



Примеры приложений

**SerialStar:
«Обмен данными со спящими узлами»**



Board Revision
Product Name
Doc Name
app_SerialStar_Communication_With_Sleeping_Devices
Revision Date 05.06.2017
Revision Number 1

1. СОДЕРЖАНИЕ

1. Содержание	2
2. Обмен данными со спящими узлами в SerialStar	3
2.1. Цель	3
2.2. Что потребуется	3
2.3. Необходимые пояснения.....	3
2.4. Практическая часть.....	4
2.5. А если не все понятно или хочется чего-то большего?	6
3. История документа.....	7
Техническая поддержка	7

2. ОБМЕН ДАННЫМИ СО СПЯЩИМИ УЗЛАМИ В SERIALSTAR

2.1. Цель

Научиться организовывать обмен данными с модемами, находящимися в спящем режиме.

2.2. Что требуется

- Модуль MBee-868-2.0 или MBee-868-3.0 с ПО SerialStar – 2 шт.
- Антенны 868 МГц с разъемом SMA - 2 шт.
- Плата MB-USBridge – 2 шт.
- USB-провод – 2 шт.
- Компьютер с терминальной программой.

2.3. Необходимые пояснения

Программным обеспечением SerialStar поддерживается несколько энергосберегающих режимов. Общим для них является то, что в таких режимах практически вся периферия, находящаяся на борту модуля, выключается. Радиоядро, как наиболее требовательная с точки зрения потребляемой энергии часть, выключается в первую очередь. Это означает, что узел, находящийся в спящем режиме не способен ни принимать данные по радио, ни передавать. Тем не менее, возможность организовать двухстороннее взаимодействие с таким узлом в SerialStar есть. Каким же образом оно осуществляется?

На самом деле ничего сложного. Просто сообщения, которые предназначены для спящих устройств, не надо сразу отправлять в эфир (ведь принять их все равно никто не в состоянии – приемник-то выключен). Вместо этого сообщение помещается в специальный буфер на передающем модуле, в котором оно и хранится до наступления определенного события. Какого? Правильно, до момента приема пакета от того узла, кому это сообщение предназначено. Это может быть пакет любого типа. Главное, чтобы адрес его отправителя совпал с адресом получателя одного из пакетов, хранящихся в буфере на отправку. Как только программа обнаружит такое совпадение, она немедленно отправит находящееся в буфере сообщение адресату. А как же, спросите Вы, сообщение дойдет до него, если спящий узел после передачи своих данных сразу засыпает? Да в том-то все и дело, что не сразу. А только после того, как пройдет время, определяемое AT-командой ST. Этот интервал начинает отсчитываться сразу после выхода модуля из сна. Если во время бодрствования произойдет какое-либо событие, типа приема пакета по радио, нажатия на кнопку «SYSTEM» или приема данных по UART, этот интервал начинает отсчитываться по-новой. Впрочем, за подробностями лучше обратиться к основному документу по проекту «sw_SerialStar.pdf».

Для нас же самым главным является то, что во время этого интервала приемник модуля включен. И вот именно пока этот интервал еще не закончился, тот модуль, у которого имелось кое-что в буфере для спящего узла, и должен отправить пакет. Если адресат пакет принял, то интервал начинает отсчитываться снова. Это дает возможность организации передачи спящему узлу нескольких сообщений подряд. Если, конечно, хост узла отправителя достаточно быстр и если параметр ST не установлен слишком малым.

Передача сообщений спящим узлам возможна только если модуль отправитель находится в одном из пакетных режимов (AP=1 или AP=3). Кроме этого, буферизация поддерживается только для трех

типов фреймов – 0x01, 0x10 и 0x17. Откуда такие ограничения? Дело в том, что хост-система перед отправкой сообщения должна знать, что узел, которому оно предназначается, является спящим узлом. И она должна иметь возможность указать модулю, что данный пакет не предназначен для немедленной отправки в эфир, а должен быть помещен в буфер. Для этого используется бит №6 байта Options, имеющийся только в вышеперечисленных 3 типах фреймов. Если в принятом модулем фрейме этот бит равен 1, то фрейм помещается в буфер. Этот бит никак не влияет на прочие управляющие биты байта Options и они на него, в свою очередь, тоже. Если же модуль находится в прозрачном режиме UART или используется фрейм 0x0F, не имеющий поля Options, то такое сообщение немедленно отправляется в эфир.

А теперь несколько слов по организации буфера сообщений. Прежде всего о его длине. Она равна десяти, то есть он способен держать сообщения для 10-и разных узлов с длиной полезной нагрузки не более 40 байт для каждого сообщения. Для одного и того же адресата может храниться только одно сообщение. Это значит, что если в буфере имеется неотправленное сообщение для какого-либо узла, то при приеме от хоста сообщения, предназначенного этому же адресату, сообщение в буфере будет переписано новым. Как только пакет из буфера ушел в эфир, он помечается как отправленный. Таким образом блокируется его повторная передача при приеме по радио очередного сообщения от того же спящего узла. А что происходит, если при приеме от хоста пакета, адресованного спящему узлу, в буфере не оказывается сообщения с таким же адресом? Тогда программа начинает искать первое же сообщение, помеченное как отправленное, и размещает поступивший пакет на его месте. Прекрасно, а если просмотрен весь буфер, и все сообщения в нем еще не отправлены, и адрес получателя не совпал? Тогда программа принимает волевое решение и начинает затирать сообщения. Начинает она с первой строки. Если затем модуль получит от хоста еще одно сообщение, которое не удалось разместить в буфере в соответствии с первыми двумя правилами, то замещается сообщение во второй строке. И так далее по кругу. Хост-система должна сама контролировать содержимое буфера и отправку сообщений из него. Команд для запроса данных о его состоянии в SerialStar не существует. Единственным способом очистить буфер от зависших сообщений является аппаратный или программный рестарт модуля.

И напоследок немного чисел. По результатам объективных измерений, время, которое требуется модулю на просмотр буфера и отправку буферизированного сообщения с максимальной полезной нагрузкой, составляет что-то около 16 мс. Именно поэтому значение параметра ST, установленное «по умолчанию», равно 20 мс. Если же Ваши сообщения, которые требуется передать спящим узлам короче, то вы можете уменьшить время ST. Таким образом, Вы получите некоторую экономию энергии, поскольку спящий модуль будет меньше ждать ответа и, соответственно, быстрее засыпать. Однако, смотрите не перестарайтесь. (?) Слишком затягивать этот параметр тоже не следует во избежание возможных потерь пакетов вследствие некоторого дрожания времени отправки буферизированных пакетов.

2.4. Практическая часть

1. В обоих модулях настройки должны быть приведены к значениям «по умолчанию». Если Вы не знаете или забыли, как это сделать, то посмотрите документ «app_SerilaStar_BootLoader.pdf».
2. Устанавливаем оба модуля на плату MB-USBridge, в соответствии с Рисунком 1.

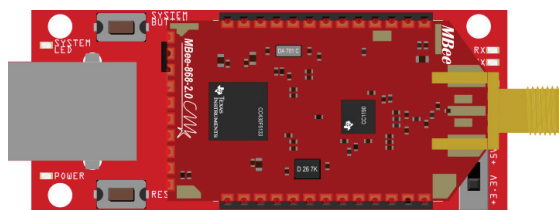


Рисунок 1

3. Если Вы экспериментируете на столе, то антенну достаточно установить на один модуль. Если требуются более значительные расстояния, то антенны надо иметь на обоих модулях.
4. Запускаем пару копий терминальной программы, выбираем соответствующие номера портов и устанавливаем для них 9600 8N1 CTS/RTS. Для наглядности неплохо было бы включить режим локального эха.
5. Переводим модули в командный режим («+++» или «SYSTEM BUTTON»).
1. Один из модулей у нас будет спящим. Для начала присвоим ему собственный идентификатор. Пусть он будет равен 0x0002. Печатаем «AT MY 2<CR>». Если Вам не понятен смысл используемых здесь и далее команд, то посмотрите про них в «sw_SerialStar.pdf» или в «app_SerialStar_IO_Control.pdf»
2. Теперь надо что-то придумать с индикацией. Проще всего удостовериться в получении пакетов спящим узлом с помощью светодиода, установленного на плате MB-USBridge и подключенного к одному из выходов модуля. В качестве такового будем использовать выход R0. С ним на плате соединен белый светодиод, который «по умолчанию» загорается при переходе модуля в командный режим.
3. Настраиваем линию R0 для работы в качестве цифрового выхода с низким уровнем после рестарта. Для этого печатаем «AT R0 4<CR>».
4. Включаем спящий режим на модуле с помощью «AT SM4<CR>». Чтобы бы было не очень утомительно ждать от него пакеты, уменьшаем период выхода в эфир с 15 секунд (значение «по умолчанию») до 3 секунд. С этой целью формируем команду «AT SP 300<CR>».
5. Сохраняем и применяем изменения «AT CN<CR>». Белый светодиод должен погаснуть.
6. Теперь внимательно смотрим на плату MB-USBridge, на которой установлен только что настроенный модуль. Если все сделано правильно, то с периодом 3 секунды должны включаться светодиоды RF TX, RF RX. Все, спящий узел мы настроили.
7. Переходим к настройке узла-отправителя. Как мы помним, он должен находиться в пакетном режиме. Включаем этот режим «AT AP3<CR>» и применяем изменения «AT CN<CR>». Больше ничего настраивать не будем. Собственный идентификатор 0x0001, установленный «по умолчанию» нас вполне устраивает.
8. Проверяем наличие связи по радио между модулями. Смотрим в окно терминала, в котором настраивали только что модуль-отправитель. Там с периодом в 3 секунды должны вываливаться пакеты от спящего узла, в которых находятся сведения о текущем состоянии его линий ввода/вывода. Разбирать их содержимое сейчас не будем – информация об этом имеется в других документах. Главное, чтобы они вообще приходили. Если Вы их не наблюдаете, значит, скорее всего дело в отсутствии антенны на одном из модулей, либо они расположены слишком близко друг к другу. Поэкспериментируйте и все должно получиться.
9. Ну вот, теперь все готово для того, чтобы зажечь светодиод по радио на спящем узле. Формируем следующую последовательность 16-тиричных чисел: 7E 00 08 17 01 00 02 44 52 30 05 1A. Пока примите эту последовательность просто на веру. Не будем пояснять, что в ней к чему – для этого есть соответствующие документы. Да, чуть не забыли сказать, что при работе в пакетном режиме не допускаются разрывы между символами более, чем удвоенная длительность одного символа. Так что ручной посимвольный ввод этой последовательности

не подойдет. Поищите, пожалуйста, терминал, который обеспечивает возможность отправки заранее подготовленного пакета. Белый светодиод должен загореться максимум через 3 секунды (ну то есть через период отправки спящего узла). Если этого не случилось, значит либо пакет составлен неправильно, либо он не был отправлен в эфир. Проверьте, пожалуйста, его еще раз и повторите попытку. Мы несколько раз прошли по пунктам практическую часть, чтобы убедиться, что порядок действий и последовательность символов корректны.

10. А сейчас мы попробует светодиод выключить. Подготавливаем и отправляем: 7E 00 08 17 01 00 02 44 52 30 04 1B. Скоро белый должен погаснуть. Если так и произошло, значит, все у нас получилось. Считается, что цель этого документа достигнута.

2.5. А если не все понятно или хочется чего-то большего?

Тогда пишите на форум или в почту. Вам обязательно помогут.

3. ИСТОРИЯ ДОКУМЕНТА

Дата	Редакция документа	Описание изменений
05.06.2017	Первая версия	

Техническая поддержка

Разработка и техническая поддержка	
СИСТЕМЫ, МОДУЛИ И КОМПОНЕНТЫ	
Разработчик систем автоматизации и телеметрии	
Телефон	+7 (495) 784 5766
Электронная почта	mbee@sysmc.ru
Сайт	www.sysmc.ru
