

Беспроводные сети компании Ember: ZigBee и EmberNet

Татьяна Кривченко, к.т.н.
tkr@efo.ru

Компания Ember специализируется на разработке и производстве приемопередатчиков, программного обеспечения и отладочных средств для беспроводных сетей сбора данных и управления.

Штаб-квартира Ember располагается в Бостоне (США) и там же находится отделение Ember, ответственное за разработку программного обеспечения и отладочных средств. Второе отделение Ember, размещенное в Кембридже (Англия), объединило спе-

В течение ряда последних лет мы являемся свидетелями поступательного развития технологии ZigBee. После анонса первоначальной идеи несколько производителей полупроводниковых кристаллов (Ember, Freescale, ChipCon, ZMD) предложили свои приемопередатчики стандарта IEEE 802.15.4, затем была опубликована первая версия спецификации ZigBee. Однако разработка реальных сетевых приложений сдерживалась незрелостью программного обеспечения. Сегодня компания Ember предлагает все необходимое для практического создания беспроводной сети устройств управления и сбора данных: приемопередатчики диапазона 2,4 ГГц, доступное программное обеспечение для реализации стека ZigBee, программно-аппаратный отладочный комплекс.

циалистов по проектированию полупроводниковых кристаллов.

Компания Ember является одной из восьми фирм-основателей альянса ZigBee. Она предлагает сегодня сертифицированную альянсом версию реализации стека ZigBee EmberZNet 1.0, но при этом подчеркивает, что предшествующая версия стека EmberNet, хоть и не соответствует спецификации ZigBee, но обладает рядом преимуществ. В анонсированной новой версии EmberZNet 2.0 функциональные возможности двух предшествующих пакетов будут совмещены.

Комплекты микросхем, на базе которых может быть реализована беспроводная технология Ember, представлены в таблице 1. Каждый узел беспроводной сети содержит приемопередатчик диапазона 2,4 ГГц и микроконтроллер, реализующий стек сетевых протоколов.

Классическим и проверенным решением стало сочетание приемопередатчика EM2420 и одного из микроконтроллеров ATmega64L или ATmega128L фирмы Atmel. Использование этих микроконтроллеров продиктовано тем, что библиотеки EmberNet и EmberZNet 1.0 были разработаны только для этих популярных кристаллов.

Новая библиотека EmberZNet 2.0 будет иметь несколько исполнений: для микроконтроллеров ATmega64/128L, микроконтроллеров MSP430 и для нового кристалла Ember EM250. Отметим, что Ember ориентируется в первую очередь на микроконтроллеры с малым энергопотреблением.

Новый кристалл EM250, появление которого на Российском рынке ожидается в начале 2006 года, представляет собой сочетание приемопередатчика диапазона 2,4 ГГц и 16-разрядного ядра XAP2, которое, с одной стороны, обладает большей производительностью и объемами памяти, чем 8-разрядные

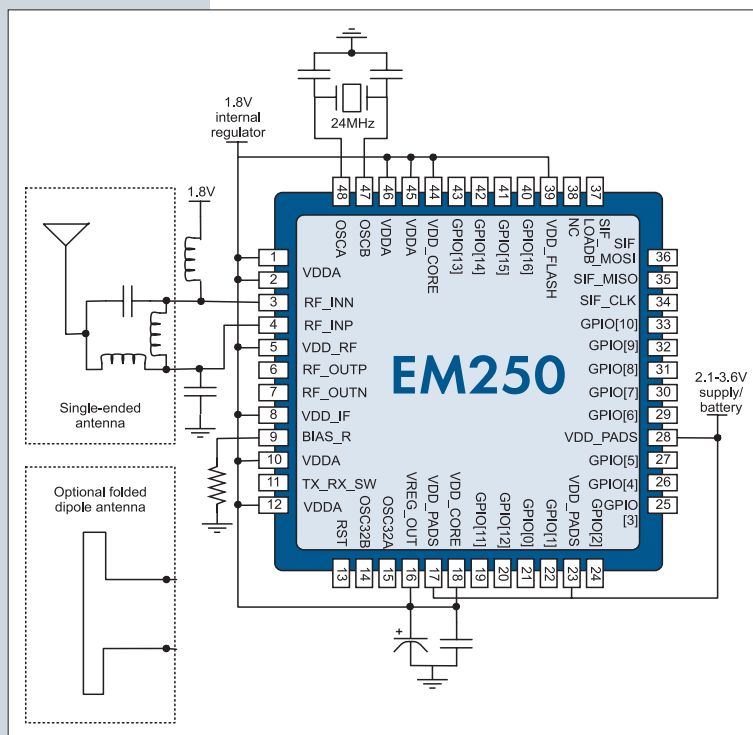


Рис. 1 Схема включения кристалла EM250

Таблица 1. Комплекты микросхем для реализации технологии Ember

N	Реализация физического уровня	Реализация стека	Реализация приложения	Поддержка пакетами ПО Ember
1	EM2420	ATmega 64/128		EmberNET3.3.3 EmberZNET1.0.2 EmberZNET2.0
2	EM2420	MSP430		EmberZNET2.0
3		EM250		EmberZNET2.0
4		EM260	любой процессор	EmberZNET2.0

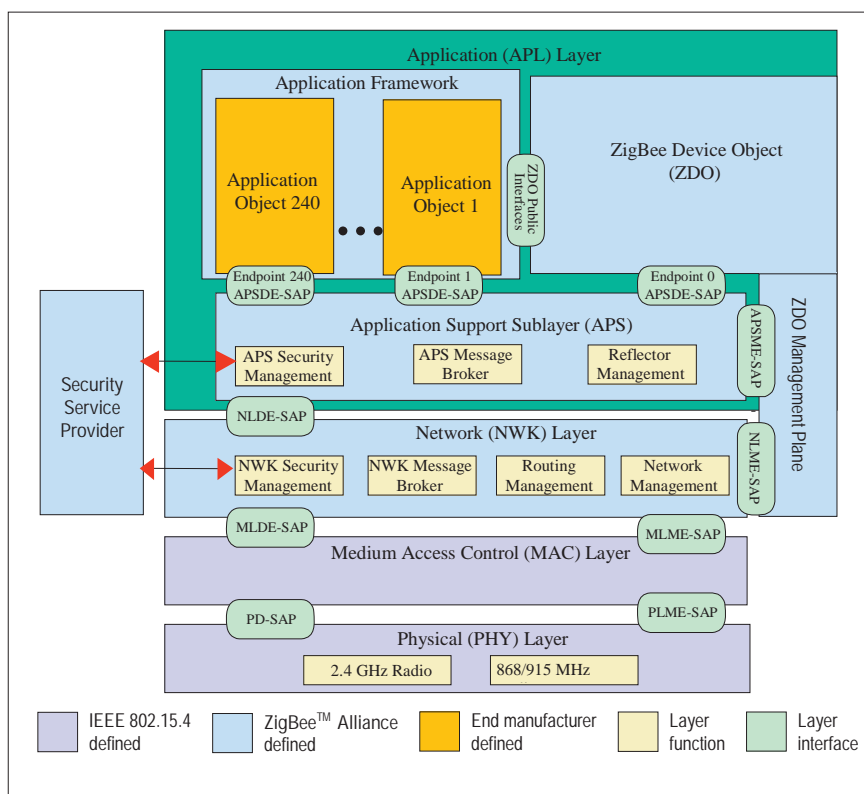


Рис. 2 Стек протоколов ZigBee

Таблица 2. Характеристики приемопередатчиков Ember

	EM2420	EM250	EM260
Чувствительность приемника, дБм	94	94	94
Программируемая выходная мощность передатчика, дБм	32...0	32...4	32...4
Энергопотребление			
прием	19,7 мА	29мА	26мА
передача	17,4 мА	30мА	27мА
спящий режим	1мкА	1мкА	1мкА
Температурный диапазон	-40°C...+85°C	-40°C...+85°C	-40°C...+85°C
Корпус	QLP48	QLP48	QLP40

микроконтроллеры, а с другой — хорошо зарекомендовало себя с точки зрения энергопотребления в мобильных устройствах Bluetooth. Таким образом, микросхема EM250 представляет собой однокристальное решение для беспроводного узла ZigBee. Пример схемы включения кристалла EM250 приведен на рис.1.

Дальнейшим шагом в развитии технологии Ember является разработка кристалла EM260, появление которого ожидается во второй половине 2006 года. Эта микросхема представляет собой сетевой процессор для устройства ZigBee. Она так же, как и EM250, содержит приемопередатчик и ядро XAP2, но ориентирована только на поддержку стека ZigBee, в то время, как для управления приложением необходимо использовать дополнительный внешний микроконтроллер, связь с которым осуществляется по SPI. Такое решение позволит разработчикам выбирать тип микроконтроллера в зависимости от требований своего приложения и снимет ряд сложностей, возникающих при совместной компиляции приложения и стека протоколов ZigBee.

Основные характеристики приемопередатчиков Ember приведены в таблице 2. Хотелось бы отметить низкое энергопотребление этих кристаллов в рабочем режиме и малое количество внешних элементов.

Спецификация ZigBee регламентирует стек протоколов взаимодействия узлов беспроводной сети (рис. 2), в котором протоколы верхних уровней используют сервисы, предоставляемые протоколами нижележащих уровней. В качестве двух нижних уровней (физического и уровня доступа к среде MAC) используется стандарт IEEE 802.15.4. Как известно, MAC-уровень в сети ZigBee реализует механизм CSMA (прослушивания несущей и устранения коллизий), сетевой уровень отвечает за маршрутизацию сообщений, а уровень APS (поддержки приложений) обеспечивает интерфейс с уровнем приложения.

Каждый узел сети имеет уникальный 64-разрядный адрес и может содержать до 240 логических получателей и источников информации, называемых конечными точками. Все сообщения в сети передаются между конечными точками.

Согласно спецификации ZigBee1.0, главным и обязательным узлом в сети является координатор, который формирует сеть, задавая номер частотного канала и идентификатор сети PAN ID. Остальные узлы (роутеры и конечные

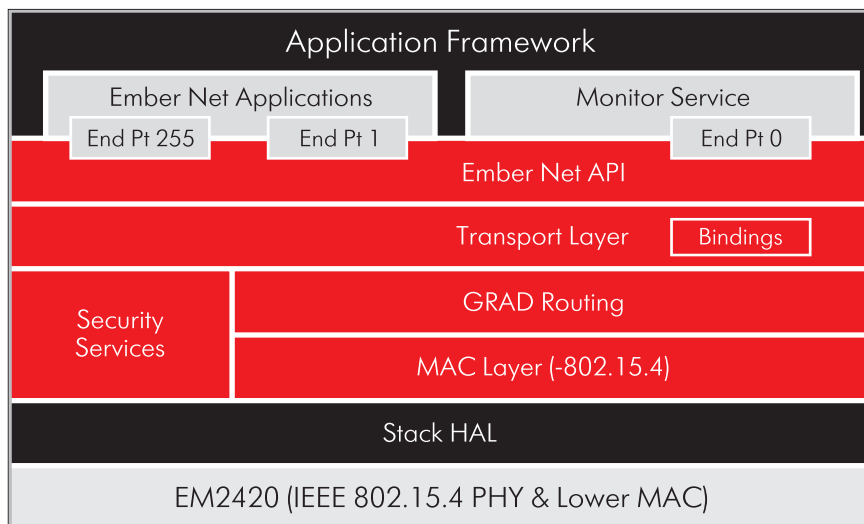


Рис. 3. Стек протоколов EmberNet

устройства) затем подключаются к координатору.

Совокупность параметров сети, определяемых на уровне приложения и обеспечивающих совместимость узлов в сети (например, количество уровней «дерева», максимальное количество присоединяемых роутеров, узлов и т. д.), называется профилем приложения.

Каждая организация может использовать частный профиль, может опубликовать свой профиль или использовать профиль, закрепленный альянсом ZigBee. В настоящее время официально стандартизован профиль HCL (Home Control Lighting), который регламентирует параметры беспроводной сети для управления осветительным оборудованием и имеет целью обеспечить совместимость конечных устройств различных производителей.

Стек протоколов EmberNet (рис. 3) был разработан до утверждения спецификации ZigBee и имеет отличия от нее, которые в первую очередь касаются топологии сети.

Стек EmberNet позволяет строить мэш-сети, в которых отсутствует координатор, все узлы равны, каждый может общаться с каждым, и каждый узел может образовывать сеть. Топологии «звезда» и «дерево» при использовании стека EmberNet рассматриваются как частные случаи мэш-сети.

Сетевой уровень стека EmberNet отличается от сетевого уровня спецификации ZigBee способом поиска маршрута. В сети с топологией «дерево» и наличием координатора маршруты для передачи сообщений жестко заданы структурой «дерева» и часто могут быть неэффективными. В мэш-сети маршрутизация является децентрализованным, распре-

деленным по сети процессом. Кроме того, сетевой уровень EmberNet реализует «градиентную» маршрутизацию, которая заключается в том, что при выборе маршрута учитывается не только количество промежуточных узлов, но и качество связи на каждом участке сети, что позволяет найти наиболее надежный путь для передачи сообщения.

Стек протоколов EmberZNet 1.0 (рис. 4) соответствует спецификации ZigBee и успешно прошел сертификацию альянса. Он может рассматриваться как реализация стека ZigBee с дополнительным транспортным уровнем Ember, который имеет связь с уровнем поддержки приложений APS через конечную точку 240. Транспортный уровень Ember предоставляет приложению дополнительный сервис, обеспечивающий, например, подтверждение доставки сообщений. При использовании транспортного уровня Ember

и стека EmberZNet1.0 возможно строить сети только с топологией «дерево».

Взаимодействие приложения и транспортного уровня во всех пакетах Ember осуществляется при помощи переменных, API-функций и callback — функций. При помощи переменных программист задает количество конечных точек для данного узла и параметры профиля сети (количество уровней «дерева», максимальное число подключаемых роутеров, максимальное количество подключаемых конечных устройств).

Основные группы API-функций пакета EmberZNet1.0 и некоторые их примеры представлены в таблице 3. В состав пакета входит подробное описание каждой функции. Видно, что программист должен позаботиться об инициализации стека и собственно беспроводной сети. Для функционирования стека в теле основного цикла должна вызываться функция EmberTick ().

Для того, чтобы из множества слышимых сообщений стек мог выделить относящуюся к данному узлу информацию, приложение перед тем, как принимать и передавать сообщения, должно установить связи (bindings), т. е. сохранить в памяти адреса своих корреспондентов. В зависимости от приложения эти связи могут быть постоянными или временными. В первом случае данные об адресатах могут записываться в энергонезависимую память, а во втором — в оперативную, что позволяет их удалять из памяти, высвобождая место для новых адресатов.

Главным узлом в сети ZigBee является координатор. Он отвечает за образование сети при помощи функции emberFormNetwork (), другие узлы затем

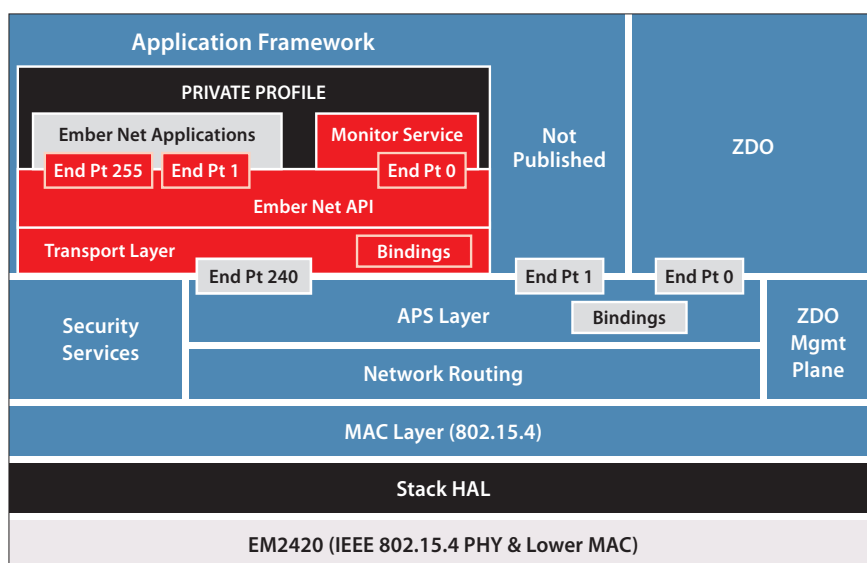


Рис. 4. Стек протоколов EmberZNet 1.0

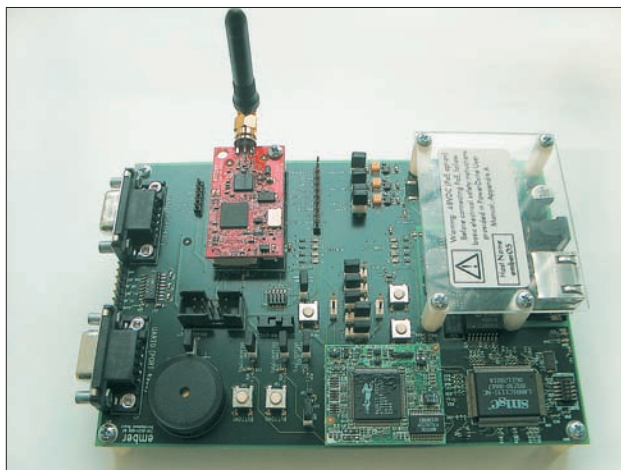


Рис. 5. Ember Developer Board

подключаются к сети при помощи функции `emberJoinNetwork()`. Координатор и роутеры могут разрешать и запрещать присоединение к сети при помощи функции `emberPermitJoining()`. Например, при разворачивании системы на объекте присоединение к сети первоначально может быть разрешено, но после того, как все узлы будут установлены и образование сети произойдет успешно, монтажник может при помощи какой-либо основной кнопки запретить дальнейшее наращивание сети.

Передача и прием сообщений осуществляются при помощи буферов, организуемых в памяти, для работы с которыми также предлагается ряд функций.

Callback-функции представляют собой процедуры, которые определяются в приложении, но вызываются стеком в определенных известных ситуациях. Например, если стек принял из сети сообщение, которое адресовано данному узлу, то он вызывает callback-функцию `emberIncomingMessageHandler()`, содержание которой заранее задал программист конечного приложения. Перечень callback — функций, которые должен

определить программист, также приведен в описании стека.

Технология программирования целевого устройства, использующего стек Ember, состоит из нескольких этапов. Предварительно в Boot-блок памяти микроконтроллера записывается предоставляемый компанией Ember загрузчик Bootloader, который затем позволяет загружать код приложения по последовательному каналу и затем модифицировать его по радиоканалу.

При помощи предоставляемого приложения «*rangetest*» разработчик целевой платы еще до разработки программного кода имеет возможность протестировать качество радиотракта своего устройства (максимальную дальность передачи при заданном уровне ошибок) и занести в энергонезависимую память параметры, необходимые для работы стека и приложения.

В состав пакетов Ember также входят примеры готовых приложений в исходных кодах для реализации узла-шлюза (*studiogateway*) между беспроводной сетью и персональным компьютером, узла-координатора, осуществляющего сбор данных с уда-

ленных датчиков, и узлов-роутеров, посылающих координатору данные со своих датчиков.

Отладочные комплекты Ember EM2420JSTK представляют собой комплекс программно-аппаратных средств, позволяющих вести одновременную разработку и отладку нескольких приложений, управляющих различными узлами сети и взаимодействующих между собой. Комплект содержит шесть отладочных плат Ember Developer Board (рис. 5).

Отладочные платы кроме возможности объединения в беспроводную сеть ZigBee имеют резервный канал Ethernet, используемый для передачи отладочной информации во время настройки сети. Существует две модификации комплекта: на основе микроконтроллеров ATmega128L и на базе кристалла MSP430.

В состав комплекта входит программный пакет EmberStudio (рис. 6), который запускается на персональном компьютере и отображает на двух разных вкладках взаимодействие узлов, объединенных в беспроводную сеть, и узлов, связанных резервным каналом Ethernet. В первом случае компьютер получает информацию о состоянии сети при помощи узла-шлюза по последовательному каналу. Во втором случае компьютер взаимодействует с узлами по резервному каналу Ethernet.

Подготовка программного кода осуществляется в среде IAR Workbench. При сборке проекта имеется возможность выполнять отладочную компиляцию (*debug*) и окончательную компиляцию (*release*). В первом случае стек передает в резервный канал отладочные сообщения-отчеты о выполненных действиях. В том числе в резервный канал дублируются все

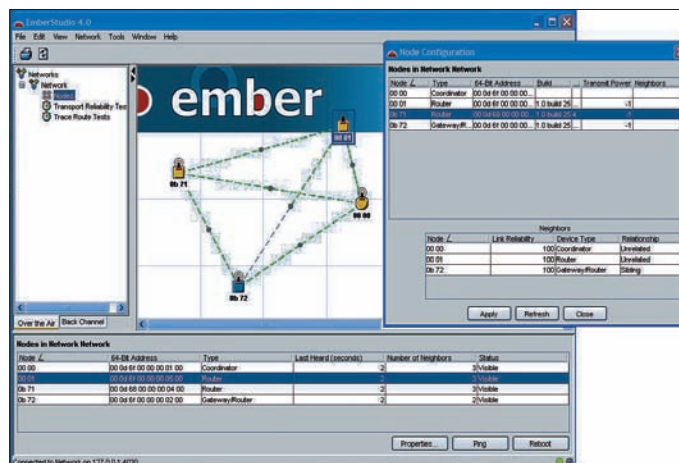


Рис. 6. Интерфейс программы EmberStudio

Таблица 3. Основные группы API-функций стека EmberZNet1.0

Поддержка работы стека	<code>emberInit()</code> , <code>emberNetworkInit()</code> , <code>emberTick()</code>
Установление связей в сети	<code>emberSetBindings()</code> , <code>emberGetBindings()</code>
Прослушивание и образование сети	<code>emberStartScan()</code> , <code>emberFormNetwork()</code>
Присоединение и удаление из сети	<code>emberJoinNetwork()</code> , <code>emberLeaveNetwork()</code> , <code>emberPermitJoining()</code>
Передача и прием сообщений	<code>emberSendDatagram()</code> , <code>emberSendMulticast()</code> , <code>emberSendSequenced()</code> , <code>emberGetLastHopLqi()</code> , <code>emberGetLastHopRssi()</code>



Рис. 7. Радиомодем ETRX1

сообщения, направляемые узлом в беспроводную сеть.

Таким образом, отладка при помощи резервного канала поддерживается на трех уровнях: на уровне стека, на аппаратном уровне в виде канала Ethernet и на уровне приложения EmberStudio.

Все это дает возможность программисту во время отладки кода при помощи программы EmberStudio наблюдать процесс обмена пакетами между узлами сети, выявлять узлы, которые работают по каким-либо причинам некорректно, и добиваться оптимальной работы всей сети.

Специалисты Ember рекомендуют выполнять отладку программного кода сначала на отладочных платах, а затем переносить его на целевые платы.

Радиомодем ETRX1, выполненный на базе технологии Ember английской компанией Telegesis (рис. 7), позволяет

значительно упростить разработку беспроводной сети, так как уже содержит запрограммированный микроконтроллер ATmega128L и позволяет осуществлять передачу и прием сообщений в сети при помощи AT-команд, передаваемых по последовательному интерфейсу.

Радиомодем имеет компактные размеры (37,75мм×20,45мм) и выпускается в двух модификациях: со встроенной антенной или с разъемом для присоединения внешней антенны.

Сохраняется возможность загрузки собственного программного кода во встроенный микроконтроллер ATmega128L. Радиомодем имеет 8 цифровых линий ввода/вывода и 2 аналоговых входа.

Предлагается также отладочный комплект ETRX1DVK, который содержит три макета, выполненных на базе модемов ETRX1 и реализующих узлы беспроводной сети ZigBee.

В заключение хочется отметить, что компания Ember не стоит на месте. В ее планах, прежде всего, дальнейшее совершенствование программного обеспечения. В новом стеке EmberZNet 2.0 анонсирована поддержка мобильных узлов и узлов с батарейным питанием, планируется усовершенствовать систему защиты информации в сети. Анонсирован комплект программного обеспечения RNAPII, который будет

запускаться на персональном компьютере и брать на себя часть функций по обслуживанию стека шлюзового узла сети. Кроме того, анонсировано программное приложение «sniffer», которое позволит использовать одну из отладочных плат в качестве анализатора сети. На смену пакету EmberStudio придет пакет EmberWorkbench, который позволит выполнять разработку программного кода и его отладку в рамках единой среды.

Вместе с тем, уже сейчас имеется все необходимое для того, чтобы можно было перейти к собственной разработке. Комплект микросхем EM2420 и ATmega128L, вместе с пакетом EmberZNet1.0, позволяют уже сегодня реализовать беспроводную сеть ZigBee. Отладочный комплект EM2420JSTK поможет это сделать в сжатые сроки. Использование радиомодемов также сокращает время разработки, но ведет, конечно, к удорожанию всей системы.

Подводя итог, хочется отметить, что основным преимуществом технологии Ember является наличие бесплатных библиотек EmberNet и EmberZNet, поддерживающих не только простые беспроводные соединения («точка-точка» и «звезда»), но и сложные самоорганизующиеся сети с ретрансляцией и маршрутизацией сообщений.