:

```
tx_semaphore_put(&g_sw4_semaphore);
```

Когда все написание кода будет завершено, создайте свой проект, щелкнув значок Build («молоток»). Если появится какое-либо предупреждение, пожалуйста, игнорируйте. Если он не скомпилирован с нулевыми ошибками, вернитесь к своему коду и устраните проблемы с помощью обратной связи компилятора, просматривая их в представлении Problems.

Если проект был успешно выполнен, нажмите на маленькую стрелку рядом с иконкой Debug, выберите Debug Configurations и разверните Renesas GDB Hardware Debugging. Выберите MyRtosProject или имя, которое вы дали своей версии, и нажмите «Debug». Это запустит отладчик. Если вам нужна дополнительная информация, просмотрите соответствующий раздел в главе 8. После того как отладчик запущен и работает, дважды нажмите «Resume». Ваша программа теперь запущена, и каждый раз, когда вы нажимаете SW4 на SK, зеленый светодиод 1 должен переключаться.

```
#include "led_thread.h"
void led_thread_entry(void);
bsp_leds_t Leds;
/* LED Thread entry function */
void led_thread_entry(void)
{
    ioport_level_t led_level = IOPORT_LEVEL_HIGH;
    R_BSP_LedsGet(&Leds);
    g_external_irq11.p_api->open(g_external_irq11.p_ctrl,
                               g_external_irq11.p_cfg);
    while (1)
    {
        g_ioport.p_api->pinWrite(Leds.p_leds[BSP_LED_LED1],
                                led_level);
        if (led_level == IOPORT_LEVEL_HIGH)
        {
            led_level = IOPORT_LEVEL_LOW;
        }
        else
        {
            led_level = IOPORT_LEVEL_HIGH;
        }
        tx_semaphore_get(&g_sw4_semaphore, TX_WAIT_FOREVER);
    }
}
void external_irq11_callback(external_irq_callback_args_t * p_args)
{
    tx_semaphore_put(&g_sw4_semaphore);
}
```

ПОЗДРАВЛЯЕМ!

Вы успешно завершили это задание!

Вопросы, которые следует вынести из этой главы:

- Использовать HAL-драйвера просто через использование всесторонних API.
- SSP позаботится о большинстве вещей, не связанных с пользователем.
- Использование ThreadX® является простым, а добавление потоков и семафоров — не трудоёмкая работа.

10 ПЕРЕДАЧА ДАННЫХ ПО USB ИСПОЛЬЗУЯ ОЧЕРЕДЬ

Что вы узнаете в этой главе:

- Как настроить USB-передачу с использованием одного из фреймворков приложений Renesas Synergy™ и как получать данные на рабочей станции.

В этой главе для отправки состояния LED1 через USB-порт на рабочую станцию Windows® мы будем использовать фреймворк в Application Frameworks. Для этого вам будет необходимо добавить новый поток и очередь в свой проект из предыдущей главы 9 и изменить существующий код в потоке(задаче) управления светодиодом, чтобы отправить состояние в виде строки в очередь. Затем новый поток(задача)связи отправит строку в терминальную программу на хостс помощью USB. Вы снова почувствуете простоту, которую предоставляет пакет услуг Synergy Software Package (SSP), даже при настройке сложных коммуникаций, таких как USB.

Если вы не выполнили упражнение из главы 9, вы можете загрузить проект со страницы книги (www.renesas.com/synergy-book). Импортируйте его в соответствии с инструкциями, приведенными в главе 5.1.3, и все будет установлено. Конечно, вы можете загрузить готовый проект для этой главы, если вы просто хотите увидеть, как достигнута цель. Если вы используете IAR Embedded Workbench® для Renesas Synergy™ (EW for Synergy), вы также можете загрузить полный проект, если вы не хотите следовать приведенным ниже инструкциям и при необходимости настраивать их.

10.1 Настройка порта USB в платформе Synergy

Если вы закрыли e² studio после последнего упражнения, повторно откройте ее и убедитесь, что ваш проект MyRtosProject активен. Если нет, просто нажмите на него, и он будет выделен, и она будет выделена жирным шрифтом. Первым шагом для вас является переход на перспективу Synergy Configuration и переход к представлению Synergy Configuration [MyRtosProject]. Если это представление было закрыто раньше, вы можете повторно открыть его, дважды щелкнув файл configuration.xml в представлении Project Explorer или щелкнув значок шестерёнки на панели главного меню. Поскольку вы уже являетесь экспертом по Synergy, я не буду описывать каждый шаг в деталях, поскольку они уже подробно описаны в предыдущем упражнении.

Перейдите на вкладку «Threads» и добавьте новый поток со свойствами “Symbol” — comms_thread и именем comms_thread и именем «Comms Thread». С выбранным добавленным потоком добавьте к нему экземпляр фреймворка связи. Нажмите на значок «Add Stack» в панели «Comms Thread Stack» и выберите New → Framework → Connectivity → Communications Framework on sf_el_ux_comms.

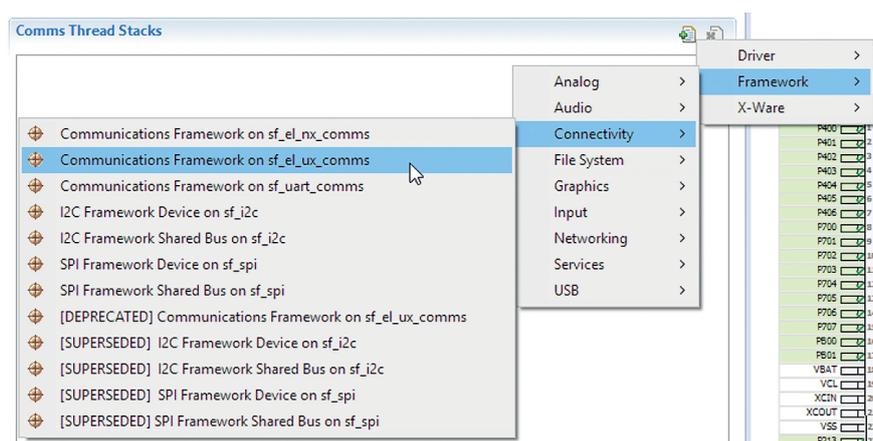


Figure 10-1: First step is to add the communications framework

Это добавит в систему полный стек `g_sf_comms0 CommunicationsFrameworksf_el_ux_comms` до уровня, где требуется вмешательство пользователя. Вы можете задаться вопросом, каково назначение у различных цветовых подсветок потоков. Это довольно просто: обычные экземпляры отмечены серым цветом, общие экземпляры, которые могут использоваться не только в одном потоке, отмечены синим цветом (всего один глобальный экземпляр на проект) и розовым отмечены модули, которые можно опционально заменять на другие, с аналогичными интерфейсами.

Для нашего проекта необходим один USBX-порт. Чтобы добавить его, щелкните по дополнительному модулю `Add USBX Port DCD` и выберите `New` и `USBX Port DCD on sf_el_ux for USBFS`. В свойствах этого модуля измените свойство приоритета прерывания полной скорости, например, приоритет 8 (см. Рис. 10-2).

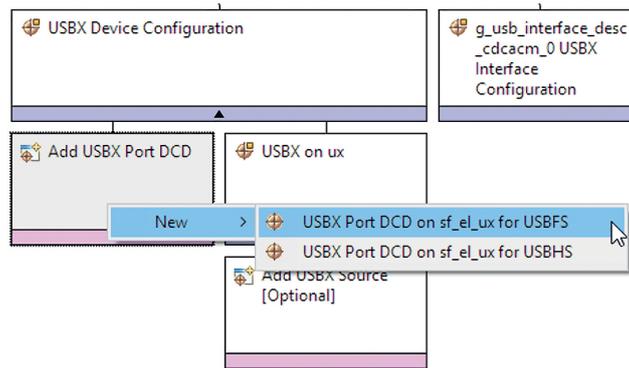


Figure 10-2: An USB Full-Speed port needs to be added to the system

Вы также можете посмотреть размер памяти USBX Pool в свойствах USBX на `ux`-модуле. По умолчанию он равен 18 кбайтам, что достаточно для большинства задач. Если вы хотите использовать разные классы USB в будущем, вам необходимо соответствующим образом изменить размер пула. С подробной информацией об этом можно ознакомиться в таблице 20 документации USBX в файле `_express_logic_u_s_b_x.html`, которая была скопирована на компьютер во время установки SSP.

Пока поток `Comms Thread` все еще выбран, добавьте нужную нам очередь (`Queue`) для отправки данных в панели объектов `Comms Thread`. Назовите его `CDC Queue` и присвойте ему символ `g_cdc_queue` в представлении «Properties». Также установите размер сообщения равным 3, так как мы хотим передать 3 слова по 4 байта в каждой передаче, а размер очереди - 24 байта, что означает, что в очереди есть место для двух (12-байтовых) сообщений.

На этом конфигурация SSP завершена. Сохраните конфигурацию и нажмите кнопку «Generate Project Content». Обратите внимание, что файл `comms_thread_entry.c` был добавлен в `src`-папку вашего проекта. Вы добавите код в этот файл во время следующих шагов.

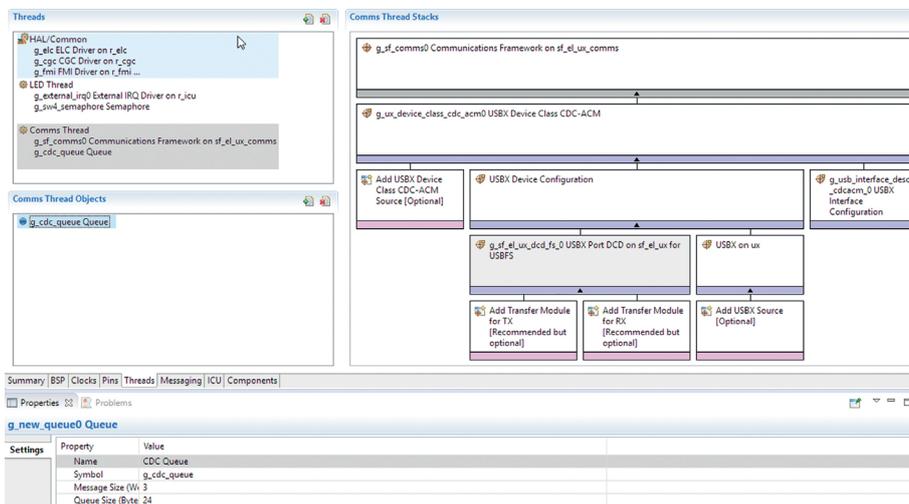


Figure 10-3: With all additions and modifications made, the Threads tab should look like this

10.2 Передача сообщений

Перед написанием кода, необходимого для передачи состояния светодиода на рабочую станцию, вам нужно будет добавить код в LED Thread, который будет копировать сообщение в очередь. Для этого вернитесь к перспективе C/C++ и откройте файл `led_thread_entry.c`. В верхней части добавьте директиву `include` для файла `comms_thread.h`, чтобы совместно с кодом в LED Thread использовать элементы из Comms Thread, такие как `g_cdc_queue`. В качестве второго шага добавьте глобальный массив с именем `send_str` с 12 символами. Помните, что это размер нашей очереди. И, наконец, добавьте следующие строки сразу после инструкции записи `ioport` внутри цикла `while(1)` в этом файле:

```
g_ioport.p_api->pinWrite(Leds.p_leds[BSP_LED_LED1], led_level);

if(led_level == IOPORT_LEVEL_HIGH)
{
    strcpy(send_str, "LED off\n\r");
    led_level = IOPORT_LEVEL_LOW;
}
else
{
    strcpy(send_str, "LED on\n\r");
    led_level = IOPORT_LEVEL_LOW;
}

/* Send the message in the queue. Wait forever for space */
/* to be available in the queue for the message. */
tx_queue_send(&g_cdc_queue, send_str, TX_WAIT_FOREVER);

tx_semaphore_get(&g_sw4_semaphore, TX_WAIT_FOREVER);
```

Последний шаг - добавить код в файл `comms_thread_entry.c`. Откройте его из Project Explorer.

Отредактируйте файл `comms_thread_entry.c` и добавьте глобальный массив из 12 элементов типа `uint8_t` с именем `rx_msg`. После этого замените строку `tx_thread_sleep(1)`. Внутри цикла `while(1)` пропишите следующее:

```
tx_queue_receive(&g_cdc_queue, rx_msg, TX_WAIT_FOREVER);

g_sf_comms0.p_api->write(g_sf_comms0.p_ctrl,
                        rx_msg,
                        strlen(rx_msg),
                        TX_WAIT_FOREVER);
```

На этом процесс написания необходимого кода завершён. Вы можете спросить: «Постойте, две строки кода — это все, что необходимо для передачи строки, полученной при помощи очереди через USB? Разве это не слишком легко?» И вы получите ответ: «Да, это так просто». SSP и фреймворки приложений позаботятся обо всем остальном. Это так просто, не так ли? Помните последний раз, когда вы писали код для передачи на другой платформе? Я предполагаю, что вы чувствуете огромную разницу в усилиях, необходимых для этого.

Теперь все, что осталось, — это построить проект. В первый раз, когда вы это сделаете, будет потрачено некоторое время, так как код для фреймворка связи должен быть скомпилирован. После создания проекта с нулевыми ошибками (вы можете получить некоторые предупреждения о неиспользуемых параметрах, просто проигнорируйте их), запустите сеанс отладки и открыв перспективу отладки Debug, дважды нажмите «Resume», чтобы запустить программу. В качестве первого теста нажмите SW4 один раз, чтобы увидеть, переключается ли LED1.

10.3 Настройка приемника на стороне хоста

При запуске программы подключите второй USB-кабель типа «А на микро-В» к USB порту, обозначенному J5 в нижней части платы. Вставьте другой конец в рабочую станцию Windows® и подождите несколько секунд, пока Windows® распознаёт плату и устанавливает драйверы для нее.

Запустите программу эмулятора терминала. Во время выполнения этого упражнения мы использовали Tera Term Pro, которую можно загрузить с <https://tssh2.osdn.jp/>, вы обнаружите, что она довольно полезна. В терминале Tera вы увидите список последовательных портов CDC. На рисунке 10-3 это COM6, но он, вероятно, будет отличаться на разных рабочих станциях. Если вы не уверены, используйте диспетчер устройств Windows®, чтобы узнать порт, к которому подключена плата.

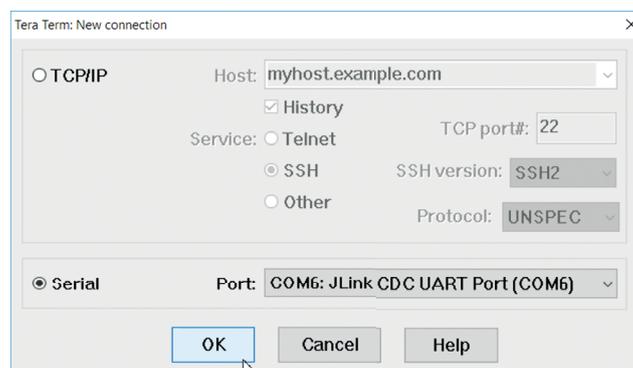


Figure 10-4: If Windows® recognized the board correctly, it will be listed in Tera Term as serial connection

В случае, если плата вообще не обнаружена или диспетчер устройств указывает на ошибку, могла возникнуть проблема с драйвером. Чтобы решить эту проблему, пожалуйста, обратитесь к следующей статье в базе знаний Renesas Synergy™: https://en-us.uknowledgebase.renesas.com/English_Content/Renesas_Synergy/_Platform/Renesas_Synergy_Knowledge_Base/Installing_USB_CDC_Driver_on_Windows_10.

При подключении и запуске Tera Term несколько раз нажмите SW4, и вы увидите переключение LED1 и состояние его выхода на терминале, как показано на рисунке 10-4.

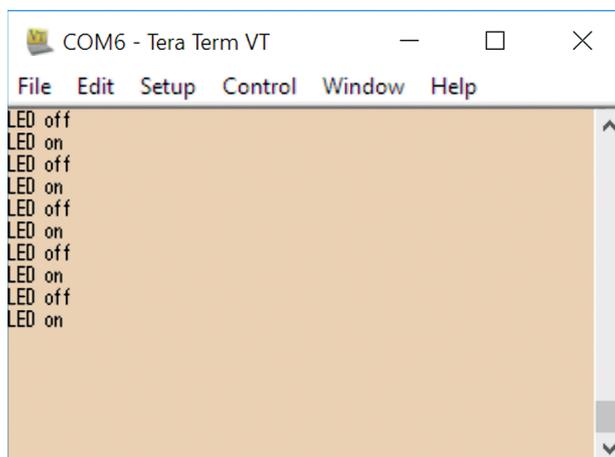


Figure 10-5: With the transfer running, the terminal program will display the state of LED1 each time SW4 is pressed

ПОЗДРАВЛЯЕМ!

Вы успешно завершили это задание! И что вы думаете сейчас: легко ли это или нет?

Вопросы, которые следует вынести из этой главы:

- Добавление USB-передачи осуществляется легко, если используются фреймворк приложений.
- Для реализации требуется всего несколько строк кода.

11 АНАЛИЗ СОБЫТИЙ В TRACEX®

Что вы узнаете в этой главе:

- Что такое TraceX® и как он может помочь вам отладить ваш код.
- Как установить и использовать TraceX® и как интерпретировать собранные данные.

Трассировка событий во время выполнения являются важной возможностью отладки, поскольку она обеспечивает видимость последовательности действий операционной системы реального времени (ОСРВ) приложения.

В этой главе мы используем программу TraceX® — инструмент, исполняемый на рабочей станции Windows®, предназначенный для отслеживания всех событий, которые происходят, когда мы нажимаем кнопку SW4 на плате Renesas Synergy™ Starter Kit (SK) в упражнении из главы 10. Для этого мы изменим программу, которую мы там использовали. Если вы не выполнили практическую работу в главе 10, это не проблема. Вы можете скачать полный проект с веб-сайта для книги (www.renesas.com/synergy-book). После загрузки вам нужно только импортировать его в соответствии с инструкциями, приведенными в главе 5.1.3.

11.1 Введение в TraceX®

TraceX® запускается на ПК и легко интегрируется с программным пакетом Synergy (SSP). Она предоставляет разработчикам графическое представление событий в реальном времени, позволяя им лучше понимать поведение своей системы. Визуализируется появление и последовательность системных событий, таких как прерывания, переключение контекста или установка семафора, события, которые обычно не видны из стандартных инструментов отладки. Таким образом, проблемы могут быть легко диагностированы, а система настроена на оптимизацию производительности и эффективности.

Сама информация о трассировке хранится в буфере целевой системы, местоположение и размера буфера, указываются в свойствах модуля TraceX® во время конфигурации. TraceX® реализует циклический буфер, который позволяет получать самые последние «N» событий для проверки даже в случае сбоя системы или других значимых событий.

Другие функции и преимущества TraceX® включают:

- Полная интеграция с ThreadX® и всеми другими компонентами X-Ware™, но она работает независимо от команд отладки с хоста или точек останова
- Обнаружение инверсий приоритета
- Профилирование выполнения задачи
- Использование стека
- Задержки (в тактах) между событиями
- Анализ эффективности и устранение узких мест
- Отображает статистику NetX™ и FileX®
- Обработывает сырые данные из буфера трассировки или его дампа

Все эти функции позволяют сэкономить много времени при разработке приложения. Нет необходимости использовать в вашей программе операторы printf () или код, переключающий светодиоды, чтобы визуализировать последовательность событий, о которых идет речь, если вам нужно изолировать определенную проблему в коде. Это действительно поможет вам сэкономить время и энергию!

11.2 Использование TraceX® в e² studio

В этой части главы мы рассмотрим использование TraceX® только вместе с e² studio. Если вы хотите использовать IAR Embedded Workbench® для Renesas Synergy™ (EW for Synergy), загрузите веб-версию этой главы.

11.2.1 Установка

TraceX® не входит в стандартную установку e² studio и необходимо загрузить отдельно от галереи Synergy. В галерее Synergy выберите Инструменты разработки, а затем TraceX. Появится главная страница TraceX®, нажмите на синюю кнопку загрузки вверху, и она начнется. Прочтите и примите появившееся лицензионное соглашение, загрузится архив установщика с именем TraceX_for_Renesas_Synergy.zip.

Во время загрузки установщика, пожалуйста, найдите время, чтобы просмотреть информацию на вкладке «Документация». Инструкции по установке, приведенные там, очень важны, и вы должны внимательно следить за ними. Они также предоставляют вам необходимую информацию, если автоматическая установка файла лицензии не удалась. Затем эту информацию можно также просмотреть, открыв файл TraceX-readme.txt, который помещается в папку установки TraceX® после распаковки архива.

Как только загрузка будет завершена, извлеките файлы из загруженного архива в новую папку по вашему выбору. Чтобы запустить процесс настройки, щелкните правой кнопкой мыши файл setup.exe и выберите «Запуск от имени администратора».

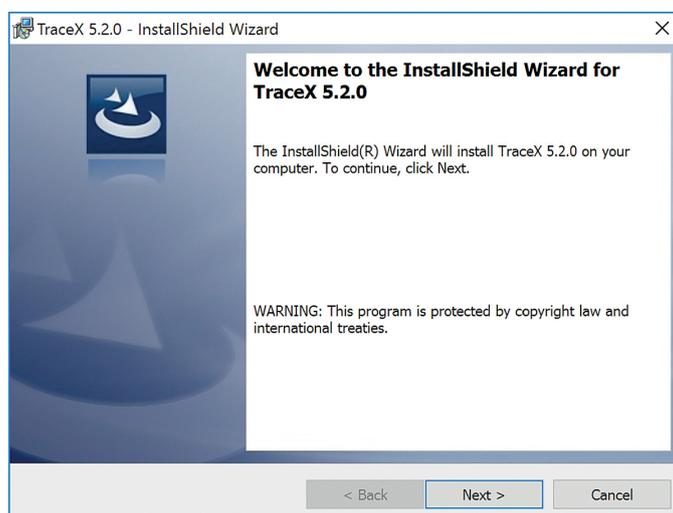


Figure 11-1: The installer for TraceX®

Примите лицензионное соглашение для программного обеспечения. Следуйте инструкциям установщика, а затем либо примите путь для установки по умолчанию, либо измените его на выбранное вами местоположение. Если вы это сделаете, обратите внимание на новое местоположение, которое вам понадобится позже, чтобы зарегистрировать инструмент внутри e² studio. На экране, отображающем сводку, нажмите «Установить», и установка начнется.

По завершении установки на последнем экране появится флажок Запустить программу. Он выбран по умолчанию, но мы сделаем это позже изнутри ISDE. В настоящий момент, пожалуйста, снимите флажок.

11.2.2 Настройка TraceX®

Поскольку TraceX® для e² studio является внешней программой, нам нужно сообщить ISDE, где можно найти исполняемый файл, чтобы мы могли запускать его, не выходя из среды разработки.

Внутри e² studio перейдите в Window → Preferences и сначала разверните C/C++, а затем ветку Renesas. Выберите TraceX и перейдите к ней или введите правильное местоположение, где вы установили исполняемый файл TraceX®. Если вы установили его в местоположение по умолчанию, это будет C:\Express_Logic\TraceX_5.2.0\TraceX.exe. Нажмите «Применить» и «ОК».

Следующим шагом является настройка приложения для выполнения фактического сбора событий трассировки. Чтобы включить трассировку, нам нужно будет добавить исходный код (Source)ThreadX® и USBX™.

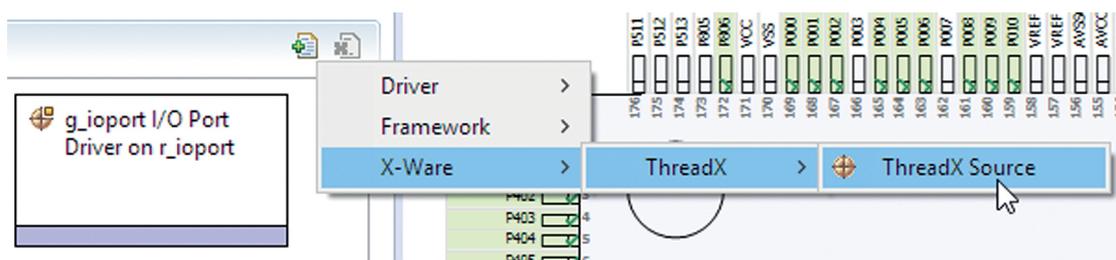


Figure 11-2: The ThreadX® source code needs to be added to the project in order to use TraceX®

Для этого переключитесь на перспективу Synergy Configuration в e² studio и убедитесь, что ваш проект с OCPB из последней главы активен. Перейдите на вкладку «Threads» и выберите «HAL/Common Module» на панели «Threads». Добавьте источник ThreadX®, нажав кнопку New → X-Ware → ThreadX → ThreadX Source в панели HAL/Common Stacks (см. Рис. 11-2). Выберите недавно добавленный источник ThreadX и в представлении «Properties» прокрутите вниз до «Event Trace» и измените его с «Отключено» на «Включено» (Enabled). Обратите внимание на имя буфера трассировки (g_tx_trace_buffer) и размер буфера (65536). Вам понадобится эта информация позже.

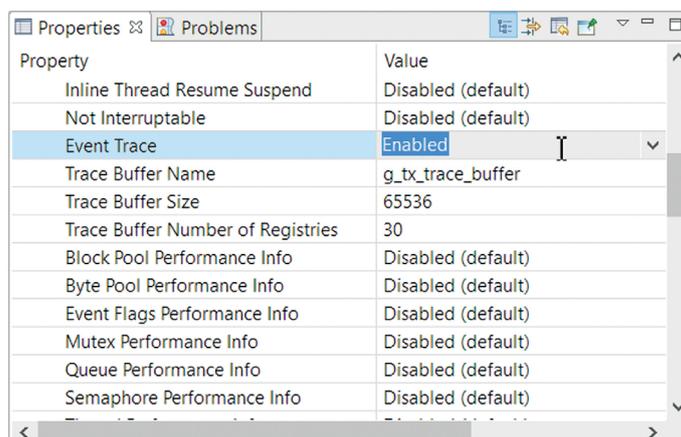


Figure 11-3: The event trace needs to be enabled in the properties of the ThreadX®

Теперь добавьте источник USBX™. Выберите пункт «Comms Thread» в панели «Thread» и щелкните правой кнопкой мыши на USBX на ux-модуле с помощью розовой панели. Выберите New →USBX Source.

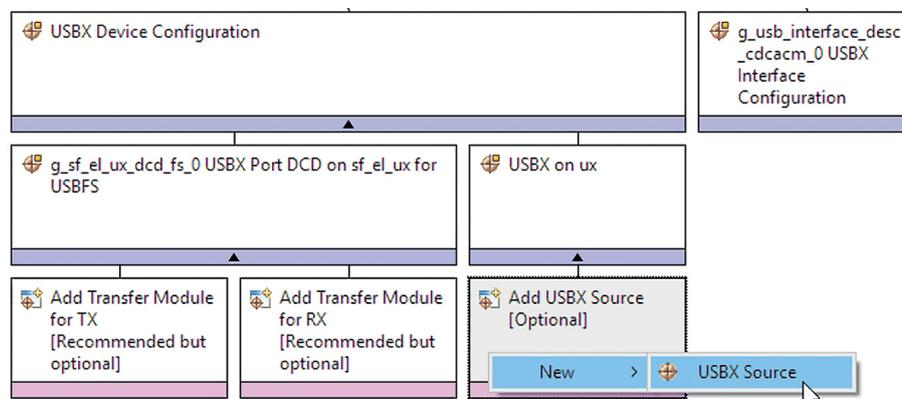


Figure 11-4: For our example, also the source code for USBX™ needs to be added

Никаких других изменений не требуется, поэтому сохраните обновленную конфигурацию SSP и нажмите кнопку «Generate Project Content», чтобы сгенерировать обновленные версии файлов. Как только генератор завершится, постройте проект, нажав на кнопку молотка. Обратите внимание, что на этот раз нужно скомпилировать большое количество файлов, поэтому сборка займет довольно много времени. На моей рабочей станции полный процесс занял чуть больше двух минут.

11.2.3 Встроенные представления и способы их использования

TraceX® предоставляет несколько средств для просмотра и анализа данных. Кнопки на панели инструментов позволяют открывать файлы с ранее записанным и сохраненным событием и перемещаться по различным захваченным событиям на основе параметра в выпадающем списке (событие, объект, переключатель или идентификатор), а также создавать разные статистические данные.

Большое окно визуализации может отображать события в режиме последовательного просмотра (Sequential View Mode), который отображается по умолчанию, и показывает события сразу после друг друга, независимо от того, сколько времени прошло между ними. Также доступен режим просмотра во времени (Time View Mode), где события отображаются в относительном по времени порядке. Переключение между этими режимами просмотра осуществляется путем выбора соответствующей вкладки в верхней части окна просмотра. Оба представления имеют свою ценность при анализе производительности во время выполнения приложения, но в большинстве случаев использование обоих из них поможет вам приблизиться к вашему решению. Режим Sequential View позволяет быстро оценить, произошли ли события в правильном порядке, а в Time View, состоялись ли они в нужное время.

Все события отображаются в том контексте, в котором они были, что можно увидеть в левой панели окна в System Context. При наведении указателя мыши на определенный контекст будет отображаться дополнительная информация. В строке события конкретного контекста различные события отображаются в отдельных цветах с характерными аббревиатурами.

Более подробную информацию о различных отображениях можно найти в документации TraceX®, доступ к которой можно получить на странице загрузки TraceX® в галерее Synergy. Некоторые из них мы также объясним в [главе 11.2.4](#).

11.2.4 Просмотр и интерпретация данных

По завершении процесса сборки запустите отладчик, но убедитесь, что ваша отладочная плата по-прежнему подключена к вашей рабочей станции с помощью двух USB-кабелей, используя разъемы DEBUG-USB и J5. e² studio загрузит код и переключится на перспективу отладки. Запустите выполнение программы, дважды нажав «Resume». Запустите свою терминальную программу, но не делайте это до того, как программа в микроконтроллере будет запущена, так как Windows® или терминальная программа может оказаться неспособной правильно определить соединение, если ее запустить слишком рано.

При запуске программы несколько раз нажмите кнопку SW4, чтобы создать достаточные системные события. Все они будут записаны в буфер трассировки, который был создан при настройке фреймворков. Приостановите программу, выбрав Run → Suspend, или нажав на символ паузы в строке главного меню.

Следующим шагом будет запуск TraceX®, выбрав Run → TraceX → Launch TraceX Debugging. В диалоговом окне, указывающем, начальный адрес буфера трассировки, а также его размер и путь к TraceX® эти параметры должны быть предварительно заполнены. Если нет, введите начальный адрес буфера трассировки и размер, указанный в главе 11.2.3. Нажмите «OK».

Если TraceX® не запускается, но выдаёт сообщение об отсутствии файла лицензии, нажмите «OK». Затем TraceX® закроется. Перейдите в папку, в которой вы распаковали установку TraceX® и скопируйте файл с именем Trace.tag в папку установки программы. Это будет c:\Express_Logic\TraceX_5.2.0\, если вы установили его в местоположение по умолчанию. Вернитесь в e² studio (нет необходимости закрывать e² studio для этой процедуры) и снова запустите TraceX®. Теперь он должен начать показывать Synergy_Use_Only как серийный номер. Эта процедура также объясняется в файле TraceX-readme.txt, расположенном в папке установщика.

Как только появится окно TraceX®, вы увидите различные события, записанные в Sequential View. Чтобы проверить те, которые связаны с нашим светодиодом и Comms Thread, переместите горизонтальный ползунок вправо, пока не увидите события, подобные тем, которые показаны на рисунке 11-5.

Контекстные изменения представлены вертикальными черными линиями, соединяющими контекстные линии. Выбранное событие представлено сплошной красной линией. В примере, показанном на рисунке 11-5, это событие 575. Наведение курсором мыши по разным событиям даст вам дополнительную информацию о них.

TraceX® также отображает статус потоков (задач). Зеленая горизонтальная линия указывает, что поток готов к исполнению, фиолетовый — что поток был приостановлен. После запуска TraceX® вы увидите только статус готовности. Чтобы увидеть все остальные статусы, вам нужно будет перейти в Options → Status Lines и выбрать «All On».

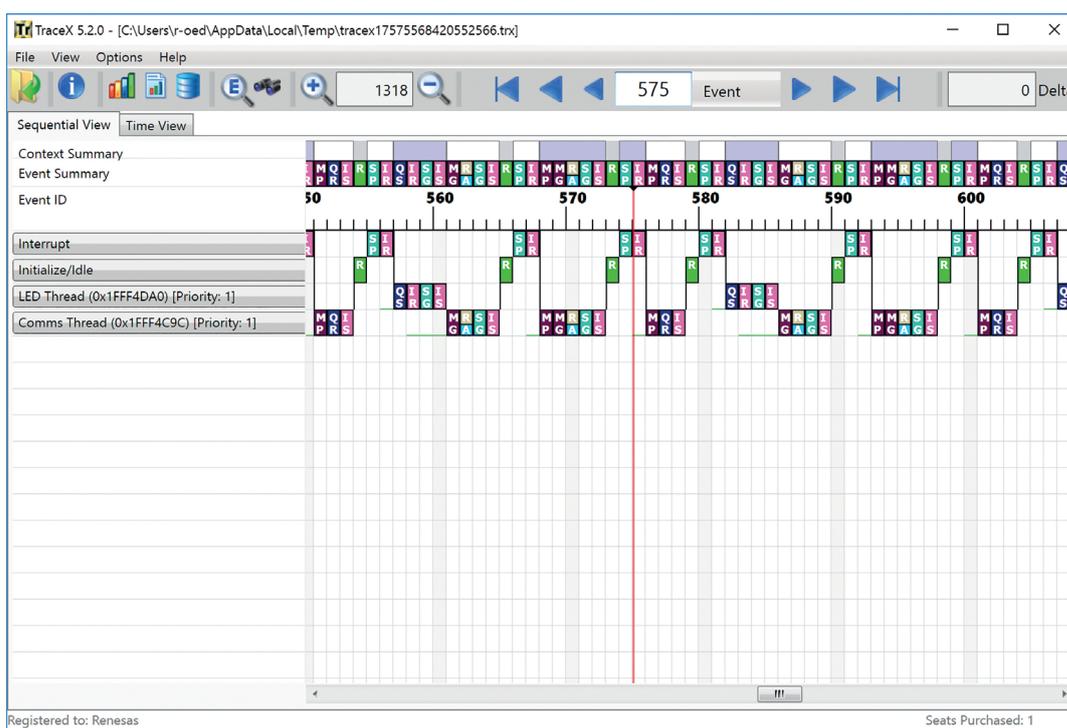


Figure 11-5: The sequence of events as visualized by TraceX®

Каждое событие представлено конкретным цветом и аббревиатурой. В нашем примере это будут следующие (начиная с события 579 на рис. 11-5):

- **R:** Running — Поток запущен (исполняется)
- **SP:** Semaphore put(SW4 semaphore) — Семафор поставлен (SW4 семафор)
- **IR:** Internal thread resume — Возобновление внутреннего потока планировщиком
- **QS:** Queue send(CDC queue) — Отправка сообщения в очередь (очередь CDC)
- **IS:** Internal thread suspend (LED thread) — Внутренний поток приостановлен планировщиком (поток LED)
- **MG:** tx_mutex_get (Comms thread) — установка мьютекса
- **RA:** Device Stack Transfer All Request Abort (USB) — Передача через стек сброс запроса прерывания (USB)
- **SG:** Semaphore get (SW4 semaphore) — Семафор запрошен (SW4 семафор)
- **IS:** Internal thread suspend (Comms thread) — Внутренний поток приостановлен планировщиком (поток Comms)

Если вы хотите просмотреть подробную информацию о событии, просто нажмите на него, и на нем отобразятся подробные сведения. Полный список возможных событий и значков можно найти внутри TraceX®, перейдя в «View» и выбрав одну из легенд (например, «ThreadXLegend») или в руководстве пользователя «User's Guide», к которому можно получить доступ через круглую синюю иконку с белой буквой «i» в главном меню TraceX®. Руководство пользователя также объяснит все другие функции, не охваченные в этом кратком руководстве, например, TimeView (см. Рис. 11-6) или различные возможности анализа производительности и статистики.

Если вы хотите собрать дополнительные данные TraceX®, возобновите выполнение кода в e² studio. Приостановите выполнение снова, как только будет записано достаточно событий. Затем нажмите Run → TraceX и выберите «Update TraceX Data». Все, что осталось в этом упражнении, заключается в том, чтобы завершить отладчик, щелкнув значок «Terminate» в строке главного меню.

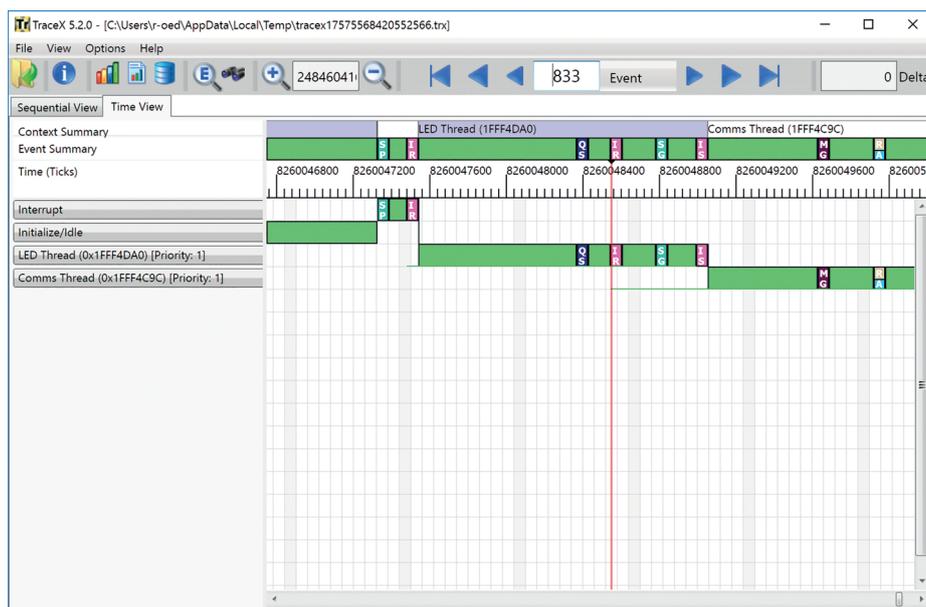


Figure 11-6: When in Time View, TraceX® displays the exact timings between the different events

ПОЗДРАВЛЯЕМ!

Вы только что завершили последнее задание в этой книге!

11.3 Использование TraceX® с IAR Workbench® для Renesas Synergy™

Печатная версия этой книги в основном охватывает набор инструментов, предоставленный Renesas. Если вы используете IAR EW для Synergy, вы можете найти расширенную версию этой главы, включая прохождение в этой среде разработки на веб-сайте книги (www.renesas.com/synergy-book).

Вопросы, которые следует вынести из этой главы:

- Возможность сбора и анализа данных трассировки делает TraceX® ценным инструментом в вашем наборе инструментов.
- Инструментарий для проекта TraceX прост и требует всего лишь нескольких щелчков мышью.

12 ЧТО ДЕЛАТЬ ПОСЛЕ ПРОЧТЕНИЯ КНИГИ

Что вы узнаете в этой главе:

- Какие существуют примеры продуктов Renesas Synergy™ Solutions и примеры приложений и как они могут вам помочь.
- Какие доступны дополнительные пакеты программного обеспечения и фреймворки для платформы Renesas Synergy™.

Теперь, когда вы узнали, из чего состоит платформа Renesas Synergy™ и насколько полезны разные части для быстрой разработки, вы можете спросить «Есть ли еще»? Ответ — да! В прошлых главах мы рассмотрели четыре из пяти основных элементов Synergy:

- **Synergy Software.** Программное обеспечение Synergy.
- **Synergy Microcontrollers.** Микроконтроллеры Synergy.
- **Synergy Gallery.** Галерея Synergy.
- **Synergy Tools and Kits.** Инструменты и наборы Synergy.
- **Synergy Solutions.** Решения Synergy.

Теперь в этой главе мы расскажем подробности о пятом элементе Synergy Solutions и о том, что он собой представляет. Кроме того, мы дадим краткое представление о квалифицированных надстройках программного обеспечения (Qualified Software Add-Ons (QSAs)) и верифицированных надстройках программного обеспечения (Verified Software Add-Ons (VSAs)), одемонстрациях партнёров и дополнительных надстройках, предоставленных Renesas. Эти решения выходят за рамки предлагаемых типовых наборов для разработки встроенных продуктов. Они являются примерами реальных продуктов и объединяют различные технологии, в том числе рабочие примеры программного обеспечения Synergy, чтобы вы могли начать свою собственную разработку продукта.

12.1 Примеры продуктов

Решения на платформе Synergy состоят из 2 категорий: примеры продуктов (Product Examples - PE) и примеры приложений (Application Examples — AE).

Примеры продуктов (PE) предлагают уникальное представление о «экземпляре разработки» конкретного конечного продукта, который точно отражает то, как будет разрабатываться фактический конечный продукт. Чтобы обеспечить быструю разработку многофункциональных приложений, части разработок примеров продуктов могут быть повторно использованы в реальных конечных продуктах.

Они поставляются с необходимым оборудованием, программным обеспечением Synergy и полной документацией, включая схемы, макеты и файлы Gerber, спецификацию материалов и специальные «проектные документы использования», описывающие методологию того, как и почему были приняты решения при разработке, и аргументация выбора различных компонентов в процессе проектирования. Это дает вам возможность модифицировать проект PE для удовлетворения конкретных потребностей вашего приложения без необходимости повторного «изобретения колеса».

Прямо сейчас, несколько PE доступны/запланированы. Одним из примеров является PE-HMI1, плата разработки для оценки микроконтроллера Synergy S7G2 Group в приложении человеко-машинного интерфейса (Human-Machine Interface — HMI). Благодаря TFT-дисплею и коммуникационным функциям, таким как Wi-Fi 802.11a/b/g, BLE, Ethernet, USB и CAN, он точно соответствует функциональности и внешнему виду реального конечного продукта HMI, что позволит быстро разработать подобные продукты. Компоненты, входящие в комплект поставки, представляют собой интегрированный 7-дюймовый широкоформатный WVGA (800 × 480 пикселей) дисплей с емкостным сенсорным экраном, отладочную плату J-Link® Lite ARM®, универсальный силовой инжектор Power over Ethernet (PoE) и необходимые кабели и документацию.



Figure 12-2: The PE-HMI1 provides everything needed to develop a feature rich HMI application

Другой PE — пример сбора данных PE-DAQ1, основанный на микроконтроллере Synergy серии S3, представляющий коммерческий или потребительский продукт для сбора данных. Он использует большую память S37A и точные аналоговые интерфейсы. Несколько датчиков движения, давления, температуры или окружающего освещения, а также пользовательский сегментный дисплей и емкостные сенсорные кнопки обеспечивают начальный набор для вашего собственного приложения. Для получения обновленной информации о других PE, пожалуйста, проверьте Synergy Explorer на странице <http://synergyexplorer.renesas.com>.

12.2 Примеры приложений

Целью примеров приложений (Application Examples - AE) является демонстрация полных приложений и предоставление примеров построения технологий, которые вы можете использовать для создания своего приложения. Они включают в себя несколько компонентов, создавая демонстрацию использования нескольких технологий, окружающих и в том числе платформу Synergy.

Например, AE подключенная к облаку системы для Internet of Things (IoT) продемонстрирует использование различных беспроводных сетей, таких как Wi-Fi, Bluetooth® low energy (BLE) Low Energy и IEEE-802.15.4 6 LoW-PAN сеть, а также HMI и облачную связь и услуги.

Другим примером, который уже доступен, является AE-CAP1 для емкостных сенсорных элементов (кнопки, слайдеры и т.д.). Он поставляется с двумя платами CPU, один с микроконтроллером Synergy S124 Group и второй - S3A7 и тремя сенсорными планшетами. Все трисенсорные панели будут работать с обоими платами CPU. Набор сертифицирован по IEC 61000 и поставляется с программным обеспечением Capacitive Touch Workbench для настройки чувствительности сенсорных датчиков.



Figure 12-3: The AE-CAP1 provides everything needed to develop a touch sensor application

AE полностью документированы и включают готовые примеры на базе SSP для среды разработки e² studio. Все это предоставляет вам информацию, необходимую для реализации вашего собственного приложения. Обновленную информацию о последних AE можно найти на странице <http://synergyexplorer.renesas.com>.

12.3 Подробнее о галерее Synergy

Не смотря на то, что мы уже говорили о галерее Synergy, это еще не все. Помните, что Renesas - это ваша точка контакта для всего, что находится на платформе Synergy. В разделе поддержки «Support» вы найдете все необходимое: доступ к функции чата для поддержки первого уровня инженерами-разработчиками из Renesas, доступными в течение 24 часов в рабочие дни, в базе знаний, которая является копилкой для всех материалов для поддержки Synergy Platformrelated и Synergy Forum расположенный в Renesas Ruiz. Также предоставляются ссылки на другие варианты поддержки.

Еще две возможности ускорить вашу разработку можно найти в разделе «Дополнительные модули ПО» в галерее Synergy. Программа надстройки программного обеспечения предоставляет множество специальных программных компонентов для разработчиков, использующих платформу Synergy. Эти компоненты охватывают множество областей технологии, включая расширенную безопасность и функциональную безопасность, облачные сервисы и специализированные алгоритмы управления.

Дополнения к программному обеспечению выпускаются в четырех разных вариантах: квалифицированные дополнения, верифицированные дополнения, партнерские проекты и дополнения разрабатываемые Renesasотдельно от SSP. Все они приносят еще большую свободу использования на платформе Renesas Synergy

12.3.1 Квалифицированное программное обеспечение

Квалифицированные надстройки программного обеспечения (Qualified Software Add-On — QSAs) дополняют программный пакет Synergy (SSP) специальными функциональными возможностями программного обеспечения, такими как цельные коммуникационные стеки, службы безопасности и контроля доступа и сложные алгоритмы управления. Они продаются и лицензируются отдельно от программного пакета SSP и обслуживаются Renesas через галерею Synergy. Компоненты QSA были выполнены Renesas и были разработаны и протестированы с соблюдением тех же строгих требований к качеству, что и SSP, и они очень интегрированы и оптимизированы для SSP и его структуры API.

Пробные версии программных компонентов QSA можно бесплатно загрузить из Synergy Gallery.

12.3.2 Верифицированное программное обеспечение

Компоненты верифицированного программного обеспечения (Verified Software Add-On — VSA) продаются и обслуживаются сторонними компаниями в составе экосистемы Synergy, и они доступны через галерею Synergy. VSA включают в себя набор специальных программных компонентов, например, стеки промышленных протоколов связи, аудиокодеков, приложений управления двигателем и т. д.

Они проверены Renesas, чтобы обеспечить совместимость с SSP и вы можете бесплатно загрузить пробные версии из Synergy Gallery. Коммерческие лицензии должны быть получены у поставщика программного обеспечения VSA напрямую.

12.3.3 Демонстрации партнёров Synergy

Доступ к категории «Synergy Partner Showcase» можно получить непосредственно из галереи Synergy и содержит так называемые партнерские проекты (Partner Projects — PP). Эти PP разрабатываются компаниями-партнерами Renesas и демонстрируются на платформе Synergy.

Они представляют собой широкий спектр продуктов и услуг, предлагаемых и поддерживаемых партнерскими компаниями Renesas, обеспечивая быстрые и простые пробные проекты для разработчиков.

12.3.4 Дополнительное программное обеспечение от Renesas

Не только партнерские компании Renesas предоставляют дополнительное программное обеспечение для платформы Synergy, но и сама Renesas. Эти надстройки для пакета программного обеспечения можно получить, нажав вкладку «Служебные программы SSP» на странице SSP в Галерее.

Одним из примеров этих надстроек является Synergy Wi-Fi фреймворк для отладочного набора Synergy SK-S7G2. Он демонстрирует, как подключение Wi-Fi можно легко добавить к сложному многопоточному приложению, используя SSP и фреймворки. Другие дополнения, например, загрузчики, будут доступны в ближайшее время.

Будущее принесет больше дополнений к платформе, поэтому часто проверяйте галерею Synergy!

Вопросы, которые следует вынести из этой главы:

- Примеры продуктов — это представление о том, как будет разрабатываться фактический конечный продукт.
- Примеры приложений — это примеры построения технологий, которые вы можете использовать для создания своего приложения.
- Галерея Synergy — это ваша единственная точка доступа для поддержки всех компонентов платформы Synergy.
- QSA — это дополнительные и специализированные программные компоненты, квалифицированные и поддерживаемые Renesas.
- VSA — это дополнительные и специализированные программные компоненты, созданные, лицензированные и поддерживаемые третьими сторонами, которые проверены Renesas для работы с SSP.
- Еще больше программного обеспечения предоставляется Renesas и их компаниями-партнерами в качестве дополнений или проектов партнеров Synergy.

13 РЕЗЮМЕ И ПРИЗНАТЕЛЬНОСТЬ

Вы сделали это! Это последняя и довольно короткая глава книги об основах платформы Renesas Synergy™. Надеюсь, вам понравилось узнавать об этой замечательной платформе столько, сколько я об этом написал.

Благодаря простоте использования, легкому доступу к компонентам и полной экосистеме, состоящей из программного обеспечения производственного класса, новых микроконтроллеров, галереи, предоставляющей единую точку доступа, инструментов и наборов, соответствующих фактическим потребностям разработчиков, и решений, которые напоминают реальные конечные приложения и вместе с большой структурой поддержки — всё это, безусловно, одна из величайших революций, которые я наблюдал более 23 лет моей работы в этой отрасли. Это смелый новый способ создания функциональных, отвечающих высоким требованиям и высоко взаимосвязанных приложений завтрашнего дня, и разработчики воспользуются им.

Я не мог представить, чтоб эта книга, которая не только включает в себя книгу как таковую, но и веб-контент и видеоуроки, стала реальностью без помощи и поддержки со стороны всех прекрасных людей в Renesas Electronics Europe.

Отдельное спасибо от всего сердца:

NellyE.

за ее приверженность и неустанное управление проектами.

AndyH.

за предоставленную мне возможность стать частью этого замечательного проекта.

KarolS. & JakubJ.

за воспроизведение каждого шага упражнений, чтобы убедиться, что они будут работать.

IanH. & GiancarloP.

за то, что они потратили свое время на проработку нескольких технических разделов и предоставили существенную обратную связь.

Steven.

за то, что сделал мой английский понятным для читателей.

Klaus R. & Britta W.

за поддержку дизайна книги.

Без них, их сильного духа, их готовности делиться идеями, их обзора для технической и языковой точности, эта книга никогда не была бы возможна!

ОБ АВТОРЕ:

Richard Oed более 23 лет работает в инженерах по применению прикладных программ и систем в области встраиваемых систем, обеспечивая поддержку и написание программного обеспечения для DSP, микроконтроллеров и микропроцессоров, а также для преобразователей данных. Теперь он работает как независимый автор, писатель и журналист. Ричард является соавтором трех патентов и является членом IEEE, ACM и VDE.

Перед покупкой или использованием любых продуктов Renesas Electronics, перечисленных здесь, пожалуйста, ознакомьтесь с последним руководством по продукту и/или спецификацией.

Renesas Electronics Europe

www.renesas.com



© 2017 Renesas Electronics Europe.
All rights reserved. Printed in Germany.
Document No. R01PF0137ED0100



САНКТ-ПЕТЕРБУРГ:

ул. Калинина, д. 13
Тел.: (812) 325-36-85
Факс: (812) 786-85-79
e-mail: info@mt-system.ru

МОСКВА:

ул. Красноармейская, д.11, корп.1
Тел.: (495) 988-20-73
Факс: (495) 988-20-74
e-mail: moscow@mt-system.ru

ЕКАТЕРИНБУРГ:

620075, ул. Шевченко,
д.18, оф.508
Тел.: +7(343) 3-111-007
e-mail: ekb@mt-system.ru

НОВОСИБИРСК:

630003, ул. Владимировская,
д. 2/1, оф. 418
Тел.: +7(383) 383-0-208
e-mail: nsk@mt-system.ru