



Примеры приложений

**SerialStar:
«Настраиваем сеть»**



Board Revision

Product Name

Doc Name app_SerialStar_Network_Control

Revision Date 31.05.2017

Revision Number 1

1. СОДЕРЖАНИЕ

1.	Содержание	2
2.	Настройка сети в проекте SerialStar	3
2.1.	Цель	3
2.2.	Что потребуется	3
2.3.	Краткий теоретический курс.....	3
2.4.	Практическая часть.....	5
2.5.	Продолжение краткого теоретического курса	6
2.6.	Продолжение практической части	7
2.7.	Окончание краткого теоретического курса.....	7
2.8.	Окончание практического курса	7
2.9.	Ответы на возможные вопросы и примеры	8
2.10.	А если вопросы все-таки остались?	8
3.	История документа.....	9
	Техническая поддержка	9

2. НАСТРОЙКА СЕТИ В ПРОЕКТЕ SERIALSTAR

2.1. Цель

Разобраться в сетевых настройках ПО SerialStar и научиться их применять на практике.

2.2. Что потребуется

Внимательно прочитать этот документ и еще кое-чего по списку:

- Модуль MBee-868-2.0 или MBee-868-3.0 – 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. Краткий теоретический курс

Для начала разберемся с тем, что мы называем сетевыми настройками. Это такие параметры, благодаря которым модуль «знает» о том, в какой именно сети ему суждено работать, а также что делать с данными, которые он принял по радио. И еще ему надо обязательно понимать, куда он должен передать информацию, которую он тем или иным способом получил сам.

Итак, начнем с принадлежности к сети. Принадлежность модулей к единой сети определяется их сетевым адресом. У всех модулей, чтобы они друг друга «видели» он должен быть одинаковым. Полный адрес состоит из 4-х байт. Старший называется идентификатором сети. Он попадает в обработку первым после приема пакета из эфира. На этом этапе задействуется радиоядро микросхемы модуля. Что такое радиоядро? По сути это независимый процессор, который связан с основным вычислительным ядром особым внутренним интерфейсом. Радиоядро работает в своем режиме и занимается низкоуровневой обработкой пакетов. Так вот, именно оно ответственно за первоначальную их фильтрацию. Только фильтрует оно их не по полному адресу, а исключительно по старшему. Тем самым освобождается ядро вычислительное, которое может не отвлекаться на каждый «мусорный» пакет.

После того, как старший байт проверен, радиоядро дает знать ядру вычислительному, что приняты данные, которые могут оказаться бесполезными, и переправляет ему принятый пакет по внутреннему интерфейсу. Вычислительное ядро, в свою очередь, проверяет оставшиеся 3 байта адреса. Два байта, располагающиеся в середине называются идентификаторами системы. С помощью них можно организовывать логические сегменты в общей зоне покрытия, то есть обеспечить изоляцию нескольких систем различного назначения. Фильтрация по младшим байтам осуществляется самыми низкими уровнями сетевого протокола, применяющегося в SerialStar.

И, наконец, младший байт адреса. Он у всех узлов SerialStar одинаковый и изменить его невозможно. Так сделано потому, что на этом этапе развития проекта, сеть является одноранговой т.е. все узлы одинаковы с точки зрения выполняемых сетевых функций. А единый младший байт адрес, как раз и предназначен для того, чтобы они знали, что с ними общается точно такой же собрат. В дальнейшем возможно появление узлов с другими задачами, в которых этот младший байт будет уже отличаться от нынешнего значения, равного 0x01.

Однако, все эти ухищрения с сетевыми адресами не должны вводить Вас в заблуждение относительно возможности полной изоляции сетей. Необходимо всегда помнить, что все узлы, если они работают на одной и той же центральной частоте, пользуются единой средой передачи, ну то есть эфиром. Это приводит к неизбежным взаимным помехам. Нельзя, например, надеяться на хороший результат, если в одной зоне покрытия работают две системы, одна из которых предназначена для передачи потока данных в режиме беспроводного удлинителя последовательного интерфейса, а вторая собирает показания от счетчиков с периодом в 1 час. Даже, если Вы назначили этим системам разные идентификаторы, хорошо работать будет только первая система.

Теперь переходим к идентификации узла в рамках одной сети. Для того, чтобы отличить одно устройство от другого существует еще один параметр – идентификатор узла. В принципе, его можно рассматривать как расширение сетевого адреса. Идентификатор узла состоит из 2 байт. Значение, которое Вы можете установить, находится в диапазоне 0x0001 ÷ 0xFFFFE. Адрес 0xFFFF, как Вы уже догадались, используется в качестве широковещательного. Фильтрация по идентификатору узла осуществляется только после того, как пакет прошел все предыдущие уровни. За нее ответственен самый высокий уровень стека протоколов – уровень приложения.

И, наконец, последнее важное замечание. Программное обеспечение SerialStar не осуществляет никакого контроля за правильностью назначения Вами сетевых параметров. Не применяются также какие-либо алгоритмы автоматического распределения адресов и разрешения конфликтов. Вам предоставляются самые широкие возможности по настройке именно Вашей уникальной сети под Вашу конкретную задачу. Однако и ответственность за то, что что-то идет не так, тоже целиком Ваша. Но, мы надеемся, что возможные трудности будут преодолены, и вскоре Вы оцените преимущества такого подхода.

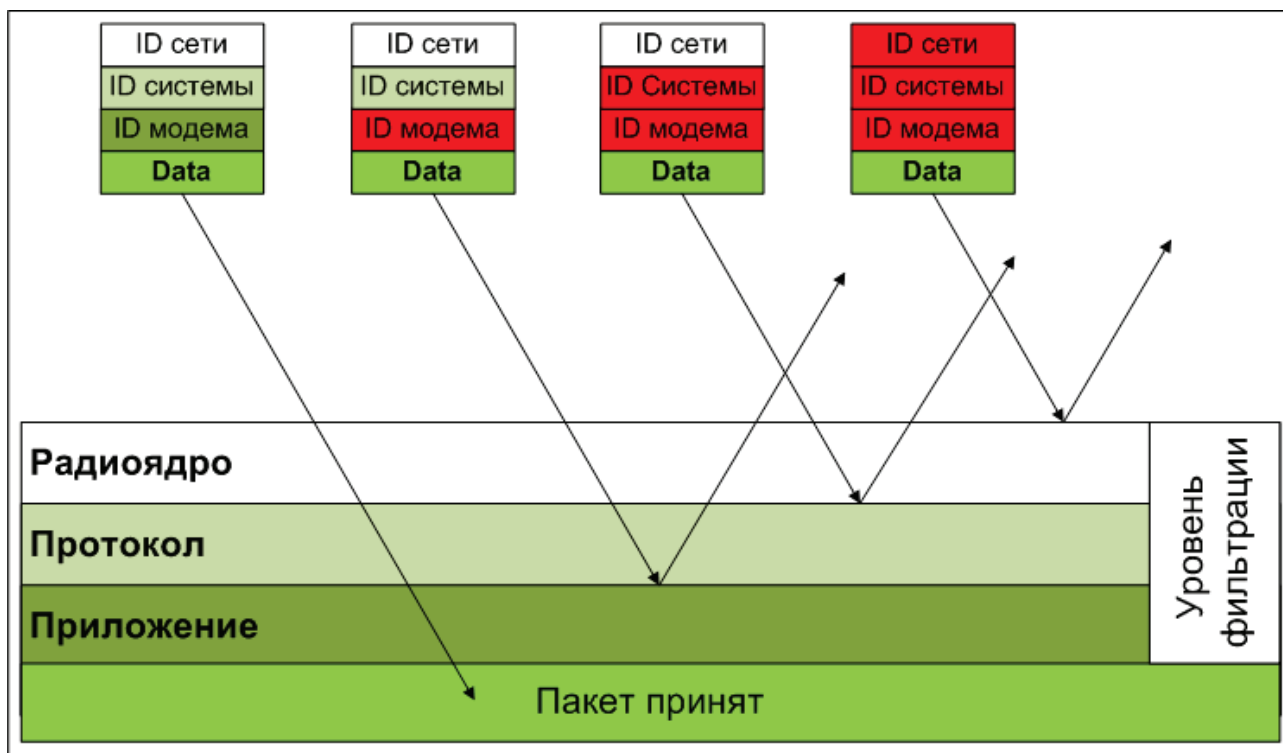


Рисунок 1

Ответим теперь на вопрос, а какие есть инструменты для настройки сетевого адреса и идентификатора узла. В SerialStar для этого имеется целых 3 варианта. Первый способ – использование программы SysmcBootLoader, второй – применение AT-команд, ну и, наконец, третий – это управление с помощью API-фреймов. Какой из них предпочесть? Все зависит от Ваших целей и задач. Настройка с помощью SysmcBootLoader проста и очень наглядна. Можно все сделать с помощью мыши. Для управления адресами с помощью AT-команд вам потребуется только простейшая терминальная программа. А вот API-фрейм незаменим там, где нужно быстро и без ошибок запрограммировать более или менее крупную серию. В ряде случаев без специальной программы собственной разработки, Вам уже не обойтись. Это ПО, например, сможет автоматически обеспечивать уникальность каждого узла с занесением их идентификаторов в базу данных и одновременным печатанием стикеров для наклейки на готовое изделие.

Кстати, Вы всегда можете пользоваться основным документом по проекту, который находится на нашем сайте «sw_SerialStar.pdf». Там изложено все тоже самое, только в сухом, академическом стиле.

2.4. Практическая часть

1. Предполагаем, что ПО SerialStar в модуль уже загружено. Если нет, то посмотрите, как это сделать в документе «app_SerialStar_BootLoader.pdf». Не забываем сбросить все настройки к заводским значениям. Там рассказано, как это сделать.
2. Устанавливаем модули на платы MB-USBridge, в соответствии с Рисунком 2.

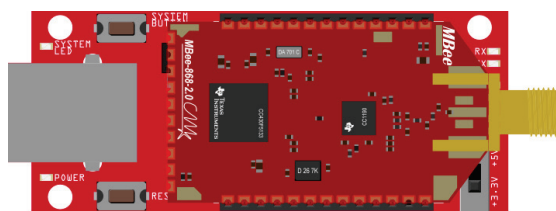


Рисунок 2

3. Если Вы проводите эксперимент на столе, то антенну достаточно иметь на одном модуле. Если же планируете исследовать систему с помощью двух компьютеров, которые находятся на некотором расстоянии, 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. Очень желательно включить режим локального эха.
5. Если все оборудование исправно и соединено правильно, то все, что Вы будете печатать в одном терминальном окне, должно появляться в другом и наоборот. Почему же это происходит? Ведь мы с Вами, кажется, ничего не настраивали, никакими адресами не управляли и сетевого планирования не осуществляли. Все дело в параметрах «по умолчанию», которые используются в SerialStar. Давайте посмотрим, что же там установлено.
6. Входим в командный режим («+++» или «SYSTEM BUTTON»).
7. Вводим команду «AT ID 0x<CR>» (<CR> здесь и далее означает нажатие клавиши «ENTER», а 0x нужен для того, чтобы ответ приходил в 16-тиричном виде – так удобнее). Получаем на экране три старших байта сетевого адреса 0x10001. Младший байт всегда же ведь один и тот же, поэтому, зачем его выводить, если его изменить все равно нельзя. Убеждаемся, что и на втором модуле сетевой адрес тоже 0x10001. Так, уже что-то понятно. Эти модули участники одной и той же сети, раз у них совпадают все три старших байта сетевого адреса. Первые уровни фильтрации преодолены.

8. А что у нас с идентификатором узла? Печатаем «AT MY 0x<CR>» по очереди в каждом окне терминала и получаем 0x01. Т.е. собственные идентификаторы тоже совпадают. Ничего удивительного, ведь модули пока еще близнецы-братья – мы же еще ничего не меняли. А программа SerialStar, как мы уже писали выше, никакого контроля распределения адресов не осуществляет. Как же тогда осуществляется прием/передача? Почему модули прекрасно друг друга видят? Об этом в следующем пункте.
9. Печатаем «AT TX 0x<CR>» и убеждаемся, что на экране 0xFFFF. Это адрес узла, которому будут передаваться данные, поступающие на UART, если модуль работает в прозрачном режиме (будет лучше, если о режимах Вы посмотрите в «sw_SerialStar.pdf»). А как Вы помните, адрес 0xFFFF является широкоэвещательным, т.е. пакеты, с таким адресом получателя будут принимать все узлы SerialStar, работающие в зоне покрытия и имеющие одну и ту же частоту несущей. Надеемся, что теперь все стало понятно.

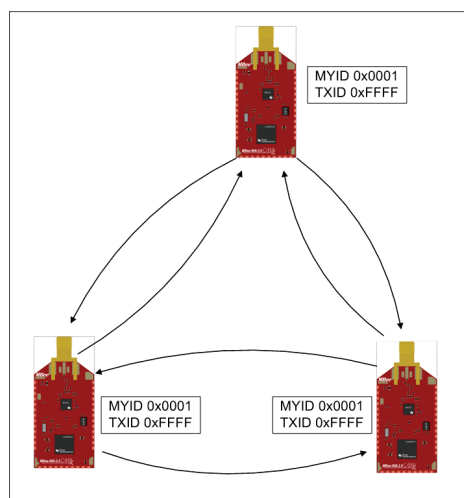


Рисунок 3

2.5. Продолжение краткого теоретического курса

А сейчас, когда мы убедились, что с помощью модулей MBees и ПО SerialStar действительно можно обмениваться данными по радио, самое время научиться делать это осознанно и не только туда, куда получается, но и туда, куда нам требуется. Тот опыт, который мы провели в практической части, можно считать примером сети с топологией «точка-точка» весьма условно. Ведь адрес для передачи оставлен нами «по умолчанию», то есть широкоэвещательным. Чтобы сделать линию связи «точка-точка» честно, придется немного подправить настройки. Прежде всего, надо сделать каждый узел уникальным внутри одной сети. Думаю, что Вы уже догадались, что надо, не меняя сетевого адреса, изменить идентификаторы узлов. А затем установить адреса для передачи на модулях равными идентификатору своего визави. Сделаем это во второй нашей практической части.

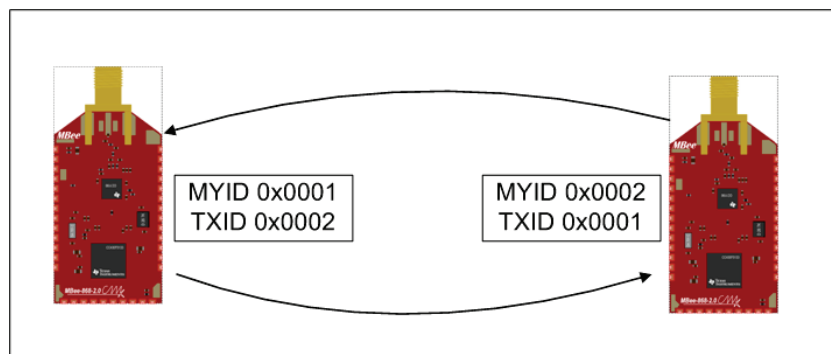


Рисунок 4

2.6. Продолжение практической части

1. Переводим модули в командный режим («+++» или «SYSTEM BUTTON»).
2. В первом окне терминала печатаем «AT TX 2<CR>», «AT CN<CR>».
3. Во втором окне вводим «AT MY 2<CR>», «AT TX 1<CR>» «AT CN<CR>».
4. Выходим из командного режима также, как и вошли.
5. Все, что печатаем в одном окне, должны получить во втором и наоборот.
6. Теперь мы корректно построили линию «точка-точка» уже на новом, осознанном уровне. А вот разобраться в смысле тех действий, которые мы предприняли в пп. 2 и 3 Вы уже можете самостоятельно или с помощью документа «sw_SerialStar.pdf».

2.7. Окончание краткого теоретического курса

Пора переходить к чему-либо более продвинутому. На системе «точка-точка» жизнь же не заканчивается. Посмотрим, как, с использованием SerialStar построить сеть с архитектурой «звезда», не переходя при этом в режим API-фрейма. Допустим, что по условиям задачи, необходимо от нескольких узлов передавать данные в направлении одного центра. Назовем этот центр Концентратором. Узлы, передающие данные, будем называть Конечными устройствами. Данные поступают на UART в прозрачном режиме. Как сделать так, чтобы данные от них получал бы только Концентратор? Все просто. Надо на всех модулях Конечных устройств установить с помощью AT-команды «TX» адрес для передачи, равным адресу Концентратора. Ничего страшного, если у всех узлов окажется один и тот же собственный идентификатор. Вот и все. Учтите только, что если потребуется передать что-нибудь в обратном направлении, то есть от Концентратора к Конечным устройствам, то данные будут получены всеми ими одновременно.

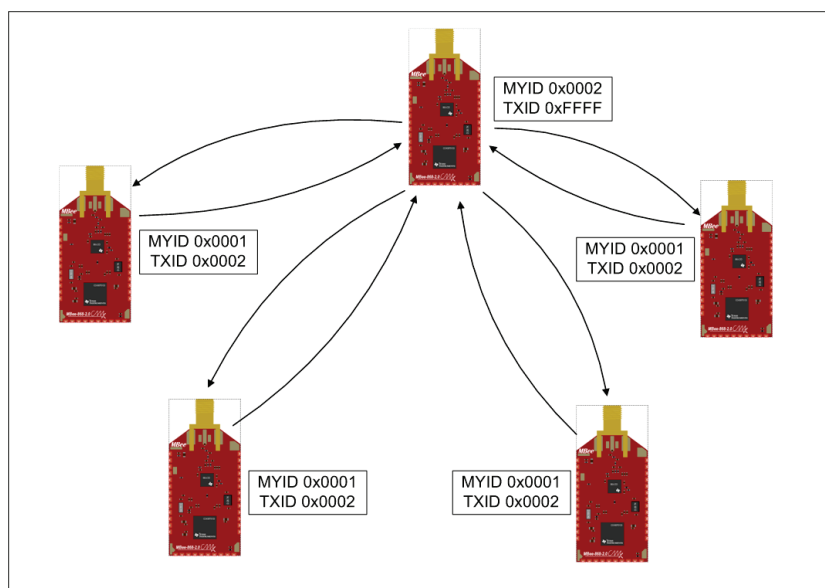


Рисунок 5

2.8. Окончание практического курса

Мы решили не проводить практической курса по развертыванию сети с архитектурой «звезда», поскольку для этого требуется несколько большее, чем 2, количество модулей и плат MB-USBridge. И нам бы пришлось указывать это количество в перечне устройств в разделе «Что потребуется». А большое количество необходимого оборудования могло бы Вас оттолкнуть от чтения этого документа,

поскольку Вы бы решили, что раз у Вас нет нужного числа плат, то и читать дальше нет никакого смысла. Сеем надеяться, что предыдущий материал изложен нами достаточно ясно, и когда у Вас появится объект, для которого будет необходима «звезда», Вы без труда справитесь со всеми необходимыми настройками.

2.9. Ответы на возможные вопросы и примеры

- 1. Могу ли я оставить все настройки «как есть» и развернуть сеть более, чем из 2-ух узлов?**
Конечно можете. Такая сеть вполне имеет право на существование. Но только если Вы самостоятельно найдете способ определить, от какого именно узла пришли данные. В такой сети все узлы видят друг друга и переданный одним из них пакет, будет получен каждым узлом. И, если не предпринимать никаких мер, то будет совершенно непонятно, кто является его отправителем. Если переходить к умным словам, то эта сеть обеспечит только физический уровень протокола. А это значит, что все более высокие уровни должны взять на себя хост-системы, к которым модули подключены. Они и должны формировать адреса, обеспечивать надежность доставки и решать вопросы маршрутизации. Допустим, есть такая практическая задача – заменить проводную линию связи между несколькими промышленными контроллерами и панелью оператора на беспроводной канал. Проводной канал был построен на интерфейсе RS485 MODBUS. Панель оператора выступает в роли мастера. Каждый контроллер имеет свой уникальный адрес в сети. Передача данных всегда инициируется мастером. В такой сети никаких предварительных настроек модулей не потребуется. Такая сеть обычно начинает работать сразу и не требует никаких перенастроек контроллеров или панели оператора. А принцип ее работы такой: панель оператора, поскольку она является мастером, передает запрос данных или команду необходимому контроллеру. Поскольку модуль, подключенный к панели оператора, работает в прозрачном режиме, этот запрос передается в эфир в неизменном виде в широкополосном режиме. Этот пакет будет получен всеми узлами и передан ими в направлении контроллеров тоже неизменным. Тот контроллер, адрес которого содержался в пакете, должен отреагировать на него ответным сообщением, которое тоже будет получено всеми узлами, то есть произойдет, в общем, все то же самое, что и при работе интерфейса RS485. Все контроллеры отбросят это сообщение, а панель оператора его получит. Подробнее о некоторых нюансах, возникающих при замене проводного канала беспроводным Вы можете посмотреть в документе «app_SerialStar_UART_Bridge.pdf». Да вот еще. Подключать модули к контроллерам и панели управления будет очень удобно с помощью плат MB-Serial.
- 2. Меня не устраивает то, что модуль принимает все подряд. Мне хотелось бы, чтобы он принимал данные, адресованные только ему. Как быть?** На самом деле в SerialStar есть возможность заблокировать прием широкополосных пакетов. Для этого надо всего лишь установить младший бит регистра режимов работы, который доступен с помощью AT-команды «DM» (Device Mode). Только надо помнить, что этот регистр представляет собой битовое поле, и перед тем, как изменить его содержимое, лучше будет его считать, чтобы убедиться в том, что Вы случайно не испортите другие его биты.
- 3. А мне хотелось бы иметь возможность отправлять данные и в широкополосном режиме и адресно. И чтобы не надо было при этом каждый раз менять параметр TXID. Как мне поступить?** В этом случае никак не обойтись без пакетного режима. Только с его помощью можно реализовать все возможности по передаче данных, заложенные в SerialStar. Посмотрите про него подробнее в «sw_SerialStar.pdf».

2.10. А если вопросы все-таки остались?

Ждем Вас с вопросами на нашем форуме. Постараемся помочь.

3. ИСТОРИЯ ДОКУМЕНТА

Дата	Редакция документа	Описание изменений
31.05.2017	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
