

Е.А. Бень

RS-485 для чайников



Е.А. Бень

RS-485 для чайников

Москва
2003

Бень Е.А.
RS-485 для чайников. Статья. – Москва: ООО "Маяк", 2003.- 20с

Данная статья эта - попытка объяснить начинающим специалистам, что к чему и помочь обойти грабли, на которые уже кто-то наступал. В этой статье автор собрал начальную информацию об устройствах, полезные добытые сведения (с ссылками) и используя собственный опыт приводит практические рекомендации по наладке линии связи, делится с начинающими как организовать связь между несколькими устройствами на базе микроконтроллеров.

Содержание

- Вступление
- 1. Универсальный асинхронный приемопередатчик (UART)
- 2. Интерфейс RS-485
- 3. Согласование и конфигурация линии связи
- 4. Защитное смещение
- 5. Исключение приема при передаче в полудуплексном режиме
- 6. "Горячее" подключение к линии связи
- 7. Рекомендации по организации протокола связи
- 8. Программные методы борьбы со сбоями
- 9. Защита устройств от перенапряжений в линии связи
- 10. Дополнительные меры защиты от помех
- Заключение
- Полезные ссылки

Вступление

Как следует из названия, статья эта - попытка объяснить начинающим что к чему и помочь обойти грабли, на которые уже кто-то наступал. Если кто-нибудь, потратив 15 минут на прочтение, сэкономит пару дней на отладке системы, я буду считать свою задачу выполненной.

Столкнувшись с необходимостью организовать связь между несколькими устройствами на базе микроконтроллеров, я стал собирать информацию по этой теме. Из конструктивных соображений определился с методом связи - последовательный канал и линия связи на основе интерфейса RS-485. Затем последовал отсев излишних теоретических рассуждений и отбор практических рекомендаций по наладке линии связи. Особое внимание я уделял описанию различных "подводных камней". На бумаге, как известно, всегда все работает, а вот на практике вскрываются неприятные особенности, притом нигде не описанные. Так и оказалось - сделали все по инструкции, а потом не одну неделю отлавливали глюки.

В этой статье я попытался собрать начальную информацию об устройствах, полезные добытые сведения (с ссылками) и собственный опыт.

1. Универсальный асинхронный приемопередатчик (UART)

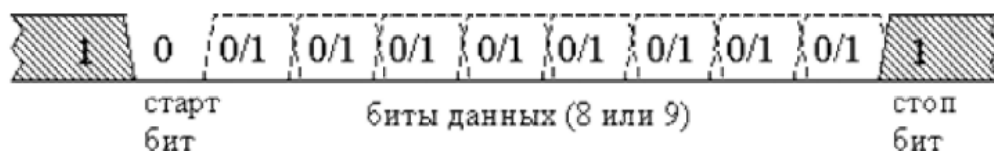
Возможно, связь через асинхронный последовательный порт уходит в прошлое, однако сложно найти контроллер, не имеющего в составе своей периферии UART. Поэтому хоронить его, думаю, рановато. Раз так, то будет нелишним сказать пару слов о том, как оно работает. Описание конкретной реализации последовательного порта есть в datasheet на каждый контроллер, поэтому опишу общее для всех.

UART можно разделить на приемник (Receiver) и передатчик (Transmitter). В состав UART входят: тактовый генератор связи (бодрейт-генератор), управляющие регистры, статусные регистры, буферы и сдвиговые регистры приемника и передатчика. Бодрейт-генератор задает тактовую частоту приемопередатчика для данной скорости связи. Управляющие регистры задают режим работы последовательного порта и его прерываний. В статусном регистре устанавливаются флаги по различным событиям. В буфер приемника попадает принятый символ, в буфер передатчика помещают передаваемый. Сдвиговый регистр передатчика - это обойма, из которой в последовательный порт выстреливаются биты передаваемого символа (кадра). Сдвиговый регистр приемника по биту накапливает принимаемые из порта биты. По различным событиям устанавливаются флаги и генерируются прерывания (завершение приема/отправки кадра, освобождение буфера, различные ошибки).

UART - полнодуплексный интерфейс, то есть приемник и передатчик могут работать одновременно, независимо друг от друга. За каждым из них закреплен порт - одна ножка контроллера. Порт приемника обозначают RX, передатчика - TX. Последовательной установкой уровней на этих портах относительно общего провода ("земли") и передается

информация. По умолчанию передатчик устанавливает на линии единичный уровень. Передача начинается посылкой бита с нулевым уровнем (старт-бита), затем идут биты данных младшим битом вперед (низкий уровень - "0", высокий уровень - "1"), завершается посылка передачей одного или двух битов с единичным уровнем (стоп-битов).

Электрический сигнал кадра посылки выглядит так:

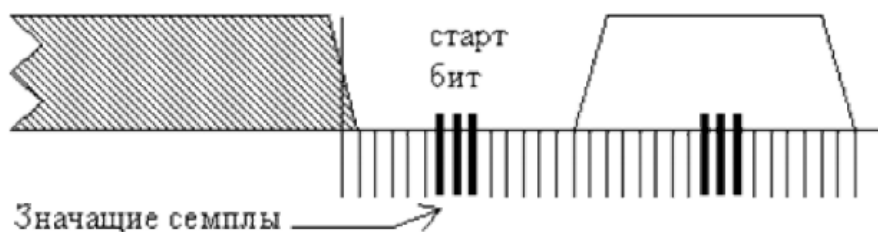


Перед началом связи между двумя устройствами необходимо настроить их приемопередатчики на одинаковую скорость связи и формат кадра.

Скорость связи или baudрейт (baudrate) измеряется в бодах - число передаваемых бит в секунду (включая старт и стоп-биты). Задается эта скорость в baudрейт-генераторе делением системной частоты на задаваемый коэффициент. Типичный диапазон скоростей: 2400 ... 115200 бод.

Формат кадра определяет число стоп-битов (1 или 2), число бит данных (8 или 9), а также назначение девятого бита данных. Все это зависит от типа контроллера.

Приемник и передатчик тактируются, как правило, с 16-кратной частотой относительно baudрейта. Это нужно для сэмплирования сигнала. Приемник, поймав падающий фронт старт-бита, отсчитывает несколько тактов и следующие три такта считывает (сэмплирует) порт RX. Это как раз середина старт-бита. Если большинство значений сэмплов - "0", старт-бит считается состоявшимся, иначе приемник принимает его за шум и ждет следующего падающего фронта. После удачного определения старт-бита, приемник точно также сэмплирует серединки битов данных и по большинству сэмплов считает бит "0" или "1", записывая их в сдвиговый регистр. Стоп-биты тоже сэмплируются, и если уровень стоп-бита не "1" - UART определяет ошибку кадра и устанавливает соответствующий флаг в управляющем регистре.

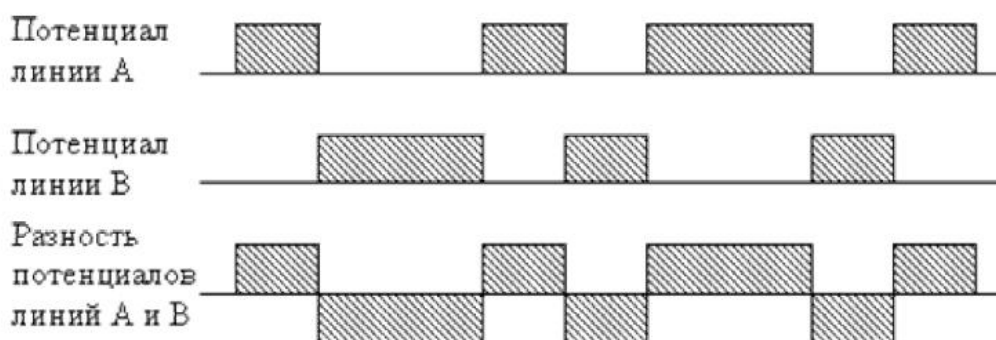


Поскольку baudрейт устанавливается делением системной частоты, при переносе программы на устройство с другим кварцевым резонатором, необходимо изменить соответствующие настройки UART.

2. Интерфейс RS-485

Интерфейс RS-485 (другое название - EIA/TIA-485) - один из наиболее распространенных стандартов физического уровня связи. Физический уровень - это канал связи и способ передачи сигнала (1 уровень модели взаимосвязи открытых систем OSI).

Сеть, построенная на интерфейсе RS-485, представляет собой приемопередатчики, соединенные при помощи витой пары - двух скрученных проводов. В основе интерфейса RS-485 лежит принцип дифференциальной (балансной) передачи данных. Суть его заключается в передаче одного сигнала по двум проводам. Причем по одному проводу (условно А) идет оригинальный сигнал, а по другому (условно В) - его инверсная копия. Другими словами, если на одном проводе "1", то на другом "0" и наоборот. Таким образом, между двумя проводами витой пары всегда есть разность потенциалов: при "1" она положительна, при "0" - отрицательна.

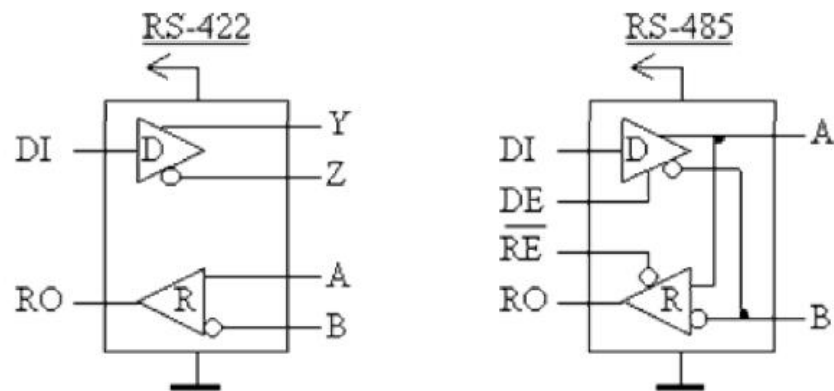


Именно этой разностью потенциалов и передается сигнал. Такой способ передачи обеспечивает высокую устойчивость к синфазной помехе. Синфазной называют помеху, действующую на оба провода линии одинаково. К примеру, электромагнитная волна, проходя через участок линии связи, наводит в обоих проводах потенциал. Если сигнал передается потенциалом в одном проводе относительно общего, как в RS-232, то наводка на этот провод может исказить сигнал относительно хорошо поглощающего наводки общего ("земли"). Кроме того, на сопротивлении длинного общего провода будет падать разность потенциалов земель - дополнительный источник искажений. А при дифференциальной передаче искажения не происходит. В самом деле, если два провода пролегают близко друг к другу, да еще перевиты, то наводка на оба провода одинакова. Потенциал в обоих одинаково нагруженных проводах изменяется одинаково, при этом информативная разность потенциалов остается без изменений.

Аппаратная реализация интерфейса - микросхемы приемопередатчиков с дифференциальными входами/выходами (к линии) и цифровыми портами (к портам UART контроллера). Существуют два варианта такого интерфейса: RS-422 и RS-485.

RS-422 - полнодуплексный интерфейс. Прием и передача идут по двум отдельным парам проводов. На каждой паре проводов может быть только по одному передатчику.

RS-485 - полудуплексный интерфейс. Прием и передача идут по одной паре проводов с разделением по времени. В сети может быть много передатчиков, так как они могут отключаться в режиме приема.



D (driver) - передатчик;

R (receiver) - приемник;

DI (driver input) - цифровой вход передатчика;

RO (receiver output) - цифровой выход приемника;

DE (driver enable) - разрешение работы передатчика;

RE (receiver enable) - разрешение работы приемника;

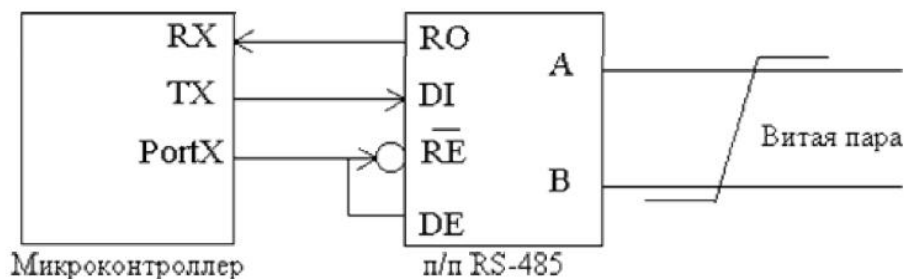
A - прямой дифференциальный вход/выход;

B - инверсный дифференциальный вход/выход;

Y - прямой дифференциальный выход (RS-422);

Z - инверсный дифференциальный выход (RS-422).

Остановлюсь поподробнее на приеме-передатчике RS-485. Цифровой выход приемника (RO) подключается к порту приемника UART (RX). Цифровой вход передатчика (DI) к порту передатчика UART (TX). Поскольку на дифференциальной стороне приемник и передатчик соединены, то во время приема нужно отключать передатчик, а во время передачи - приемник. Для этого служат управляющие входы - разрешение приемника (RE) и разрешения передатчика (DE). Так как вход RE инверсный, то его можно соединить с DE и переключать приемник и передатчик одним сигналом с любого порта контроллера. При уровне "0" - работа на прием, при "1" - на передачу.



Приемник, получая на дифференциальных входах (AB) разность потенциалов (U_{AB}) переводит их в цифровой сигнал на выходе RO. Чувствительность приемника может быть разной, но гарантированный пороговый диапазон распознавания сигнала производители микросхем приеме-передатчиков пишут в документации. Обычно эти пороги составляют ± 200 мВ. То есть, когда $U_{AB} > +200$ мВ - приемник определяет "1", когда $U_{AB} < -200$ мВ - приемник определяет "0". Если разность потенциалов в линии настолько мала, что не выходит за пороговые значения - правильное распознавание сигнала не гарантируется. Кроме того, в линии могут быть и не синфазные помехи, которые исказят столь слабый сигнал.

Все устройства подключаются к одной витой паре одинаково: прямые выходы (А) к одному проводу, инверсные (В) - к другому.

Входное сопротивление приемника со стороны линии (R_{AB}) обычно составляет 12 КОм. Так как мощность передатчика не бесконечна, это создает ограничение на количество приемников, подключенных к линии. Согласно спецификации RS-485 с учетом согласующих резисторов передатчик может вести до 32 приемников. Однако есть ряд микросхем с повышенным входным сопротивлением, что позволяет подключить к линии значительно больше 32 устройств.

Максимальная скорость связи по спецификации RS-485 может достигать 10 Мбод/сек. Максимальное расстояние - 1200 м. Если необходимо организовать связь на расстоянии больше 1200 м или подключить больше устройств, чем допускает нагрузочная способность передатчика - применяют специальные повторители (репитеры).

Стандартные параметры интерфейсов	RS-422	RS-485
Допустимое число передатчиков / приемников	1 / 10	32 / 32
Максимальная длина кабеля	1200 м	1200 м
Максимальная скорость связи	10 Мбит/с	10 Мбит/с
Диапазон напряжений "1" передатчика	+2...+10 В	+1.5...+6 В
Диапазон напряжений "0" передатчика	-2...-10 В	-1.5...-6 В
Диапазон синфазного напряжения передатчика	-3...+3 В	-1...+3 В
Допустимый диапазон напряжений приемника	-7...+7 В	-7...+12 В
Пороговый диапазон чувствительности приемника	± 200 мВ	± 200 мВ
Максимальный ток короткого замыкания драйвера	150 мА	250 мА
Допустимое сопротивление нагрузки передатчика	100 Ом	54 Ом
Входное сопротивление приемника	4 кОм	12 кОм
Максимальное время нарастания сигнала передатчика	10% бита	30% бита

3. Согласование и конфигурация линии связи

При больших расстояниях между устройствами, связанными по витой паре и высоких скоростях передачи начинают проявляться так называемые эффекты длинных линий. Причина этому - конечность скорости распространения электромагнитных волн в проводниках. Скорость эта существенно меньше скорости света в вакууме и составляет

немного больше 200 мм/нс. Электрический сигнал имеет также свойство отражаться от открытых концов линии передачи и ее ответвлений. Грубая аналогия - желоб, наполненный водой. Волна, созданная в одном конце, идет по желобу и, отразившись от стенки в конце, идет обратно, отражается опять и так далее, пока не затухнет. Для коротких линий и малых скоростей передачи этот процесс происходит так быстро, что остается незамеченным. Однако, время реакции приемников - десятки/сотни нс. В таком масштабе времени несколько десятков метров электрический сигнал проходит отнюдь не мгновенно. И если расстояние достаточно большое, фронт сигнала, отразившись в конце линии и вернувшись обратно, может исказить текущий или следующий сигнал. В таких случаях нужно каким-то образом подавлять эффект отражения.

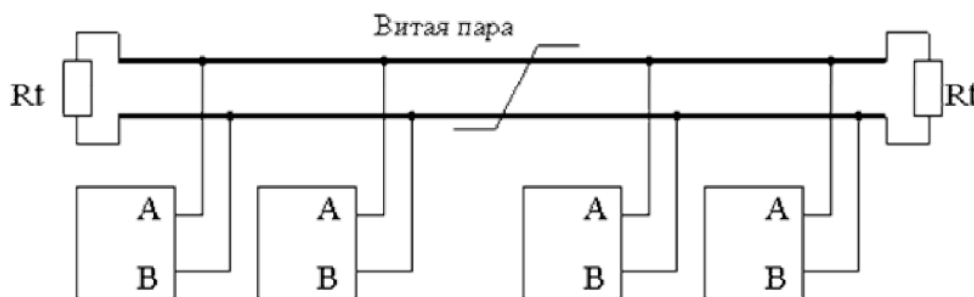
Наука Электротехника предлагает решение этой проблемы. У любой линии связи есть такой параметр, как волновое сопротивление Z_v . Оно зависит от характеристик используемого кабеля, но не от длины. Для обычно применяемых в линиях связи витых пар $Z_v=120$ Ом. Оказывается, что если на удаленном конце линии, между проводниками витой пары включить резистор с номиналом равным волновому сопротивлению линии, то электромагнитная волна дошедшая до "тупика" поглощается на таком резисторе. Отсюда его названия - согласующий резистор или "терминатор".

Большой минус согласования на резисторах - повышенное потребление тока от передатчика, ведь в линию включается низкоомная нагрузка. Поэтому рекомендуется включать передатчик только на время отправки посылки. Есть способы уменьшить потребление тока, включая последовательно с согласующим резистором конденсатор для развязки по постоянному току. Однако, такой способ имеет свои недостатки. Для коротких линий (несколько десятков метров) и низких скоростей (меньше 38400 бод) согласование можно вообще не делать. Подробнее можно почитать в статье ["Обрежьте жирок с RS-485"](#).

Эффект отражения и необходимость правильного согласования накладывают ограничения на конфигурацию линии связи.

Линия связи должна представлять собой один кабель витой пары. К этому кабелю присоединяются все приемники и передатчики. Расстояние от линии до микросхем интерфейса RS-485 должно быть как можно короче, так как длинные ответвления вносят рассогласование и вызывают отражения.

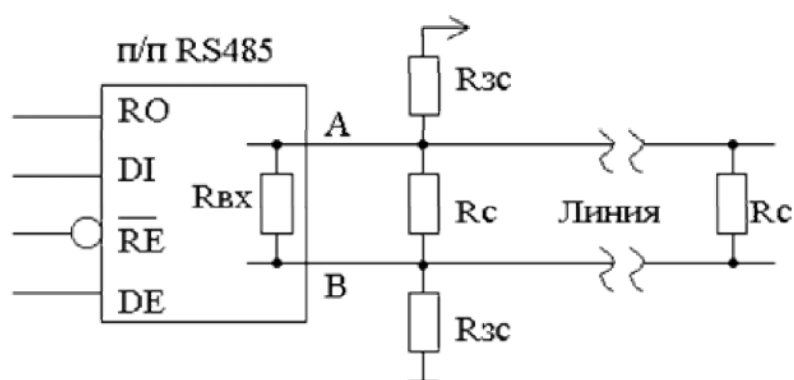
В оба наиболее удаленных конца кабеля ($Z_v=120$ Ом) включают согласующие резисторы R_t по 120 Ом (0.25 Вт). Если в системе только один передатчик и он находится в конце линии, то достаточно одного согласующего резистора на противоположном конце линии. Более подробно о правильных и неправильных конфигурациях сети можно прочитать в статье ["Правильная разводка сетей RS-485"](#).



4. Защитное смещение

Как уже упоминалось, приемники большинства микросхем RS-485 имеют пороговый диапазон распознавания сигнала на входах А-В - $\pm 200\text{мВ}$. Если $|U_{ab}|$ меньше порогового (около 0), то на выходе приемника RO могут быть произвольные логические уровни из-за несинфазной помехи. Такое может случиться либо при отсоединении приемника от линии, либо при отсутствии в линии активных передатчиков, когда никто не задает уровень. Чтобы в этих ситуациях избежать выдачи ошибочных сигналов на приемник UART, необходимо на входах А-В гарантировать разность потенциалов $U_{ab} > +200\text{мВ}$. Это смещение при отсутствии входных сигналов обеспечивает на выходе приемника логическую "1", поддерживая, таким образом, уровень стопового бита.

Добиться этого просто - прямой вход (А) следует подтянуть к питанию, а инверсный (В) - к "земле". Получается делитель:



$R_{vх}$ - входное сопротивление приемника (обычно 12 кОм);

R_c - согласующие резисторы (120 Ом);

R_{3c} - резисторы защитного смещения.

Величины сопротивлений для резисторов защитного смещения (R_{3c}) нетрудно рассчитать по делителю. Необходимо обеспечить $U_{ab} > 200\text{мВ}$. Напряжение питания - 5В.

Сопротивление среднего плеча - 120Ом//120Ом//12КОм на каждый приемник - примерно 57 Ом (для 10 приемников). Таким образом, выходит примерно по 650 Ом на каждый из двух R_{3c} . Для смещения с запасом - сопротивление R_{3c} должно быть меньше 650 Ом. Традиционно ставят 560 Ом.

Обратите внимание: в расчете номинала R_{3c} учитывается нагрузка. Если на линии висит много приемников, то номинал R_{3c} должен быть меньше. В длинных линиях передачи необходимо так же учитывать сопротивление витой пары, которое может "съесть" часть смещающей разности потенциалов для удаленных от места подтяжки устройств. Для длинной линии лучше ставить два комплекта подтягивающих резисторов в оба удаленных конца рядом с терминаторами.

Многие производители приемопередатчиков заявляют о функции безотказности (failsafe) своих изделий, заключающейся во встроенном смещении. Следует различать два вида такой защиты:

Безотказность в открытых цепях. (Open circuit failsafe.) В таких приемопередатчиках применяются встроенные подтягивающие резисторы. Эти резисторы, как правило, высокоомные, чтобы уменьшить потребление тока. Из-за этого необходимое смещение

обеспечивается только для открытых (ненагруженных) дифференциальных входов. В самом деле, если приемник отключен от линии или она не нагружена, тогда в среднем плече делителя остается только большое входное сопротивление, на котором и падает необходимая разность потенциалов. Однако, если приемопередатчик нагрузить на линию с двумя согласующими резисторами по 120 Ом, то в среднем плече делителя оказывается меньше 60 Ом, на которых, по сравнению с высокоомными подтяжками, ничего существенного не падает. Поэтому, если в нагруженной линии нет активных передатчиков, то встроенные резисторы не обеспечивают достаточное смещение. В этом случае, остается необходимость устанавливать внешние резисторы защитного смещения, как это было описано выше.

Истинная безотказность. (True failsafe.) В этих устройствах смещены сами пороги распознавания сигнала. Например: -50 / -200 мВ вместо стандартных порогов ± 200 мВ. То есть при $U_{ab} > -50$ мВ на выходе приемника RO будет логическая "1", а при $U_{ab} < -200$ - на RO будет "0". Таким образом, и в разомкнутой и в пассивной линии при разности потенциалов U_{ab} близкой к нулю, приемник выдаст "1". Для таких приемопередатчиков внешнее защитное смещение не требуется. Тем не менее, для лучшей помехозащищенности все-таки стоит дополнительно немного подтягивать линию.

Сразу виден минус внешнего защитного смещения - через делитель постоянно будет протекать ток, что может быть недопустимо в системах малого потребления. В таком случае можно сделать следующее:

а). Уменьшить потребление тока, увеличив сопротивления R_{zc} . Хотя производители приемопередатчиков и пишут о пороге распознавания в 200 мВ, на практике вполне хватает 100 мВ и даже меньше. Таким образом, можно сразу увеличить сопротивления R_{zc} раза в два-три. Помехозащищенность при этом несколько снижается, но во многих случаях это не критично.

б). Использовать true failsafe приемопередатчики со смещенными порогами распознавания. Например, у микросхем MAX3080 и MAX3471 пороги: -50 мВ / -200 мВ, что гарантирует единичный уровень на выходе приемника при отсутствии смещения ($U_{ab}=0$). Тогда внешние резисторы защитного смещения можно убрать или значительно увеличить их сопротивление.

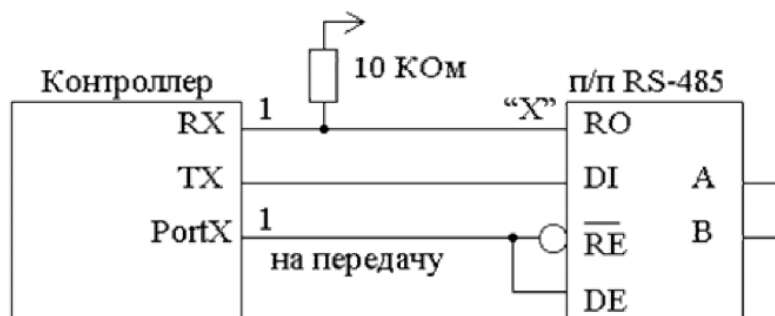
в). Не применять без необходимости согласование на резисторах. Если линия не будет нагружена на 2 по 120 Ом, то для обеспечения защитного смещения хватит подтяжек в несколько килоом в зависимости от числа приемников на линии.

Для опторазвязанной линии подтягивать следует к питанию и "земле" изолированной линии. Если не применяется опторазвязка, подтягивать можно к любому питанию, так как делитель создаст лишь небольшую разность потенциалов между линиями А и В. Нужно только помнить о возможной разности потенциалов между "землями" устройств, расположенных далеко друг от друга.

5. Исключение приема при передаче в полудуплексном режиме

При работе с полудуплексным интерфейсом RS-485 (прием и передача по одной паре проводов с разделением по времени) можно забыть, что UART контроллера - полнодуплексный, то есть принимает и передает независимо и одновременно.

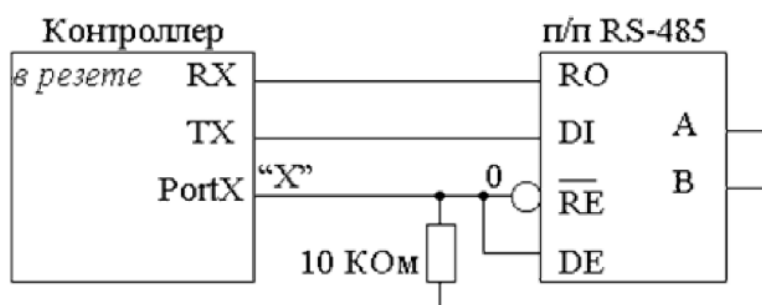
Обычно во время работы приемопередатчика RS-485 на передачу, выход приемника RO переводится в третье состояние и ножка RX контроллера (приемник UART) "повисает в воздухе". В результате, во время передачи на приемнике UART вместо уровня стопового бита ("1") окажется неизвестно что, и любая помеха будет принята за входной сигнал. Поэтому нужно либо на время передачи отключать приемник UART (через управляющий регистр), либо подтягивать RX к единице. У некоторых микроконтроллеров это можно сделать программно - активировать встроенные подтяжки портов.



Примечание: у микроконтроллера AT90S8535 (AVR Atmel) есть глючок - при отключенном UART он все равно принимает, и после включения на прием первый принятый байт может быть испорчен. Так что активировать подтяжку RX ему нужно обязательно.

6. "Горячее" подключение к линии связи

Насколько я знаю, спецификацией RS-485 не предусмотрено "горячее" подключение - включение новых приемопередатчиков в линию связи во время работы системы. Тем не менее, подобную операцию система переносит практически безболезненно, если учесть один нюанс. Это важно, когда питание на устройство подается в момент подключения, например, когда плата в виде кассеты вставляется в разъем. Дело в том, что во время любого сброса: по включению питания, по сигналу на входе "Reset", по срабатыванию сторожевого таймера - контроллеру требуется время на инициализацию, которое может составлять до нескольких десятков миллисекунд. Пока контроллер находится в состоянии сброса, он принудительно настраивает все порты на вход. Получается ситуация, при которой питание на микросхему приемопередатчика RS-485 уже подано, но входы разрешения приемника /RE и передатчика DE "висят в воздухе". В результате, приемопередатчик может по помехе открыться на передачу и все время пока микроконтроллер в отключке пускать в работающую линию мусор. Избежать этого легко - достаточно через резистор в несколько килоом подтянуть вход разрешения приемника /RE к нулю. Этим приемопередатчик сразу по включении питания настраивается на прием и не лезет на линию.



7. Рекомендации по организации протокола связи

На физическом уровне линия связи готова к работе, однако, нужен еще и протокол - договоренность между устройствами системы о формате посылок.

По природе интерфейса RS-485 устройства не могут передавать одновременно - будет конфликт передатчиков. Следовательно, требуется распределить между устройствами право на передачу. Отсюда основное деление: централизованный (одномастерный) обмен и децентрализованный (многомастерный).

В централизованной сети одно устройство всегда ведущее (мастер). Оно генерирует запросы и команды остальным (ведомым) устройствам. Ведомые устройства могут передавать только по команде ведущего. Как правило, обмен между ведомыми идет только через ведущего, хотя для ускорения обмена можно организовать передачу данных от одного ведомого к другому по команде ведущего.

В децентрализованной сети роль ведущего может передаваться от устройству к устройству либо по некоторому алгоритму очередности, либо по команде текущего ведущего к следующему (передача маркера ведущего). При этом ведомое устройство может в своем ответе ведущему передать запрос на переход в режим ведущего и ожидать разрешения или запрета.

Последовательный канал по меркам контроллера - штука медленная. На скорости 9600 бод передача одного символа занимает больше миллисекунды. Поэтому, когда контроллер плотно загружен вычислениями и не должен их останавливать на время обмена по UART, нужно использовать прерывания по завершению приема и передачи символа. Можно выделить место в памяти для формирования посылки на передачу и сохранения принятой посылки (буфер посылки), а также указатели на позицию текущего символа. Прерывания по завершению приема или передачи символа вызывают соответствующие подпрограммы, которые передают или сохраняют очередной символ со сдвигом указателя и проверкой признака конца сообщения, после чего возвращают управление основной программе до следующего прерывания. По завершению отправки или приема всей посылки либо формируется пользовательский флаг, обрабатываемый в основном цикле программы, либо сразу вызывается подпрограмма обработки сообщения.

В общем случае посылка по последовательному каналу состоит из управляющих байтов (синхронизация посылки, адресов отправителя и получателя, контрольной суммы и пр.) и собственно байтов данных.

Протоколов существует множество и можно придумать еще больше, но лучше пользоваться наиболее употребительными из них. Одним из стандартных протоколов последовательной передачи является [MODBUS](#), его поддержку обеспечивают многие производители промышленных контроллеров. Но если Вам нужно буквально "два байта переслать" или просто освоить методы связи и не хочется из-за этого изучать систему команд модбаса и писать для него драйвер, предлагаю варианты относительно простых протоколов. (И все-таки в дальнейшем стоит ориентироваться именно на MODBUS.)

Основная задача в организации протокола - заставить все устройства различать управляющие байты и байты данных. К примеру, ведомое устройство, получая по линии поток байтов, должно понимать, где начало посылки, где конец и кому она адресована.

1). Часто встречаются протоколы на основе ASCII-кода. Управляющие символы и данные передаются в виде обыкновенных ASCII символов. Посылка может выглядеть так:

В HEX виде: 3Ah 31h 32h 52h 53h 34h 38h 35h 0Dh
В ASCII виде: ":" "1" "2" "R" "S" "4" "8" "5" /ПС/

В начале управляющий символ начала посылки ":", следующие две цифры - адрес получателя (12), затем символы данных (RS485) и в конце - управляющий символ конца посылки 0Dh (перевод строки). Все устройства на линии, приняв символ ":", начинают записывать в память посылку до символа конца строки 0Dh. Затем сравнивают адрес из посылки со своим адресом. Устройство с совпавшим адресом обрабатывает данные посылки, остальные - игнорируют посылку. Данные могут содержать любые символы, кроме управляющих (":", 0Dh).

Достоинство этого протокола в удобстве отладки системы и простоте синхронизации посылок. Можно через преобразователь RS485-RS232 подключить линию к COM-порту компьютера и в любой терминалке увидеть всю проходящую информацию "на человеческом языке". Недостатки - относительно большой размер посылки при передаче большого количества двоичной информации, ведь на передачу каждого байта нужно два ASCII символа (7Fh - "7", "F"). Кроме того, надо преобразовывать данные из двоичного вида в ASCII и обратно.

2). Можно организовать протокол с непосредственной передачей двоичных данных. При этом управляющие символы и байты данных различаются с помощью настройки дополнительного девятого бита в UART. Для управляющих символов этот бит устанавливается в "1". Первым в посылке передается управляющий символ с единичным девятым битом - остальные его "нормальные" биты могут содержать адрес устройства-получателя, признак начала/конца посылки и что-нибудь еще. Затем передаются байты данных с нулевым девятым битом. Все принимающие устройства узнают по девятому биту управляющий символ и по содержанию его остальных битов определяют, кому адресованы последующие данные. Адресуемое устройство принимает данные, а все остальные игнорируют их до следующего управляющего символа.

UART некоторых контроллеров, например C167 (Infineon) может в особом режиме (wakeur) автоматически распознавать в полученном байте девятый бит и генерировать прерывание при получении только управляющего символа. Адресуемое устройство при этом нужно переключить в режим обычного приема до следующего управляющего символа. Это позволяет остальным устройствам экономить время на обработке прерываний при получении байтов данных, адресованных не им.

Если требуется сопряжение системы и компьютера с Windows, такой протокол лучше не применять, так как у Windows могут быть проблемы с распознаванием девятого бита в UART.

3) Протокол может быть "чисто" двоичным, то есть без выделения специальных управляющих символов. Синхронизация посылок в этом случае может осуществляться за счет отслеживания паузы между принятыми байтами. Принимающее устройство отсчитывает время с момента последнего приема байта до следующего, и если эта пауза оказывается больше какой-то величины (например, 1.5 - 3.5 байта), делается вывод о потере предыдущей посылки и начале новой. Даже если предыдущая посылка была незакончена - приемный буфер сбрасывается. Можно также синхронизировать посылки по уникальной стартовой последовательности байтов (по аналогии со стартовым символом в

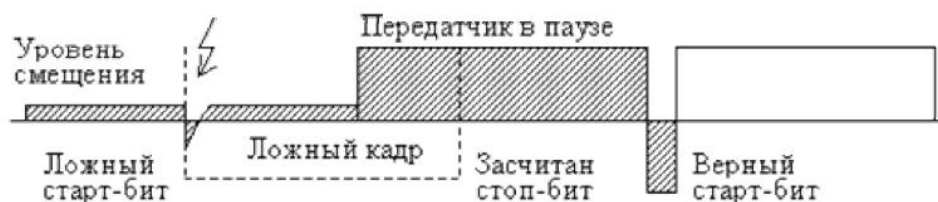
ASCII протоколе). В таких протоколах надо принимать особые меры для защиты от приема ложной посылки, начатой из-за помехи.

8. Программные методы борьбы со сбоями

Для повышения надежности связи обязательно нужно предусмотреть программные методы борьбы со сбоями. Их можно условно разделить на две группы: защита от рассинхронизации и контроль достоверности.

1). Защита от рассинхронизации. Несмотря на защитное смещение, сильная помеха может пробиться в линию без активных передатчиков и нарушить правильную последовательность приема посылок. Тогда возникает необходимость первой же нормальной посылкой вразумить принимающие устройства и не дать им принять помеху за посылку. Делается это с помощью синхронизации кадров (активная пауза) и синхронизации посылок (преамбула).

Защита от рассинхронизации кадров. *Обязательная мера! Все последующие меры синхронизации посылок имеют смысл только совместно с этой.* Помеха ложным стартовым битом может сбить правильный прием кадров последующей посылки. Чтобы вернуться к верной последовательности, нужно сделать паузу между включением передатчика на передачу и посылкой данных. Все это время передатчик удерживает в линии высокий уровень, через который помехе трудно пробиться (активная пауза). Паузы длительностью в 1 кадр на данной скорости связи (10-11 бит) будет достаточно для того, чтобы любое устройство, принимавшее помехи приняло стоп-бит. Тогда следующий кадр будет приниматься с нормального стартового бита.



Того же эффекта можно добиться передачей символа FFh перед первым байтом посылки, так как кроме стартового бита, все его биты - "1". (Если стартовый бит символа FFh попадет на стоп-бит ложного кадра, будет просто засчитана ошибка кадра).

Защита от рассинхронизации посылок. *Применяется совместно с предыдущей защитой!* Особо подлая помеха может замаскироваться под управляющий символ и сбить принимаемую затем посылку. Кроме того предыдущая посылка может быть прервана. Из-за этого крайне желательно в подпрограмме приема и сохранения данных предусмотреть меры по опознанию настоящего начала посылки и сбросу приемного буфера посылки (области памяти, куда сохраняются принимаемые байты). Для этого служит преамбула - предварительный признак начала посылки.

Стартовый символ. В ASCII протоколе роль преамбулы играет специальный управляющий символ начала посылки. По каждому приему такого символа нужно сбрасывать буфер: обнулять число принятых байт, перемещать указатель на начало буфера и т.п. То же самое нужно делать при переполнении буфера. Это позволит настоящему управляющему символу сбросить предыдущую "посылку", начатую ложным символом.

Пример. Последний управляющий символ ":" сбросит предыдущую ложную посылку:

_____ :) _____ : 1 2 R S 4 8 5 /ПС/ _____

Стартовая пауза. В двоичном протоколе, где не предусмотрен уникальный управляющий символ, и синхронизация посылок идет по заданной паузе между байтами, достаточно увеличить активную паузу, описанную в синхронизации кадров, до длительности паузы между байтами, по которой начинается прием новой посылки. То есть, между включением приемопередатчика на передачу и отправкой первых байтов посылки нужно сделать паузу длительностью в 1.5 - 3.5 кадра UART. При активном передатчике во время такой преамбулы помехе трудно будет прорваться к приемникам, они зафиксируют нужную паузу, сбросят буфер посылки и настроятся на прием новой посылки. Этот метод применяется, в частности, для протокола MODBUS RTU.

Стартовая последовательность. Если в двоичном протоколе синхронизация осуществляется лишь по корректному началу посылки, то отфильтровать ложную посылку можно только по логике ее структуры. Преамбула в данном случае - некоторая стартовая последовательность символов, которая не может встретиться в данных посылки, и которую вряд ли сформирует помеха. Преамбула отсылается перед основной посылкой. Принимающее устройство отслеживает в поступающих данных эту стартовую последовательность. Где бы она не состоялась, принимающее устройство сбрасывает буфер посылки и начинает принимать новую.

Вариант 1. Посылка начинает заново приниматься после приема "go!" (вместо символов могут быть любые 8-битные данные):

_____ : - Ъ _____ g o ! 1 2 R S 4 8 5 _____

Вариант 2. Посылка начинает заново приниматься после приема *не менее* трех "E" подряд и стартового байта ":" (вместо символов могут быть любые 8-битные данные):

_____ > : - E _____ E E E : 1 2 R S 4 8 5 _____

Даже если до стартовой последовательности было два таких символа подряд, посылка начнет сохраняться только за последовательностью из не менее чем трех подряд (лишние игнорируются) и стартового символа. Если вместо "E" использовать байт FFh - можно совместить синхронизацию кадров и посылок. Для этого посылаются четыре FFh, а принимающее устройство ожидает не менее трех, с учетом того, что первый байт FFh может уйти на синхронизацию кадров.

2). Контроль достоверности. Особо сильная помеха может вклиниться в посылку, исказить управляющие символы или данные в ней, а то и вовсе уничтожить ее. Кроме того, одно из подключенных к линии устройств (абонент) может выйти из строя и перестать отвечать на запросы. На случай такой беды существуют контрольная сумма, тайм-ауты и квитирование.

Контрольная сумма - в общем случае 1-2 байта кода, полученного некоторым преобразованием из данных посылки. Самое простое - "исключающее или" всем байтам данных. Контрольная сумма рассчитывается и включается в посылку перед отправкой. Принимающее устройство производит ту же операцию над принятыми данными и сверяет рассчитанную контрольную сумму с полученной. Если посылка была повреждена, то,

скорее всего, они не совпадут. В случае применения ASCII протокола - код контрольной суммы также передается ASCII-символами.

Тайм-аут - максимальное время ожидания ответа от запрашиваемого устройства. Если посылка была повреждена или запрашиваемое устройство вышло из строя, то ведущее устройство не повиснет в ожидании ответа, а по истечении определенного времени признает наличие сбоя. После чего можно еще пару раз повторить запрос и, если сбой повторяется, перейти на обработку аварийной ситуации. Тайм-аут отсчитывается с момента завершения передачи запроса. Его длительность должна с небольшим запасом превышать максимальное время ответной передачи плюс время, необходимое на обработку запроса и формирование ответа. Ведомому устройству тоже не мешает обработка тайм-аутов. Особенно в ситуациях, когда отсутствие регулярного обновления данных или новых команд от ведущего устройства критично для работы устройств системы. Самая простая реализация для ведомого - сброс сторожевого таймера по приему посылки. Если по какой-либо причине данные перестали поступать - устройство сбросится по переполнению сторожевого таймера. После сброса устанавливается безопасный режим до приема первой команды.

Квитирование - подтверждение доставки (квитанция). Когда важно, чтобы ведомый обязательно получил данные или команду, возникает необходимость проконтролировать получение им посылки. Ведущее устройство, отправив ведомому данные, ждет ответа с подтверждением. Ведомое устройство, получив данные, в случае их корректности посылает ответ, подтверждающий доставку. Если по истечении тайм-аута ведущее устройство не получает подтверждение, делается вывод о сбое в связи или в ведомом устройстве. Дальше обычные меры - повтор посылки. Но тут есть нюанс: повреждена и не получена может быть сама квитанция. Ведущее устройство, не получив квитанцию, повторяет посылку, и ведомое обрабатывает ее повторно. Не всегда это существенно, но если перепосылалась команда типа "увеличить параметр на 1" это может привести к незапланированному двойному увеличению параметра. В таком случае надо предусмотреть что-нибудь типа циклической нумерации посылок, чтобы ведомое устройство отличало повторные посылки от новых и не обрабатывало их.

9. Защита устройств от перенапряжений в линии связи

Разность потенциалов между проводниками линии и между линией и "землей" приемопередатчика, как правило, не должна выходить за пределы $-7...+12$ В. Следовательно, может потребоваться защита от разности потенциалов между "землями" и от перенапряжений из-за замыкания на высоковольтные цепи.

Разность потенциалов между "землями". При организации сети на основе интерфейса RS-485 следует учитывать неявное присутствие третьего проводника - "земли". Ведь все приемопередатчики имеют питание и "землю". Если устройства расположены недалеко от начального источника питания, то разность потенциалов между "землями" устройств в сети невелика. Но если устройства находятся далеко друг от друга и получают местное питание, то между их "землями" может оказаться существенная разность потенциалов. Возможные последствия - выход из строя приемопередатчика, а то и всего устройства. В таких случаях следует применять гальваническую развязку или дренажный провод.

Гальваническая развязка линии и устройств осуществляется либо опторазвязкой цифровых сигналов (RO, DI, RE, DE) с организацией изолированного питания микросхем приемопередатчиков, либо применением приемопередатчиков со встроенной гальванической развязкой сигналов и питания (например, MAX1480). Тогда вместе с дифференциальными проводниками прокладываются провод изолированной "земли" (сигнальной "земли") и, возможно, провод изолированного питания линии.

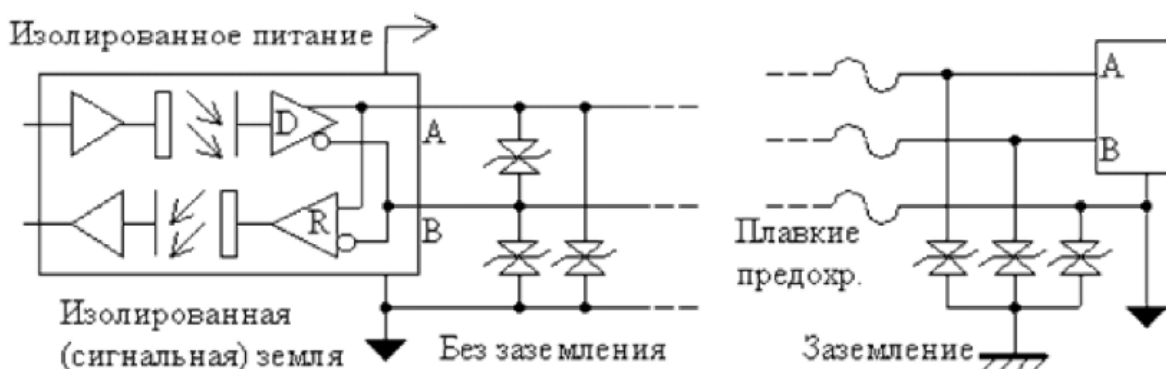
Дренажный провод - провод, прокладываемый вместе с витой парой и соединяющий "земли" удаленных устройств. Через этот провод уравниваются потенциалы "земель". При включении устройства в линию дренажный провод следует подсоединять первым, а при отключении - отсоединять последним. Для ограничения тока через дренажный провод его заземляют в каждом устройстве через резистор в 100 Ом (0.5 Вт).



Замыкание на высоковольтные цепи. Если существует опасность попадания на линию или одну из местных "земель" высокого напряжения, следует применять опторазвязку или шунтирующие ограничители напряжения. А лучше и то и другое.

Напряжение пробоя опторазвязанного интерфейса составляет сотни и даже тысячи вольт. Это хорошо защищает устройство от перенапряжения, общего для всех проводников линии. Однако, при дифференциальных перенапряжениях, когда высокий потенциал оказывается на одном из проводников, сам приемопередатчик будет поврежден.

Для защиты от дифференциальных перенапряжений все проводники линии, включая изолированный общий, шунтируются на локальные "земли" при помощи ограничителей напряжения. Это могут быть варисторы, полупроводниковые ограничители напряжения и газоразрядные трубки. Физический принцип их действия разный, но суть одна - при напряжении выше порогового их сопротивление резко падает, и они шунтируют линию. Газоразрядные трубки могут шунтировать очень большие токи, но имеют высокий порог пробоя и низкое быстродействие, поэтому их лучше применять по трехступенчатой схеме вместе с полупроводниковыми ограничителями. Когда заземление линии невозможно, проводники линии шунтируются ограничителями между собой. Но это защитит только от дифференциальных перенапряжений - защиту от общего должна взять на себя опторазвязка.



Защита ограничителями напряжения действенна при кратковременных перенапряжениях. При длительных - токи короткого замыкания могут вывести ограничители из строя, и устройства на линии окажутся без защиты. Для защиты от коротких замыканий в линию можно последовательно включить плавкие предохранители. Подробнее о защите от перенапряжений можно прочитать в руководстве B&B Electronics ["RS-422 and RS-485 Application Note"](#) (англ.).

10. Дополнительные меры защиты от помех

Диагностика. Если есть возможность выбора маршрута прокладки кабеля с замером уровня помех - не стоит ей пренебрегать. Даже если программная коррекция ошибок успешно справляется со сбоями, нужно сделать все, чтобы физически снизить уровень помех в линии. Полезно предусмотреть в программе диагностический режим, в котором накапливалась бы статистика сбоев, обрабатываемых программной коррекцией (провал по контрольной сумме или тайм-ауту). Если сбоев слишком много, желательно поработать над поиском и устранением их причины. Снижение скорости связи (бодрейта) во многих случаях повышает помехоустойчивость. Не имеет смысла устанавливать скорость обмена больше, чем необходимо для нормальной работы системы, если только не требуется запас на модификацию.

Прокладка кабеля. По возможности не следует проводить витую пару вдоль силовых кабелей, тем более в общей оплетке, так как существует опасность наводок от силовых токов через взаимную индуктивность. Силовое оборудование, коммутирующее большие токи, также является источником помех. Сигнальные цепи питания оптоизолированной линии лучше не использовать для питания чего-либо еще, так как протекающие по сигнальной "земле" лишние токи могут вносить в линию дополнительный шум.

Некачественная витая пара с асимметричными характеристиками проводников - еще один источник проблем. Чем меньше шаг витой пары (чаще перевиты провода) - тем лучше. Даже если не применяется опторазвязанная линия или дренаж, стоит сразу провести кабель с запасной витой парой - на случай, если произойдет обрыв первой или все же понадобится провести сигнальную землю.

Экранирование и заземление. В промышленных условиях, тяжелых в плане электромагнитного шума, рекомендуется применять экранированный кабель с витой парой. Экран, охватывающий проводники линии, защищает их от паразитных емкостных связей и внешних магнитных полей. Экран следует заземлять только в одной из крайних точек линии. Заземление в нескольких точках недопустимо: из-за разности потенциалов местных "земель" по экрану могут протекать существенные токи, которые будут создавать наводки на сигнальные проводники. Некоторые разработчики рекомендуют для защиты от

радиопомех дополнительно включать в нескольких местах между экраном и заземлением специальные высокочастотные конденсаторы емкостью 1...10 нФ.

Индуктивные фильтры. Если в линию все же попадают высокочастотные помехи, их можно отсеять индуктивными фильтрами. Существуют специальные индуктивные фильтры, предназначенные для подавления высокочастотных помех в линиях связи. Они последовательно включаются в линию непосредственно у приемников. Например, В82790-S**** фирмы Еrcos, выполненный в виде четырехполюсника, через который витая пара подсоединяется к приемнику.



Заключение

Я не претендую на полноту сведений о физических и программных тонкостях связи по интерфейсу RS-485. Однако, полагаю, что еще одно изложение темы, немного по другому сформулированное, и к тому же дополненное личным опытом не будет лишним для разработчиков, только начинающих разбираться в этой области. Надеюсь, приведенная информация поможет Вам в организации беспроблемной и надежной связи.

Полезные ссылки

1. ["Обрежьте жирок с RS-485"](#)

Maxim's Application Note 386 (A184, март 2001 года), пер. И.Н. Бирюков.

<http://www.gaw.ru/html.cgi/txt/interface/rs485/power.htm> (рус)

http://www.maxim-ic.com/appnotes.cfm/appnote_number/386 (англ)

2. ["Правильная разводка сетей RS-485"](#)

Maxim's Application Note 763 (январь 2001 года), пер. И.Н. Бирюков.

<http://www.gaw.ru/html.cgi/txt/interface/rs485/app.htm> (рус)

http://www.maxim-ic.com/appnotes.cfm/appnote_number/763 (англ)

3. ["Интерфейсы последовательной передачи данных. Стандарты RS-422/RS-485"](#)

Александр Локотков. Журнал "СТА" 3/97.

http://www.cta.ru/pdf/1997-3/note1_1997_3.pdf

4. ["Описание протокола MODBUS"](#)

http://www.idom.ru/files/Schneider/Info/Networks/MODBUS/PI_MBus_300.Pdf (англ)

http://www.idom.ru/files/Schneider/Info/Networks/MODBUS/Modbus_Rus.Doc (рус)

5. ["RS-422 and RS-485 Application Note"](#)

B&B Electronics Mfg. Co. Inc (Rev. 1997)

<http://www.bb-elec.com/bb-elec/literature/tech/485appnote.pdf> (англ)

6. ["Руководство по применению DMX512"](http://dsl.msk.ru/rus/around/dmx512/dmx512.htm) (управление диммерами по RS-485)
Adam Bennette. Copyright © PLASA 1994. (Пер. АО "ДСЛ" 1995-97)
<http://dsl.msk.ru/rus/around/dmx512/dmx512.htm>
7. ["Selecting and Using RS-232, RS-422, and RS-485 Serial Data Standards"](http://www.maxim-ic.com/appnotes.cfm/appnote_number/723)
Maxim's Application Note 723
http://www.maxim-ic.com/appnotes.cfm/appnote_number/723
8. ["Explanation of Maxim RS-485 Features"](http://www.maxim-ic.com/appnotes.cfm/appnote_number/367)
Maxim's Application Note 367
http://www.maxim-ic.com/appnotes.cfm/appnote_number/367