

Виктор Бугаев (г. Долгопрудный), Максим Мусиенко, Ярослав Крайнык (г. Николаев)

ВИЗУАЛИЗАЦИЯ ВОЗМОЖНОСТЕЙ: ГРАФИЧЕСКИЙ ГЕНЕРАТОР КОДА STM32CUBEMX

До сих пор среды разработки для микроконтроллеров значительно уступали специализированным средам разработки прикладного ПО. Но визуальный редактор конфигурации и генератора кода **STM32CubeMX**, предлагаемый **STMicroelectronics**, позволяет осуществлять тонкую автоматизированную настройку инициализации и генерировать базовые проекты. Статья пошагово описывает создание такого проекта для микроконтроллера **STM32L0** с низким энергопотреблением.

Средства автоматизации, связанные с процессом создания программного обеспечения, позволяют значительно сократить время, затрачиваемое на решение типовых проблем при реализации проекта. В таких случаях используются общепринятые практики программирования, основанные на опыте широкого круга разработчиков, что позволяет быстро создать базу для дальнейшего проектирования.

Особую популярность и распространение подобные средства получили среди разработчиков приложений для бизнеса, мобильных и настольных приложений. Зачастую среды разработки, которые используются в данном сегменте, буквально перенасыщены различными рода инструментами, мастерами, дополнениями, которые обеспечивают требуемый функционал. Более того, часто сами среды разработки играют роль своеобразной энциклопедии, позволяющей ознакомиться с новыми возможностями и технологиями, что немаловажно для начинающих программистов.

Тем не менее, как ни парадоксально, среди программных продуктов, ориентированных на использование при разработке встраиваемых систем, тенденция к автоматизации рутинных процессов прослеживается заметно слабее. Специализированные среды разработки для микроконтроллеров (МК) зачастую заметно уступают своим аналогам из мира разработки прикладного программного обеспечения (ПО), даже несмотря на то, что могут иметь с ними общую базу (например, **Eclipse**). Это можно объяснить меньшей популярностью данной сферы и ослабленным вниманием вендоров к данному фактору.

Результатом такого положения дел является то, что микроконтроллеры, находящиеся в основе многих встраи-

ваемых систем, имеют ограниченный доступный для использования функционал, который, в свою очередь, зачастую имеет ограниченный набор функций. При организации структуры проекта используются типовые шаблоны организации программного кода вне зависимости от его конечного предназначения. Именно эти моменты являются ключевыми в плане дальнейшего развития специализированных программных инструментов для работы с такими устройствами.

Указанная выше концепция автоматизации настроек инициализации и создания базового проекта положена в основу визуального редактора конфигурации и генератора кода **STM32CubeMX**.

Преимущества и недостатки генераторов кода и использования библиотек

Безусловно, наиболее эффективной с точки зрения использования памяти и степени контроля разработчика за реализацией является прямая работа с регистрами устройства в соответствии с документацией производителя. Это также дает возможность быстрее изучить предоставляемую программную архитектуру, понять основные принципы работы с периферийными устройствами. Программист, имеющий подобный опыт реализации для конкретной платформы, ознакомлен с большим количеством деталей, которые позволяют обеспечивать более тонкий контроль над созданием решения. В то же время такой подход требует от программиста высокого уровня дисциплины и готовности к увеличению времени разработки из-за необходимости связывания низкоуровневой работы с функциональными абстракциями.

Другой стратегией разработки можно считать использование библиотек. В основе создания программных библиотек лежит концепция повторного ис-

пользования кода. Наиболее удачные и универсальные решения оформляются в виде отдельного модуля, который в случае необходимости может подключаться к проекту. Это позволяет избежать временных затрат на реализацию стандартных требований. В то же время необходимо учитывать универсальность решения, которая в определенных случаях (обеспечение максимального быстродействия, неординарная реализация и другие) может являться и недостатком. Обычно, в силу данного фактора, библиотеки обеспечивают меньшую производительность по сравнению с решениями, оптимизированными для конкретной задачи. Также необходимо помнить, что даже несмотря на широкое использование и постоянную проверку, библиотеки могут содержать ошибки, которые сложно выявить.

Начинающему программисту особенно важно понимать различие между данными подходами. Ни один из них не является абсолютной панацеей, поскольку содержит как положительные, так и отрицательные моменты.

Генераторы программного кода предназначены для того, чтобы создавать отдельные файлы исходного кода или всего проекта на основе заданных настроек. Чаще всего генераторы используют популярные библиотеки, что позволяет унифицировать процесс генерации исходных текстов. В зависимости от количества настраиваемых параметров они могут — за счет реализации части функционального обеспечения — значительно минимизировать время, затрачиваемое на создание основы для дальнейшей разработки. Практика использования подобных генераторов для разработки прикладного ПО во многих отраслях стала стандартной, а знание соответствующих инструментов является обязательным. В то же время, в сфере разработки встраиваемых систем подобные генераторы применяются благодаря преимуществам, которые они предоставляют, однако все еще находятся на этапе развития и получения признания среди разработчиков.

К недостаткам генераторов кода можно отнести ограниченное количество настроек, которое не позволяет учесть все

особенности проекта. Также при разработке проекта для динамической системы, которая часто изменяет параметры собственной работы в зависимости от внутренних и внешних условий, генераторами можно описать лишь ограниченное количество состояний данной системы. Необходимость учета большого количества критериев влечет за собой внесение большого количества изменений в программную реализацию.

Тем не менее, при всех указанных недостатках, положительные стороны использования генераторов кода очевидны.

Вендоры рынка микроконтроллеров пока только начинают обращать свое внимание на данный тип инструментов. Одним из наиболее интересных решений является визуальный редактор настроек инициализации **STM32CubeMX**.

Возможности STM32CubeMX

Загрузить инсталлятор **STM32CubeMX** можно с сайта производителя [1]. Установка программы требует наличия на компьютере одной из последних версий среды выполнения Java 7, поэтому при ее отсутствии необходимо предварительно ее установить.

Визуальный редактор настроек **STM32CubeMX** является составной частью инфраструктуры **STM32Cube**, которая также включает специализированные библиотеки для работы с различными линейками микроконтроллеров **STM32**. Стоит отметить, что интеграция между составными частями выполнена

удачно, поскольку устанавливать необходимые библиотеки можно прямо из редактора. Для этого необходимо подключение к Интернету, поскольку загрузка производится с сайта компании STMicroelectronics. Также есть возможность отслеживать обновления версий уже загруженных библиотек. Кроме того, библиотеки могут использоваться самостоятельно при работе с другим программным обеспечением.

Пакет программного обеспечения, который предлагается для использования, включает в себя следующие основные компоненты:

- библиотеку CMSIS;
- библиотеку для работы с периферийными устройствами (*Hardware Abstraction Layer – HAL*);
- библиотеки, упрощающие использование возможностей популярных плат (*Board Support Package – BSP*);
- библиотеки более высокого прикладного уровня, позволяющие реализовать специализированные возможности (система реального времени FreeRTOS, поддержка файловой системы FAT, работа со стеком протоколов TCP/IP благодаря библиотеке lwIP, использование возможностей USB и др.). Не все библиотеки данного уровня могут быть доступны для подключения при использовании конкретного микроконтроллера.

Особенно следует отметить библиотеку HAL, которую можно назвать продолжением развития известной библиотеки Standard Peripheral Library (SPL), с которой она имеет много общего в

концепции организации программного кода. Также архив с библиотеками содержит документацию, примеры кода и полезные дополнения.

Основными функциями, которые предоставляет визуальный редактор конфигурации, являются:

- настройка использования выводов периферийными устройствами в составе микроконтроллера;
- настройка тактирования;
- настройка периферийных устройств;
- расчет потребления энергии в соответствии с планируемыми режимами работы.

Интерфейс самой программы прост, удобен и понятен. Несмотря на наличие большого количества настроек, он не перегружен ненужными функциями, а визуальное представление различных систем МК позволяет полностью сложить картину о конечном результате выполненных настроек.

Пример создания проекта на основе МК STM32L0

Проведем более детальное ознакомление с возможностями визуального редактора настроек на примере создания проекта для микроконтроллера семейства STM32L0. Данная серия устройств ориентирована на обеспечение решений с низким энергопотреблением. Ядро микроконтроллера основано на архитектуре **Cortex M0+**. Также в составе устройства присутствуют периферийные модули с низким энергопотреблением.

В качестве устройства для тестирования проекта выбрана отладочная плата NUCLEO-L053R8 которая основана на МК **STM32L053R8T6** [2]. В примере будут использоваться модули приемопередатчика USART, последовательного интерфейса SPI, часов реального времени RTC. Устройство будет переходить в режим низкого потребления Stop, из которого будет выходить по сигналу часов реального времени. Для работы таймера воспользуемся внешним источником сигнала 32768 Гц, который присутствует на выбранной плате [3].

Как уже упоминалось, для работы с устройствами определенной серии необходимо загрузить набор специализированных библиотек. Сделать это можно прямо из программы **STM32CubeMX**, выполнив команду *Help/Install New Libraries*. В списке доступных библиотек необходимо отметить библиотеку *Firmware Package for Family STM32L0* последней версии и нажать кнопку *Install Now* для произведения установки. После ее окончания доступность библиотеки будет обозначена в списке, как показано на рисунке 1. При использовании для доступа к сети прокси-сервера следует произвести необходимые настройки в окне, которое доступно



Рис. 1. Окно установки библиотек

по команде *Help/Updater Settings...* на вкладке *Connection Parameters*.

После загрузки необходимой библиотеки можно перейти к созданию проекта посредством команды *File/New Project...* Данное действие откроет окно выбора устройства (рисунок 2), для которого предполагается создание проекта. Благодаря применению фильтров можно сократить количество устройств в основном списке и ускорить поиск.

После создания проекта пользователю для редактирования доступны четыре вкладки:

- Pinout;
- Clock Configuration;
- Configuration;
- Power Consumption Calculator.

В данном окне пользователю предлагается различный выбор платформенной базы — либо выбор контроллера по параметрам в левой части окна, либо выбор по заранее определенной линейке микроконтроллера. В первом случае инженер простым добавлением меток напротив нужной периферии может подобрать себе микроконтроллер из семейства STM32. Во второй вкладке «boards» инженер может выбрать одну из отладочных плат STMicroelectronics и уже под нее уже сконфигурировать конечный проект. Распределение функций между вкладками позволяет пользователю работать с одной группой настроек, но, в то же время, присутствует интеграция между вкладками: действия, выполненные на одной, влияют на доступные действия в другой. Это гарантирует целостность всего решения и отсутствие конфликтов в коде проекта. Созданный проект следует сохранить в отдельную директорию.

Настройка выводов

Настройка функционального назначения выводов производится в первой из четырех рабочих вкладок. Рабочая область вкладки разделена на две части: в первой отображается дерево конфигурации периферийных модулей и специализированных программных компонентов, а во второй организовано графическое представление МК и использования его выводов. Дерево конфигурации позволяет не только выбрать, какие модули будут использоваться, но и, например, выбрать режим использования, дополнительные настройки и так далее. По мере выбора используемых компонентов в части отображения будут происходить изменения относительно используемых выводов. Также будет происходить автоматическая проверка того, какие модули не могут использоваться или какие возможности могут быть ограничены в связи с произведенными настройками. Уведомление об этом появляется в виде предупреждающих знаков желтого или красного цвета в дереве конфигурации,



Рис. 2. Окно выбора МК

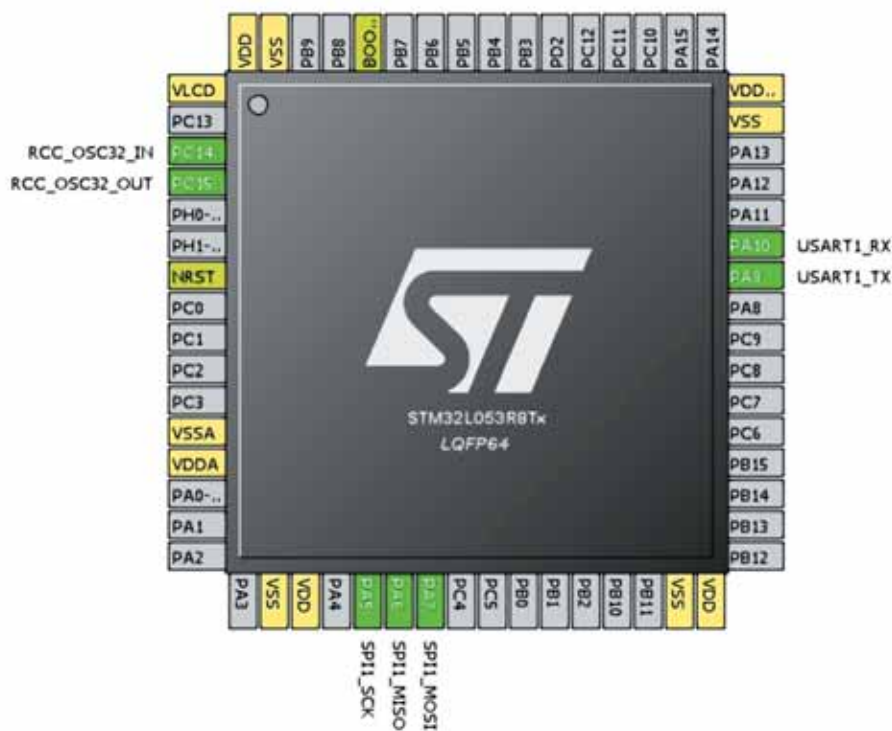


Рис. 3. Визуальное представление используемых выводов

а соответствующие настройки становятся недоступными для редактирования. Таким образом, есть возможность сразу проверить доступность одновременной работы выбранного набора модулей, не прибегая к изучению документации, поскольку возможный конфликт будет сразу выявлен. Кроме того, функциональное назначение выводов может быть произведено и с помощью визуального представления. Для этого необходимо левой клавишей мыши нажать на

вывод, после чего появится подсказка о возможных функциях вывода в составе периферийных устройств. Выбор одной из функций приводит к тому, что соответствующее устройство в дереве конфигурации становится активным.

В нашем случае установим следующие опции для работы необходимых устройств: USART будет работать в асинхронном режиме, SPI — в полнодуплексном режиме в качестве ведущего устройства, в RTC будет использоваться

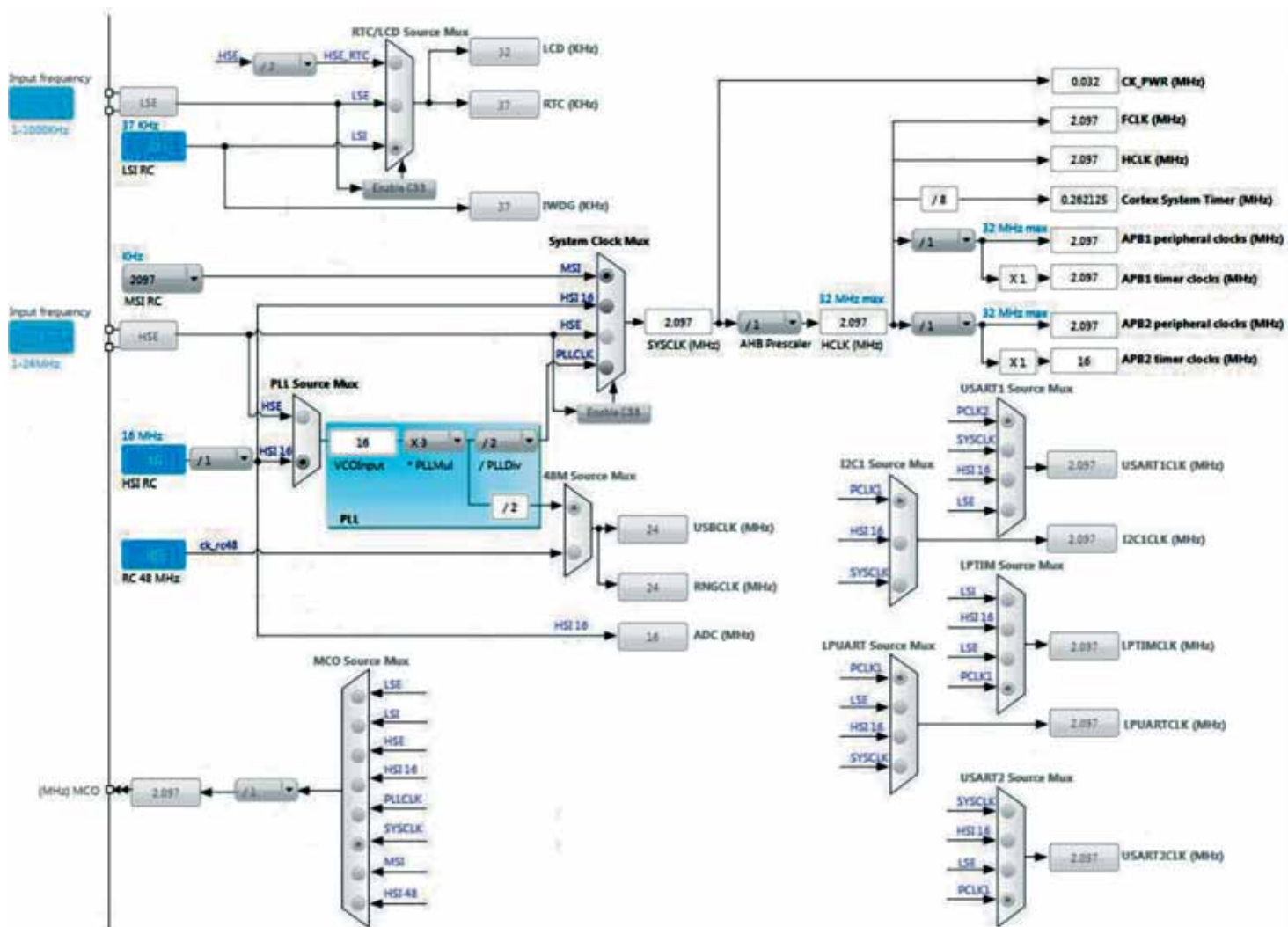


Рис. 4. Конфигурация источников тактового сигнала

сигнал *Alarm A*. Все это можно выполнить, раскрыв соответствующие узлы в дереве IP и изменив значение *Disable* на указанные выше для каждого модуля. Кроме того, поскольку в качестве источника тактового сигнала для RTC будет выступать внешний низкочастотный источник, то в модуле RCC следует выбрать опцию для его включения (*Low Speed Clock (LSE) – Crystal/Ceramic Resonator*). Результат выполнения указанных настроек должен соответствовать тому, что показано на рисунке 3.

Используемые выводы отображены зеленым цветом, рядом с ними – подсказка об их функциональном назначении.

Настройка тактирования

Вкладка настроек тактирования (*Clock Configuration*) является одной из самых интересных возможностей. Данный инструмент представляет собой логическое продолжение развития подобных мастеров, которые ранее распространялись компанией STMicroelectronics в виде файлов Excel с макросами. В данном случае он тесно интегрирован с другими параметра-

ми настроек, что является шагом вперед, поскольку необходимость внесения дополнительных изменений в сгенерированный проект сводится к минимуму в случае использования одного режима тактирования.

Интерфейс данной вкладки во многом повторяет интерфейс упомянутого выше инструмента-предшественника. Преимущество такого представления – возможность отслеживать связи между различными составными компонентами системы тактирования и контроллерами все этапы данного процесса. Более того, для устройств серии STM32L0 характерной особенностью является возможность более широкого выбора сигналов тактирования для различных внутренних устройств. В качестве источника тактового сигнала могут использоваться внутренние и внешние генераторы. В общем случае источники тактирования разделяют на высокоскоростные (*High-speed*), низкоскоростные (*Low-speed*) и мультискоростные (*Multi-speed*). При этом все они могут использоваться для обеспечения работы различных составляющих. Потребность в работе на высоких частотах обеспечивает наличие фазовой автоподстройки частоты. Однако

следует помнить, что максимальная тактовая частота работы не может превышать 32 МГц. При нарушении данного требования, а также других требований относительно настройки рабочих частот модулей, в рабочей области редактора проблемные участки будут выделены красным цветом. Данное действие выполняется автоматически, аналогично проверке одновременного использования выбранного набора модулей на первой вкладке.

Важной характеристикой редактора настроек тактирования является то, что в качестве источника частоты могут использоваться только модули, которые были предварительно выбраны среди устройств на вкладке *Pinout*. Аналогичным образом, тактовый сигнал предполагается использовать только для устройств, которые обозначены активными, иначе опции настроек будут просто недоступны. Например, чтобы иметь возможность генерации кода для внешнего источника тактирования, необходимо указать для модуля RCC использование высоко- или низкоскоростного осциллятора. После того, как данные параметры будут активированы, перейдя на вкладку *Clock Configuration*,

можно увидеть, что появилась возможность использовать внешний источник тактового сигнала.

Настройка тактирования периферийного модуля, не являющегося активным, также недоступна в редакторе. В качестве примера рассмотрим RTC — часы реального времени (*Real-Time Clock*), для которых предполагается отдельный источник тактового сигнала. Выбор одной из рабочих функций RTC на вкладке Pinout приводит к тому, что становится доступным для выбора мультиплексор тактового сигнала. Набор сигналов, доступных для выбора, определяется условием, описанным выше. Результат проведенных настроек должен соответствовать рисунку 4.

В целом, инструмент настройки системы тактирования очень удобен и позволяет избежать возможных ошибок при задании параметров.

Настройка периферии

Помимо уже рассмотренной настройки периферии в дереве конфигурации на первой вкладке, где производится базовая настройка и, в основном, используются параметры, влияющие на использование выводов, присутствует также отдельная вкладка Configuration, предназначенная для этих целей. По своей структурной организации она аналогична первой: присутствуют дерево конфигурации используемых модулей, а также модулей, которые не требуют использования выводов МК, и часть с визуальными компонентами, которые предоставляют доступ к более детальной настройке системы. Визуальные компоненты сгруппированы в соответствии с функциями устройства для настроек (системные, функции управления, обмена данными и другие). Кроме того, в данном случае дерево конфигураций обеспечивает значительно меньше настроек, чем в предыдущем случае: основная часть настроек перенесена на индивидуальное окно выбора параметров работы модуля, которое открывается нажатием кнопки с обозначением модуля.

В правой части вкладки выбранные ранее устройства сгруппированы так, как показано на рисунке 5.

О текущем состоянии настроек можно узнать по иконке, отображаемой на соответствующей модулю кнопке. О том, что не все параметры работы модуля указаны, свидетельствует черный цвет названия модуля на кнопке. Выявление ошибок сопровождается появлением красного значка на кнопке и диалоговым окном с сообщением об ошибке. Несмотря на то, что некоторые устройства могут быть обозначены как не полностью сконфигурированные, это не означает, что сгенерированный проект будет содержать ошибки. Типичным примером

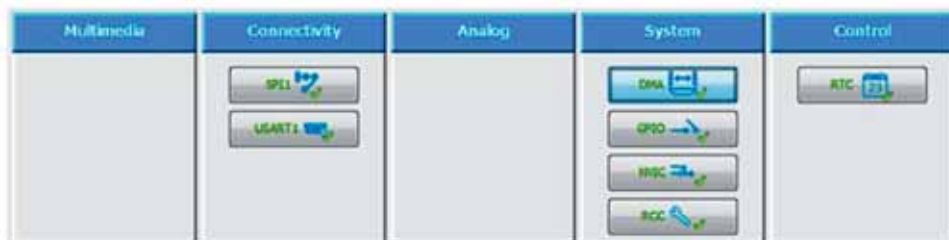


Рис. 5. Представление устройств, доступных для конфигурации

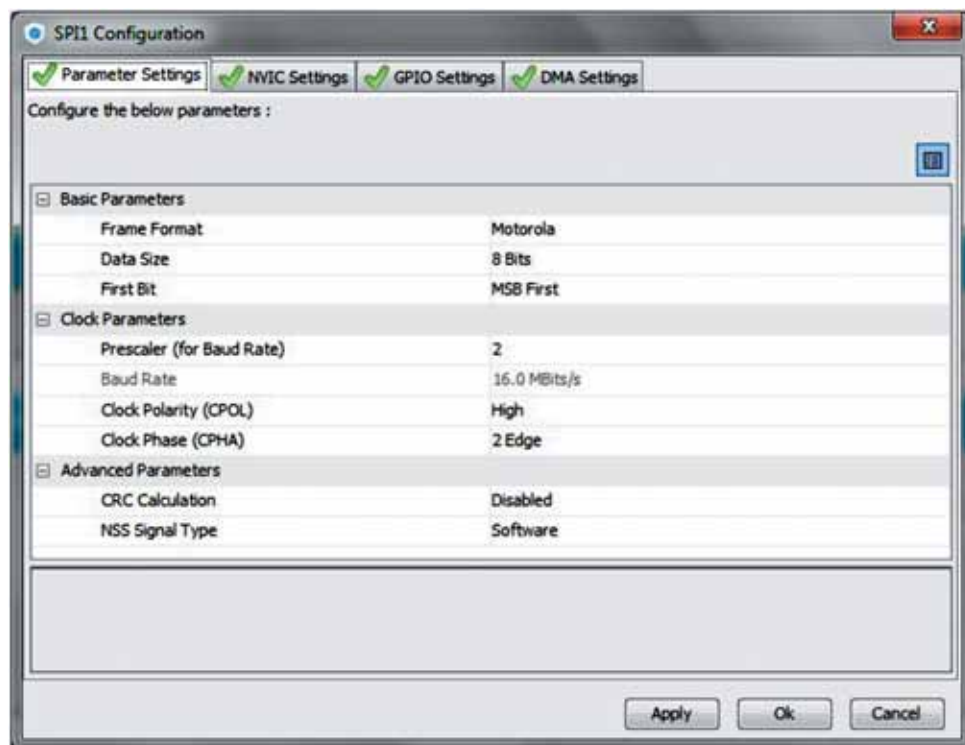


Рис. 6. Окно настройки параметров модуля SPI

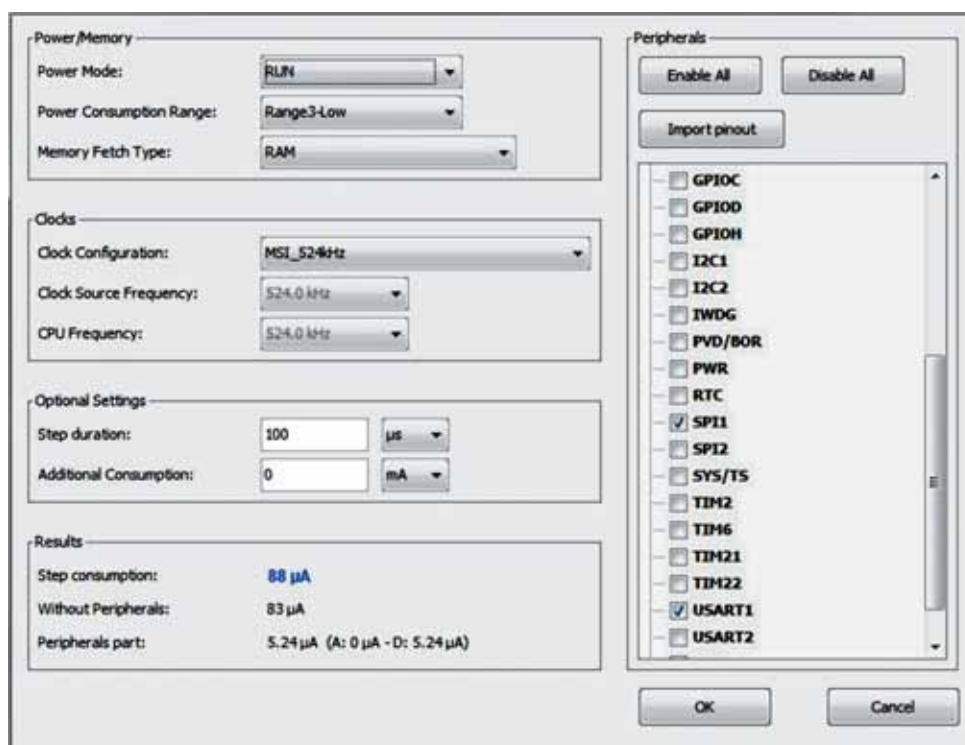


Рис. 7. Настройка режима работы микроконтроллера для оценки потребления

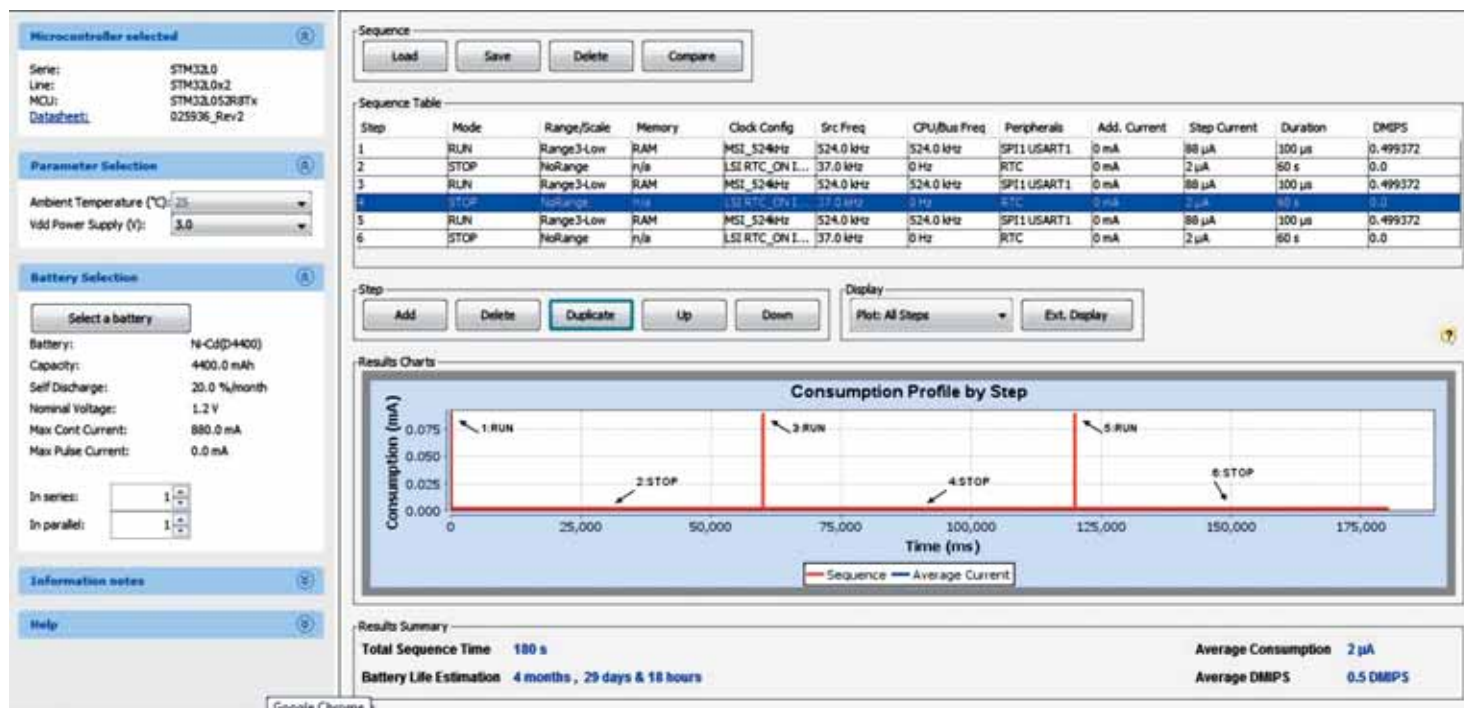


Рис. 8. Показатели потребления для выбранных режимов работы

является настройка прямого доступа к памяти (*Direct Memory Access, DMA*), которая может не выполняться при отсутствии использования данной функции. Тем не менее, в определенных случаях необходимо выполнить полную конфигурацию модуля для обеспечения его корректной работы. Устройство, параметры которого полностью установлены, отображается зеленой надписью на кнопке и иконкой об успешной проверке заданных параметров.

Большинство устройств, выбранных для использования на вкладке *Pinout*, по умолчанию получают конфигурацию, которая не содержит ошибок и готова к использованию. Для того чтобы выполнить настройку модуля в соответствии с потребностями проекта, необходимо нажать на соответствующую ему кнопку. После нажатия пользователю доступно окно настроек данного модуля, которое, по аналогии с основным окном, организовано по принципу вкладок. Количество и тип вкладок зависят от конкретного модуля.

Рассмотрим более детально настройку модуля SPI. Откроем соответствующее ему окно. В нем присутствуют четыре вкладки:

- Parameter Settings.
- NVIC Settings.
- GPIO Settings.
- DMA Settings.

На первой вкладке установим, что при отсутствии передачи тактовый сигнал устройства имеет высокий логический уровень (параметр, *Clock Polarity (CPOL)* — значение *High*), а считывание/запись происходит по фронту (параметр *Clock Phase (CPHA)* — значение *2 Edge*), как по-

казано на рисунке 6. Параметры по умолчанию, установленные на вкладке настроек пинов ввода/вывода (*GPIO Settings*), могут использоваться без изменений. Поскольку прерывания SPI использоваться не будут, то настройки для вкладки *NVIC Settings* не изменяем. На последней вкладке, *DMA Settings*, добавим, что по мере приема данных они будут сохраняться в памяти. Последовательно добавим новую настройку DMA (кнопка *Add*), выберем из списка доступных событий *SPI1_RX*, а остальные параметры установятся автоматически. Наличие автоматической установки параметров значительно упрощает работу, поскольку для DMA имеется большое количество настроек, из которых для получения корректного результата необходимо отметить соответствующие требованиям. Это можно сделать и на основе документации, однако автоматизация данной настройки позволяет значительно ускорить конфигурирование и избежать ошибок, связанных с неточностями установки параметров.

Подобные настройки доступны для всех выбранных модулей, что позволяет быстро выполнить процесс задания необходимых параметров.

Расчет потребления

С учетом того, что устройства серии STM32L0 позиционируются как устройства с ультранизким энергопотреблением, возможность предварительной оценки данного параметра является востребованной и актуальной.

В основу расчета потребляемой энергии в STM32CubeMX положена следующая концепция: устройство может рабо-

тать в различных режимах потребления (используемое для примера устройство — в семи). От текущего режима работы зависит то, какие устройства могут быть активными, и, соответственно, могут потреблять энергию. В зависимости от продолжительности различных режимов работы и от используемых устройств, можно оценить среднее потребление системы, а также время ее автономной работы для выбранного источника питания.

Перейдя на вкладку *Power Consumption Calculator* (рисунок 7), вы получаете возможность задать последовательность режимов, в которых будет работать устройство, а также выбрать задействованные периферийные устройства для каждого режима. Для расчета времени жизни устройства пользователь может создать свой тип батарейки или выбрать ее в выпадающем списке слева. Таким образом, можно создать полный профайл устройства, сформировать отчет и сохранить его в одном из известных форматов для отчета.

Добавить новый режим можно с помощью кнопки *Add* в группе *Step*. В открывшемся окне следует выбрать режим работы относительно использования питания из выпадающего списка *Power Mode*. В примере будет использоваться всего два режима — *RUN* и *STOP*. Для каждого из них необходимо настроить режим питания ядра, выполнение программы из FLASH или RAM, выбрать источник тактирования, указать длительность работы микроконтроллера в данном режиме и в правой колонке выбрать периферию, которая будет активна в данном режиме (рисунок 7). Также можно учесть дополнительные источ-

ники потребления в окне «Additional Consumption», например, светодиоды на плате в том или ином режиме работы. Конечные графики потребления и время жизни от батарейки будут показаны на рисунке 8

Как показано на рисунке, для выбранного режима описанная система по расчетам способна работать автономно более четырех месяцев.

Подобный инструмент очень удобен и позволяет быстро оценить показатели автономности разрабатываемой системы, что особенно важно в разработках с акцентом на энергосбережении.

Генерация проекта

STM32CubeMX обеспечивает генерацию проектов для трех различных сред разработки/наборов инструментов: MDK-ARM, EWARM, TrueSTUDIO. Данный параметр можно установить с помощью настроек, выполнив команду *Project/Settings...* Выполнить генерацию проекта можно командой меню *Project/Generate Code*. Сгенерированный проект можно найти в папке с файлом проекта визуального редактора. Если все действия выполнены правильно, проект должен собраться без ошибок в соответствующей среде разработки. После этого он готов ко внесению дальнейших изменений.

Если в процессе проектирования приложения пользователю необходимо добавить дополнительный периферийный модуль, это не составляет никаких проблем. Нужно просто выбрать модуль, сконфигурировать его и вызвать команду *Project/Generate Code*. Система добавит необходимый функционал в исходный проект, не повредив уже сделанную работу программиста.

Сами библиотеки периферии, которые использует программа, находятся на диске C, в папке STM32Cube, в разделе «Мои документы».

В одной из следующих ревизий программы STMicroelectronics обещает добавить поддержку связки Eclipse/GCC

Заключение

Появление визуального редактора настроек инициализации STM32CubeMX является свидетельством того, что инструменты разработчика для компании STMicroelectronics представляются важной частью общего развития. Сам визуальный редактор предоставляет простой и понятный доступ к различным системам устройства для их настройки. Особенно важным является то, что на основе выполненных настроек пользователь может сгенерировать готовый проект, который можно использовать для разработки

дальнейшего решения. В рассмотренном примере важной частью является оценка потребления энергии. Таким образом, данный программный продукт является мощным инструментом, предоставляющим широкие возможности для упрощения работы программиста и позволяющим сократить общее время, необходимое на разработку.

Литература

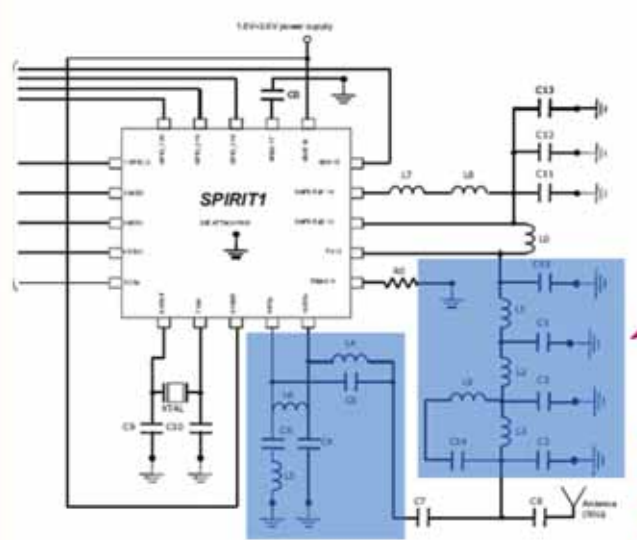
1. http://www.st.com/web/catalog/tools/FM147/CL1794/SC961/SS1743/PF259242?icmp=stm32cubemx_pron_prcube_feb2014&sc=stm32cube-pr
2. <http://www.st.com/web/en/resource/technical/document/datasheet/DM00105960.pdf>
3. http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/DM00105823.pdf

Получение технической информации,
заказ образцов, поставка –
e-mail: mcu.vesti@compel.ru



Согласующие SMD-трансформаторы (BALUN)

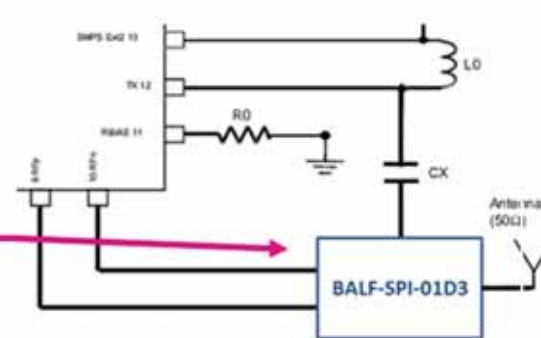
Для трансиверов ST, Texas Instruments, Nordic Semiconductor





ОСОБЕННОСТИ:

- Упрощают схемотехнику цепочки согласования с антенной
- Выпускаются для диапазонов 868 МГц и 2400 МГц
- Встроенный фильтр
- Рабочая температура: -40...85°C



Поддержка разработчиков:

E-mail: st@compel.ru

www.compel.ru/projects-support



www.compel.ru