

## Внутриприборный интерфейс на микросхемах фирмы Micrel

Иосиф КАРШЕНБОЙМ  
iosifk@eltech.spb.ru

**Цель данной статьи — показать возможности применения микросхем Ethernet-трансиверов, Ethernet-контроллеров и микросхем Ethernet-коммутаторов, выпускаемых фирмой Micrel [1], для построения внутриприборного интерфейса.**

### Введение

Обычно далеко не всякий разработчик приборов проявляет интерес к микросхемам для построения сетей Ethernet — трансиверам, контроллерам и коммутаторам. Обсудив вопросы построения внутриприборного интерфейса, мы определим перспективные направления и посмотрим, сможем ли мы получить более эффективные решения при новом взгляде на эту рутинную проблему. А для начала решим, какой тип интерфейса будет для нас более предпочтительным.

### Описание и сравнение параллельных и последовательных шин

Переход от параллельных шин к последовательным. Когда-то в качестве внутриприборных шин рассматривались в основном только параллельные шины. Деление прибора на составные части было выполнено таким образом, чтобы максимально использовать возможности, предоставляемые простейшими параллельными шинами. Так, например, память была выполнена отдельно от процессора и подключалась к нему по стандартной (или не очень стандартной) параллельной шине. Однако с увеличением степени интеграции микросхем разработчики получили возможность размещать на одной плате не только вычислительные узлы, но и память, а также периферийные устройства (или если не всю периферию, то некоторую ее часть). При этом расширялись и возможности параллельных шин. Они становились более функциональными и более быстрыми. И теперь уже шины использовались для того, чтобы переписать результаты вычислений, произведенных в периферийной плате, в какой-либо распределенный ресурс. Например, в общую память. Но и в этом случае, используя параллельные шины, приходилось мириться с их недостатками.

Достоинства параллельных шин:

- Относительная простота аппаратной и программной реализации.
- Большое число открытых проектов и справочных материалов.

Недостатки параллельных шин:

- Реализация шины требует дорогого многоконтактного разъема.
- Реализация шины требует дорогой аппаратной поддержки: надо иметь буферные каскады на большое число линий, выравнивать задержку в линиях.
- Необходимо задействовать большую площадь печатной платы.
- Довольно трудно осуществить режим «горячей» замены плат.
- Удлинитель интерфейса громоздкие и дорогие.
- Нет возможности (в большинстве случаев) сделать гальваническую развязку.
- Довольно трудно осуществить резервирование шины, так как для этого необходимо множество микросхем, разъемов и значительная площадь платы.

Конечно, уважаемые читатели, и сами могут продолжить этот перечень достоинств и недостатков параллельных шин. Мы не приводим здесь описание конкретных шин потому, что их довольно много. И так исторически сложилось, что кроме стандартных часто применяли и нестандартные по набору сигналов и конструкции шины.

Теперь от параллельных шин перейдем к рассмотрению достоинств и недостатков последовательных шин.

Достоинства последовательных шин:

- Реализация шины не требует дорогого разъема.
- Реализация шины не требует дорогой аппаратной поддержки: нет необходимости передавать сигналы на большое число линий.
- Нет необходимости задействовать большую площадь печатной платы.
- Режим «горячей» замены плат выполняется довольно легко.
- Удлинитель интерфейса (на необходимое расстояние) недорогие.
- Есть возможность (в большинстве случаев) сделать гальваническую развязку.
- Довольно легко осуществить резервирование шины, так как это не требует больших затрат микросхем, разъемов и площади платы.

Недостатки последовательных шин:

- Относительная сложность аппаратной реализации.
- Относительная сложность программной реализации.
- Сравнительно мало открытых проектов и справочных материалов.

Так почему же в настольных компьютерных системах мы видим повсеместную замену параллельных шин на последовательные? Скорее всего, дело в тех самых «вентилеях», которые находятся внутри микросхемы и составляют ее физическую «сущность». Если до определенного периода времени технологи и разработчики микросхем боролись с тем, как разместить на кристалле микросхемы как можно больше этих самых «вентилей», то теперь эта проблема уже решена. «Вентилей» на кристалле размещено так много, что теперь разработчики микросхем решают совершенно другую задачу: что можно сделать с этими вентилями на кристалле.

Да, интерфейс последовательной линии сложнее. И он требует большего числа регистров, триггеров, преобразователей кодов и т. д. Но за счет усложнения внутренней «начинки» микросхем удается сократить затраты на дорогостоящее конструирование печатной платы, на разъемы, уменьшить габариты изделия. Все это вместе приводит к увеличению надежности и повышению пользовательских характеристик изделия. То есть к повышению качества изделия, а зачастую и к снижению цены на него. Вот почему применение в изделиях последовательных интерфейсов сейчас становится предпочтительным.

Каковы дополнительные возможности, предоставляемые последовательными шинами? При рассмотрении параллельных интерфейсов мы не обсуждали вопрос о преобразовании одного типа параллельного интерфейса в другой тип параллельного же интерфейса. Например, если мы хотим преобразовать межприборный интерфейс в интерфейс принтера. Такие преобразователи использовались редко, и большей частью они не были стандартизованы. Другое дело, если в приборе применяется последовательный интерфейс, да еще если этот интерфейс стандартизованный.

Любой преобразователь можно взять готовый (как модуль или как компонент) и встроить его в прибор. Хотите Ethernet «медный» превратить в оптоволоконный? Пожалуйста. Хотите в USB — тоже без проблем. Хотите в набор ИКМ трактов? Ну и так далее...

А теперь — ближе к микросхемам.

### Простейшие варианты реализации интерфейса

Если бы речь шла только о передаче данных с одного компьютера на другой, то здесь, как говорится, «не стоило бы и огород городить»... Все интерфейсы стандартизованы, что называется, «подключил и готово»... Мы же будем рассматривать только внутриприборные интерфейсы, то есть те варианты, когда нам нужно связать две части прибора. Например, сам прибор и выносной блок. Или цифровые датчики, расположенные в разных частях здания. Или что-то еще в этом роде. И выносные блоки обязательно должны быть компьютеризированные. Поэтому и применение стандартных интерфейсов как внутриприборных интерфейсов может быть не всегда оправданно.

Представим себе ситуацию, когда мы хотим передать несколько сигналов на сотню метров. И хотим иметь гальваническую развязку. Мы знаем стандартное решение вопроса. Например, RS-232 или RS-485. Теперь же давайте рассмотрим решение на микросхемах Ethernet-трансиверов. Будем использовать Ethernet-трансивер как сериализатор-десериализатор. Сейчас мы будем рассматривать только те микросхемы, которые работают по стандарту 10/100 Мбит/с, хотя, конечно, можно было бы рассмотреть и гигабитные трансиверы, такие как KSZ9021 [2]. Берем «штатное» включение микросхемы трансивера KSZ8041 (рис. 1) [1, 2].

Как известно, установка режимов трансивера может быть выполнена двумя способами — программно или аппаратно. В первом случае контроллер, находящийся «рядом» с микросхемой трансивера, по служебному интерфейсу заносит информацию, соответствующую требуемому режиму работы, в микросхему. Во втором случае требуемый режим устанавливается путем подключения со-

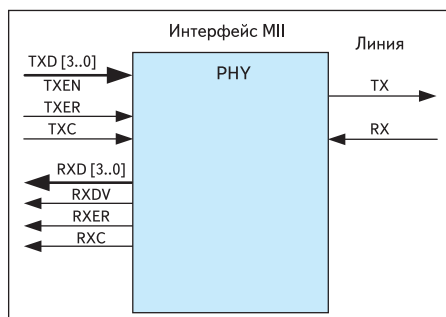


Рис. 1. Пример использования Ethernet-трансивера как сериализатора-десериализатора

Таблица 1. Описание сигналов для интерфейса MII

Название сигнала	Направление передачи данных по отношению к PHY	Направление передачи данных по отношению к MAC	Использование сигнала
Transmit Data: TXD[3..0]	Вход	Выход	Группа параллельных сигналов данных, которые поступают в трансивер из MAC. Они выдаются синхронно относительно TXC
Transmit Enable: TXEN	Вход	Выход	MAC устанавливает этот сигнал, когда установлены достоверные данные на TXD. Этот сигнал должен быть синхронизирован с TXC
Transmit Error: TXER	Вход	Выход	Сообщает о том, что в передаваемом потоке данных произошла ошибка. Этот сигнал должен быть синхронизирован с TXC. Трансивер имеет возможность передавать в линию сигнал ошибки TXER, получаемый от MAC. Когда MAC устанавливает TXER, трансивер установит кодовую комбинацию «Н» на выводах TX. Типичная ситуация для установления сигнала ошибки по передаче такова: хост не успел заполнить буфер передачи данными, и кадр еще не закончен, а буфер считан весь. Передача еще не остановлена, и в линию передаются «пустые» данные, которые не будут соответствовать кадру верхнего уровня. Если не устанавливать сигнал TXER, то неправильно сформированный кадр уйдет в линию и будет принят приемником. Далее приемник, произведя проверку кадра по контрольной сумме, все же отбракует принятый кадр. Но если установить сигнал TXER, то на приемной стороне кадр будет отбракован еще до конца его приема, то есть в момент получения сигнала ошибки
Transmit Clock: TXC	Вход	Выход	Вырабатывается в трансивере и передается в MAC. TXC = 2,5 МГц для операций 10 Мбит/с, TXC = 25 МГц для операций 100 Мбит/с
Receive Data: RXD [3..0]	Выход	Вход	Группа параллельных сигналов данных, которые выдаются из трансивера в MAC синхронно относительно RXC
Receive Data Valid: RXDV	Выход	Вход	Трансивер устанавливает этот сигнал, когда он получает достоверный пакет данных и, соответственно, выдает достоверные данные на RXD. Сигнал на этом выводе синхронный с RXC. Изменения сигнала по времени зависят от того, какой режим по быстройдействию используется в линии: - Для режима 100TX, RXDV устанавливается с первым нибблом Start of Frame Delimiter (SFD) «5D» и остается установленным до последнего ниббла пакета данных. - Для режима 10 Вт полная преамбула усечена
RXER: Receive Error	Выход	Вход	Трансивер сообщает о том, что в приемном потоке данных произошла ошибка. Сигнал на этом выводе синхронный с RXC. На приемной стороне, когда трансивер находится в режиме 100 Мбит/с и получает недопустимый символ из сети, он устанавливает RXER код «1110» на выводах RXD
RXC: Receive Clock	Выход	Вход	Вырабатывается в трансивере и передается в MAC. RXC = 2,5 МГц для операций 10 Мбит/с, RXC = 25 МГц для операций 100 Мбит/с

Таблица 2. Описание сигналов для интерфейса RMII

Название сигнала	Направление передачи данных по отношению к PHY	Направление передачи данных по отношению к MAC	Использование сигнала
REF_CLK	Вход	Вход или выход	Сигнал синхронизации. Используется для приема, передачи и управления (Synchronous clock)
CRS_DV	Выход	Вход	Определение несущей / Принимаемые данные достоверны (Carrier Sense / Receive Data Valid)
RXD[1:0]	Выход	Вход	Принимаемые данные (Receive Data)
TX_EN	Вход	Выход	Разрешение передачи (Transit Enable)
TXD[1:0]	Вход	Выход	Передаваемые данные (Transit Data)
RX_ER	Выход	Вход (не требуется)	Ошибка приема (Receive Error)

Примечание. Когда используется интерфейс RMII, неиспользуемые сигналы интерфейса MII — TXD[3:2], TXER — должны быть подключены к GND.

ответствующих выводов микросхемы к нулю или единице. Поскольку в даташите на микросхему KSZ8041 приведено полное описание подключения выводов в соответствии с требуемым режимом работы, то на рис. 1 эти цепи не показаны.

Со стороны хоста можно получить:

- 4 линии на прием и 4 линии на передачу на частоте 25 МГц — интерфейс MII.
- 2 линии на прием и 2 линии на передачу на частоте 50 МГц — интерфейс RMII.
- 1 линию на прием и 1 линию на передачу на частоте 100 МГц — интерфейс SMII.

Описание основных сигналов для интерфейсов MII, RMII и SMII приведено в таблицах 1–3.

Кроме линий приема/передачи данных, можно также дополнительно задействовать и такие сигналы интерфейса MII, как, например, «Ошибка передачи» и «Ошибка приема».

В стандартном кабеле UTP5 имеется 8 проводов. 4 из них задействованы для подключения одного трансивера. А это значит, что,

Таблица 3. Описание сигналов для интерфейса SMII

Название сигнала	Источник сигнала	Приемник сигнала	Назначение сигнала
RX	PHY	MAC	Принимаемые данные и управление (Receive Data and Control)
TX	MAC	PHY	Передаваемые данные и управление (Transmit Data and Control)
SYNC	MAC	PHY	Синхронизация (Synchronization)
Clock	System	MAC&PHY	Синхронизация (Synchronization)

если расстояние не очень большое и помехи в окружающей среде невелики, то и 4 оставшиеся провода тоже можно задействовать для подключения второго трансивера.

А если мы хотим увеличить расстояние до нескольких сотен метров? И здесь тоже нет никаких проблем. Если необходимо и в дальнейшем использовать медный провод, то мы можем поставить ретрансляторы, тоже выполненные на микросхемах трансиверов KSZ8041. Для одного ретранслятора требуется только 2 такие микросхемы. Схема ретранслятора показана на рис. 2.

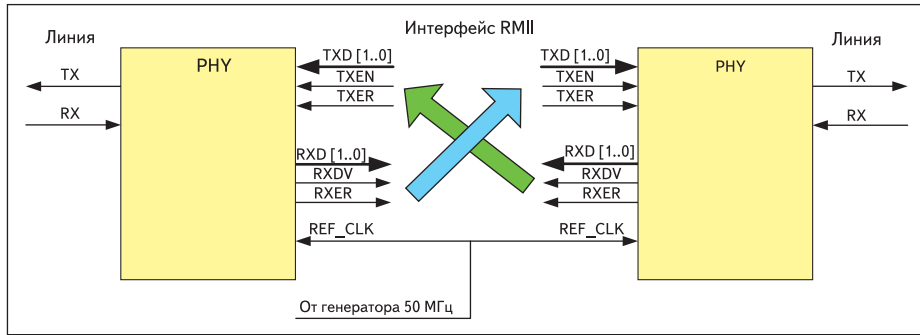


Рис. 2. Пример использования двух Ethernet-трансиверов как ретранслятора

И теперь несколько слов о технологии POE. Питание для трансиверов ретрансляторов, а, возможно, и для приемника, находящегося на дальней стороне, может подаваться по тем же самым проводам, по которым передается и информация. Пример POE показан на рис. 3.

**Немного о POE**

В настоящее время технология питания по кабелю Ethernet стандартизована в IEEE 802.3af. Согласно спецификации стандарта, напряжение питания, передаваемого по кабелю (рис. 3), лежит в пределах 48-В «телекоммуникационного» диапазона, уровень потребляемой абонентом мощности — до 13 Вт.

Что делать, если мы хотим передавать данные не на сто метров, а гораздо дальше? Тогда нужно добавить в тракт приема/передачи оптические преобразователи. Теперь во-

прос о расстоянии будет определяться в большей степени тем, какое оптоволокно мы взяли. Заметим, что рост продаж дешевого пластикового оптоволокна позволяет получить достаточно недорогие и эффективные решения. Поскольку преобразователи интерфейса и ретрансляторы выполняются на одних и тех же микросхемах, то, следовательно, можно сделать и комбинированные устройства — преобразователи с медной линии передачи на оптическую. Это позволит выполнять комбинированную линию, в которой часть сегментов будет выполнена на медных, а другая часть сегментов — на оптических кабелях. Итак, если в выносном блоке прибора у нас стоит маломощный микроконтроллер, то совсем необязательно выполнять в нем сложную программную надстройку. Вполне достаточно взять выход этого микроконтроллера, например, SPI, и подать его на транс-

вер. Если тактовая частота микроконтроллера близка к 25 или 50 МГц, то синхрочастота для SPI и МП может быть общей. А в том случае, когда тактовая частота микроконтроллера значительно ниже, выходные сигналы микроконтроллера могут быть простробированы тактовой частотой МП.

Теперь решим другую проблему. Целесообразно ли будет применять трансиверы, если мы хотим использовать передачу данных только локально внутри прибора, и при этом гальваническая развязка нам не нужна? Конечно, да! Убираем трансформаторы и вместо них ставим конденсаторы, чтобы произвести согласование уровней сигналов. Примеры применения трансиверов без гальванической развязки показаны на рис. 4.

То, что мы рассматривали до сих пор, — это передача данных «точка-точка». Сейчас же мы рассмотрим такой вариант передачи данных, когда данные надо передавать от одного абонента внутриприборного интерфейса всем другим абонентам, или когда мы хотим выполнять несколько передач данных типа «точка-точка» одновременно для двух или более пар абонентов.

**Структура интерфейса. Описание и сравнительные характеристики интерфейсов микроконтроллеров**

Представим себе, что мы имеем прибор, содержащий, допустим, 5–10 плат. На каждой из этих плат находится свой собственный ми-

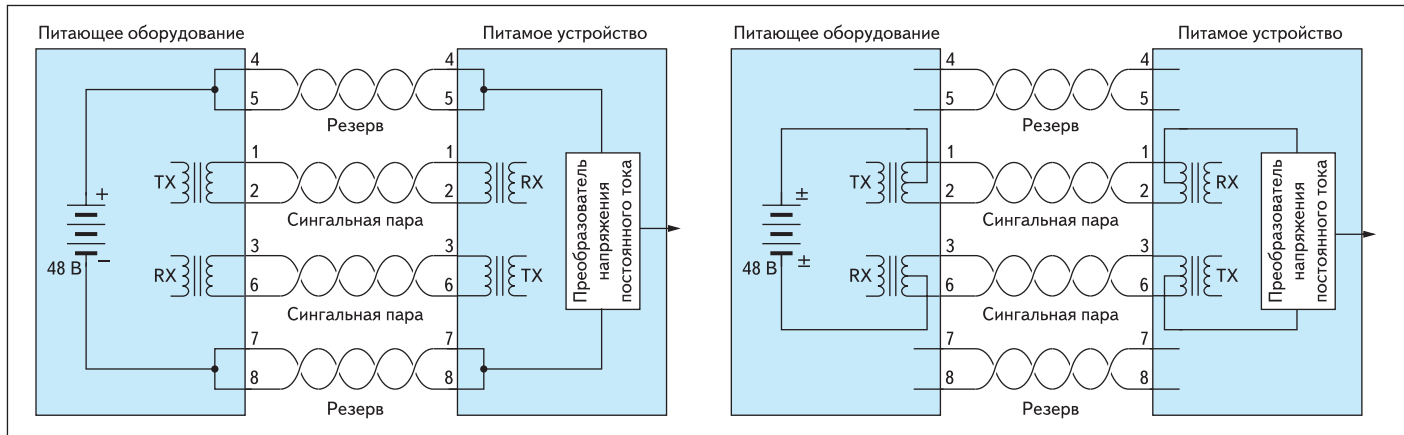


Рис. 3. Примеры передачи питания для ретранслятора и приемника по Ethernet-кабелю

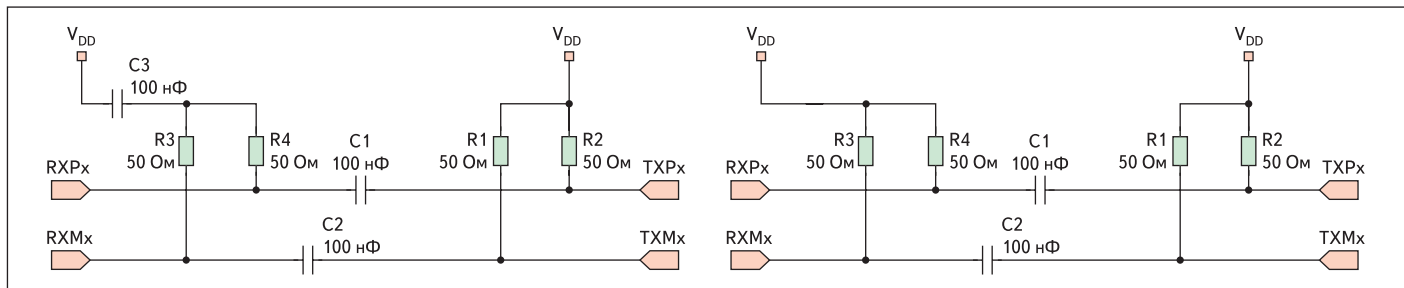


Рис. 4. Примеры использования Ethernet-трансиверов для передачи данных без гальванической развязки

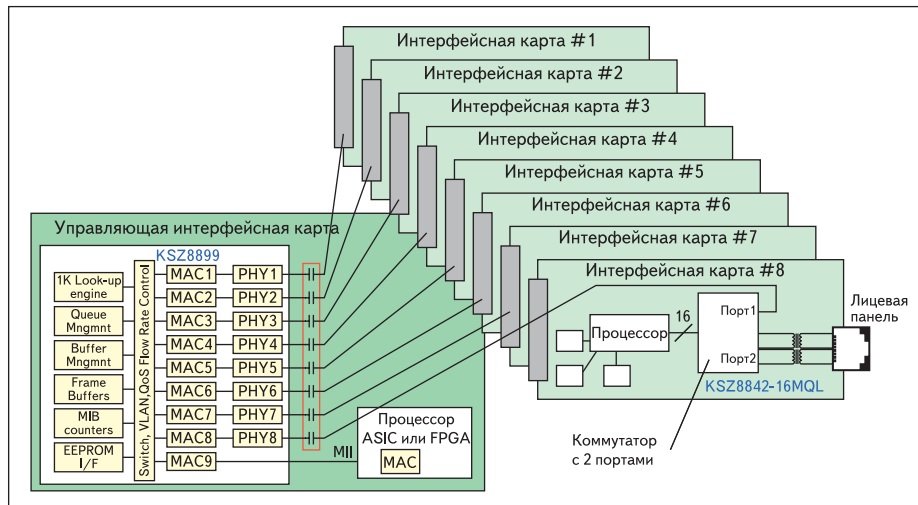


Рис. 5. Пример внутриприборного интерфейса, организованного по обычной сетевой структуре (в центре — коммутатор, он связан с абонентами сети)

кроконтроллер, занимающийся предварительной обработкой данных. Как их связать?

Как было сказано ранее, для связи модулей внутри прибора целесообразно применить последовательный интерфейс. И желательно, чтобы этот интерфейс был стандартный. Кроме того, желательно не делать на уровне модуля преобразователь из параллельного интерфейса на плате в последовательный интерфейс, связывающий эти платы между собой.

Начнем рассмотрение «со стороны» микроконтроллера, находящегося на плате. Если рассмотреть стандартные последовательные интерфейсы, которыми оснащены современные микроконтроллеры, то вот что есть у нас в распоряжении: USART (UART), SPI, I<sup>2</sup>C и т. п., но и не более. Порты CAN, MII, как и порты Ethernet-PHY, хотя и появились у некоторых типов микроконтроллеров, но все же они имеются лишь у некоторых моделей и только у отдельных производителей. Интерфейсы типа USART (UART) напрямую применять неудобно, так как это байт-ориентированные интерфейсы. И хотя есть производители, аппаратно поддерживающие разновидность этого интерфейса, — LIN, все же применение данного интерфейса нецелесообразно: передавать данные по нему медленно и сложно.

Интерфейс I<sup>2</sup>C также применяется только для достаточно медленных устройств, поэтому здесь мы его тоже рассматривать не будем.

Есть еще один интерфейс — SPI. Это самый быстрый последовательный интерфейс, которым обладают микроконтроллеры, но у него нет аппаратной поддержки для эффективной организации внутриприборного интерфейса. Для того чтобы сделать внешние арбитры и коммутаторы, требуется достаточно много ресурсов при крайне низкой эффективности.

Что касается CAN, то это вполне быстрый интерфейс. И его рассмотрение — это отдель-

ная тема. У этого интерфейса все довольно хорошо, кроме одного: не существует таких микросхем, как свичи, хабы и т. д. Поэтому мы его сейчас обсуждать не будем, а перейдем к рассмотрению следующего «кандидата». Но предварительно отметим, что из всех рассмотренных интерфейсов, имеющихся у микроконтроллеров, «второе место» будет занимать SPI — как самый быстрый и самый распространенный интерфейс.

А теперь давайте в качестве кандидата «номер один» на внутриприборный интерфейс рассмотрим интерфейс Ethernet. Предложение, конечно, не новое, но для большинства читателей, скорее всего, будет необычным. Ведь до сих пор все разработчики были приучены к тому, что «все эти Ethernet'ы, TCP/IP и многое другое» связаны только с большой головной болью...

### Реализация внутриприборного интерфейса

#### Первый шаг — интерфейс Ethernet

Итак, давайте выберем для внутриприборного интерфейса Ethernet. Что привлекательного в таком решении? Конечно, это отработанная номенклатура микросхем для аппаратной поддержки интерфейса. Это трансиверы, контроллеры и, конечно, свичи или коммутаторы. Поставляются микросхемы как для коммерческого, так и для промышленного диапазонов.

Для начала представим себе самый простой вариант: будем передавать данные, «запакованные» в стандартные кадры только уровня Ethernet. Это позволит нам применить микросхемы-коммутаторы, поскольку они работают с данными, находящимися в адресных полях кадра, и, кроме того, коммутаторы проверяют контрольные суммы кадров. Скорости передачи данных мы получим, соответственно, 10 или 100 Мбит/с.

Представим, что мы выполнили прибор по обычной сетевой структуре. В центре — коммутатор, он связан с абонентами сети (рис. 5).

Какие возможности мы имеем в таком варианте? При использовании уникальных адресов пакеты будут передаваться только от одного абонента к другому. При использовании широковещательных адресов пакеты будут передаваться от одного абонента ко всем другим абонентам. Мы можем применить неуправляемый или управляемый коммутатор. Неуправляемый коммутатор пересылает пакеты только так, «как сказано» в заголовках пакетов. Применив управляемый коммутатор, мы сможем запрещать или разрешать передавать данные на какие-то направления, копировать пакеты, предназначенные одному абоненту, и пересылать их другому абоненту.

Далее. Коммутаторы могут быть включены каскадно. При этом число абонентов соответственно увеличивается. Точно так же при этом увеличиваются возможности каскадирования. Более того, прибор может быть выполнен по распределенной структуре. Мы помним, что для медного провода длина кабеля может достигать 130 метров. И мы уже рассмотрели то, как можно сделать ретранслятор (что позволит нам существенно увеличить расстояние), а также случаи применения оптических преобразователей. Таким образом, прибор из настоящего без особых проблем «превращается» в распределенный. И это же решение удобно и для стоечного исполнения приборов.

#### Аппаратная реализация модулей.

##### Примеры аппаратной реализации интерфейса

Итак, в центре сетевого интерфейса будут находиться коммутаторы. А что будет находиться на модулях? Мы знаем, что уже появились микроконтроллеры со встроенными портами Ethernet. Например, микроконтроллеры от Freescale, такие как MCF5282, MCF52232. Или можно взять микроконтроллеры только с портом MAC и добавить к ним трансивер, например, микроконтроллер от ADI — BF537. Ну, а если мы хотим использовать какой-либо другой микроконтроллер, в котором нет этих портов? Тогда нам нужен «отдельно стоящий» контроллер Ethernet! Контроллер — как сетевая карта для PC, и он тоже выполнен на одной микросхеме. Но, в отличие от сетевой карты, где применяются микросхемы только с интерфейсным выходом на шину PCI, мы для нашей платы можем выбрать микросхему, имеющую выход на параллельную шину общего назначения. Мало того, теперь наши возможности значительно возросли с появлением микросхемы KCSZ8851SNL [2], которая имеет интерфейс SPI для связи с хостом. Это очень удобно, поскольку нет необходимости применять микроконтроллер в корпусе с большим числом выводов. И еще раз хочется напомнить, что мы рассматриваем микросхемы для Ethernet 10/100.

## Краткое описание микросхем для реализации интерфейса

### Микросхемы трансиверов

Фирма Micrel, занимающая лидирующее положение в промышленности по производству микросхем для Ethernet, объявила о начале поставок микросхем KSZ8041TL, KSZ8041NL и KSZ8041FTL, которые являются новейшими Ethernet 10/100 PHY-трансиверами и полностью соответствуют спецификации IEEE 802.3u. Они имеют такие дополнительные функции, как автопереключение MDI/MDI-X по Hewlett-Packard, могут быть подключены напрямую одна к другой, чтобы работать в режиме ретранслятора или преобразователя с медного кабеля на оптоволоконный. Кроме того, как и все новые микросхемы, выпускаемые этой компанией, они имеют режим для кабельной диагностики LinkMD, что позволяет выявить не только обрывы в кабеле, но даже изменения его импеданса. Микросхемы изготавливаются в компактном корпусе TQFP с 48 входами, размеры корпуса — 9×9 мм. Микросхемы KSZ8041TL/FTL имеют самую низкую потребляемую мощность — всего 180 мВт. Они выпускаются в коммерческом и промышленном исполнении.

Трансивер KSZ8041TL поддерживает режим 10Base-T и 100Base-TX при работе на медный провод. Микросхема KSZ8041FTL поддерживает еще и 100-Base-FX и может быть объединена с KSZ8041NL или с KSZ8041TL так, что эти микросхемы будут подключены напрямую одна к другой по интерфейсу MII, что позволяет сформировать очень дешевое решение конвертера медь-оптика. Микросхемы KSZ8041TL/FTL предлагают различные варианты для подключений интерфейса к узлу MAC процессора — MII, RMII, или SMII.

Кроме двух перечисленных трансиверов есть еще один — KSZ8041NL. Он имеет все характеристики, идентичные KSZ8041TL, кроме одной. Микросхема выпускается в ультракомпактном 32-выводном корпусе QFN (5×5 мм).

### Стартовые наборы для трансиверов

Рассмотрим стартовый набор на более ранний трансивер — KSZ8001 (рис. 6). Он состоит из двух микросхем — собственно KSZ8001

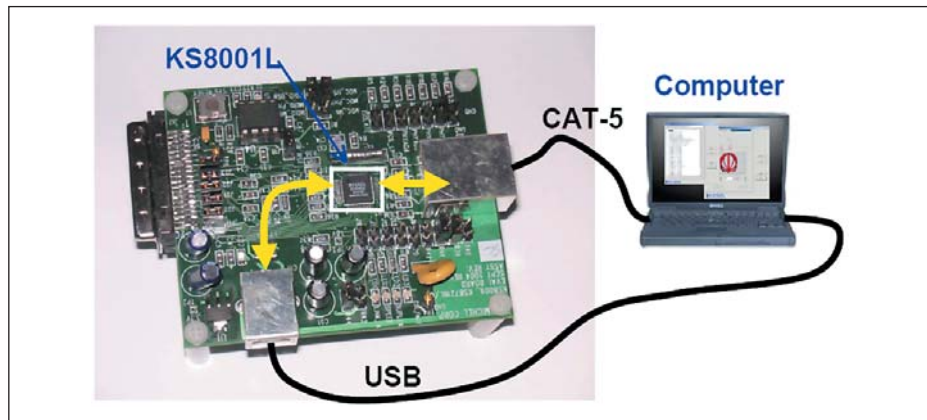


Рис. 6. Стартовый набор на трансивер KSZ8001. При помощи программного обеспечения, работающего на хосте, пользователь имеет возможность передавать данные между USB- и Ethernet-портами компьютера

и микросхемы, преобразующий интерфейс USB в интерфейс MII. Данные из MII передаются в трансивер. Пользователь имеет возможность подключить к этому набору хост-компьютер, одним кабелем к разъему USB, другим — к Ethernet. Тестовое программное обеспечение, входящее в состав набора, позволяет передавать пакеты данных от одного интерфейса в другой. Этот стартовый набор можно использовать те разработчики, которые хотели бы сделать аналогичный преобразователь интерфейсов. Кроме приема/передачи данных, этот стартовый набор позволяет проводить диагностику кабеля Ethernet в режиме LinkMD. Пользователь может увидеть на экране компьютера сообщения об аварии в кабеле и расстояние до места, где возникла неисправность. Если прибор, который вы будете разрабатывать, нуждается в такой диагностике, то для этого в трансиверах фирмы Micrel «уже все готово».

### Микросхемы контроллеров

Микросхемы контроллеров KSZ8851 представляют собой новейшую разработку фирмы Micrel. Кратко рассмотрим их основные преимущества. Что же касается того, «почему и зачем я выбираю контроллер», то здесь можно обратиться к материалам, находящимся на сайте [2], где в разделе, посвященном предыдущим изделиям фирмы Micrel — контроллерам KSZ8841/2, подробно рассмотре-

ны достоинства и недостатки схемного решения с контроллерами Ethernet.

Характеристики новых контроллеров KSZ8851MQL и KSZ8851MLL в основном близки к характеристикам микросхем KSZ8841/2, конечно, за исключением добавленных новых функций (рис. 7), а характеристики контроллера KSZ8851SNL несколько отличаются от предыдущих микросхем, так как эта микросхема имеет последовательный интерфейс по шине обмена данными с хостом (рис. 8, 9). Поэтому дальнейшее рассмотрение характеристик контроллеров KSZ8851 проводится на примере микросхемы KSZ8851SNL.

Корпус микросхемы контроллера KSZ8851SNL — 32-выводный корпус MLF (5×5 мм).

Особенности:

- Интегрированный на кристалле контроллер Ethernet, MAC и PHY, полностью соответствующий стандарту IEEE.802/3.802.3u.
- Интерфейс SPI с тактовыми частотами до 50 МГц, для высокой пропускной способности по данным.
- Поддерживаются режимы 10BASE-T/100BASE-TX.
- Поддерживаются режимы — полный дуплекс, полудуплекс, управление потоком данных при столкновениях и режим протитового давления.
- Поддерживается RXQ и TXQ FIFO DMA для быстрой передачи данных при чтении и записи.



Рис. 7. Блок-схема контроллеров KSZ8851MQL и KSZ8851MLL

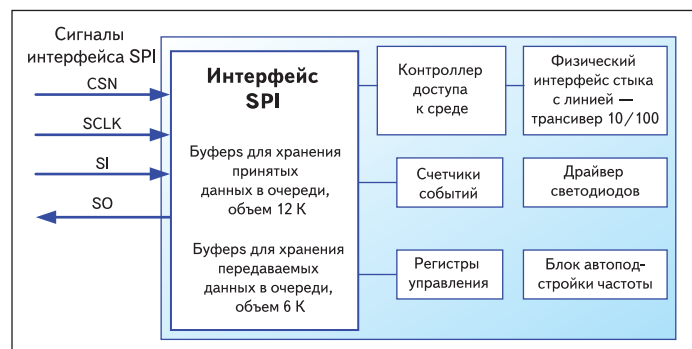
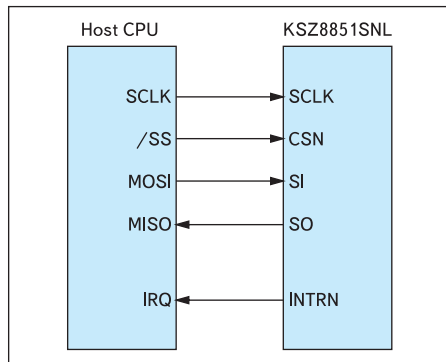


Рис. 8. Блок-схема контроллера KSZ8851SNL



**Рис. 9.** Схема подключения микросхемы KSZ8851SNL к хосту по последовательному интерфейсу SPI: CSN, SCLK, SI, SO — сигналы интерфейса SPI микросхемы KSZ8851; SCLK, SS, MOSI, MISO — сигналы интерфейса SPI хост-процессора; INTRN — выход сигнала прерывания микросхемы KSZ8851; INTRN — вход запроса прерывания хост-процессора

- Поддерживается генерация контрольной суммы и проверка контрольной суммы заголовка для IP (IPv4)/TCP/UDP/ICMP.
  - Поддерживается генерация контрольной суммы и проверка контрольной суммы заголовка для IPv6 TCP/UDP/ICMP.
  - Автоматическая генерация и проверка 32-разрядной CRC.
  - Поддерживается простая система передачи команд и данных в цикле SPI для RXQ/TXQ FIFO и в регистры.
  - В TXQ FIFO и RXQ FIFO может быть помещено несколько кадров данных без дополнительных команд.
  - Поддерживает гибкий режим доступа для чтения/записи ко внутренним регистрам — Byte (8-битовый), Word (16-разрядное) и Double word (32-разрядное).
  - Большая внутренняя память — 12 кбайт для RX FIFO и 6 кбайт для TX FIFO. Несколько режимов по управлению потоком данных для RX FIFO.
  - Конфигурируемая схема прерываний от хоста позволяет уменьшить загрузку процессора хоста.
  - Мощная и гибкая схема фильтрации адресов.
  - Для задания адреса MAC можно дополнительно использовать внешнюю микросхему EEPROM с последовательным интерфейсом.
  - Единственная опорная синхрочастота — 25 МГц и для PHY, и для MAC.
- Режимы потребления мощности, источника питания и корпус:
- Единственный источник питания 3,3 В с возможностью установки напряжения VDD — питание выводов ввода/вывода 1,8; 2,5 и 3,3 В.
  - Встроенный интегрированный маломощный источник питания — регулятор напряжения, используется для ядра и аналоговых блоков от 3,3 или 2,5 в 1,8 В.

- Расширенные режимы управления питанием позволяют получить очень малое потребление энергии на холостом ходу.
- Светодиодные индикаторы (2 светодиода) могут быть запрограммированы пользователем в следующие режимы — link, «активность», «скорость 10/100».
- Коммерческий температурный диапазон: 0...+70 °C.
- Индустриальный температурный диапазон: -40...+85 °C.
- Выпускается в 32-выводном корпусе MLF (5×5 мм).

В дополнение к функциям коммутатора 2-го уровня микросхемы KSZ8851SNL выполняют следующие функции:

- Выравнивается IP-заголовок по границе двойного слова, для чего есть возможность добавки двух байтов перед заголовком кадра.
- Технология Micrel LinkMD позволяет определить длину кабеля, диагностировать дефектные кабели и определить расстояние до ошибки.
- Функции «Wake-on-LAN» включают в себя Magic Packet, пробуждение по приему кадра, определение состояния линии связи — link и технологию обнаружения энергии сигнала.
- Автопереключение входов на прямой или перекрестный режим по Hewlett-Packard MDI-X с опцией запрета и разрешения.
- Способность передать и получить пакеты до 2000 байтов.

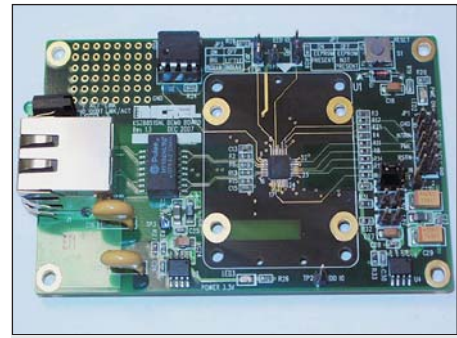
#### Стартовые наборы для контроллеров

На рис. 10, 11 показаны стартовые наборы KSZ8851SNL и KSZ8851MLL. Традиционно представлен полный комплект документации, необходимой для разработчика. Это и схемы, и исходные коды программ, и файлы конструирования печатной платы.

#### Микросхемы коммутаторов

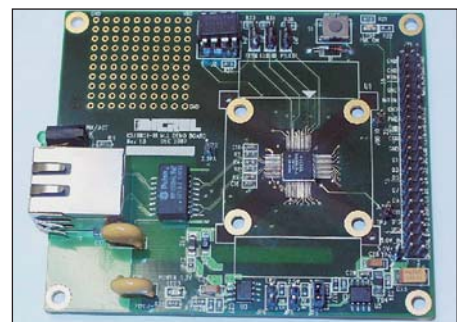
Фирма Micrel производит коммутаторы для Ethernet 10/100 со встроенными PHY — KSZ8695, KS8993, KS8995, KS8997, KS8999. Выпускаются управляемые и неуправляемые коммутаторы на различное число портов: 3-, 5-, 8- и 9-портовые. В том числе и на индустриальный диапазон температур. Коммутаторы имеют несколько вариантов интерфейса для связи с контроллером (МП/PCI/SPI). Отдельно необходимо отметить микросхему KSZ8695, представляющую собой систему на кристалле. В данной микросхеме к 5-портовому коммутатору со встроенными PHY добавлен микроконтроллер с ядром ARM922.

Как работает коммутатор? Неуправляемый коммутатор сам производит свое обучение и запоминает, на каком его порте какой из абонентов находится. Процесс проходит следующим образом. Сервер, управляющий сетью, передает в сеть широкоэвещательный пакет. Все абоненты сети обязаны ответить на этот



**Рис. 10.** Стартовый набор контроллера KSZ8851SNL.

В правой части платы находится разъем для подключения интерфейса SPI



**Рис. 11.** Стартовый набор контроллера KSZ8851MLL.

В правой части платы находится разъем для подключения параллельного интерфейса общего назначения

запрос. Когда абонент отвечает серверу, то он в своем пакете данных, в поле «от кого», подставляет свой MAC-адрес. Вот этот-то адрес и заносится в таблицу поиска адресов для того порта, к которому подключен абонент. Далее, если в коммутатор на любой его порт поступит пакет с уникальным адресом, то в поле «кому» будет стоять MAC-адрес того абонента, которому адресован данный пакет, и все поисковые таблицы в портах коммутатора начнут искать запись о том, что к этому порту подключен абонент с таким адресом. В том случае, если такой записи нет, то в данный порт этот пакет данных не передается. Если же запись с таким адресом есть, то этот пакет передается в данный порт.

Таким образом, неуправляемый коммутатор работает автономно. Режимы работы коммутатора могут быть «защиты» аппаратно или загружаться самой микросхемой коммутатора из внешнего ПЗУ с последовательным доступом.

Неуправляемый коммутатор не имеет возможности перенаправлять пакеты или каким-либо образом менять их содержимое. В отличие от неуправляемого, управляемый коммутатор имеет возможность перенаправлять пакеты, дублировать пакеты на другой порт. Но, как и неуправляемый, не имеет возможности менять содержимое пакетов. Для управления коммутатором обычно используют микроконтроллер. На рис. 12 представле-

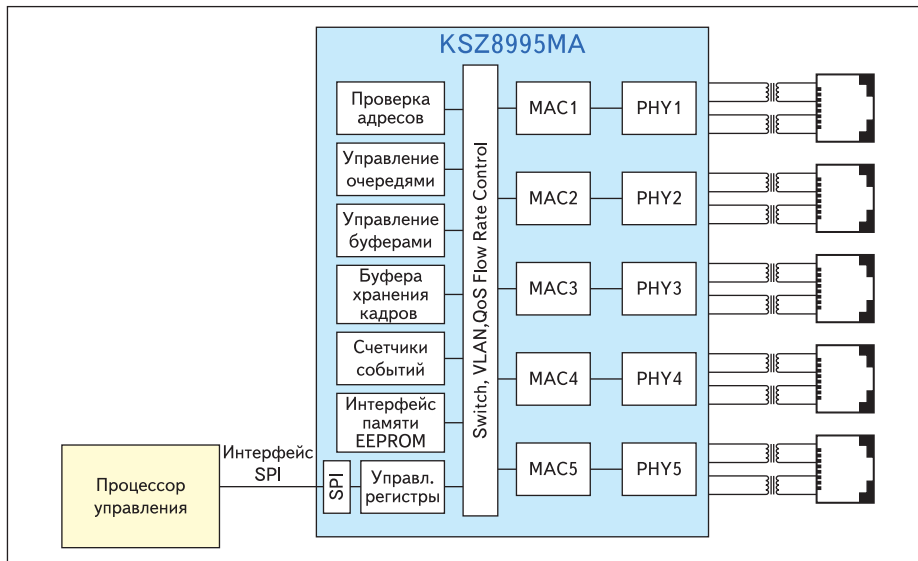


Рис. 12. Блок-схема управляемого коммутатора KSZ8995MA, к которому подключен процессор, управляющий коммутатором

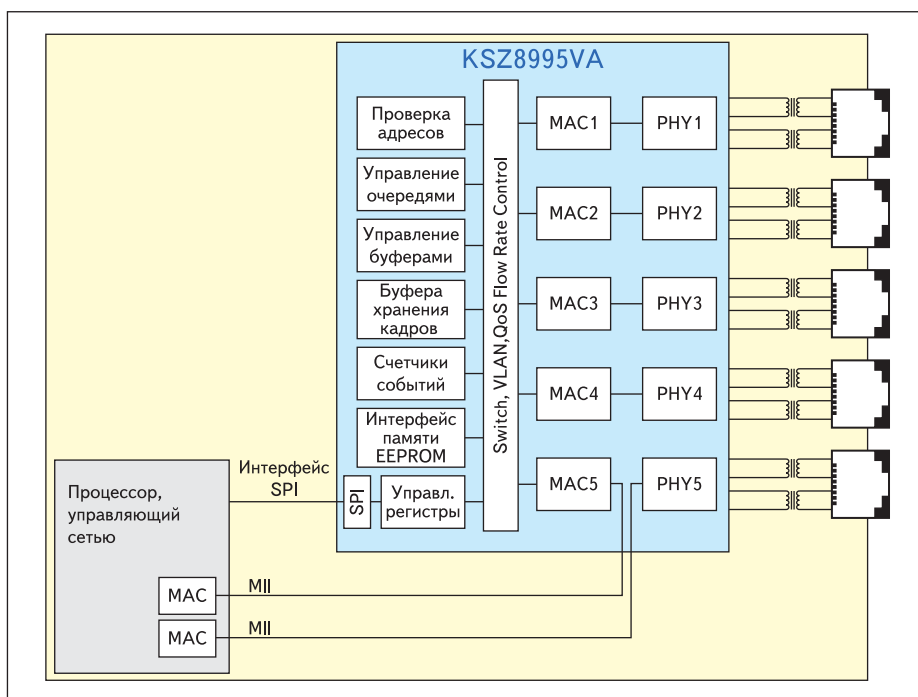


Рис. 13. Блок-схема управляемого коммутатора KSZ8995VA, к которому подключен процессор, управляющий коммутатором и сетью

на блок-схема управляемого коммутатора KSZ8995MA, к которому подключен микроконтроллер, управляющий коммутатором.

Для того чтобы при работе сети не только менять адреса портов приема/передачи коммутатора, но и содержимое пакетов, можно «врезать» дополнительный процессор в один из трактов приема/передачи данных на каком-либо порте коммутатора. Тогда этот процессор сможет получить пакет, обработать его и произвести необходимые манипуляции с данными. На рис. 13 показано такое решение. Но, конечно, кроме этого, есть и другие аналогичные решения.

Например, мы хотим производить обработку большого числа пакетов. И для этого нам одного тракта данных будет недостаточно. Что делать в этом случае? Давайте применим микросхему KSZ8695 [4–6], о которой мы говорили ранее. В данной микросхеме к 5-портовому коммутатору со встроенными PHY добавлен микроконтроллер с ядром ARM922. Но и это еще не все. До сих пор мы говорили о последовательных внутриприборных интерфейсах. Теперь пришла пора обсудить еще и тот случай, когда у нас в приборе есть «нечто» с интерфейсом PCI. Как сделать переходник от интерфейса PCI на после-

довательный внутриприборный интерфейс? Для этого удобно использовать микросхему KSZ8695. Микросхема выпускается в трех модификациях. Микросхемы KSZ8695P и KSZ8695PX имеют соответственно 3 и 1 шину интерфейса PCI, а микросхемы KSZ8695X такой шины не имеют. Причем по шине PCI микросхемы могут выдавать свой тактовый сигнал для абонентов шины PCI, и тогда они работают подобно чипсету материнской платы. Так же можно сконфигурировать микросхему, чтобы она получала тактовый сигнал извне. И тогда микросхема будет работать подобно PCI-карте. Блок-схема KSZ8695P приведена на рис. 14. Дополнительно можно рассмотреть еще одну возможность, заложенную в KSZ8695, — это VLAN (виртуальная локальная сеть).

VLAN обычно используется для того, чтобы разделить всех сетевых пользователей на несколько групп, создать логическую сегментацию рабочих групп и установить защиту информации для каждого логического сегмента сети. Таким образом, разделение сети на VLAN позволяет оградить пользователей LAN от несанкционированного доступа (firewall).

Поскольку KSZ8695P имеет встроенный 5-портовый Ethernet, то его можно рассматривать как 5-портовый переключатель сетевых интерфейсов, с поддержкой как 802.1Q VLAN на основе технологии тегов, так и VLAN, привязанную к определенному порту. Благодаря тому, что пакеты, передаваемые разным сегментам сети, имеют отличительные признаки, сеть LAN можно представить так, как показано на рис. 15.

#### Стартовые наборы для коммутаторов

Для того чтобы оперативно начать работу, можно заказать стартовый набор на требуемый коммутатор. В качестве примера можно привести наиболее распространенный набор на микросхему коммутатора KSZ8995 (рис. 16).

Теперь давайте сделаем еще один шаг «вверх».

#### Второй шаг — интерфейс TCP/IP

В предыдущей главе мы ознакомились с тем, какие преимущества мы получим, если выполним внутриприборный интерфейс на основе интерфейса Ethernet. Как видим, даже без применения интерфейсов протокола передачи данных TCP/IP можно получить довольно удобное и выгодное решение проблем. Ну, а если мы применим и TCP/IP или UDP, то что мы получим дополнительно, и сколько за это надо заплатить?

Без сомнения, нет необходимости рассказывать о бесплатном сыре. Но вот об очень «бюджетном» варианте рассказать не помешает. Представим себе, что на вашем модуле стоит микроконтроллер с установленной на нем операционной системой. И в составе этой операционной системы уже есть драйверы для TCP/IP или UDP. Тогда этот модуль

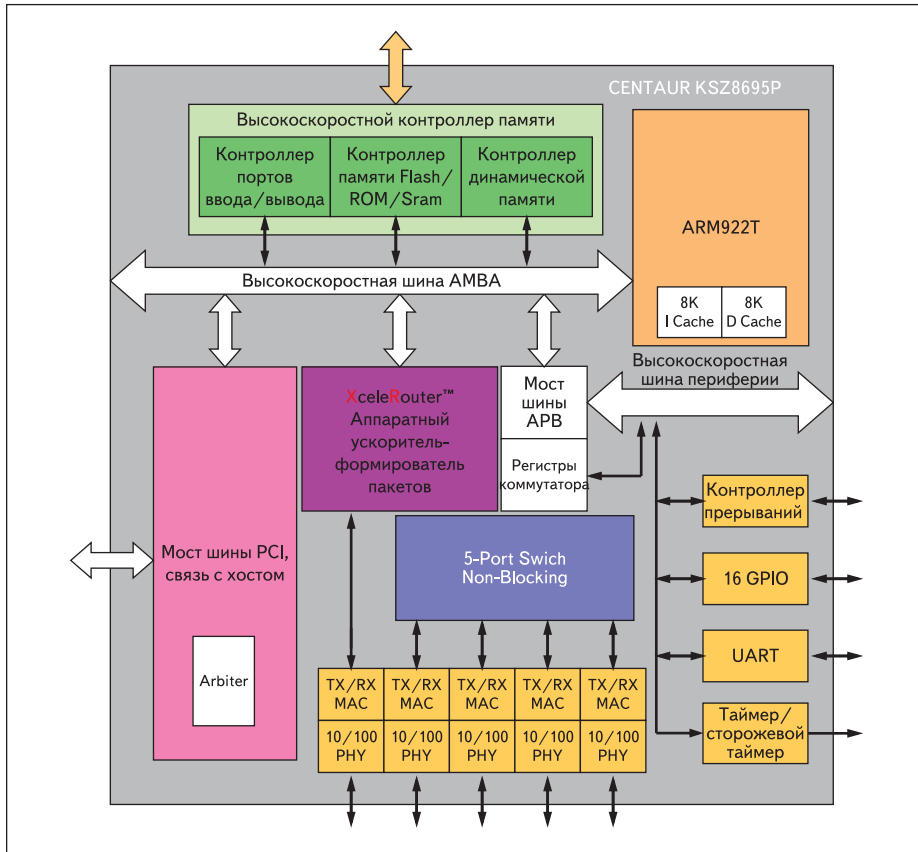


Рис. 14. Блок-схема KSZ8695P и KSZ8695PX

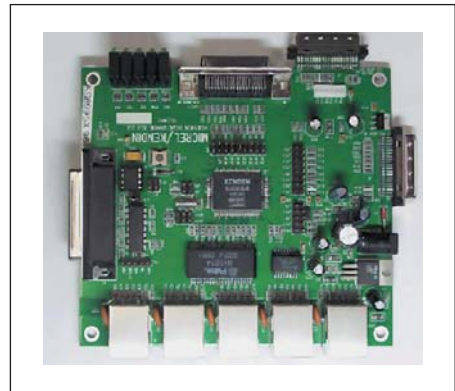


Рис. 16. Стартовый набор на микросхему KSZ8995

ло, если этот модуль находится в составе прибора. Тут список преимуществ будет гораздо больше. Провести удаленное тестирование модуля или всего прибора в целом — без проблем! Дистанционно перепрограммировать или изменить настройки — да, конечно! Но и не только это.

Представьте себе, что вам позвонил клиент и пожаловался на то, что «полчаса назад что-то с прибором было не так». Что будем делать? В браузере набираем web-адрес прибора, вводим пароль супервизора и проверяем файл логирования работы прибора. Скорее всего, в файле логирования обнаружится сбой по сети или что-то еще, вызвавшее сбой в работе прибора. Следовательно, вы сможете сообщить об этом клиенту. Причина будет оперативно найдена, а значит, при дальнейшей работе прибора ее можно устранить. И, как следствие, вы никогда более не услышите от клиентов, что, мол, «прибор-то неплохой, да вот иногда что-то с ним случается».

Хорошо, есть один модуль, в котором стоит операционная система. Но в других модулях ее может и не быть? Конечно, часть модулей может и не иметь возможности напрямую подключаться к сети. Но главное, что есть хотя бы один модуль, который может передать в сеть информацию о состоянии других модулей. Поэтому, даже если часть вашего прибора использует стандартный TCP/IP, то это уже дает прибору дополнительные преимущества.

А что еще можно получить, используя в приборе стандартный TCP/IP? До сих пор, говоря о «распределенном» приборе, мы имели в виду, что часть его будет выполнена отдельно и подключена к основному блоку посредством закрытого протокола. Если же мы применим открытый стандартный протокол, то это позволит нам сделать действительно распределенный прибор. Например, мы хотим делать какую-либо обработку результатов измерений дистанционно. Или обработку с сохранением. Или мы хотим подключить к прибору дистанционные терминалы. И этот список может быть увеличен. Напомним об агрегатировании приборов. Допустим, мы хотим применить на одном объекте несколь-

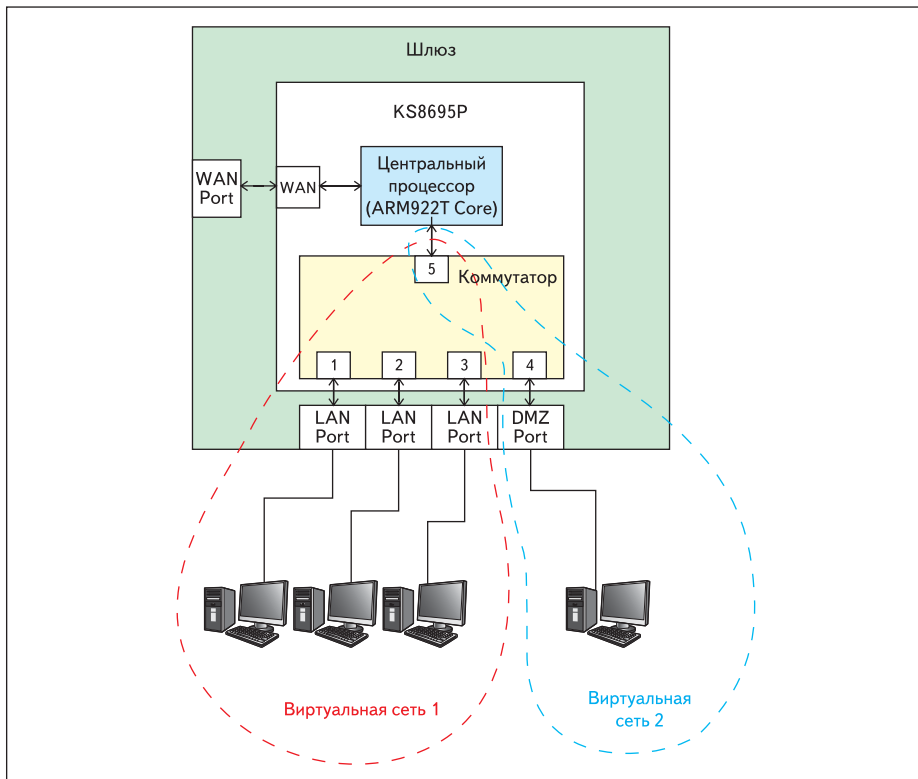


Рис. 15. Разделение локальной сети на сегменты по технологии VLAN

становится доступен по сети, как при работе внутри прибора, так и при автономной работе вне его. Что это дает? При автономной ра-

боте вы можете отлаживать модуль, который находится где угодно. Это небольшие преимущества при отладке, но все же. Другое де-



ко устройств, причем алгоритм их работы будут зависеть от результатов обработки данных во всех этих приборах. И далее, и самим приборам, и оператору, который с ними работает, становится абсолютно неважно, что один из них получает данные по спутниковому каналу связи, а другой — по оптоволоконному. Стандартные каналы связи, стандартные хабы, свичи и т. д.

### Первые шаги

Всем известно, как трудно даются первые шаги. Что нужно в первую очередь? Конечно, помощь в выборе стратегического направления. И для этого следует просто зайти на сайт компании Micrel и посмотреть размещенные там материалы или позвонить по телефону, чтобы получить консультацию. Стартовые наборы всегда находятся на складе компании, и поэтому нет необходимости долго ждать их доставки в Россию. Образцы могут быть оперативно предоставлены по запросу клиента. И это позволит перейти к разработке практически сразу же после принятия решения о выборе направления. И, наконец, склад компании позволяет ускорить разработку, поскольку микросхемы с него че-

рез региональный офис в запланированное время поступают к клиенту. Это исключает задержки в поставке компонентов.

Все это вместе дает возможность сделать первые и последующие шаги совершенно безболезненно, и, соответственно, ускоряет и упрощает вашу разработку.

### Заключение

Применение микросхем фирмы Micrel позволяет значительно упростить аппаратную и программную реализацию интерфейса. При наличии дополнительной вычислительной мощности появляется возможность выполнять «прозрачные» приборы и использовать стандартные каналы связи с ними. ■

### Литература

1. [www.micrel.com](http://www.micrel.com)
2. [www.eltech.spb.ru/micrel\\_ethernet.html](http://www.eltech.spb.ru/micrel_ethernet.html)
3. [www.freescale.com](http://www.freescale.com)
4. KS8695P DS v1.1.pdf — CENTAUR KS8695P Data Sheet Integrated Multi-Port PCI Gateway Solution, [www.micrel.com](http://www.micrel.com)
5. <http://linuxdevices.com/news/NS2611150039.html>
6. <http://www.arm.com/products/CPUs/ARM922T.html>