

СРЕДА BeeKit™ – УНИВЕРСАЛЬНЫЙ ИНСТРУМЕНТ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ БЕСПРОВОДНЫХ ПРИЛОЖЕНИЙ СТАНДАРТА 802.15.4/ZigBee. ЧАСТЬ 1

Михаил Соколов, инженер по применению, ООО Freescale Semiconductor
Александр Гришин, инженер, МЭИ (ТУ)

Разработка программного обеспечения для беспроводных систем стандарта 802.15.4/ZigBee является, возможно, наиболее сложной задачей из всех этапов проектирования. Компания Freescale предлагает универсальную среду для создания программного обеспечения беспроводных устройств — BeeKit™, призванную облегчить освоение технологии ZigBee и упростить труд программистов.

В статье предлагается обоснование выбора программной библиотеки, которая будет заложена в основу собственного ПО, описываются доступные примеры и упражнения для знакомства с технологией и сокращения времени разработки. Приводятся ключевые возможности и функции среды проектирования BeeKit.

ВВЕДЕНИЕ

Беспроводная технология ZigBee и стандарт 802.15.4 известны на российском рынке более 3 лет — знакомство состоялось. В сознании инженеров сформировались четкие понятия о назначении, возможностях, преимуществах и отличительных особенностях. Все большее число компаний начинают внедрять радиоканал на базе стандарта 802.15.4 в свои устройства и системы.

В России можно выделить несколько областей, где идет наиболее активное освоение новой технологии — жилищно-коммунальное хозяйство, охранно-пожарные системы, коммерческая автоматизация, автоматизация зданий и системы типа «умный дом». Не стоит забывать и про множество других применений, в которых радиоканал способен эффективно заменить провода, например, промышленная автоматизация и telemetry, медицинская техника и т.д. Перечисленные области подразумевают создание систем, включающих большое число различных устройств. Именно благодаря возможности относительно легко создавать сети из тысяч и более устройств появилась перспектива повсеместного распространения радиоканала там, где прочные позиции уж заняли проводные коммуникации.

Одним из основных сдерживающих факторов до недавнего момента было отсутствие в России обобщенного решения ГКРЧ на стандарт 802.15.4.

Это ограничение не позволяло свободно приобретать и использовать беспроводные системы и устройства конечным пользователям, необходимо было проходить через процедуру частного решения, подразумевающего получение разрешений и сертификатов не только для производителей, но и для потребителей. По предварительной информации, ратификация стандарта в России ожидается весной 2007 года, что повлечет за собой бурный рост числа устройств, изготовленных в России. Причем для потребителей этих изделий не потребуется никаких разрешительных документов на их использование, а для производителей значительно упростится и удашевится процедура получения сертификатов на свою продукцию в государственных регулирующих органах.

Как мы видим, остается все меньше препятствий для производителей при проектировании беспроводных ZigBee устройств. В каждом конкретном случае необходимо решить только технические аспекты реализации.

ЭТАПЫ ПРОЕКТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ. С ЧЕГО НАЧАТЬ РАЗРАБОТКУ?

Любое беспроводное устройство стандарта 802.15.4/ZigBee содержит в себе как приемопередающий радиотракт для доступа к среде передачи данных, так и микропроцессорную часть, отвечающую за обработку пакетов данных, обеспечение сетевых и

системных функций, необходимых для работы в составе беспроводной сети. Аппаратная реализация радиотракта беспроводного ZigBee устройства подробно описана в [1–3]. Разработка программного обеспечения беспроводного устройства, как правило, требует существенно большего времени и усилий разработчиков, нежели проектирование радиотракта. Данная статья является первой из цикла статей, посвященных вопросам практической реализации программного обеспечения беспроводных устройств и систем стандарта 802.15.4/ZigBee.

Какие этапы включает в себя разработка ПО, какие доступны программные средства проектирования и отладки, и существуют ли примеры, позволяющие ускорить процесс создания собственного ПО? Попробуем ответить на эти и другие вопросы, с которыми столкнется программист на начальных этапах.

Прежде чем начинать разработку программного обеспечения беспроводных устройств стандарта 802.15.4/ZigBee, программист должен определить для себя некоторый набор критериев и требований, следующих из технического задания на разработку. Факторы, влияющие на проектирование программного обеспечения беспроводных устройств 802.15.4/ZigBee:

- необходимо или нет обеспечить совместимость разрабатываемых устройств на уровне профилей стека протоколов ZigBee с устройствами других производителей?

- необходимо ли максимально полно использовать уже готовое ПО для сокращения времени разработки и минимизации усилий программистов?

- существует ли возможность самостоятельной разработки сетевого ПО для минимизации объема программного кода и оптимизации скоростных и других параметров функционирования беспроводной сети?

- при проектировании собственного сетевого ПО — удовлетворяет

ли стандартный протокол передачи данных 802.15.4MAC [4] всем требованиям проектируемой системы либо необходим свой, специфический протокол?

— какие сетевые топологии предполагается поддерживать? Достаточно ли будет, например, топологии «звезды», или необходимо построение многоячейковой сети?

В качестве отправной точки при разработке беспроводного ПО компания Freescale Semiconductor предлагает универсальную среду разработки BeeKit™ [5], которая позволяет выбрать базовую библиотеку программного обеспечения и произвести первоначальные настройки функционирования беспроводных систем и устройств посредством простого и понятного графического интерфейса пользователя. В среду BeeKit™ интегрированы различные наборы библиотек ПО, удовлетворяющие требованиям проектируемой системы, с учетом перечисленных факторов.

Таким образом, вначале необходимо определиться с базовой библиотекой беспроводного ПО, исходя из особенностей проектируемой системы. Доступны три варианта базовых библиотек, входящих в среду BeeKit™: SMAC (Simple MAC), 802.15.4 MAC, BeeStack. Выбор библиотеки является одним из важнейших этапов проектирования, т.к. здесь закладывается тот фундамент, который будет определять начальную функциональность каждого из устройств беспроводной сети и всего беспроводного ПО в целом. Какова степень влияния каждого из приведенных выше факторов на выбор той или иной библиотеки? Мы не будем рассматривать возможные комбинации факторов, встречающиеся в реальных системах ввиду значительного объема информации, а подробно рассмотрим их в отдельности.

ОБЕСПЕЧЕНИЕ СОВМЕСТИМОСТИ НА УРОВНЕ ПРОФИЛЕЙ СТЕКА ZigBee.

Необходимость обеспечения совместимости на уровне профилей устройств стека протоколов ZigBee с устройствами других производителей влечет за собой наибольшие финансовые вложения как на этапе разработки, так и во время серийного выпуска изделий. В этом случае однозначно выбирается библиотека стека протоколов BeeStack, уже «в базе» обеспечивающая весь функционал ZigBee сетей, полностью отвечающая спецификации альянса ZigBee версии 2006 года. Необходимо только использовать стандартный профиль устройств, утверж-

денный альянсом, и дописать часть ПО, обеспечивающую связь приложения с сетевым ПО стека протоколов. Преимущества — наименьшее время проектирования больших распределенных беспроводных сетей, возможность выпускать изделия с торговым знаком альянса ZigBee, выпуск устройств с возможностью их использования по всему миру в системах, состоящих из устройств от различных производителей. К недостаткам можно отнести значительные финансовые затраты на сертификационные испытания устройств на совместимость в альянсе, членские взносы в альянс ZigBee и использование стандартных профилей и прочие накладные расходы. Сюда же входят расходы на применение наиболее дорогой элементной базы — микропроцессоров с большим объемом внутренней флэш-памяти (минимум 32...128 Кбайт, в зависимости от класса ZigBee устройства) и повышенной производительностью, предпочтительнее 16- либо 32- разрядные МК для полнофункциональных устройств ZigBee (FFD — Full Function Device) в случае больших систем.

Чаще возникают ситуации, когда нет необходимости в обеспечении совместимости на уровне профилей стека ZigBee, однако и нет достаточных инженерных ресурсов для реализации собственных программных сетевых решений, либо когда необходимо как можно быстрее спроектировать устройство или систему. В этом случае возможно выбрать библиотеку стека BeeStack, настроить параметры собственного профиля и использовать полностью или частично реализованный функционал технологии ZigBee. Это значительно упростит разработку беспроводных устройств, потребует существенно меньше усилий со стороны программистов (по сравнению с собственной реализацией сетевого решения), ускорит время выхода на рынок, однако производитель не имеет право использовать торговый знак альянса на своей продукции, и по-прежнему необходимо использовать наиболее дорогие и производительные микроконтроллеры.

РЕАЛИЗАЦИЯ СОБСТВЕННОГО СЕТЕВОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Идеальный по экономической эффективности вариант — когда компания самостоятельно разрабатывает собственные сетевые программные решения под свои системы и устройства, используя библиотеки SMAC или MAC. При этом у разработчиков уже должен быть опыт работы с

беспроводной технологией стандарта 802.15.4/ZigBee. В этом случае возможно достаточно быстро создавать беспроводные решения, однако не совместимые ни с какими другими решениями стандарта 802.15.4 других производителей. Основное преимущество — возможность создания сложных систем, с оптимальными для системы протоколом и параметрами по быстродействию, с большим количеством устройств, используя дешевые микроконтроллеры. Но, если разработчик только начинает знакомство со стандартом 802.15.4/ZigBee, может потребоваться несколько месяцев на освоение технологии и изучение всех ее тонкостей. Кроме того, ввиду отсутствия опыта практической реализации, программист столкнется с множеством проблем, на преодоление которых также уйдет какое-то время.

При самостоятельной реализации сетевого ПО разработчику доступны два набора библиотек — SMAC и MAC, также входящие в состав среды BeeKit. Отличаются библиотеки значительно. Во-первых, в библиотеке MAC полностью реализован протокол передачи, отвечающий стандарту 802.15.4 [4]. Данный протокол имеет все необходимые сервисные битовые области и соответствующие служебные байты для организации систем любой топологии и любой сложности. Однако его универсальность напрямую отражается на пропускной способности. Кроме того, разработчик не может изменить сам протокол, существует возможность только конфигурировать отдельные его параметры, такие, как длину байт данных, наличие/отсутствие шифрования, передачу пакетов синхронизации и некоторые другие. Сложный протокол тянет за собой множество сервисных функций, которые не могут быть удалены из библиотеки MAC, иначе теряется не только смысл библиотеки, но пропадает и соответствие стандарту 802.15.4. Во-вторых, помимо протокола, библиотека MAC предоставляет разработчику набор служебных функций и процедур, необходимых на этапе организации сети, этапе объединения устройств в сеть, для восстановления сети после сбоев/переподключения питания, поддержки механизма маршрутизации, поддержки синхронизированных сетей с гарантированными временными интервалами (GTS), доступа к среде передачи по механизму CSMA-CA, шифрования AES128 и многое другое. Все возможности библиотеки MAC перечислены в описании стандарта 802.15.4 [4].

Если необходимо обеспечить максимальную пропускную способность системы, либо использовать микроконтроллер с малым объемом флэш-памяти (2...32 Кбайт), выбор разработчика – библиотека SMAC. По сути, в библиотеку SMAC входит небольшой, но достаточный набор функций в исходных кодах, необходимый для работы с радио приемопередатчиком стандарта 802.15.4/ZigBee. При этом у программиста свободны руки в плане как реализации собственного протокола передачи, так и реализации необходимых в конкретной системе сервисных и сетевых функций, без каких-либо лишних, обязательных для библиотеки MAC, программных блоков и модулей.

ПОДДЕРЖКА СЕТЕВЫХ ТОПОЛОГИЙ

С точки зрения поддержки разнообразных сетевых топологий – ситуация более гибкая. Практически возможно реализовать системы со сложной многоячейковой топологией, используя любую из базовых библиотек. Здесь необходимо понимать две общие тенденции. Во-первых,

при реализации одной и той же сетевой структуры практически всегда объем программного кода будет тем меньше, чем проще базовая библиотека заложена в основе. Например, для реализации многоячейковой сети со строго определенной функциональностью и возможностью маршрутизации в каждом из узлов системы при использовании стека ZigBee понадобится минимум 64 Кбайта флэш-памяти, а лучше 128 Кбайт при большом количестве устройств, а используя библиотеку SMAC, легко поместить весь программный код, плюс несложный код по обработке задач приложения, в 32 Кбайта флэш-памяти. В итоге разница в стоимости за счет микроконтроллеров может достигать нескольких долларов, и в случае крупносерийных изделий экономия получается ощутимой. Во-вторых, чем проще базовая библиотека, тем больше работы программистов потребуется для реализации одной и той же задачи. Соответственно, необходимо оценивать эти факторы в комплексе, прогнозировать объемы выпуска изделий, затраты времени программистов на реализацию зада-

чи при использовании той или иной базовой библиотеки. В результате каждый разработчик сможет для себя определить, когда оправдано использовать стек ZigBee, а когда выгоднее разрабатывать собственное сетевое решение на базе SMAC или MAC библиотек. В таблице 1 приведена краткая классификация базовых библиотек ПО SMAC, MAC и BeeStack. В таблице 2 приведен перечень примеров, упражнений и стартовых проектов, доступных разработчикам для быстрого освоения беспроводной технологии 802.15.4/ZigBee и начала разработки ПО для собственных беспроводных устройств и систем.

СРЕДА ПРОЕКТИРОВАНИЯ BeeKit™

Определившись с выбором базовой библиотеки ПО, перейдем к рассмотрению сред проектирования, необходимых для реализации ПО беспроводных приложений. С точки зрения Windows приложений разработчику понадобятся две среды проектирования: BeeKit и CodeWarrior IDE. Остановимся подробнее на среде BeeKit.

Таблица 1. Основные характеристики базовых библиотек ПО, входящих в состав среды BeeKit

Параметры	SMAC	802.15.4 MAC	BeeStack™
Совместимость на уровне профилей стека протоколов ZigBee	Нет	Нет	Да
Совместимость на уровне протокола/стандарта IEEE802.15.4	Нет	Да	Да
Возможность реализации собственного протокола передачи	Да	Нет	Нет
Пропускная способность при передаче данных	Высокая	Высокая/средняя	Средняя/низкая
Поддержка устройств с батарейным питанием (рекоменд.)	Да	Да (RFD)	Да (RFD)
Поддерживаемые аппаратные платформы	1) MC1319x/20x+HCS08 2) MC1321x 3) MC1319x/20x+S12(X) 4) MC1319x/20x+DSC/DSP 5) MC1319x/20x+ColdFire 6) MC1322x *	1) MC1319x/20x+HCS08 2) MC1321x 3) MC1319x/20x+ColdFire 4) MC1322x *	1) MC1319x/20x+HCS08 2) MC1321x 3) MC1319x/20x+ColdFire * 4) MC1322x *
Среды проектирования и отладки			BeeKit™ + CodeWarrior™
Трудозатраты на реализацию систем с топологией:			
1) «точка–точка»	Низкие	Низкие	Низкие
2) «звезда»	Низкие	Низкие	Низкие
3) «дерево»	Высокие	Средние	Низкие
4) «многоячейковая сеть»	Высокие	Высокие	Низкие
Рекомендации по использованию микроконтроллеров: разрядность, объем внутренней Flash памяти, Кбайт	8-, 16-, 32-разрядные 2...32	8-, 32-разрядные 32...64	8-разрядный (RFD, FFD) 32-разрядный (FFD) 64...128
Затраты на разработку беспроводной системы	Средние(оплата работы программистов)	Средние(оплата работы программистов)	Низкие (стоимость ПО стека BeeStack). Высокие (стоимость ПО стека BeeStack + сертификация на совместимость и использование торгового знака альянса ZigBee)
Наличие примеров, упражнений, стартовых проектов	13 примеров/стартовых проектов	16 примеров/упражнений, 1 стартовый проект	Профиль ZigBee Home Automation (13 типов устройств/примеров проектов), 1 стартовый проект

* Поддержка аппаратной платформы ожидается в третьем квартале 2007 г.

Таблица 2. Примеры проектов и упражнения, входящие в состав базовых библиотек SMAC, MAC и BeeStack.

SMAC	802.15.4 MAC	BeeStack™
Accelerometer — Передача данных с датчика ускорения в ПК	MyWirelessApp Demo 01 — Инициализация ПО MAC	GenericApp — Базовый проект стека BeeStack
Lighting Demo — Controller — Управление освещением — выключатель	MyWirelessApp Demo 02 — Сканирование RF каналов, выбор рабочего канала для организации PAN	WirelessUART — Реализация беспроводного последовательного интерфейса
Lighting Demo — Device — Управление освещением — лампа	MyWirelessApp Demo 03 (Coordinator) — Активация поддержки запросов на объединение	HA GenericApp — Базовый проект стека BeeStack для профиля НА
OTAP Programmer — Программатор Flash памяти МК по радио эфиру	MyWirelessApp Demo 03 (End Device) — Активация поиска координатора и генерирование запроса на объединение	HA OnOffLight — Лампа с функцией Вкл./Выкл.
PER Test RX — Тестирование PER — приемная часть	MyWirelessApp Demo 04 (Coordinator) — Функция добавления устройства в сеть	HA OnOffSwitch — Устройство Вкл/Выкл освещения
PER Test TX — Тестирование PER — передающая часть	MyWirelessApp Demo 04 (End Device) — Функция подключения к координатору	HA DimmableLight — Лампа с функцией Вкл/Выкл и регулируемой мощностью свечения
Range Demo Plus RX — Оценка дальности радиопередачи — приемная часть	MyWirelessApp Demo 05 (Coordinator) — Активация функции передачи данных	HA DimmerSwitch — Управление освещением с функцией регулирования мощности свечения
Range Demo Plus TX — Оценка дальности радиопередачи — передающая часть	MyWirelessApp Demo 05 (End Device) — Функция передачи данных	HA Thermostat — Термостат с ЖКИ индикатором
Repeater — Реализация ретранслятора	MyWirelessApp Demo 06 (Coordinator) — Функция отложенной передачи данных к устройству	HA TempSensor — Датчик температуры
Simple PTC — Тестирование SMAC пакетов	MyWirelessApp Demo 06 (End Device) — Функция запроса данных, отложенных ранее для устройства	HA RangeExtender — Сетевой ретранслятор
Test Mode — Проект по тестированию различных радиопараметров приемо-передачи	MyWirelessApp Demo 07 (Coordinator) — Организация синхронизированной сети	HA ConfigurationTool — Устройство конфигурирования и мониторинга сети
Wireless UART — Реализация беспроводного последовательного интерфейса	MyWirelessApp Demo 07 (End Device) — Поддержка синхронизированной сети с запросом данных	HA Combined Interface — Реализация шлюза/диагностирующего устройства сети
Empty project — Стартовый проект	MyWirelessApp Demo 08 (Coordinator) — Активация функции безопасности/шифрования	Test Profile2 — Пример реализации собственного профиля
	MyWirelessApp Demo 08 (End Device) — Устройство использует функции безопасности/шифрования	Empty project — Стартовый проект
	Star Network Demo (Coordinator) — Пример создания сети «звезда»—координатор	
	Star Network Demo (End Device) — Пример создания сети «звезда»—оконечное устройство	
	Empty project — Стартовый проект	

Среда BeeKit обладает простым в использовании графическим интерфейсом и представляет собой инструмент для конфигурирования и предварительной настройки создаваемых проектов для беспроводных устройств и

систем стандарта IEEE802.15.4. Среда не является законченным средством отладки и программирования, где генерируются исходные коды для МК, а только добавляет необходимый для реализации беспроводных приложений

функционал к стандартным средствам разработки ПО, формируя файлы проекта с заданными свойствами.

В состав пакета входят ZigBee-совместимый набор протоколов BeeStack, базовые библиотеки SMAC и MAC, а также необходимый инструментарий, позволяющий создавать программное обеспечение на базе микроконтроллеров с ядром HCS08 компании Freescale. Среда обладает уникальной масштабируемостью, позволяющей поддерживать новые программные библиотеки и функциональные возможности, которые будут доступны в дальнейшем, при поддержке других ядер процессоров, следующих версий стека протоколов ZigBee и стандарта 802.15.4. Поддержка других аппаратных микроконтроллерных платформ ожидается в третьем квартале 2007 г., когда будут доступны соответствующие версии библиотеки BeeStack с поддержкой ядер ColdFire и ARM7. Кроме того, в состав среды входят предварительно сконфигурированные примеры приложений и шаблоны программного кода, а каждый новый проектируемый проект ПО создается отдельно от базовой библиотеки, будь то SMA, MAC или BeeStack, что обеспечивает легкое обновление и повторное их использование.

По завершению настройки параметров и режимов беспроводной связи BeeKit позволяет экспортовать проект в интегрированную среду разработки Freescale CodeWarrior IDE через XML формат для последующей доработки и отладки конечных программных продуктов. Разработчику также предоставляется возможность экспортовать и импортировать изменения параметров, внесенные как в среде BeeKit, так и в среде CodeWarrior IDE.

Таким образом, как уже отмечалось ранее, первым этапом разработки программного обеспечения является работа со средой BeeKit, чему и посвящена эта статья. Будут рассмотрены все аспекты работы с BeeKit: создание проектов, настройка и конфигурирование проектов и экспортование в среду CodeWarrior IDE.

КОМПОНЕНТЫ СРЕДЫ BeeKit, ОСНОВНЫЕ ПОНЯТИЯ

Структура BeeKit представлена на рисунке 1. Графический пользовательский интерфейс BeeKit GUI позволяет создавать и управлять рабочей средой для разработки приложения. Пользователь настраивает рабочую среду, выбирая целевую платформу, тип устройства, структуру приложения, параметры конфигурации сети

и другие критерии. Данная настройка будет подробно рассмотрена в статье. Рабочая среда — это, по сути, формирование файлов проекта, необходимых для компиляции в CodeWarrior IDE.

Редактор проектов (Project Editor) позволяет пользователям редактировать исходные файлы и добавлять внешние файлы к рабочей среде. Компилятор (Compiler) генерирует бинарные файлы для целевых аппаратной платформы. Отладчик (Debugger) предоставляет пользователям возможность отлаживать программу через стандартный отладочный интерфейс микроконтроллера.

Интегрированные библиотеки (Codebase) включают в себя [6]:

- библиотеки SMAC, MAC и BeeStack;
- конфигурационные файлы;
- готовые примеры и упражнения, стартовые проекты;
- исходный код;
- компоненты для поддержки различных аппаратных платформ.

Набор примеров, упражнений и стартовых проектов на базе интегрированных библиотек SMAC, MAC и BeeStack (BeeApps) приведен в таблице 2.

Библиотека ПО SMAC доступна в исходных кодах, содержит минимальный набор программных функций, позволяющих работать с аппаратной платформой стандарта 802.15.4 и создавать любые беспроводные приложения на базе собственных протоколов передачи. Эта библиотека особенно актуальна для приложений с жесткими стоимостными критериями, предоставляя возможность проектировать ПО для МК со встроенной Flash памятью от 4КБ.

Библиотека ПО MAC (IEEE 802.15.4 MAC/PHY 2.0) включает в себя набор программных функций для уровней PHY и MAC стандарта 802.15.4, предоставляет доступ к аппаратному уровню PHY стандарта и обеспечивает транспортный механизм для стека протоколов ZigBee.

Библиотека стека протоколов ZigBee (BeeStack) полностью соответствует спецификации альянса от 2006 года. Данное ПО обеспечивает уровни стека протоколов от транспортного до уровня приложений, позволяя через профили устройств конфигурировать работу беспроводной системы и отдельных ее компонентов. ПО поддерживает любые сетевые топологии («точка–точка», «звезда», «дерево», «многоячейковая сеть»), функции ретрансляции и мар-

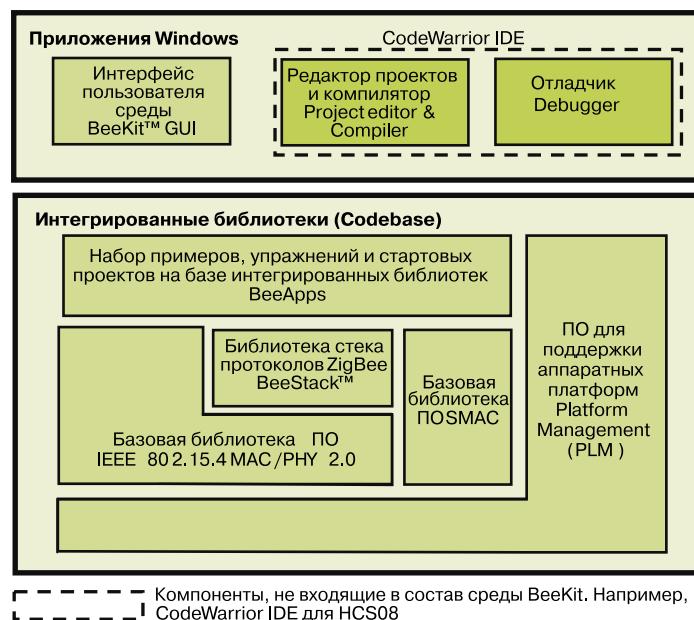


Рис. 1. Структура среды проектирования BeeKit™

шрутизации, безопасности, функции по организации сетей их обеспечению их надежной работы.

ПО для поддержки аппаратных платформ (PLM — Platform Management SW) обеспечивает не только возможность переноса проектов на различные типы МК и радио приемопередатчиков, но и позволяет задавать и управлять режимами энергосбережения, модулями таймеров и часов реального времени, ЖКИ индикатором, светодиодами, кнопками и источниками внешних прерываний, внешними периферийными модулями, активными радиокомпонентами (RxTx переключателем, внешними усилителями) и прочее.

Основные понятия, встречающиеся в среде BeeKit:

- проект (Project) состоит из компонентов, входящих в состав Codebase, которые в итоге формируют единое ПО для беспроводного приложения;
- рабочая среда (Workspace) является набором всех файлов, необходимых в CodeWarrior IDE для линковки приложения. В рабочей среде разработчик также может добавлять свои собственные файлы к проектам;
- решение (Solution) представляет собой «проектный контейнер» и состоит из группы связанных проектов (Projects) или ZigBee-приложений. Используя BeeKit, пользователи работают именно с решениями, даже если решение состоит из одного проекта. Текущее решение можно сохранять и продолжить работу с ним позже;
- обозреватель решений (Solution Explorer) — основная область глав-

ного окна BeeKit, содержащая все связанные проекты и позволяющая разработчику осуществлять навигацию по активным на данный момент решениям.

КЛЮЧЕВЫЕ ВОЗМОЖНОСТИ СРЕДЫ BeeKit

В рамках одной статьи невозможно рассказать о всех функциях и режимах работы среды BeeKit, также как и о последовательности действий при разработке проекта ПО на базе библиотек SMAC, MAC и BeeStack, это будет подробно рассмотрено в последующих публикациях. Сейчас предлагаем познакомиться с основными возможностями среды, позволяющими минимизировать непосредственно работу по программированию при создании собственных программных решений [6].

Стартовое окно программной оболочки показано на рисунке 2. Именно с него начинается работа с BeeKit, здесь возможно выбрать базовую библиотеку ПО (File > Select Codebase), также начать новый проект (File > New project).

Создание нового проекта подразумевает определенные действия:

- выбор проекта — новый проект либо использование готового примера;
- выбор и настройка аппаратной платформы (см. рис. 3 и 4);
- задание типа устройства, функций безопасности и режимов энергосбережения (см. рис. 5);
- задание разрешенных частотных каналов, сетевого идентификатора и MAC-адреса устройства (см. рис. 6).

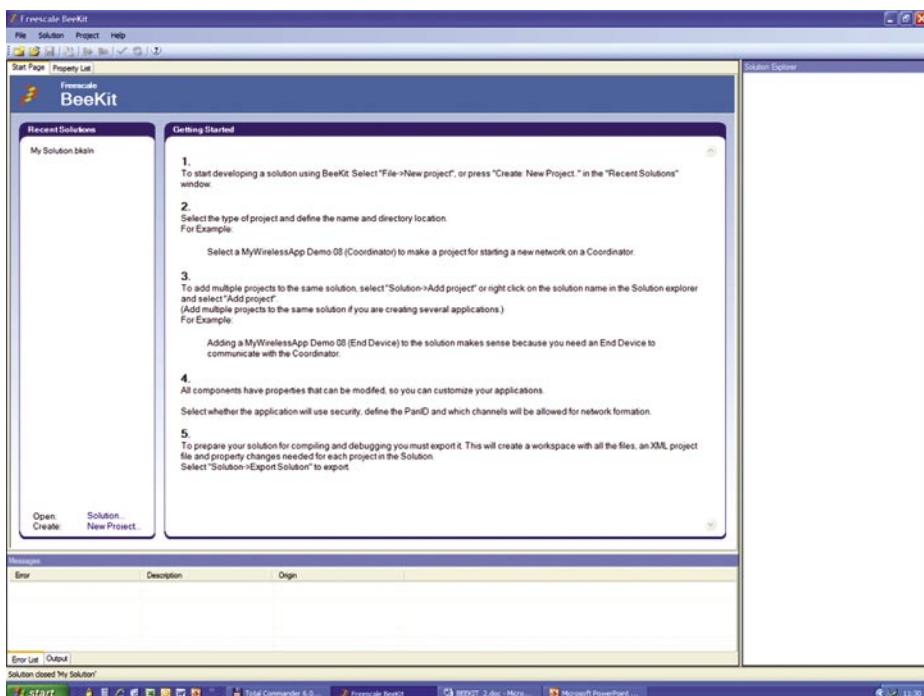


Рис. 2. Стартовое окно среды BeeKit



Рис. 3. Пример окна выбора аппаратной платформы

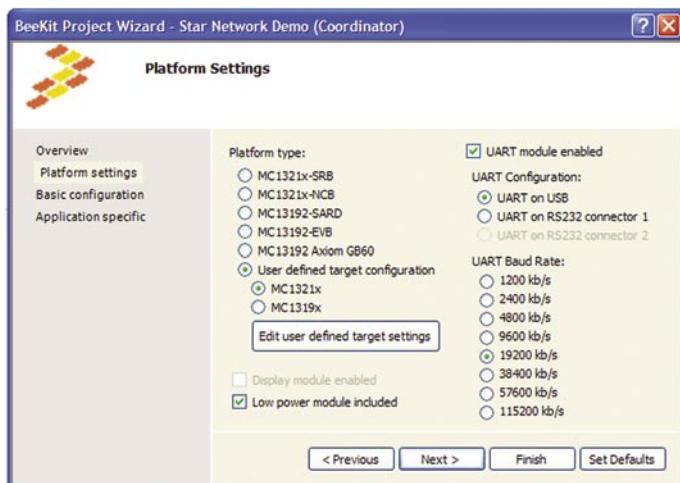


Рис. 4. Окно тонкой настройки ПО под особенности аппаратной части

Последовательность действий и набор параметров для конфигурирования может варьироваться в зависимости от используемой библиотеки и/или типа примера проекта.

После того, как проект создан, возможно либо продолжить конфигурирование ПО уже непосредственно в окне обозревателя решений, либо сразу экспортовать проект в среду CodeWarrior IDE. В любом случае для осуществления этих действий необходимо щелчком правой кнопки мыши на различных подуровнях в области «Решения», в появляющихся контекстных меню выбрать соответствующее действие. Импорт проекта в среду CodeWarrior IDE показано на рисунке 7.

Как только решение было успешно создано и импортировано в CodeWarrior IDE, полученный проект можно использовать различными способами. Первый вариант состоит в том, чтобы проект, сконфигурированный в BeeKit, скомпилировать и загрузить в выбранную аппаратную платформу. Другая возможность заключается в том, чтобы использовать импортированный проект как шаблон для разработки более сложного программного обеспечения в соответствии с собственными требованиями и продолжить программирование уже в среде CodeWarrior IDE.

ЗАКЛЮЧЕНИЕ

Цель статьи — облегчить выбор базовых библиотек программного обеспечения, познакомить с набором примеров, упражнений и стартовых проектов, входящих в состав различных библиотек среды BeeKit. Также авторы начали знакомить читателей со средой проектирования BeeKit, ее возможностями, функциями и режимами работы. В последующих публикациях планируется рассказать непосредственно об использовании среды BeeKit при реализации собственных программных решений на базе примеров из библиотек SMAC, MAC и BeeStack.

ЛИТЕРАТУРА

1. Соколов М., Гришин А. Аппаратные средства реализации беспроводных решений ZigBee/802.15.4. Современная электроника, №9, 2006, с. 28–35.

2. Алый А., Разработка модуля беспроводной передачи телеметрических данных в диапазоне частот 2,4 ГГц. Современная электроника, №9, 2006, с. 36–40.

3. М. Соколов. Программно-аппаратное обеспечение беспроводных сетей на основе технологии ZigBee/802.15.4.

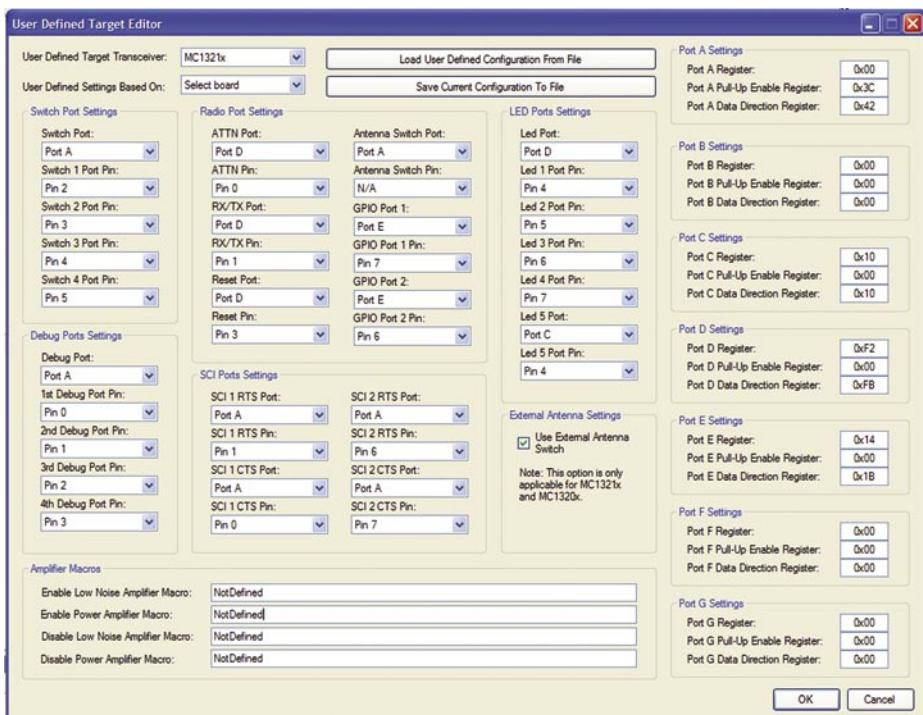


Рис. 5. Пример окна выбора типа устройства, параметров безопасности и энергосберегающих режимов



Рис. 6. Пример окна задания разрешенных частотных каналов, ID-сети и MAC-адреса устройства

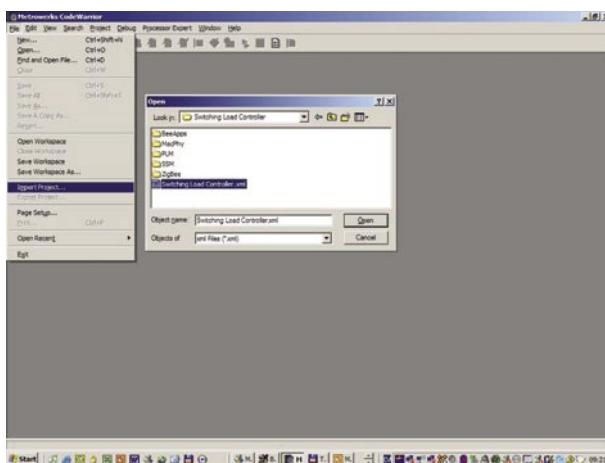


Рис. 7. Импорт проекта из среды BeeKit в среду CodeWarrior IDE

Электронные компоненты, №11, 2004,
с. 80—87.

4. [grouper.ieee.org/groups/802/15/
pub/TG4.html](http://grouper.ieee.org/groups/802/15/pub/TG4.html).

5. www.freescale.com/zigbee.
6. [BKWCTKUG.pdf](#) — файл из комплекта документации среды BeeKit.

МИКРОСХЕМЫ

TEXAS INSTRUMENTS



- Микроконтроллеры
- Стабилизаторы
- Стандартная логика
- Операционные усилители
- Микросхемы памяти

ANALOG
DEVICES

MICROCHIP

ЧИП
ИНДУСТРИЯ

129110, г. Москва,
ул. Гиляровского, д. 39

Тел.: (495) 780-95-07
(495) 780-95-05
Факс: (495) 631-31-45

postavka@chipindustry.ru
www.chipindustry.ru

Контрактное производство электроники

ISO 9001:2000



- Весь спектр услуг по производству электроники
- Производство электроники в любых количествах – от одного изделия до крупных серий
- Индивидуальный подход к заказчикам
- Выполнение срочных заказов в течение 1-2 дней
- Самые современные технологии, оборудование и материалы
- Поставка электронных компонентов с собственного склада
- Входной контроль компонентов и комплектующих
- 100-процентный выходной контроль качества выпускаемой продукции

Четыре автоматизированные линии поверхностного монтажа с суммарной производительностью 150 000 компонентов в час

Автоматическая оптическая инспекция и рентген-контроль

АЛЬТОНИКА

115230 Москва, Варшавское шоссе, 42
Тел.: (495) 787-41-09, 787-11-16
E-mail: ems@altonika.ru
www.altonika.ru

Heraeus

Мировой производитель платиновых датчиков температуры

 **ПЛАТАН**
www.platan.ru

**ПЛАТИНОВЫЕ
датчики
температуры**

- диапазон измерения -196...+1000°C
- отличная линейная зависимость сопротивления от температуры
- высокая точность измерений
- SMD и выводные корпуса
- корпусированные датчики температуры

Москва, ул. Ивана Франко, д. 40, стр. 2 Тел./факс: (495) 97-000-99
Почта: 121351, Москва, а/я 100 E-mail: platan@aha.ru



ЭЛАСТИЧНЫЕ ТЕПЛОПРОВОДЯЩИЕ ДИЭЛЕКТРИЧЕСКИЕ ПРОКЛАДКИ "НОМАКОН"

Материал КПТД-2-60/250

- интенсивный теплоотвод от нагреваемой поверхности;
- монтаж полупроводниковых элементов без нанесения теплопроводящей пасты, что гарантирует чистоту и сокращает время сборки;
- снижение себестоимости за счет уменьшения трудоемкости сборки и замены дорогостоящей керамики.

Удельное объемное сопротивление

0,9-1,1 Вт/(мхК)

Теплопроводность

3,5 кВ

(4-4,5) $\times 10^{-3}$

Тангенс угла диэлектрических потерь (при 1000 Гц)

от -60° С до 250° С

Рабочая температура

от 0,15 до 3 мм

Толщина

КОМПАУНД ЗАЛИВОЧНЫЙ ТЕПЛОПРОВОДЯЩИЙ ДИЭЛЕКТРИЧЕСКИЙ ДВУХКОМПОНЕНТНЫЙ

- предназначен для капсулирования и герметизации электронных схем и микросборок путем полной или частичной заливки;
- эффективный теплоотвод от греющихся элементов и микросборок;
- обладает эластичностью, химической инертностью и хорошей адгезией.

ТУ РБ 100009933.004-2001

ИЗОЛИРУЮЩИЕ ВТУЛКИ

- Обеспечивают изоляцию полупроводниковых приборов от корпуса радиатора при креплении винтами.
- Втулки под корпус ТОЗ, ТО220. Крепление винтами M2,5 и M3

Цены снижены
на 50%
правос-лист на сайте
www.nomakon.ru

220013, г. Минск, РБ, а/я 185
тел. (+375 17) 202 13 51
тел./факс (+375 17) 211 35 49

Представительство в Москве:
ООО "Номакон-Центр", тел.: (495) 729 59 23, 729 59 24

ООО "Додзка", тел.: (495) 797 55 45

Представительство в Санкт-Петербурге:
ООО "ЭКСТ", тел.: (812) 712 65 12