



Олег Пушкарев (КОМПЭЛ)

ПЕРЕДАЧА ДАННЫХ В ZIGBEE-СЕТИ С ПОМОЩЬЮ МОДУЛЕЙ XBEE ZNET 2.5

Применение ZigBee-модулей с предустановленным программным обеспечением существенно сокращает время разработки ZigBee-сети. В статье рассмотрены практические аспекты работы с модулями XBee ZNet 2.5 (Series 2) компании Digi (бывшая компания MaxStream) в режиме API. Разработчикам пригодится в работе пример построения беспроводного датчика на базе модуля XBee ZNet 2.5 для сбора данных о температуре.

Модули ZigBee XBee ZNet 2.5 (Series 2) (рис. 1) являются законченными узлами, способными самостоятельно подключаться к сети с MESH-топологией и передавать данные, поступающие от внешнего хост-процессора. Благодаря встроенному ZigBee-стеку все операции по формированию сети, присоединению новых устройств, прокладке оптимальных маршрутов сообщений осуществляются автоматически, без участия внешнего микроконтроллера. Для передачи данных существует ограниченный набор простых управляющих команд, которые могут выполняться даже на самом недорогом 8-разрядном внешнем микроконтроллере. Модули XBee ZNet 2.5 могут выступать и как

абсолютно самостоятельные узлы. В этом случае источником или приемником информации выступают имеющиеся на модуле периферийные узлы — порты ввода-вывода, АЦП и ЦАП. При использовании модуля без внешнего микроконтроллера прием или передача данных будет происходить под управлением команд, поступающих по эфиру от координатора или любого другого узла сети.

Работа в API-режиме

Подача команды ATND в режиме API и ответ от конечного устройства

Модули XBee могут работать в режиме управления с помощью AT-команд или в API-режиме [2,3]. Для управления с помощью вне-



Рис. 1. Модули XBee ZNet 2.5 (Series 2)



Компания Digi проводит ребрендинг MaxStream

В связи с покупкой компании MaxStream, компания Digi производит смену логотипа и вводит новые названия для продукции MaxStream. Логотип MaxStream выводится из употребления, и вся продукция теперь будет выпускаться только с логотипом Digi. Сайт www.maxstream.net теперь существует лишь в виде одной страницы, все ссылки с которой ведут в соответствующие разделы сайта www.digi.com. Компания Digi вводит новые имена для ZigBee-модулей, теперь модули XBee первой серии имеют названия XBee 802.15.4 (Series 1) и XBee-PRO 802.15.4 (Series 1). Модули второй серии именуются как XBee ZNet 2.5 (Series 2) и XBee-PRO ZNet 2.5 (Series 2). Данные изменения касаются только смены названий и никак не отражаются на технических параметрах модулей.

шнего микроконтроллера режим API гораздо более удобен, так как позволяет передавать и данные, и команды управления в общем потоке. Кроме того, некоторые возможности в режиме AT-команд просто недоступны. Например, послать по ZigBee-сети AT-команду удаленному модулю можно только в API-режиме. Работа с API-пакетами требует вычисления контрольных сумм, что не очень удобно при ручном формировании пакета в окне программы X-CTU (рис. 2а, б), однако не представляет большой сложности при управлении XBee-модулем с помощью внешнего микроконтроллера. Для практических экспериментов с API-фреймами оказалось более удобным использовать программу-терминал «Terminal v1.9b» [4], которая позволяет заранее составить набор отсылаемых API-фреймов в виде макросов и отсылать их нажатием одной кнопки. Например,

Таблица 1. API-фреймы, полученные на выходе UART модуля XBee ZNet 2.5

Ответы на API-пакет «ND»

```

7E 00 19 88 11 4E 44 00 E1 90 00 13 A2 00 40 0A 0C 22 38 00 FF FE 01 00 C1 05 10 1E 0C
7E 00 1D 88 11 4E 44 00 08 55 00 13 A2 00 40 0A 0D 4F 4E 4F 44 45 32 00 FF FE 01 00 C1 05 10 1E D2
7E 00 19 88 11 4E 44 00 9D FE 00 13 A2 00 40 0A 0B 4C 33 00 FF FE 01 00 C1 05 10 1E BE
7E 00 22 88 11 4E 44 00 25 73 00 13 A2 00 40 0A 0F 47 20 53 4C 45 45 50 5F 53 4D 41 00 25 71 02 00 C1 05 10 1E 82

```

Расшифровка структуры фрейма (для первой строки Табл. 1)

Данные Пояснения

- 7E – стартовый разделитель (признак начала API-фрейма)
- 00 19 – длина API-фрейма
- 88 – ID-номер API-пакета (ответ на AT-команду)
- 11 – номер фрейма, на который получен ответ
- 4E 44 – ASCII-имя AT-команды, на которую пришел ответ (здесь: «ND»)
- 0 – байт статуса (00 – ОК)
- E1 90 – 16-битный сетевой адрес MY
- 00 13 A2 00 – 64-битный уникальный адрес SH
- 40 0A 0C 22 – 64-битный уникальный адрес SL
- 38 00 – ASCII строковый идентификатор узла NI (здесь: «3», строка заканчивается 00)
- FF FE – адрес «родительского» узла. Действительно только для конечных устройств
- 01 – тип устройства (0-координатор, 1-роутер, 2-конечное устройство)
- 00 – статус
- C1 05 – профиль устройства (profile ID)
- 10 1E – производитель (manufacturer ID)
- 0C – контрольная сумма

Таблица 2. Примеры API-фреймов

Действие	API-фрейм
Отправка AT-команды «ND»	7E 00 04 08 11 4e 44 54
Отправка AT-команды «IS» как фрейм запроса удаленной команды	7E 00 0F 17 11 00 13 A2 00 40 0A 0F 47 25 73 02 49 53 4C
Ответ узла на запрос IS идущий от координатора	7E 00 15 97 E9 00 13 A2 00 40 0A 0D 4F 44 C5 49 53 00 01 00 00 02 03 3A 3F
API-фреймы, получаемые на выходе координатора при аппаратном сбросе	7E 00 02 8A 00 75 – фрейм статуса. Состояние «Перезагрузка» 7E 00 02 8A 06 6F – фрейм статуса. Состояние «Координатор начал работу»
На выходе координатора при подсоединении 1 роутера	7E 00 20 95 00 13 A2 00 40 0A 0B 4C 61 59 01 61 59 00 13 A2 00 40 0A 0B 4C 33 00 FF FE 01 02 C1 05 10 1E 22
На выходе координатора при подсоединении конечного устройства	7E 00 24 95 00 13 A2 00 40 0A 0D 4F 00 02 01 00 02 00 13 A2 00 40 0A 0D 4F 4E 4F 44 45 32 00 00 00 02 02 C1 05 10 1E 5F
На выходе координатора при нажатии кнопки DIO1 на конечном устройстве	7E 00 24 95 00 13 A2 00 40 0A 0D 4F 00 02 01 00 02 00 13 A2 00 40 0A 0D 4F 4E 4F 44 45 32 00 00 00 02 01 C1 05 10 1E 60
Прием данных в API-режиме. Каждый пакет содержит один ASCII-код символа цифры от 0 до 5	7E 00 0D 90 00 13 A2 00 40 0A 0E 18 76 C2 01 30 E1 7E 00 0D 90 00 13 A2 00 40 0A 0E 18 76 C2 01 31 E0 7E 00 0D 90 00 13 A2 00 40 0A 0E 18 76 C2 01 32 DF 7E 00 0D 90 00 13 A2 00 40 0A 0E 18 76 C2 01 33 DE 7E 00 0D 90 00 13 A2 00 40 0A 0E 18 76 C2 01 34 DD 7E 00 0D 90 00 13 A2 00 40 0A 0E 18 76 C2 01 35 DC

чтобы обнаружить все узлы сети, нужно составить фрейм AT-команды «ND» и отослать его с любого модуля сети. Ниже приведены API-фреймы, полученные на выходе UART модуля XBee ZNet 2.5, с которого был отправлен запрос на обнаружение всех узлов сети. Всего было получено 4 ответа. Как видно из приведенных ответов, в сети было обнаружено 2 роутера и 2 конечных узла.

Ниже, в таблице 2, приведены дополнительные приме-

ры API-фреймов, полученные от различных узлов в ZigBee-сети в различных реальных ситуациях. Различные блоки фрейма выделены цветом и типом шрифта. Приведенные примеры могут быть полезными при изучении раздела, посвященному разделу по работе с API-фреймами в оригинальном описании производителя [5] (русскоязычный перевод данного документа доступен по запросу на wireless@compel.ru).

Практическая часть — делаем беспроводной датчик температуры

Мы воспользуемся возможностью отсылки API-команды для удаленного измерения температуры с помощью ZigBee-модуля. Схема измерительного модуля очень проста (рис. 3). Измерение температуры производится с помощью аналогового датчика LM19, преобразующего температуру в диапазоне от -55 до 130°C в выходное напряжение, измеряемое с помощью АЦП модуля XBee

жащие информацию о температуре удаленного узла. Для простейшего эксперимента лучше ограничиться сетью из 2 узлов – только координатора и нашего температурного датчика (работающего постоянно, т.е. не в спящем режиме). Удаленный модуль должен включаться после того, как координатор успешно стартовал (красный светодиод мигает один раз в секунду). При включении наш модуль с температурным датчиком самостоятельно подключится к координатору, на UART-выходе которого мы получим API-фрейм идентификации подключенного узла (рис. 5). Данный фрейм мы также будем получать на выходе координатора при нажатии кнопки «Идентификация» на нашем температурном датчике.

Для отправки запроса на удаленное считывание значения АЦП (температуры), нам необходимо знать 64-разрядный уникальный и 16-разрядный сетевой адрес нашего удаленного узла. Эти значения можно взять из полученного сообщения о присоединении (00 13 A2 00 40 0A 0F 47 и 22 62 соответственно). Теперь мы можем сформировать API-пакет на удаленное считывание АЦП.

Контрольную сумму для отправляемых пакетов в данном случае мы рассчитываем вручную. После отправки пакета мы получим ответный API-фрейм, содержащий значение напряжения АЦП, которое можно пересчитать в температуру (рис. 6). Для преобразования аналого-цифровых данных в милливольты, используйте следующий алгоритм:

$$AD \text{ (мВ)} = (\text{значение ADIO} / 0x3FF) * 1200 \text{ мВ [5]}$$

Из полученного пакета мы видим, что значение АЦП = 02 8F (предпоследние 2 байта в API-фрейме), что в абсолютном исчислении составляет 0,768 В. С учетом делителя, напряжение на выходе LM19 было $0,891 \times 2 = 1,536 \text{ В}$, что соответствует температуре $24,5^\circ\text{C}$. Формулы для пересчета выходного напряжения в значения температуры можно найти в документации на LM19.

По мере освоения API-команд можно добавлять в сеть новые узлы и тестировать передачу данных с ретрансляцией пакетов. Для минимизации энергопотребления модуль можно перевести в спящий режим (SM=4) с периодическим пробуждением по встроенному таймеру, например каждые 3 секунды

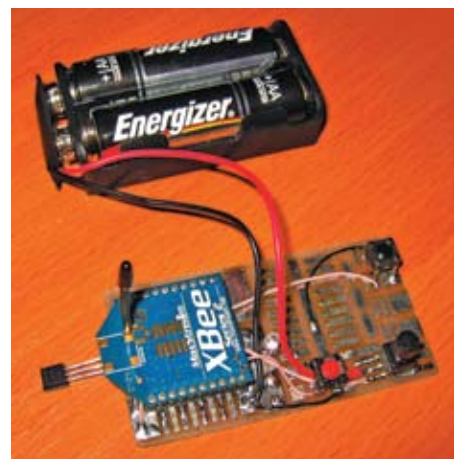


Рис. 4. Температурный датчик на базе модуля XBee

(SP=0x12C, ST=0x1F4). В этом случае модуль будет периодически просыпаться, отсылая запрос «родительскому» узлу на проверку наличия информации для себя. При этом рекомендуется задать с некоторым запасом время хранения сообщений на родительском узле, например SP=0x2BC (7 секунд).

Пропускная способность и задержки

Разработка конечного устройства, которое будет использовать MESH-сеть для передачи данных, требует повышенного внимания

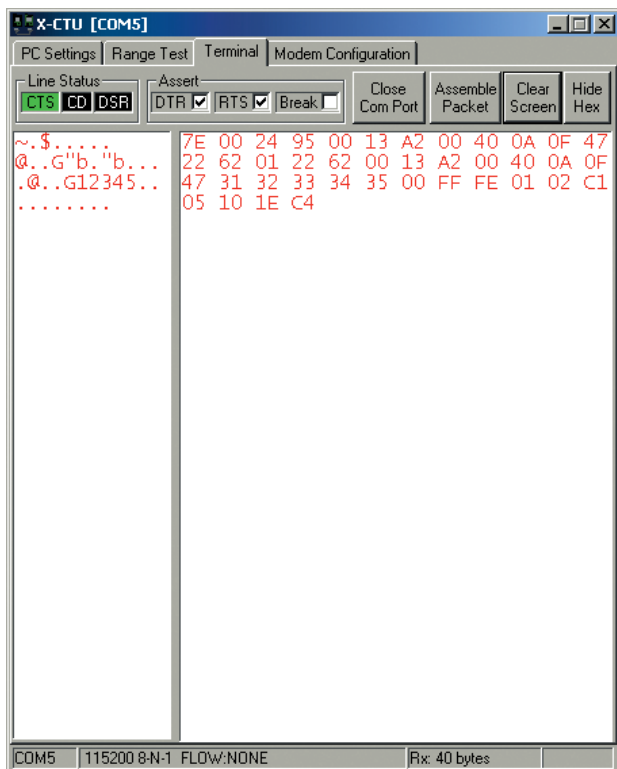


Рис. 5. API-фрейм с параметрами присоединенного узла сети

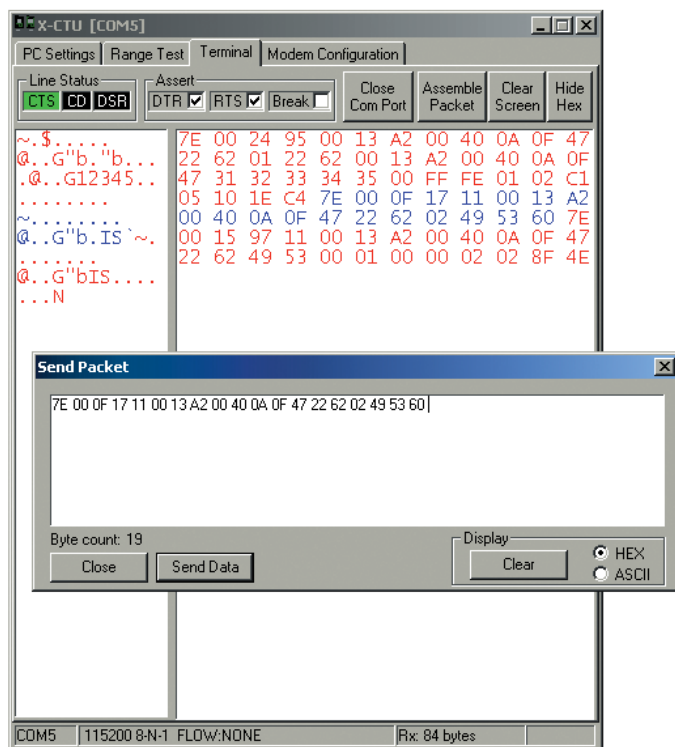


Рис. 6. Ответный API-фрейм

разработчика на этапе определения параметров будущей сети. Например, одно из частых заблуждений относительно свойств ZigBee-сети – это завышенные представления о скорости передачи данных между узлами сети. Несмотря на относительно высокую фиксированную скорость передачи данных в радиоканале, равную 250 кБит/сек, реальная скорость может быть меньше на порядок! Для объяснения этого факта необходимо понимать логику сетевого взаимодействия узлов сети и не забывать про задержки на подтверждение пакетов. Кроме того, обработка данных на нижних уровнях стека также занимает определенное время. Производители с неохотой отвечают (если вообще отвечают) на вопросы о реальной скорости передачи данных в ZigBee-сети и возникающие задержки (latency). Обычный ответ сводится к туманной фразе что «эти параметры зависят от топологии сети и могут существенно варьироваться в зависимости от текущего уровня помех и напряженности трафика». Как бы нам нравился или не нравился подобный ответ, он отражает реальное положение дел – можно только грубо оценивать реальные параметры скорости передачи данных и возникающие задержки.

Исходя из практического опыта работы с модулями Digi (MaxStream) (рис. 1), можно говорить о средней скорости передачи данных порядка 4-5 кБит/сек на уровне 802.15.4 (связь точка-точка, модули XBee 802.15.4 (Series 1), передача файла ≈ 100 кБит, скорость UART 115200).

При работе в MESH-сети реальная скорость будет ниже, однако, с моей точки зрения, не это является критичным параметром, т.к. ZigBee-сети не предназначены для передачи больших объемов данных. В системах сбора информации с беспроводных датчиков объем полезных данных составляет десятки-сотни байт, что не предъявляет высоких требований к средней скорости передачи данных. Однако кроме скорости передачи, в сетях с MESH-топо-

логией при передаче данные будут передаваться к узлу-получателю с переменной задержкой, которую необходимо учитывать в алгоритме поведения центрального узла сбора информации. Прежде всего, необходимо правильно выбирать величину времени ожидания ответа от удаленного узла, если пакет данных ретранслируется несколькими маршрутизаторами (роутерами). Грубо оценим максимальную задержку при следующих начальных условиях:

1. Допустим, что расстояние от координатора-отправителя до удаленного узла – 8 хопов (ретрансляций).

2. Время передачи пакета между двумя узлами (одна попытка) ≈ 30 мсек (точное значение зависит от конкретной программной реализации ZigBee-стека, MAC-уровня и микросхемы трансивера).

3. Число ретрансляций на уровне 802.15.4 – 3 (типовая фиксированная величина).

Таким образом, максимальное время передачи сообщения от одного узла к другому составляет ≈ 90 мс (30 мс $\times 3$ попытки), столько же потребуются для передачи подтверждения (уровня приложения). В маршруте с 8 ретрансляциями максимальная задержка составит $8 \times 90 \times 2 = 1,44$ секунд на одну попытку отправки сообщения. Разумеется, если мы отправили данные и не получили ответ, то можно попытаться повторить отправку, скажем 3 раза. Тогда наше приложение будет делать вывод о недоступности узла получателя (нет связи) только через примерно 5 секунд. Сама по себе эта цифра не большая и не малая – все зависит от того приложения, которое использует MESH-сеть для передачи данных. Для системы АСКУЭ, где пакеты данных передаются один раз в час, задержка в 5 секунд не является критической. В то же время, такая задержка будет совершенно неприемлемой, если нужно синхронно управлять комплексом светофоров на крупной транспортной развязке. Именно задержки при ретрансляции пакета являются главным ограничителем при построении «вытянутых» ZigBee-

сетей. При количестве ретрансляций равном 100 максимальное время ожидания реакции удаленного узла будет достигать одной минуты! По-видимому, именно из этих соображений даже в самой последней спецификации ZigBee-Pro максимальная глубина ZigBee-сети ограничена 30 ретрансляциями. В модулях XBee ZNet 2.5 максимальное время ожидания ответов для команд ND (поиск узлов) и DN (прокладка маршрута) не может превышать 25 секунд (параметр NT). При использовании модулей XBee ZNet 2.5 глубина сети (число хопов) формально не ограничена, однако число ретрансляций при рассылке широковещательных сообщений не может быть более 15. Это означает, что конкретный узел просто не сможет обнаружить другие узлы, расположенные на расстоянии более 15 хопов от него.

Заключение

Практическое построение беспроводной сети на базе модулей XBee ZNet 2.5 не требует глубоких знаний ZigBee-спецификации и может быть выполнено в сжатые сроки разработчиком средней квалификации. Однако при разработке ZigBee-сетей с MESH-топологией необходимо понимать процедуры сетевого взаимодействия и учитывать реальную скорость передачи данных и возникающие задержки.

Список литературы

- [1] Новости Электроники, №2, 2006 стр. 15-20
- [2] Новости Электроники, №2, 2007 стр. 25-27
- [3] Новости Электроники, №16, 2007 стр. 12-16
- [4]<http://www.mikrocontroller.net/attachment/24594/term20041226.zip>
- [5] Product Manual v1.x.2x – ZigBee Protocol http://www.compel.ru/images/catalog/868/product-manual_XBee_Series2_OEM_RF-Modules_ZigBee.pdf.

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