Разработка программного обеспечения для сети ZigBee

на базе библиотеки EmberZNet. Часть I

В последние годы все большую популярность при разработке беспроводных систем сбора данных и управления завоевывает технология ZigBee. Первые сети, использующие эту технологию, как правило, проектировались на базе модулей со встроенным программным обеспечением. Такой подход существенно упрощает и ускоряет разработку. Вместе с тем для больших проектов, в которых количество устройств достигает десятков тысяч, экономически целесообразно разрабатывать собственные радиомодули и программное обеспечение.
Многие производители микросхем для сетей ZigBee предлагают сопутствующие библиотеки для разработки встраиваемого программного обеспечения. Так, компания Ember, ведущий производитель однокристальных решений для сетей ZigBee, предоставляет для своих кристаллов библиотеку EmberZNet, позволяющую создавать полностью ZigBee-совместимые устройства. В статье рассмотрены особенности разработки программного обеспечения на базе этой библиотеки. Первая часть статьи посвящена вопросам планирования сети, а вторая — конкретной реализации приложения.

Сергей Солодунов
sys@efo.ru

Введение

Компания Ember известна многим разработчикам благодаря своей популярной серии микросхем EM250/EM260. Это были первые появившиеся на рынке системы на кристалле для построения ZigBee-сетей. Они содержат приемопередатчик ZigBee, процессорное ядро и периферийные узлы. Популярная серия радиомодулей ETRX2 компании Telegesis также выполнена на базе кристалла EM250.
В этом году компания Ember выпустила новое поколение высокоинтегрированных микропотребляющих микросхем EM351/EM357, обладающих на данный момент передовыми характеристиками.

Микросхемы EM351/EM357

Микросхемы новой серии представляют собой систему на кристалле на базе популярного 32-разрядного ARM-процессора Cortex-M3. Встроенный приемопередатчик IEEE802.15.4 на 2,4 ГГц обеспечивает более высокую, чем у микросхем EM250, выходную мощность до 8 дБм и чувствительность –101 дБм. Микросхемы содержат встроенную Flash-память (EM351—128 кбайт и EM357—192 кбайт) с возможностью включения защиты от несанкционированного доступа. Увеличенный по сравнению с предыдущими моделями объем памяти и возросшая производительность процессорного ядра позволяют реализовать на базе этих микросхем компактные однокристальные устройства со стандартными профилями ZigBee, такими, например, как Home Automation (домашняя автоматизация) и Smart Energy (автоматическое считывание показаний счетчиков). Одной из важных характеристик новой серии, выделяющих продукцию Ember на рынке однокристальных ZigBee-устройств, стало снижение тока потребления в спящем режиме до 400 нА. За счет этого снижения, а также благодаря повышению скорости выхода из режима сна время жизни батареи питания может быть увеличено на 25%. Основные характеристики микросхем EM351/EM357 приведены в таблице 1.

Средства разработки

Компания Ember предлагает все необходимые отладочные средства для разработки приложений на базе своих кристаллов. Основным аппаратным отладочным средством является многофункциональное устройство InSight Adapter, позволяющее создать альтернативный проводной отладочный интерфейс Ethernet для программирования узлов и отслеживания данных, передаваемых в эфир. Для ускорения разработки предлагаются отладочные платы и мезонинные радиомодули, выполненные с использованием микросхем EM351/357. Радиомодули имеют различные варианты исполнения высокочастотной части: с интегрированной антенной или с разъемом для присоединения внешней антенны, с дополнительным внешним усилителем мощности или без него. Программные средства включают в себя встраиваемое программное обеспечение для микросхем Ember, компилятор, набор

программ и утилит для внутрисхемного программирования и отладки как на уровне одного узла сети, так и для отладки взаимодействия узлов на сетевом уровне.

Пакет встраиваемого программного обеспечения поставляется бесплатно вместе с микросхемами Ember. Он включает в себя библиотеку EmberZNet, реализующую все уровни стека протоколов ZigBee, исходные коды демонстрационных проектов, использующих эту библиотеку, а также широкий набор полезных утилит, включающих загрузчик по радиоканалу и функции для тестирования аппаратных узлов во время производства. Библиотека EmberZNet для семейства EM35x сертифицирована на соответствие спецификации ZigBee PRO Feature Set и портирована для среды разработки IAR EWARM IDE. Стоит отметить, что среда IAR EWARM достаточно популярна среди разработчиков, и многие используют ее в своих текущих проектах, поэтому возможность ее применения в новом проекте для многих фирм означает исключение затрат на приобретение и освоение нового программного обеспечения.

Для разработки устройств, реализующих стандартные профили ZigBee, предлагается программа, генератор конфигурационных файлов, — Application Builder [1, 2]. Эта программа, используя исходные коды демонстрационных проектов, входящие в состав пакета EmberZNet, определяет, какие порции исходных кодов будут входить в проект для реализации требуемых стандартных операций, а какие нет. При помощи простого и удобного интерфейса можно задать тип устройства, профиль приложения, конфигурационные параметры стека, выбрать необходимые кластеры из библиотеки кластеров [1] и сконфигурировать приемопередатчик Ember. Результатом работы программы Application Builder являются конфигурационные заголовочные файлы и файлы проекта для используемого компилятора.

Основной программой для отладки проектов на сетевом уровне является программа InSight Desktop, которая представляет собой анализатор трафика беспроводной сети ZigBee. Она позволяет записывать, декодировать и фильтровать пакеты, передаваемые в сети, а также отображает взаимодействие узлов в графической форме. Сообщения, передаваемые по сети, записываются в лог-файл в порядке их поступления. Важной особенностью является возможность при отображении лог-файла группировать пакеты в транзакции заданного уровня (MAC-уровня, сетевого уровня или транспортного). Выполняя пошаговый просмотр лог-файла, разработчик имеет возможность по мере необходимости вникать в подробности более низких уровней и быстро выявлять причины неправильной работы сети. Графическое окно (рисунок) позволяет загружать план местности, где эксплуатируется сеть (чертеж здания, садового участка, гаража в формате JPEG). Наблюдение взаимодействия узлов на фоне реального плана делает отладочный процесс еще более наглядным, что можно использовать уже в готовой системе для контроля работы сети.

InSight Desktop взаимодействует с сетью через резервный канал Ethernet. Связь целевого узла с резервным каналом Ethernet обеспечивается многофункциональным устройством InSight Adapter, которое подключается к отладочному SIF-порту микросхемы Ember. Библиотека EmberZNet выполнена таким образом, что приемопередатчик все отсылаемые и принимаемые пакеты дублирует в отладочный SIF-порт. Поэтому, если узел подключен при помощи устройства InSight Adapter к локальной сети, то все пакеты, отсылаемые узлом в эфир, будут переданы также в центральную отладочную программу InSight DeskTop. Кроме того, при загрузке в узел специального программного кода он становится сниффером сети, то есть узлом, прослушивающим эфир на определенном частотном канале и собирающим все пакеты, которые ему слышны.

Таблица 1. Основные характеристики микросхем EM351/EM357

Параметры		EM351	EM357
Микроконтроллер	Процессорное ядро	ARM Cortex-M3	
	Flash-память	128 кбайт	192 кбайт
	ОЗУ	12 кбайт	
	Эмулированная EEPROM	8 кбайт	
Радиочастотный тракт	Максимальная мощность передатчика (режим Boost)	8 дБм	
	Чувствительность приемника (режим Boost)	−101 дБм	
Периферийные узлы	Входы/выходы	24	
	Последовательные интерфейсы	Два последовательных канала с поддержкой UART, SPI, TWI, DMA	
	АЦП	Сигма-дельта АЦП (14-разрядов, 6 каналов, 3 дифф. входа)	
	Отладочный интерфейс	JTAG	
	Таймеры	Два 16-разрядных таймера с поддержкой различных источников тактового сигнала	
Ток потребления	Передача данных	40 мА (+8 дБм)	
	Прием данных	25 мА	
	Спящий режим (при работающем таймере)	800 нА	
	Спящий режим (при отключенном таймере)	400 нА	
Габариты	Корпус (размеры)	QFN48 (7×7 мм)	
Рабочие условия	Температурный диапазон	−40... +85 °C	
	Напряжение питания	2,1−3,6 В	

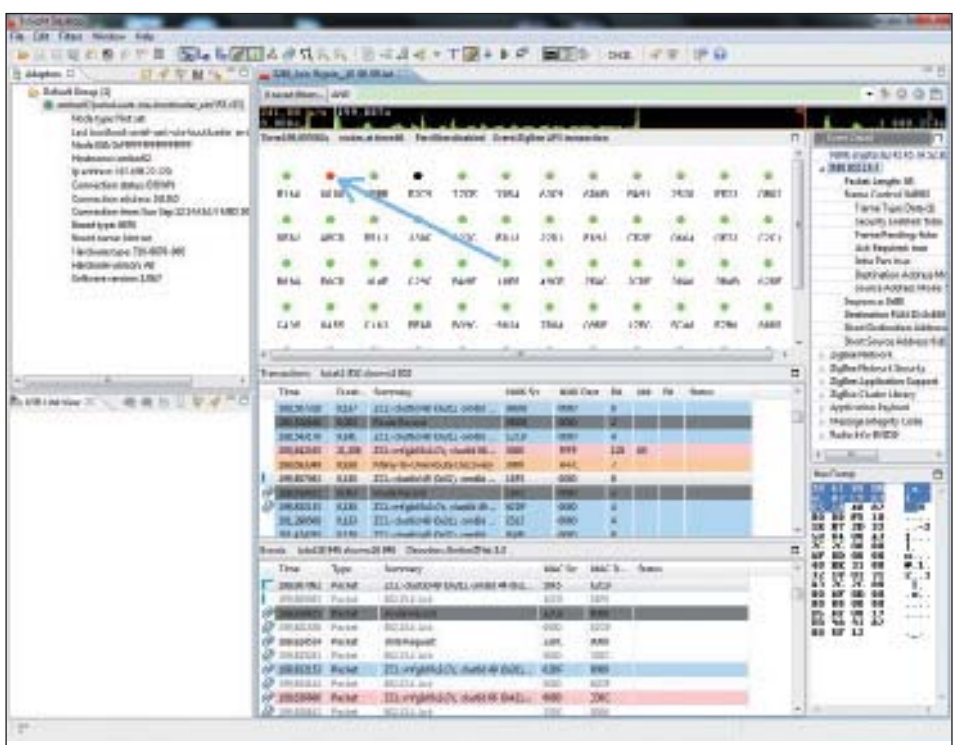

Рис. Графическое окно программы InSight Desktop

Разработчик, посылая сообщения узлам через резервный канал, может управлять и «перепрошивать» узлы беспроводной сети через резервный канал Ethernet или по радиоканалу. Таким образом, разработчик распределенной сети может на экране одного компьютера наблюдать взаимодействие всех беспроводных узлов и перепрограммировать их, не покидая своего рабочего места.

Все перечисленные отладочные средства Ember доступны по цене и могут поставляться как отдельно, так и в составе отладочного комплекта. Разработчикам, которые приобретают отладочный комплект, предоставляется доступ на интернет-портал технической поддержки, где они имеют возможность задавать свои вопросы напрямую инженерам Ember.

Планирование сети ZigBee

Как уже упоминалось, пакет EmberZNet содержит демонстрационные примеры приложений, которые могут быть использованы разработчиками в своих проектах в качестве основы для разработки собственных приложений. Поэтому, как правило, разработчики начинают работу над проектом с изучения данных примеров. Но демонстрационные приложения Ember не могут быть универсальными, и их необходимо адаптировать под индивидуальные нужды каждого приложения. Чтобы выяснить, какие изменения надо внести в базовый пример для того, чтобы ZigBee-сеть полностью отвечала заданным требованиям, необходимо заранее продумать и спланировать детали реализации будущей сети. Это позволяет избежать многих трудностей и на дальнейших этапах разработки системы.

На этапе планирования необходимо решить, какие типы устройств в сети будут использоваться, и оценить объемы передаваемых ими данных, чтобы убедиться, что пропускающей способности сети хватает для обслуживания всех устройств. Выбор оптимальных методов маршрутизации позволит сократить объемы служебного трафика, сократить время реакции на событие и повысить надежность сети. Предварительная оценка объемов памяти позволит сделать правильный выбор аппаратного обеспечения для ZigBee-устройств.

Типы узлов и объемы передаваемых данных

Начать планирование сети логично с определения типов используемых узлов и их количества. Существует четыре возможных типа узлов: координатор, роутер, дочернее спящее устройство и дочернее мобильное устройство. Координатор выполняет функции по формированию сети, а также является одновременно доверительным центром (trust-центром). Доверительный центр устанавливает политику безопасности и задает настройки во время подключения устройства к сети. Спящие и мобильные устройства используют режимы пониженного энергопотребления. Как правило, это узлы с батарейным питанием. Обычно они играют роль датчиков или контроллеров каких-либо исполнительных устройств. Их количество диктуется потребностью конкретного приложения. Роутеры, кроме того, что они могут передавать в сеть свою собственную информацию, также осуществляют маршрутизацию пакетов по сети и должны быть готовы к передаче данных в любой момент времени. Поэтому такие узлы не используют режимов пониженного энергопотребления и имеют стационарное питание. Их количество в сети должно быть достаточным для обслуживания требуемого количества спящих и мобильных узлов, а также достаточным для охвата всей территории объекта, на котором устанавливается сеть. При этом для повышения надежности сети желательно обеспечить некоторую избыточность роутеров, чтобы при выходе из строя одного узла данные могли бы достичь своей цели обходным путем.

Для поддержки и формирования маршрутов все роутеры в своей памяти хранят таблицу соседних узлов — Neighbor Table. В эту таблицу записываются адреса и уровень сигнала от узлов, находящихся в радиовидимости роутера, то есть на расстоянии в один хоп. Это означает, что максимальное количество соседних узлов, через которые данный роутер может проложить маршрут, равно количеству записей в этой таблице. Максимальное количество записей — 16. Если соседних узлов будет больше, чем может вместить в себя таблица Neighbor Table, то стеком EmberZNet будут автоматически отбираться узлы, от которых поступают сигналы с более высоким уровнем. Таким образом, для передачи сообщений на узлы, близко расположенные, но не входящие в таблицу Neighbor Table, потребуется ретрансляция через другие узлы. Следовательно, малый размер этой таблицы может привести к увеличению длины маршрутов (и потенциально к более высоким задержкам при передаче сообщений). Поэтому необходимо на этапе проектирования сети оценить максимально возможное число соседей для каждого роутера и определить оптимальный размер таблицы Neighbor Table. Каждый элемент таблицы занимает в памяти RAM 18 байт.

Информация о дочерних устройствах (спящих и мобильных) сохраняется в другую таблицу, называемую таблицей дочерних устройств — Child Table. Общее количество спящих или мобильных узлов, обслуживаемых одним роутером, не должно превышать 32. Это число также должно быть спланировано и задано программистом на начальном этапе проектирования.

Пропускная способность ZigBee-сети составляет 250 кбит/с. Но полоса пропускания делится между всеми участниками информационного обмена: когда какой-либо узел ведет передачу, все остальные ждут, пока канал освободится. Кроме того, пакеты данных содержат служебные заголовки, а также ZigBee-роутеры периодически обмениваются служебными пакетами с информацией, необходимой для поддержания сетевых функций. Реальная скорость данных зависит от выбранных методов маршрутизации, количества ретрансляций, наличия или отсутствия шифрования, помеховой обстановки. Для оценки, из 128 байт, передаваемых за 1 транзакцию, 37 задействованы под служебные заголовки. Таким образом, размер полезных данных составляет 91 байт. При использовании шифрования для защиты данных размер служебной нагрузки на каждый пакет увеличивается, и размер полезных данных при этом сокращается с 91 байта до 73. Опытным путем установлено, что ZigBee-сеть способна функционировать при нагрузке в 50 кбит/с на каждый узел. При этом рекомендуется вдвое уменьшить это значение для обеспечения высокой надежности передачи данных.

Методы маршрутизации и служебный трафик

Рассмотрим методы маршрутизации, предлагаемые стандартом ZigBee. Первый метод — Table Routing — используется для адресной передачи сообщений от одного узла в сети любому другому. Он применяется в сетях, в которых информация не консолидируется одним или несколькими узлами — концентраторами данных. Рассмотрим его более подробно.

Для обнаружения маршрута от узла-источника к узлу-получателю узел-источник инициирует процесс Route Discovery — исследование сети, который начинается с широковещательного запроса Route Request. После того как процесс поиска маршрута завершится, каждый узел, участвующий в маршруте, запишет в свою маршрутную таблицу два сетевых адреса: узла-назначения и следующего роутера, которому необходимо ретранслировать сообщение, предназначенное для соответствующего получателя. Механизм Table Routing обеспечивает 100%-ное обнаружение маршрута, если таковой существует, при этом при утрате старого маршрута ищется новый.

Этот вид маршрутизации был первым, который использовался в ячеистых ZigBee-сетях. На практике оказалось, что ZigBee-сети часто применяются в сенсорных сетях, где собранная информация отправляется в единый центр сбора данных (синк). Недостатком маршрутизации Table Routing для таких систем является то, что узлы, находящиеся вблизи центрального узла, участвуют в большом количестве маршрутов, и поэтому их маршрутная таблица при большом количестве узлов быстро переполняется и не позволяет построить большую сеть. Поэтому был разработан другой механизм маршрутизации, учитывающий специфику информационных потоков, стремящихся консолидироваться в одной или нескольких точках. Этот вид маршрутизации называется Many-to-One Source Routing.

При использовании этого механизма множество широковещательных запросов Route Request к центральному узлу заменяется одним широковещательным запросом SINK_ADVERTISE, после получения которого все узлы в сети получают маршруты до центрального узла-синка. При этом каждый узел хранит в своей памяти только один адрес узла, которому нужно передать сообщение, чтобы оно достигло центрального устройства. Информация об обратных маршрутах в сети не сохраняется, то есть центральный узел, не имеющий достаточно памяти для хранения полных маршрутов до всех узлов, не имеет маршрутов к удаленным узлам. Знание этих маршрутов, однако, центральному узлу необходимо для посылки узлу-источнику квитанций о доставке

сообщения. Для того чтобы центральный узел мог послать такую квитанцию, узел-источник, перед тем как послать собственно данные, инициирует в адрес центрального узла сообщение Route Record. Каждый роутер, ретранслируя это сообщение, добавляет в него информацию о маршруте. Таким образом, центральный узел получает полную информацию о маршруте до узла-источника и использует ее при посылке квитанции о получении последующего пакета данных. В этот момент, сразу после получения пакета от удаленного узла, имея полный маршрут к нему, центральный узел может послать удаленному узлу кроме квитанции и какую-либо дополнительную, например управляющую, информацию.

При использовании метода Many-to-One сокращается объем служебного трафика, так как процесс поиска маршрута с широковещательными рассылками стартует только от узла-концентратора, в то время как в механизме Table Routing широковещательные рассылки генерировались бы каждым удаленным узлом для поиска маршрута к центральному. Кроме того, промежуточные ретрансляторы не хранят информацию о пути от концентратора к удаленным узлам, что предохраняет их таблицы от переполнения. Пакет SINK_ADVERTISE рассылается концентратором с периодичностью, которая регулируется настройками стека. В демонстрационных приложениях Ember этот интервал задается константой TIME_BEFORE_SINK_ADVERTISE. Следует иметь в виду, что, если маршрут от удаленного узла к концентратору испортился, он восстановится только со следующей рассылкой.

Заканчивая обсуждать механизм Many-to-One Source Routing, следует сказать, что в сети может быть несколько узлов-синков, собирающих данные и рассылающих время от времени служебные пакеты SINK_ADVERTISE для перерасчета маршрутов.

Еще два метода маршрутизации, используемые ZigBee-сетями, — это Broadcast Routing, применяемый для широковещательной рассылки, и Multicast Routing, предназначенный для групповой доставки сообщений.

Метод маршрутизации Broadcast Routing используется для рассылки широковещательных сообщений. При отправке широковещательного сообщения никогда не используются механизмы подтверждения получения, так как это может привести к перегрузке сети. Для повышения надежности доставки широковещательное сообщение повторяется трижды. Использование широковещательных рассылок нагружает сеть, так как каждое из них ретранслируется и повторяется трижды. Поэтому введен дополнительный параметр, называемый радиусом, ограничивающий количество ретрансляций широковещательного пакета.

Multicast Routing предназначен для доставки групповых сообщений. Multicast-сообщение доставляется только членам группы. Остальные устройства только ретранслируют его. Multicast-режим фактически является фильтрованным широковещательным режимом со всеми вытекающими последствиями. Поэтому использовать его, как и широковещательный режим, надо крайне осмотрительно. При небольшом размере группы зачастую выгоднее отправить несколько адресных посылок вместо одной групповой.

Важно также помнить, что ZigBee-спецификация (а значит, и стек Ember) накладывает ограничение на количество одновременно рассылаемых широковещательных сообщений. Дело в том, что согласно спецификации каждый роутер, получив и ретранслируя такое сообщение, должен занести его в свою broadcast-таблицу. Это сделано для того, чтобы широковещательные сообщения не циркулировали по сети бесконечно. А так как количество мест в таблице ограничено (ввиду ограниченности RAM) и широковещательное сообщение (в большой сети) может циркулировать по сети несколько секунд, то само собой накладывается ограничение на количество отсылаемых сообщений за определенный период времени. Безопасная оценка — это 8 broadcast-сообщений в течение 9 секунд.

Как стало ясно по итогам разбора существующих в ZigBee методов маршрутизации, служебный обмен информацией в сети достаточно интенсивен. Поэтому он влияет на пропускную способность в сети. Для оценки объемов служебной информации желательно иметь представление о видах служебных сообщений, передаваемых в сети. В их число входят широковещательные пакеты Route Request и адресные Router Record и Route Error, а также пакеты-квитанции MAC Ack и APS Ack, которые используются для подтверждения успешного получения пакета на уровне приложения. Еще один вид служебного пакета — это Link Status. Он тоже широковещательный, но имеет ограничение по радиусу в 1 хоп. Пакеты этого типа используются для обмена информацией между роутерами о качестве связи.

Объем служебного трафика также можно регулировать при помощи некоторых настроек стека. Рассмотрим эти настройки.

Можно регламентировать использование механизма Route Discovery. Включение опции Force Route Discovery при отсылке пакета приводит к обязательному автоматическому поиску нового маршрута перед отправкой каждого пакета. Такой режим целесообразно использовать при отправке сообщений мобильным устройствам, так как велика вероятность того, что старый маршрут устарел. При включенной опции Enable Route Discovery маршрут ищется только тогда, когда обнаружен его обрыв. И, наконец, режим Disable Route Discovery используется в том случае, когда применяется метод маршрутизации Many-to-One Source Routing.

При использовании механизма маршрутизации Many-to-One Source Routing и включенной опции Enable Route Discovery механизм поиска маршрута Table Routing включается, когда ранее проложенный маршрут к концентратору не работает. То есть утраченный маршрут может быть восстановлен до того, как появится следующий пакет Many-to-One Route Discovery.

Для снижения количества служебного трафика необходимо выбирать оптимальные методы маршрутизации, применяя для доставки адресных пакетов, в зависимости от задачи, Table Routing, Many-to-One Source Routing или их комбинацию, а также по возможности избегать широковещательных и групповых рассылок.

Таблица 2. Распределение объема памяти

Расход RAM	Тип узла	Байт/запись	Минимум	Максимум
Binding Table*	R, S, E	2		
Address (Cache) Table	R, S, E, TC	10		
Route Table	R, S	6	20	
Discovery Table	R, S	9	8	
Multicast Table	R, S, E	3		
Source Route Table	S, TC	4		255
Child Table*	R, S	4		32
Neighbor Table	R, S	18		16
Key Table*	R, S, E, TC	24		
Буферы	R, S, E	40		

Примечание: S — синк; R — роутер; E — дочернее утройство; TC — доверительный центр; * — сохраняется также в Simulated EEPROM.

Оценка объемов памяти

Для оценки требуемых объемов памяти полезно использовать таблицу 2. В ней указано, сколько памяти RAM используется на каждую запись в служебных таблицах. Количество записей в каждой таблице может варьироваться. Для некоторых таблиц определены верхние или нижние границы для количества записей. Так, в стеке ZigBee PRO определено, что роутер обязательно должен иметь как минимум 20 записей в таблице маршрутов и 8 записей в таблице Discovery Table. Чем больше записей могут хранить таблицы, тем надежнее связь в больших сетях.

Еще одна важная опция при оценке объемов памяти — это тип используемого концентратора в сетях сбора данных. Если в качестве концентратора применяется устройство

с большим объемом памяти, способное хранить маршруты до всех узлов в сети, необходимо использовать опцию HIGH_RAM_CONCENTRATOR. В этом случае размер маршрутной таблицы Source Route Table концентратора должен быть равен максимальному количеству удаленных узлов, отправляющих на него данные. При выбранной опции HIGH_RAM_CONCENTRATOR удаленные узлы освобождаются от необходимости каждый раз при передаче данных предварительно посылать служебный пакет Route Record. Если же концентратор имеет ограниченные ресурсы по памяти, используется опция LOW_RAM_CONCENTRATOR. В этом случае концентратор хранит не все пути до удаленных узлов, а только до тех, которые недавно прислали ему сообщение, и хранит их в течение короткого времени, достаточного для отсылки подтверждения о получении сообщения. Размер маршрутной таблицы (Source Route Table) в этом случае должен быть равен максимально возможному количеству удаленных устройств, одновременно отправляющих сообщения на концентратор. При использовании такого механизма экономится память концентратора, но увеличивается сетевой трафик. Кроме того, при возникновении срочной необходимости передать данные, концентратору придется восстанавливать маршрут при помощи Table Routing или ждать следующего входящего сообщения от интересующего узла.

Надежность сети, режимы работы спящих устройств

Для того чтобы обеспечить надежное функционирование разрабатываемой ZigBee-сети, следует использовать подтверждение доставки сообщений, обеспечить запас по полосе пропускания и по времени реакции на событие.

Разные приложения имеют разные требования к времени реакции на событие. Для системы сбора данных достаточно большие задержки могут быть приемлемыми, а для систем безопасности время реакции на событие должно быть как можно меньше. Основные задержки в сети определяются скоростью продвижения информации по сети и длиной пути. Безопасная оценка для ненагруженных сетей — 10 мс/хоп. Максимальное значение обычно не превышает 50 мс/хоп.

Еще один аспект, влияющий на время реакции на событие, — это режимы сна и бодрствования спящих узлов. Существуют различные варианты режимов работы спящих устройств. В зависимости от приложения спящие устройства могут просыпаться по событию и немедленно передавать сообщение, а могут просыпаться с определенной периодичностью для отправки регулярного дежурного сигнала или для приема управляющих и информационных сообщений. Время, в течение которого сообщение для спящего устройства хранится в памяти его родительского роутера, может конфигурироваться, но при этом максимальное значение этого времени составляет 30 секунд. Поэтому интервал, с которым спящие узлы должны производить опрос своего родительского узла на наличие входящих сообщений для них, не должен превышать этот максимум.

Также желательно иметь резервные устройства, выполняющие функции концентратора данных. При выходе из строя одного из них узлы, передававшие свои сообщения пропавшему концентратору, автоматически переключатся на концентратор, продолжающий работать в сети. Использование нескольких концентраторов особенно оправданно в больших сетях, где за счет этого можно сократить путь данных от конечных узлов к концентратору и, следовательно, разгрузить сеть.

Завершив этап планирования сети, можно переходить к разработке программного кода приложения.

Во второй части статьи будет рассмотрена архитектура стандартного приложения ZigBee, использующего пакет EmberZNet, перечислены и описаны основные константы, требующие конфигурирования; функции, используемые для формирования сети и передачи данных; рассмотрены основные обработчики событий стека.

Литература

1. Солодунов С. Реализация стандартного профиля ZigBee Home Automation на базе платформы Ember. Часть 1. // Беспроводные технологии. 2009. № 3.
2. Солодунов С. Реализация стандартного профиля ZigBee Home Automation на базе платформы Ember. Часть 2. // Беспроводные технологии. 2009. № 4.