

УДК 681.58:681.32

ПРОТОКОЛ И РЕАЛИЗАЦИЯ СЕТЕВОГО ВЗАИМОДЕЙСТВИЯ В МОДУЛЬНЫХ СИСТЕМАХ УПРАВЛЕНИЯ

Матвеев Дмитрий Алексеевич

ООО «Электронные и программные системы», Российская Федерация, г. Саранск,
E-mail: Dimitry88@mail.ru, тел. +7(917)9983957, 430000, ул. Коммунистическая, д. 13

Аннотация. В статье приводится способ создания микро сетевого взаимодействия модулей распределенной системы управления, топология сетевого соединения, схемотехническое решение аппаратной части, рассматривается протокол обмена командами, а также оценивается теоретическая пропускная способность.

Ключевые понятия: микроконтроллер, манчестерское кодирование, NRZ-кодирование, программируемая логика, волоконно-оптическая линия связи, оптический трансивер, модель OSI, модуль UART, сетевой адаптер

Введение

В современных условиях развития промышленной электроники, когда от общей концепции построения системы управления до создания макетного образца необходимо затратить минимальное количество времени, исчисляемое несколькими месяцами, твердо укоренилась тенденция модульного дизайна. Такое обстоятельство позволяет разработчикам распределить работу по проектировке целостной системы, где каждый конструктор занят созданием своего функционально законченного модуля. При таком подходе основную трудность составляет общее объединение модулей, что зачастую требует не стандартных решений. Наиболее сложно стоит вопрос для пространственно распределенных модульных систем, работающих в условиях сильных ЭМ (электромагнитных) помех. Применение микро сетевых принципов организации соединения модулей, позволяет разработчику, руководствуясь принятым стандартом схемотехнического решения сетевого адаптера, а также протоколом обмена, упростить работу по организации взаимосвязи.

Теория

Основным подходом при проектировке межмодульных соединений является минимизация числа линий связи, что повышает надежность, помехоустойчивость линии передачи данных, а также экономическую целесообразность. С целью выполнения данного условия, в качестве топологии сетевого соединения было выбрано кольцевое соединение станций, такой подход позволяет отказаться от узла агрегирования, упростить работу по прокладке соединительных линий, а также ввести функции самодиагностики, как сетевого тракта так и системы в целом. В данной функции подразумевается идеология неразрывности кольца передачи данных, при котором в случае обрыва соединения или неработоспособности одной из станций, данные не могут быть переданы по кругу от первого модуля к последнему.

Структурно состав разработанного микро сетевого протокола представлен на рисунке 1, здесь вводятся следующие понятия:

Линия передачи данных - физическое соединение модулей, выполненное на основе медного или волоконно-оптического кабеля.

Адаптер – аппаратный комплекс, осуществляющий взаимодействие модуля с сетью передачи данных включающий, трансивер для выбранной среды передачи данных, а логику управления.

Модуль – функционально законченное устройство, требующее взаимодействия с себе подобными блоками.

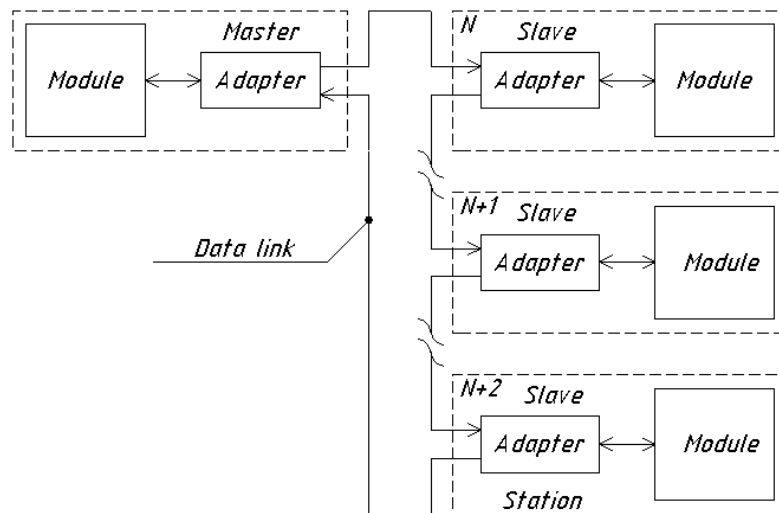


Рис 1. Общая модель организации сети

В данном случае (и для большинства встраиваемых решений) реализуется метод: “один ведущий множество ведомых” при котором одна из станций сетевого взаимодействия является ведущей, инициирует обмен, формирует команды и контролирует эхо ответы, остальные являются ведомыми, но могут осуществлять асинхронную передачу данных к ведущей станции. Структурная схема сетевого адаптера представлена на рисунке 2, рассмотрим более детально логику работы данного сетевого адаптера. Здесь байт символа и дискретные параметры, необходимые для передачи поступают на вход мультиплексора, который организует принцип временного разделения (TDMA), после чего данные побайтно отправляются в кодер, где происходит преобразование к требуемому формату, далее битовый поток через трансивер и переключатель связанности поступает в канал связи. В зависимости от требуемой полосы пропускания канала связи, сложности схемотехнического решения и необходимой надежности, кодер работает в NRZ или манчестерском коде, а также осуществлять контроль избыточным кодом CRC16 или битом паритета. Битовый поток, формируемый кодером, может быть передан по медной витой паре в соответствии с электрическими параметрами RS485 или оптическому волокну. Прием данных осуществляется в обратной последовательности.

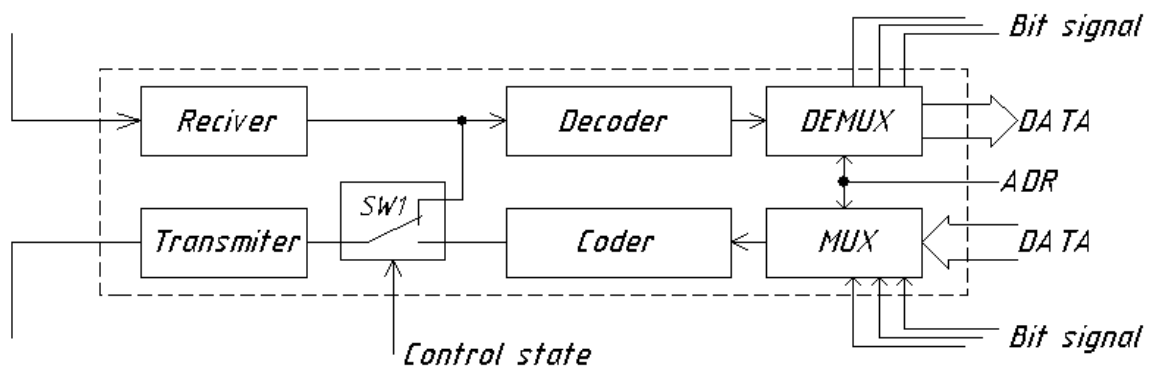


Рис 2. Структура сетевого адаптера

На рисунке 3 приведены диаграммы NRZ и манчестерского кода, получение NRZ осуществляется последовательным сдвигом параллельных данных в регистре и асинхронной передаче на удаленную сторону. У данного кода в виду его простоты есть существенные ограничения не позволяющие передавать длинные последовательности нулей или единиц из-за возможно срыва синхронизации приемника, а так же скоростные ограничения, вызванные необходимостью применения высокостабильных тактовых генераторов. Более совершенным является метод манчестерского кодирования, в котором передаваемые биты, определяют направление перепада логического уровня, в результате чего код обладает хорошими самосинхронизирующимися свойствами. Применение данного метода позволяет осуществлять синхронную передачу длинных битовых посылок, с достаточно высокой скоростью (например, для витой пары 5 категории может составлять 12Мбит\с). Также становится возможным использование импульсных трансформаторов для осуществления гальванической изоляции от линии передачи данных.

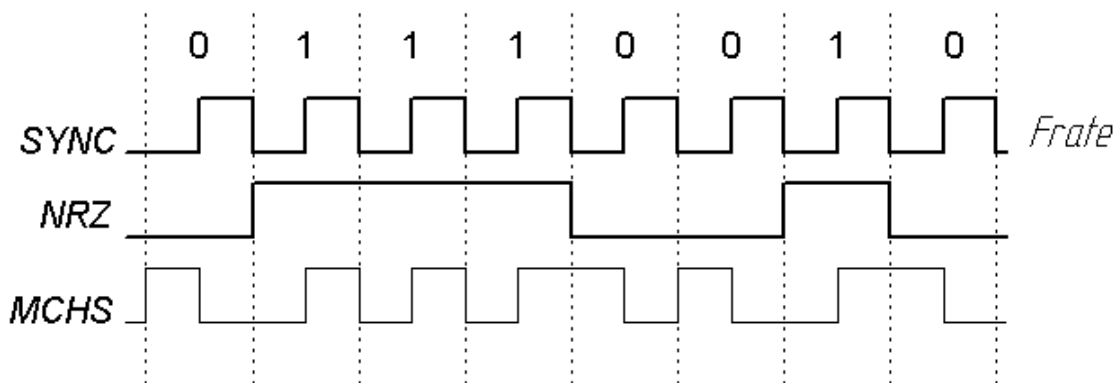


Рис 3. Методы кодирования последовательных данных

Важным элементом в логике работы адаптера является переключатель связанности, предназначенный для перевода адаптера в режим “прозрачности”, необходимость такого действия будет определена при описании работы уровня доступа к среде передачи данных (MAC). Наиболее простым в реализации является адаптер, работающий в коде NRZ с контролем целостности данных по биту паритета, при этом кодер представляет собой сдвиговый регистр, а декодер схему захвата с цифровой фазовой автоподстройкой частоты (ФАПЧ) синхронизации. Структурная схема такого декодера приведена на рисунке 4.

Данный декодер имеет вход последовательных данных, выход параллельных, а также сигнал байтовой синхронизации. Кроме этого декодер обнаруживает одиночные ошибки, и имеет соответствующий выход “Data error”, указывающего на некорректность принятой информации. Описанный простой тип адаптера может быть сконфигурирован в массиве небольшого CPLD кристалла и занимает от 30 до 64 конфигурируемых логических блоков (CLB), что позволяет использовать CPLD уровня EPM7032, EPM7064, ATF1500, XC9500 и им подобные.

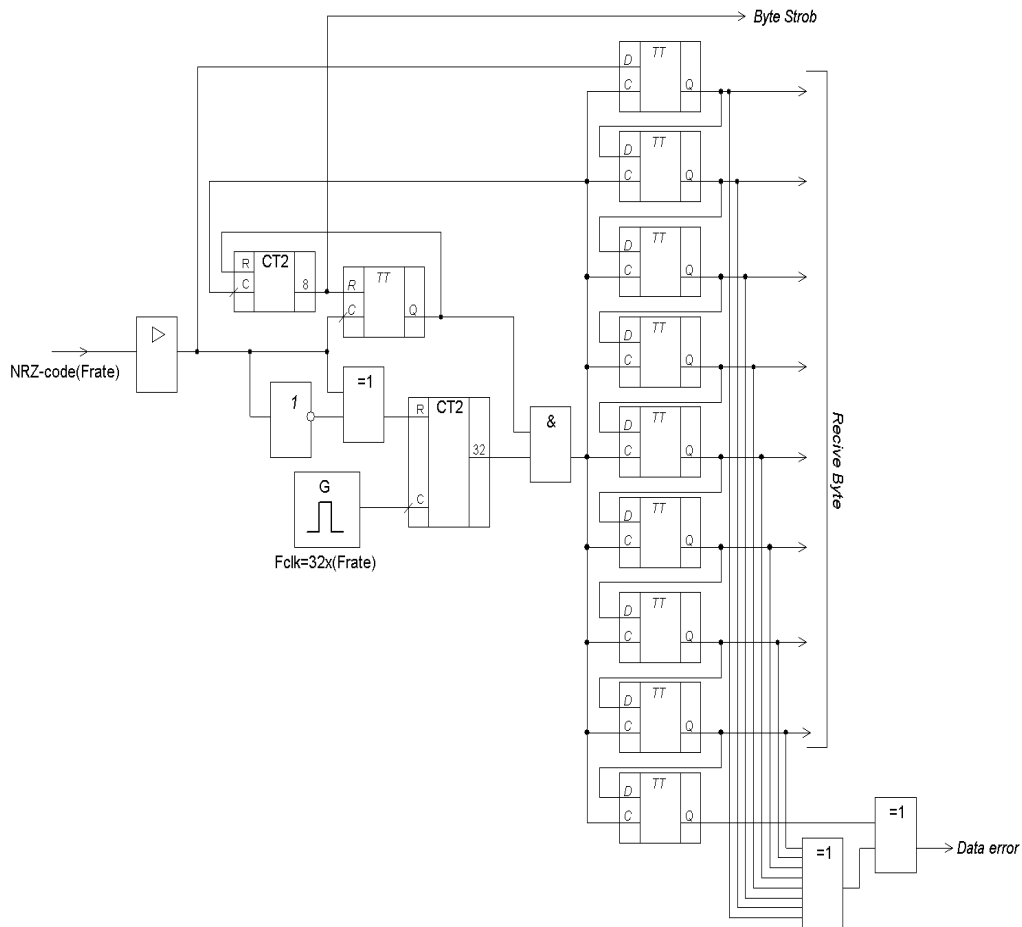


Рис 4. Структура декодера NRZ

Данный декодер имеет вход последовательных данных, выход параллельных, а также сигнал байтовой синхронизации. Кроме этого декодер обнаруживает одиночные ошибки, и имеет соответствующий выход “Data error”, указывающего на некорректность принятой информации. Описанный простой тип адаптера может быть сконфигурирован в массиве небольшого CPLD кристалла и занимает от 30 до 64 конфигурируемых логических блоков (CLB), что позволяет использовать CPLD уровня EPM7032, EPM7064, ATF1500, XC9500 и им подобные.

Также имеется возможность создать простой адаптер на базе внутренних ресурсов микроконтроллера (МК) (см. рисунок 5), здесь используется микросхема буфера с Z-состоянием (K1533ЛП8, 74HC125) для реализации ключа связанности, в качестве трансиверов используются приборы SFH551, SFH756. Кодер, декодер выполнены на базе внутреннего модуля UART МК (AT89S8253), контроль корректности данных и управление ключом связанности осуществляется программным путем. Доступ к среде передачи данных осуществляется следующим образом, в момент, когда модулю необходимо передать сообщение он разрывает связанность петли, передает сообщение, после чего восстанавливает связанность. Тем самым, позволяя другим станциям осуществлять “сквозную передачу” через собственный сетевой адаптер.

Дальнейшее описание протокола сетевого взаимодействия будем производить, опираясь на данную схему.

Разработка протокола осуществлялась с учетом нескольких требований:

1. Возможность производить отладку, как протокола, так и устройств работающих в “петле передачи данных” в терминальных программах.
2. Простота реализации в микропроцессорных системах с небольшой производительностью, организуя как процедуру прерывания.
3. Возможность формирования не ограниченного числа команд
4. Читаемость передаваемых служебных команд и данных
5. Избыточность для контроля корректности запрошенного действия

С учетом всех требований команды и служебные символы представляются в кодировке ASCII, что позволяет давать командам осмысленные обозначения, тем самым выполняя 2 задачи: читаемость протокола при отладке и введение запрещенных сочетаний символов, позволяющих повысить помехозащищенность.

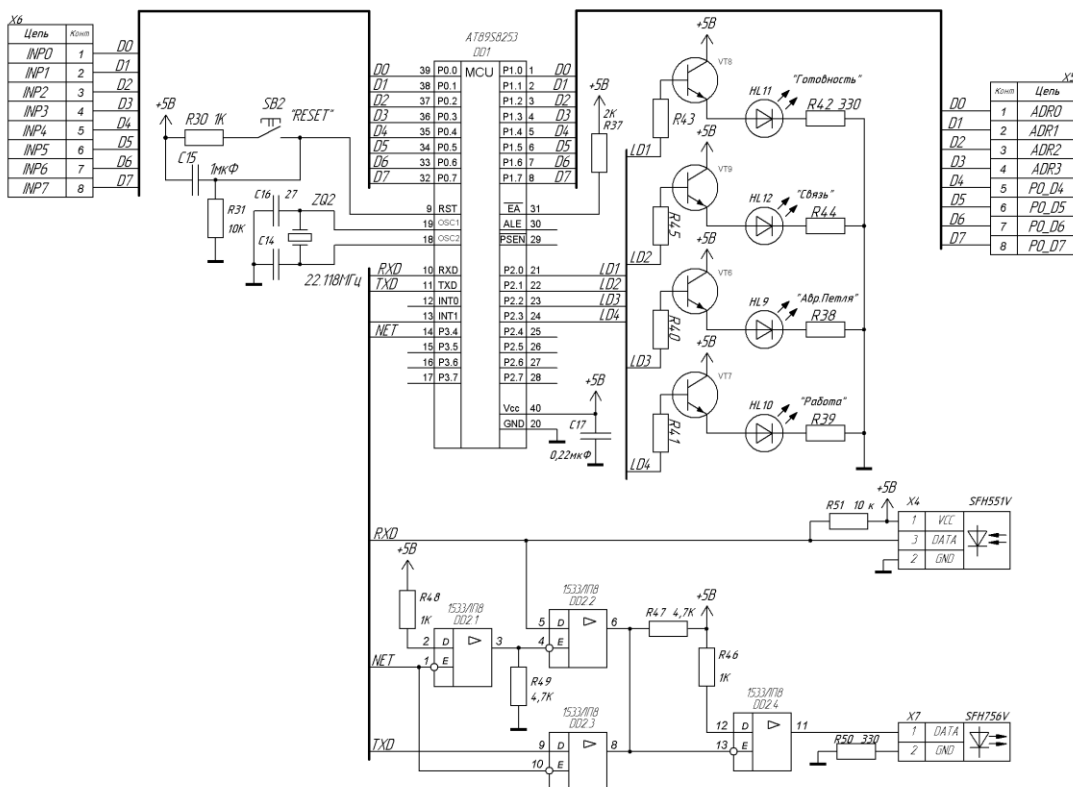


Рис 5. Простая аппаратная реализация адаптера на основе встроенного последовательного порта МК

Рассматриваемый далее протокол сетевого взаимодействия является асимметричным, и ориентирован на передачу количества информации в направлении от ведомого к ведущему модулю большего, чем в обратном. При этом ведущий модуль, в основном осуществляет формирование и выдачу команд, или циклов записи одиночных байтовых констант. Для детального ознакомления необходимо обратиться к рисунку 6. В протоколе предусмотрено два служебных символа: “V” (Variant) с кодом 56h, обозначающим начало константы данных и “CR” (Return) - обозначающим завершение посылки. В протоколе введены следующие исключения: не одна команда не должна начинаться с символа “V” иначе остальные символы команды будут восприняты как операнды, команда содержит только алфавитный набор латинских символов и цифр, в поле данных не должно содержаться символа “CR”, команда “CNF” является зарезервированной и осуществляет загрузку объявленной константы, команда “ID” также является зарезервированной и необходима для выполнения широкопередатель-

ных команд. Существует следующий набор инструкций регламентированных протоколом передачи данных:

1. последовательность передачи команды к ведомой станции и ответ
2. передача одиночной константы к ведомой станции
3. передача константы к ведущей станции
4. передача 8ми байтового массива констант к ведущей станции
5. передача 15 байтового массива констант к ведущей станции
6. последовательность записи одиночной константы и выполнения команды
7. последовательность выполнения команды и передачи константы к ведущей станции
8. последовательность выполнения команды и чтения 15 байтового массива констант из ведомой станции
9. последовательность выполнения широковещательной команды и ответ ведомых станций

Дадим общее пояснение принципа на примере первой инструкции показанной на рисунке 6. Каждый цикл обмена начинается с передачи ведущим модулем символа “V”, адреса модуля к которому необходимо обратиться и завершающего символа “CR”. Следующая посылка содержит строку команды (число символов ограничивается размером буфера) представленную символами ASCII, и завершающий символ “CR”. После чего модуль, распознавший свой адрес, выполняет команду и отправляет ответ ведущей станции об успешном выполнении. Формирование пакета начинается с символа “V” и собственного адреса, завершает посылку символ “CR”. Далее следует зарезервированная команда “ID” указывающая станциям сети, что была осуществлена передача идентификатора готовности. По данному принципу организовано большинство транзакций в сети. Некоторым более сложным алгоритмом формирования пакета обладают команды передачи одиночных констант. Процесс загрузки одиночной константы состоит из двух посылок. Первая отправляет байт константы, обрамленный символами “V” и “CR”, где, в случае равенства константы коду 0Dh (CR) заключительный служебный символ не отправляется. Вторая содержит команду “CNF” предписывающую всем станциям воспринимать первый пакет не как маркер адреса, а как константу. Далее повторяется выше описанная последовательность для передачи пакета с строковой командой.

Некоторые команды предусмотренные разработчиком могут вместо пакета готовности высылать одиночную константу или массив из 8 -15 байт. Пример подобной транзакции приведен на рисунке 6 (8). При запросе массива данных ведомое устройство выставляет символ “V” далее 15 байт данных и 2 байта битовой карты. Как было сказано выше, константы со значением 0Dh не должны присутствовать в массиве данных, дабы не вызывать ложное детектирование конца пакета. По этому, перед отправкой все байты массива содержащие “CR” заменяются на, увеличенное на единицу значение, а их позиционные места отмечаются на битовой карте, что позволяет скорректировать их значения на стороне приемника. Кроме этого данное решение позволяет ввести некий контроль состояния данных, например наличие единицы в битовой карте за порядковым номером байта свидетельствует о том, что этот байт должен быть равен 0Eh, отличное от этого значение свидетельствует о сбое в передаче пакета. Байты битовой карты высылаются в порядке старшинства, при этом старший байт никогда не равен 0Dh т.к. его младший значащий бит всегда равен нулю, а в случае равенства младшего байта коду “CR” заключительный символ не высылается.

Адресация модулей заслуживает отдельного внимания, т.к. она может начинаться с любого значения, но для удобства работы используются коды цифр ASCII (30h-39h), что является удобным при проведении отладочных работ в терминале. Адрес 0 в символьном представлении или 30h в численном, является широковещательным и предписывает всем станциям принять команду к исполнению. В этом случае ответ ведущей станции будет носить организованный характер, при котором станции будут подтверждать готовность выполнения в порядке своих адресов. Более детально данный процесс представлен на рисунке 6 (9)

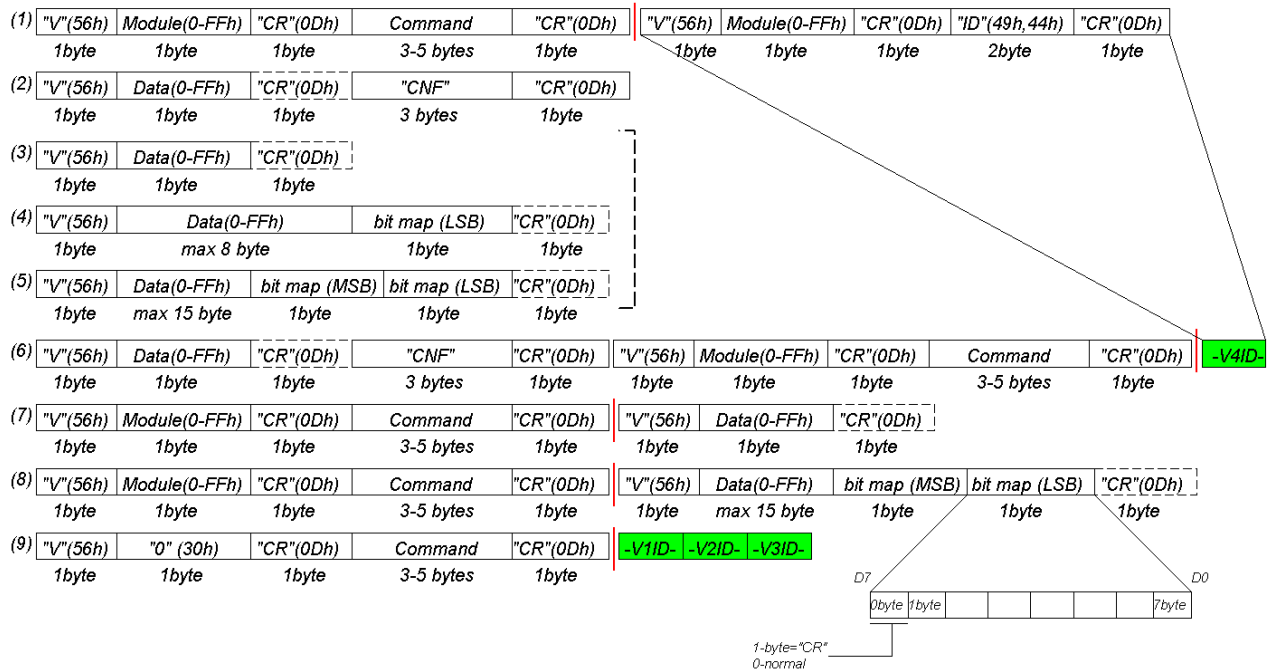


Рис 6. Протокол передачи данных\команд

Последним не определенным пунктом является присвоение адресов станциям сети, данный процесс проиллюстрирован на рисунке 7. Изначально разрабатываемый микро сетевой протокол предусматривает два состояния петли передачи данных: режим полной связанности, при котором приемные тракты всех стаций подключены к петле и ожидают команд и режим ожидания команды присвоения адресов.

В режиме ожидания присвоения адресов внутренний переключатель связанности находится в отключенном состоянии, в результате чего только соседний с ведущей станцией ведомый модуль может получать команды. На этом принципе основан процесс присвоения адресов, при котором модуль получающий команду $V0\langle CR\rangle\text{ADR}\langle CR\rangle$ назначает себя первым модулем в сети (прибавляет 1 к заголовку) и восстанавливает связанность. Далее отправляет следующей станции команду $V1\langle CR\rangle\text{ADR}\langle CR\rangle$ и так далее пока последняя станция не получит адрес, а ведущая станции по ее ответу не определит сколько узлов находится в сети. Как только данная процедура выполнена, петля готова к работе в нормальном режиме.

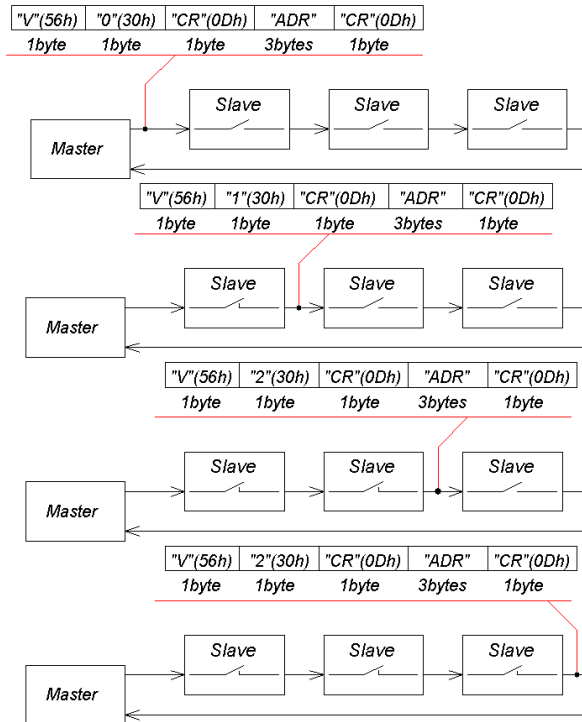


Рис 7. Процесс присвоения адресов

Выводы и заключение

Данный протокол передачи данных был промоделирован в среде ISIS 7, а также реализован в макетных образцах построенных на базе МК AT89S8253. Данный макет применен в модульной системе управления преобразователем частоты ВПЧА, где показал успешную работу, позволяя существенно сократить число внешних межмодульных связей, а также повысить функциональную насыщенность системы управления.

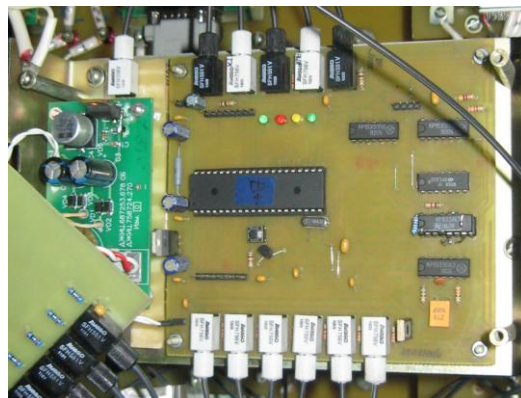


Рис 8. Макет станции для работы в кодировке NRZ по волоконно-оптической линии связи

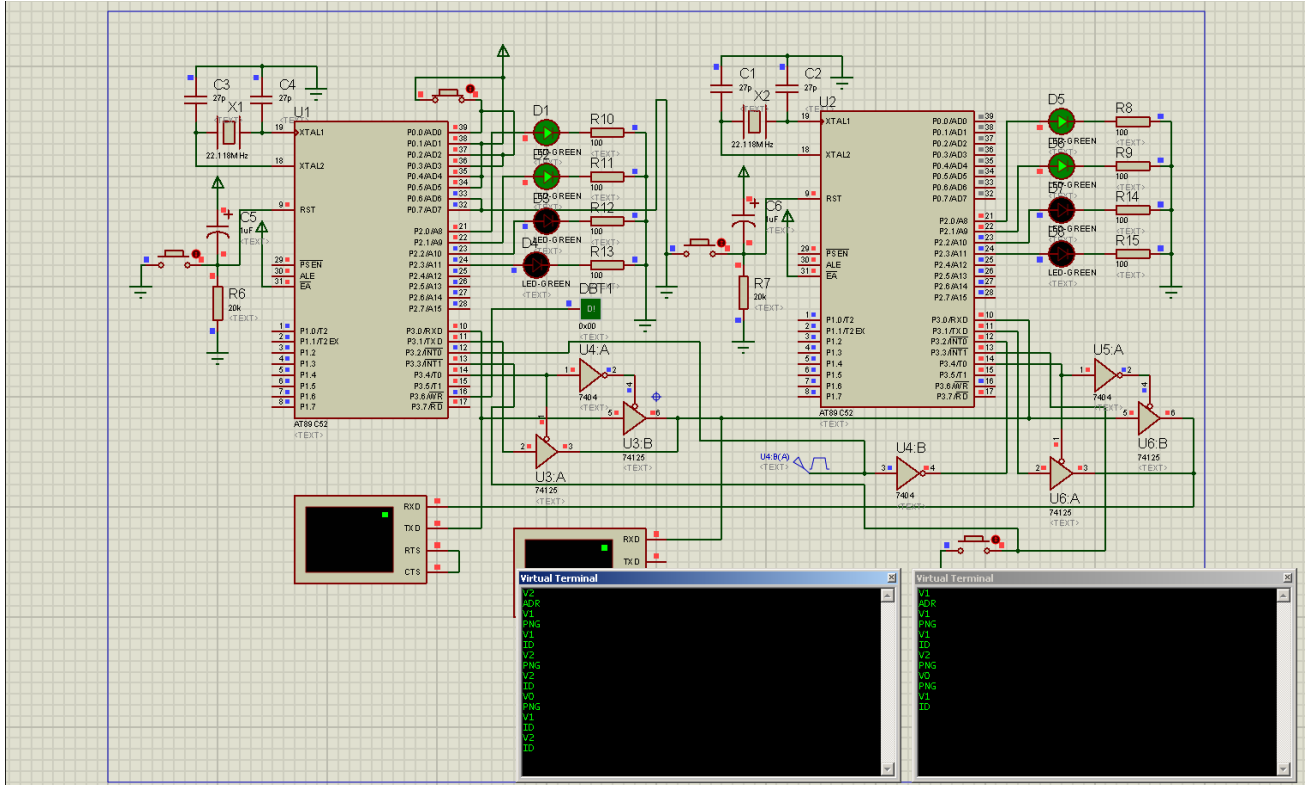


Рис 9. Модель в среде ISIS 7 состоящая из двух ведомых станций и терминала в роли ведущей станции

PROTOCOL AND REALIZATION NETWORKING IN MODULAR SYSTEMS CONTROL

Matveev A. Dmitri

«Electronic and program systems», Russian Federation, Saransk city, Mordovia, E-mail:
Dimitry88@mail.ru, tel. +7(917)9983957, 13, Kommunisticheskaya str., 430000.

Annotation. The article provides a way to create a micro-networking modules distributed control system, the topology of the network connection, hardware circuit solution is considered protocol commands, and estimated theoretical capacity.

Key words: the microcontroller, Manchester encoding, NRZ-coding, programmable logic, fiber-optic communications, optical transceiver, the model OSI, the module UART, network adapter.



Матвеев Дмитрий Алексеевич,
Россия, г. Саранск,
Ведущий инженер
ООО «Электронные и программные системы»,
Российская Федерация, г. Саранск,
E-mail: Dimitry88@mail.ru,
Web-site: www.el-prog-sys.ru
тел. +7(917)9983957

Закончил Мордовский государственный университет им. Н. П. Огарёва в 2010 г. по специальности промышленная электроника. В настоящее время является ведущим специалистом ООО «Электронные и программные системы».