



## МЕЖСЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ В ПРОМЫШЛЕННЫХ СЕТЯХ

А.А. Сахнюк, ХОЛИТ Дэйта Системс", г.Киев

**Д**ля обеспечения работы в одной сети информационных и технологических систем, выпускаемых различными производителями, используется Модель Взаимодействия Открытых Систем (OSI), разработанная Международной Организацией Стандартов (ISO) в 1979 г., которая координирует:

- взаимодействие прикладных процессов;
- формы представления и хранения данных;
- управление сетевыми ресурсами;
- безопасность данных и защиту информации;
- диагностику программ и технических средств.

На базе этой модели описываются правила и процедуры передачи данных между открытыми системами. Основными элементами модели являются уровни, объекты, соединения, физические средства соединения. Модель стала основой для разработки широкого комплекса международных стандартов открытых систем.

В модели OSI выделяют семь уровней информационного взаимодействия:

- 7 прикладной уровень: передача информации между программами;
- 6 уровень представления: шифрование, кодирование и сжатие данных;
- 5 сеансовый уровень: установка, поддержка и разрыв соединения;
- 4 транспортный уровень: точность доставки, уровень качества услуг;
- 3 сетевой уровень: маршруты передачи, обработка и передача сообщений;
- 2 канальный уровень: управление каналом связи, доступ к среде передачи и адресация;
- 1 физический уровень: связь на уровне аппаратуры.

Уровни не зависят друг от друга и состоят из активных объектов, которые:

- взаимодействуют с другими объектами того же уровня;

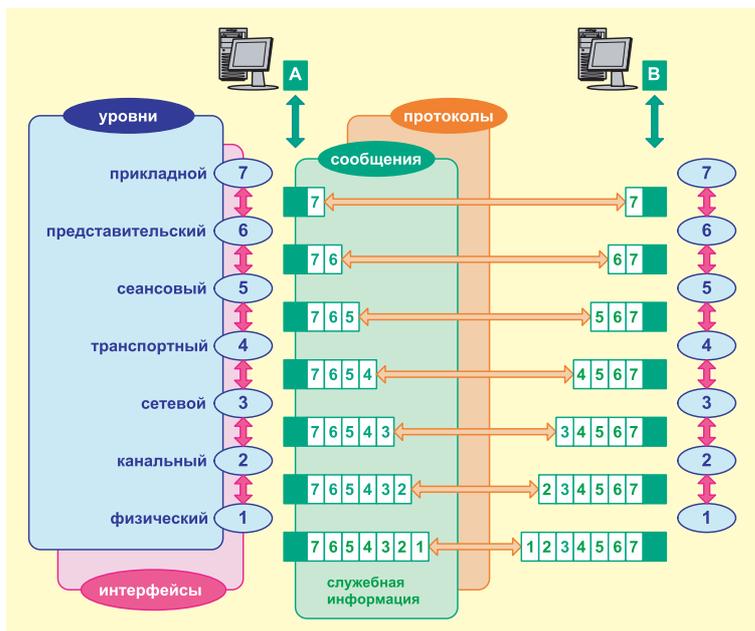
- предоставляют сервис смежному с ним верхнему уровню;
- получают сервис от смежного с ним нижнего уровня;
- обмениваются блоками данных с целью выполнения возложенных на них задач.

Процедура взаимодействия двух узлов может быть описана в виде набора правил для каждой пары соответствующих уровней обеих участвующих сторон. Формализованные правила, определяющие формат и последовательность сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне, но в разных узлах, называются протоколом.

Модули, реализующие протоколы соседних уровней и находящиеся в одном узле, также взаимодействуют друг с другом в соответствии с четко определенными правилами с помощью стандартизованных форматов сообщений. Эти правила называют интерфейсом. Интерфейс определяет последовательность и формат сообщений, а также и набор предоставляемых услуг, которыми обмениваются сетевые компоненты, лежащие на соседних уровнях в одном узле.

Иерархически организованный набор протоколов, достаточный для взаимодействия узлов в сети, называется стеком коммуникационных протоколов. Коммуникационные протоколы могут быть реализованы как программно, так и аппаратно. Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средств, а протоколы верхних уровней, как правило, программными средствами. Протоколы реализуются не только узлами сети, но и другими сетевыми устройствами - концентраторами, мостами, коммутаторами, маршрутизаторами и шлюзами.

Самые простые сети используют только некоторые из уровней этой семиуровневой модели. Первый уровень - самый нижний и представляет



физическую линию и электрические сигналы, передаваемые по ней. Седьмой уровень - пользовательский, он позволяет приложениям и различным устройствам обмениваться данными и понимать друг друга. Примером может служить иностранная речь: для понимания смысла сказанного необходимо знать язык говорящего. Устройства, которые используют общий 7-й уровень, говорят на одном языке, могут свободно обмениваться данными и правильно их интерпретировать.

Обычный RS-232 определяет только 1-й уровень, подразумевая, что два устройства могут быть соединены без электрических конфликтов. Для передачи и интерпретации данных в каждое устройство необходимо добавить дополнительное программное обеспечение, поддерживающее протокол обмена. Оно определит правила соединения и форматы данных.

Для работы RS-485 необходимо дополнительно определение канального уровня, который задает способ доступа к физической среде и переключение направления работы приёмопередатчиков устройств, входящих на интерфейсную линию (симплексный или полудуплексный режимы). Для RS-485 канальный уровень определяет дополнительную физическую среду (витую пару), позволяющую передавать сигналы в дуплексном режиме.

Популярный CAN определяет соединения только по двум первым уровням модели OSI. Поэтому, например, внутри автомобиля разработчики используют CAN в качестве коммуникационной связи и дописывают поверх него свой протокол 7-го уровня, поддерживающий взаимодействие и понимание между узлами автомобиля. Любой другой узел с интерфейсом CAN вне автомобиля не сможет участвовать во взаимодействии, пока для него не будет реализован совместимый протокол 7-го уровня.

В реальных производственных условиях возникает необходимость взаимодействия устройств от различных производителей и поэтому существует несколько открытых протоколов 7-го уровня, базирующихся на CAN, например: DeviceNet и CANopen. Два устройства от разных производителей смогут взаимодействовать потому, что используют один и тот же протокол 7-го уровня.

Ethernet и протокол TCP/IP охватывают 1-4 уровни модели OSI, которые определяют больше функций и свойств чем технология CAN. Два узла

Ethernet могут взаимодействовать друг с другом, но только протокол TCP/IP обеспечивает транспортный механизм для передачи данных и предоставляет такие услуги как установка соединения и поддержка повторной передачи в случае ошибки. Однако два узла Ethernet могут только переговариваться, но они не понимают друг друга поскольку не используют один и тот же "язык", т.к. уровень 7 не определен протоколом TCP/IP. Поверх TCP/IP должны работать приложения вроде SNMP, HTTP, FTP для обмена и обработки данных из сети. Эти протоколы являются протоколами 7-го уровня. В промышленной сети имеются дополнительные протоколы 7-го уровня, поддерживающие обмен данными в реальном времени, профили устройств и т.п. Как пример, можно привести Ethernet/IP (IP=industrial protocol) и Modbus-IDA (Interface for Distributed Automation).

Рассмотрим, как конвертировать сигналы из одной сети в другую, учитывая все вышесказанное.

### Преобразователи и повторители интерфейсов

Преобразователи интерфейсов обычно работают на 1-м и 2-м уровнях модели OSI. Все что они делают, так это преобразовывают электрические сигналы из одной физической среды в другую. Например, преобразователь RS-232 в RS-422 только конвертирует и переформатирует электрические сигналы между этими сетями. Другой пример это Ethernet-преобразователь 100Мбит категории 5 в 100Мбит на оптоволокне. Преобразователи интерфейсов полезны при переходе от одной физической среды к другой, но они только преобразовывают электрические сигналы и не поддерживают никаких других функций.

По мере распространения в сетевом кабеле сигналы деградируют и искажаются. Если кабель достаточно длинный, то за счет ослабления сигнал может стать совершенно нераспознаваемым. Для восстановления сигнала используются повторители. Они работают на физическом уровне и позволяют регенерировать сигнал, а затем пересылать его в другой сегмент сети. Повторители позволяют увеличить макси-

мальное расстояние между узлами сети. Повторители ничего не фильтруют и не преобразовывают. Для того, чтобы повторитель мог нормально работать, соединяемые участки сети должны использовать одинаковый метод доступа к среде передачи. Например, повторитель не может соединить сегмент, использующий CDMA/CD (Ethernet) с сегментом Token Ring, использующим Token Passing.

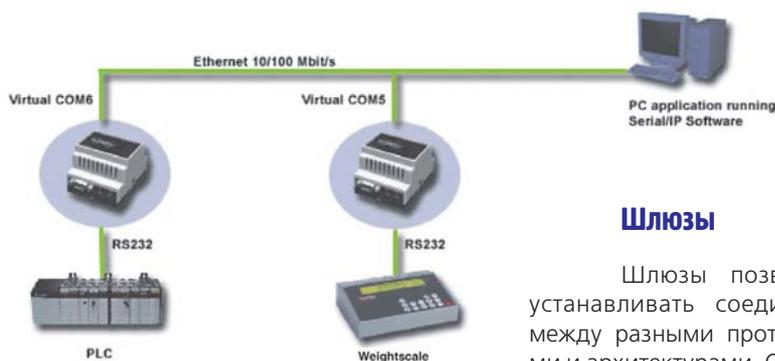
### Коммутаторы

Коммутаторы можно рассматривать как интеллектуальные повторители. Они усиливают сигналы физической сети, а также обеспечивают интеллектуальный анализ полученных пакетов. Коммутатор просматривает и анализирует полученные пакеты данных по всем портам. Прежде всего проверяется контрольная сумма. Если она "не в порядке", то пакет отбрасывается. Если сумма "в порядке", коммутатор проверяет адрес назначения и посылает пакет на порт, содержащий целевой адрес.

### Серверы устройств

Мосты работают на уровнях 1-4 модели OSI. Они устанавливают соединения, обеспечивают ретрансляцию и обработку ошибок. Данные свободно проходят через мост, но они пересылаются, как части анонимных данных. 4-й уровень не имеет собственного механизма обработки данных. Серверы устройств, используемые для перехода между последовательным интерфейсом вроде RS-232 и Ethernet, достаточно популярны. Сервер устройств поддерживает полный стек протоколов TCP/IP, что позволяет инкапсулировать данные с последовательного порта в кадры TCP/IP и транспортировать их через Ethernet. В точке назначения последовательные данные извлека-





### Шлюзы

Шлюзы позволяют устанавливать соединения между разными протоколами и архитектурами. Они переупаковывают и преобразовывают данные, поступающие из одной сети в другую так, что они становятся понятными приложению в другой сети.

Так как шлюз соединяет две системы, которые могут использовать различные коммуникационные протоколы и форматы данных, он может изменять формат сообщения таким образом, чтобы оно соответствовало приложению на приемной стороне. Шлюзы используют полный набор коммуникационной модели OSI от первого до седьмого уровней, что позволяет действительно передавать данные и делать их обработку в различных сетях. Например, шлюз между Ethernet и другими сетями, такими как DeviceNet, Profibus или просто последовательным портом, позволяет отображать данные сети Fieldbus непосредственно на встроенном Web-сервере, производить Web-управление и мониторинг сети.

Тот факт, что последовательный порт может не поддерживать уровень

7 модели OSI, создает дополнительные трудности. Например, считыватель штрих-кодов просто передает данные в ASCII кодах, которые встроены в специальные кадры. Проблему преобразования данных можно решить, применив устройство AnyBus-Communicator (HMS, Швеция), используя выделенную временную память внутри шлюза. Поток данных из последовательного устройства отображается в локальную память шлюза, который с помощью конфигурационного ПО определяет, какие байты данных из потока являются данными, а какие - командной и контрольной информацией. Выбранные данные отображаются на седьмом уровне сети и таким образом могут быть считаны и поняты другим узлом, использующим уровень 7 для обработки данных.

*Рассмотрим несколько примеров:*

1. Считыватель штрих-кода RS-232 в Profibus (Generic Mode/ASCII) без обработки данных. В данном примере сканер передает данные в виде последовательности ASCII-символов. Кадры последовательных данных передаются по интерфейсу RS-232 на вход AnyBus-коммуникатора. Коммуникатор использует стандартный режим для чтения всего кадра данных из последовательного порта и сохраняет их в своей памяти. При неустановленном режиме обработки данных все данные копируются вверх в устройство Profibus-DP Master внутри PLC через канал ввода/вывода Profibus I/O. Эти данные рассматриваются как входные

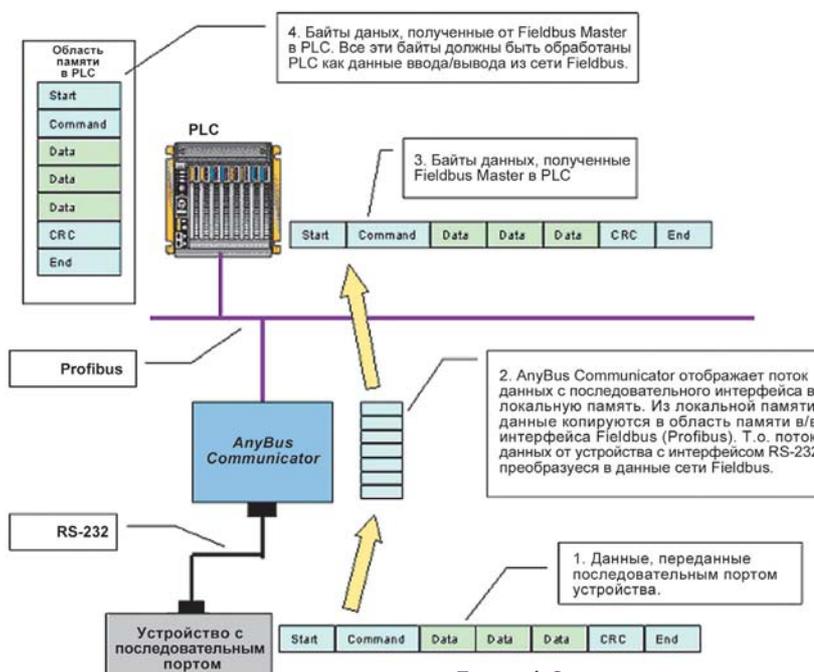
ются из кадра TCP/IP. Это может показаться похожим на работу преобразователя интерфейса, но на самом деле на сервере устройств производится сложная обработка протоколов и временных соотношений передаваемого сигнала.

Кроме того, на компьютере обычно запускается специальное программное обеспечение, создающее виртуальный последовательный порт и поддерживающее извлечение последовательных данных из кадров TCP/IP. Виртуальный локальный порт используется приложением, работающим с последовательными портами компьютера. При этом, как правило, нет необходимости вносить изменения и дополнения в работающее приложение. В большинстве случаев конечный пользователь может не заботиться о том, что прямое соединение по порту заменяется виртуальным через сеть Ethernet. Это позволяет располагать устройства, работающие через COM-порты, на значительном удалении от компьютера.

При работе устройства с PLC-контроллером необходимо использовать два сервера устройств. Один располагается возле устройства с последовательным портом, другой - между Ethernet и последовательным портом PLC для извлечения данных из пакетов TCP/IP.

Серверы устройств хорошо работают в сетях, передающих нерегулярные сообщения. Однако если требуется пересылка и обработка данных в реальном времени следует использовать приложения, работающие на 7-м уровне модели OSI.

Многие сервера устройств имеют статический Web-сервер для собственной конфигурации. Но поскольку 4-й уровень не позволяет обрабатывать данные, оказывается невозможным отображать данные из приложения на встроенном Web-сервере с помощью SSI-скриптов и Java приложений.

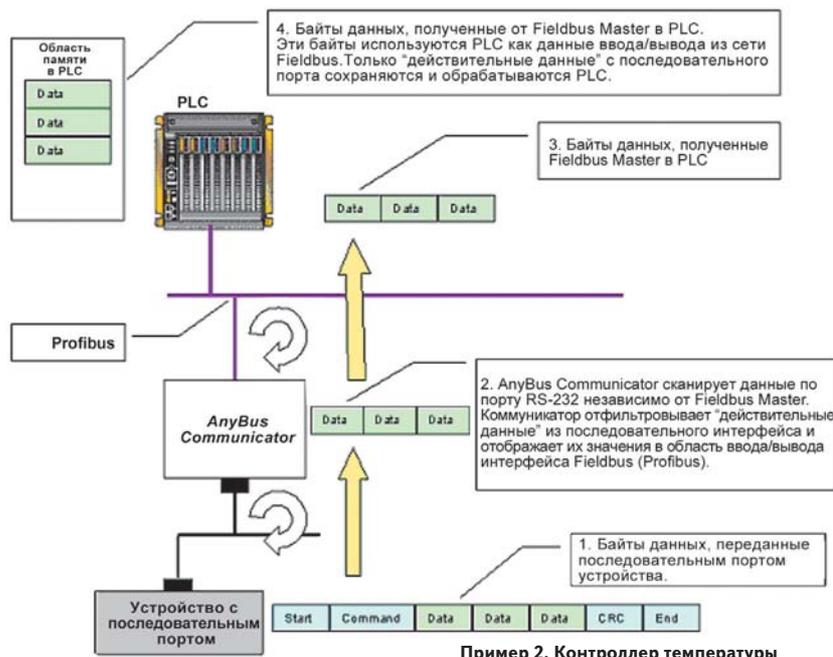


Пример 1. Считыватель штрих-кода

для устройства Master на шине FieldBus. Если вместо Profibus использовать сеть Ethernet, станет возможным считывать данные из локальной памяти коммутатора на встроенный Web-сервер через SSI-скрипты или Java-апплеты. Это сделает сканер доступным через Интернет. Такой режим передачи полезен только для простых последовательных протоколов передачи данных. Тем не менее, вся работа по управлению и тактированию обмена данными должна выполняться программой, работающей внутри PLC. Это задача для системного интегратора. Например, реализация функции проверки контрольной суммы внутри PLC - задача нетривиальная. Большим достоинством предлагаемого решения является то, что последовательное устройство выглядит так, как будто оно содержит встроенную сетевую карту и работает как настоящее сетевое устройство. Таким образом, шлюз конвертирует данные из последовательной сети в формат, понятный PLC с соответствующей шиной Fieldbus, и устройство с последовательным интерфейсом выглядит как любое другое устройство такой сети.

2. Контроллер температуры RS-232 в Profibus (Generic Mode/ASCII) с обработкой и фильтрацией данных. В этом примере коммутатор использует стандартный режим для чтения полного кадра. Данные обрабатываются в соответствии с выбранными функциями и сохраняются в локальной памяти коммутатора. Байты данных, выбранные при настройке в ABC конфигураторе, извлекаются и копируются вверх в ведущее устройство Profibus DP внутри PLC через канал данных Profibus I/O. Эти данные рассматриваются как входные для FieldBus Master. Поток выходных данных FieldBus "течет" в противоположном направлении. Он направлен через шину Profibus вниз, в AnyBus-communicator. Из своей локальной памяти выбирается кадр последовательных данных для канала RS-232 и в него добавляются управляющие данные и значения контрольной суммы. Эта информация используется контроллером температуры. Если вместо Profibus использовать сеть Ethernet, то можно сделать контроллер температуры Интернет-доступным, аналогично рассмотренной выше ситуации.

3. Контроллер управления движением RS-232 в Profibus, режим ModBus Master. Рассмотрим как ком-

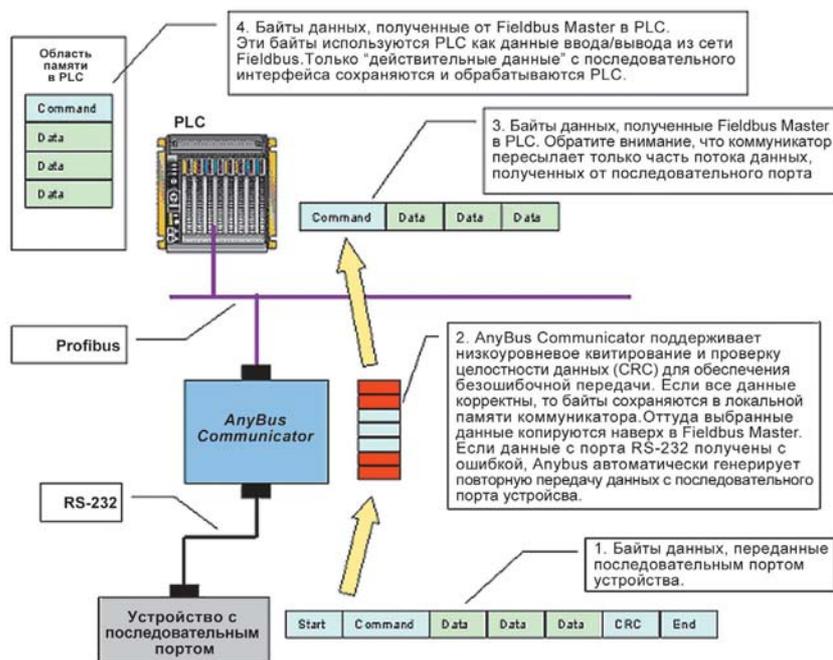


Пример 2. Контроллер температуры

муникатор в режиме Modbus Master работает с контроллером, имеющим последовательный интерфейс с протоколом Modbus RTU. При настройке с помощью ABC-конфигуратора выбирается список сканирования для устройства Modbus Master. Определяются все команды Modbus, регистры, отображение данных и временные соотношения. Коммутатор контролирует связь с контроллером по Modbus и обменивается выбранными данными с интерфейсом Profibus, с которого данные поступают в Fieldbus Master PLC-контроллера.

С точки зрения PLC контроллер движения выглядит так, как будто он встроен в интерфейс Profibus. PLC может игнорировать то, что контроллер на самом деле имеет последовательный интерфейс. При использовании сети Ethernet контроллер движения также легко становится Web-доступным.

В наиболее совершенных режимах AnyBus-коммуникатор может также работать как активный управляющий шлюз, соединяющийся с несколькими последовательными устройствами с использованием общего протокола, например, ModBus RTU, и



Пример 3. Контроллер движения

передающий данные в такие промышленные сети, как ProfiBus или Ethernet.

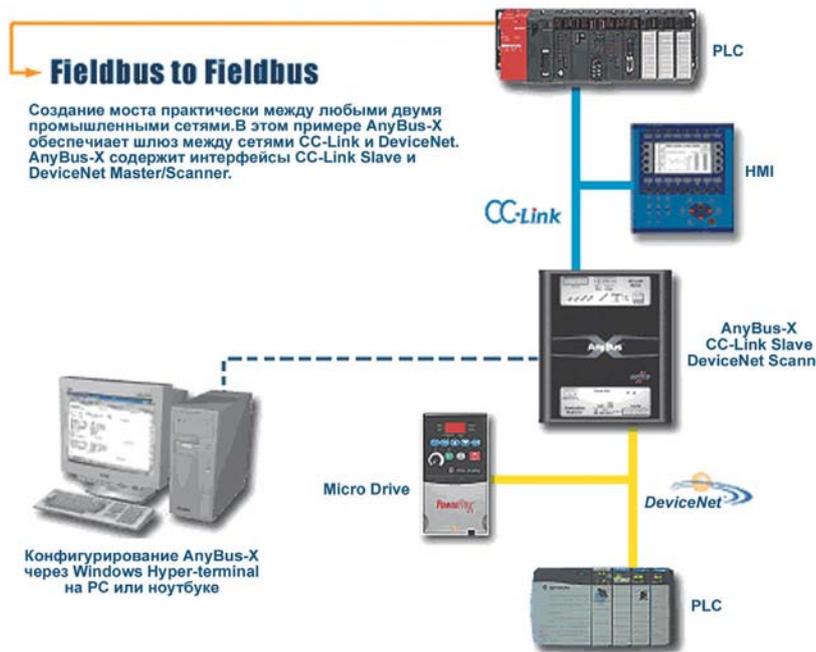
### Мосты

В последнее время все более возрастают требования к гибкости и универсальности оборудования на производственных участках. Индустриальные мосты позволяют связывать старое и новое оборудование, работающее в разных сетях на одном производственном участке или за его пределами. Устройства AnyBus-X позволяют связывать две различные промышленные сети во всевозможных комбинациях.

AnyBus-X - мосты/шлюзы между двумя сетями. Семейство AnyBus-X насчитывает 130 различных изделий. Они поддерживают 15 промышленных сетей таких как: Profibus, DeviceNet, CANopen, CC-Link и др., наряду с протоколами ModBus-TCP и Ethernet/IP. Устройства AnyBus-X облегчают системным интеграторам задачу обмена информацией между различными сетями. Они являются существенной частью любой сетевой инфраструктуры и идеально подходят для дополнения и модернизации имеющейся на предприятии сети.

Работа AnyBus-X основана на передаче периодических данных ввода-вывода между сетями. Универсальность AnyBus-X может быть проиллюстрирована тремя различными концепциями использования шлюза:

- между двумя различными промышленными сетями;



- между промышленной сетью и Ethernet;

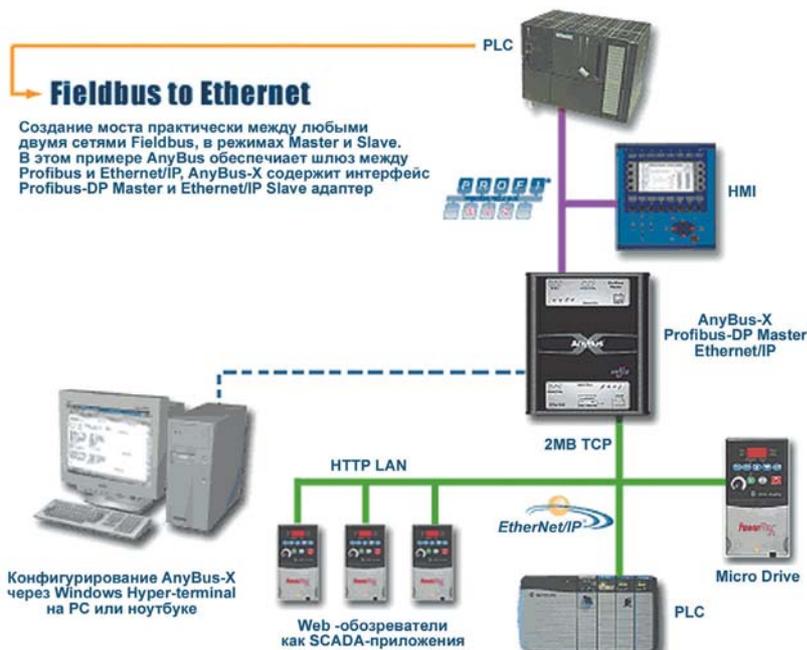
- из промышленной сети в Bluetooth.
- Модуль Bluetooth используется для поддержки беспроводного доступа через стандартный пользовательский Web-интерфейс к промышленному оборудованию и обеспечивает оптимальную функциональность при передаче данных между промышленной сетью и Bluetooth. Дополнительно в него интегрирован Ethernet-интерфейс (витая пара) для поддержки связи с Интернет или локальной сетью.

Web-интерфейс позволяет легко производить управление и конфигурирование оборудования с использованием компьютеров типа notebook или PDA без установки специального клиентского программного обеспечения.

### Конфигурирование AnyBus-X

AnyBus-X был разработан для получения универсальности, гибкости и надежности, но прежде всего для облегчения установки и конфигурирования. Шлюз копирует данные ввода-вывода в двух направлениях между двумя сетями, поддерживая тем самым межсетевой обмен. По умолчанию копируется по 20 байт данных. Изменить установки достаточно легко, что делается при использовании специального конфигурационного порта RS-232 и стандартной терминальной программы, например Hyper Terminal for Windows.

При конфигурировании шлюза пользователь выбирает максимальный объем данных В/В, которые он собирается передать из Сети А в Сеть В. Поскольку различные промышленные сети поддерживают разный объем данных В/В, параметр определяется по меньшему значению. Для стандартных приложений ввода-вывода конфигурирование происходит достаточно просто и не требуется дополнительного программирования. Но при нерегулярной передаче данных, предварительной обработке или преобразовании данных, шлюз может быть оснащен по заказу дополнитель-

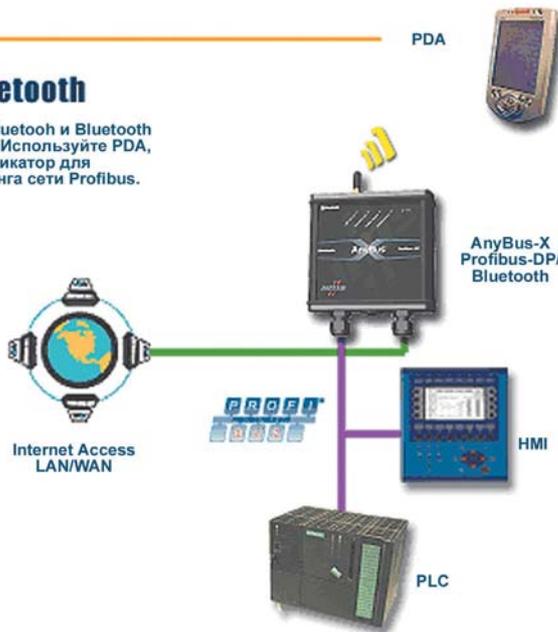


## Fieldbus to Bluetooth

Точка доступа Profibus к Bluetooth и Bluetooth к локальной сети Ethernet. Используйте PDA, ноутбук и сотовый коммуникатор для дистанционного мониторинга сети Profibus.

ными настраиваемыми программными функциями.

Большинство модулей в Slave-режиме поддерживают автоматическое определение скорости передачи или установку скорости DIP-переключателями. Через порт RS232 с помощью терминальной программы можно производить выгрузку ПО, устанавливать статус и объём данных В/В. Можно также использовать конфигурационные файлы EDS&GSD, которые доступны с интернет-узла HMS. Конфигурационная утилита HMS NetTool может быть использована совместно с Master-устройствами для Profibus и DeviceNet. Конфигурационная утилита RSNetWorx компании Rockwell Automation также может быть использована со всеми версиями, которые содержат интерфейсы DeviceNet, ControlNet и Ethernet/IP. Версия с интерфейсом Ethernet поддерживает Web-интерфейс для управления и диагностики.



Если нужно преодолеть разницу между электрическими сигналами в соединяемых сетях - применяем преобразователи интерфейсов.

Серверы устройств привлекательны при использовании сетей на основе Ethernet при управлении от ПК.

Шлюзы способны решать все задачи связи между двумя сетями. При этом они являются достаточно сложными устройствами и требуют конфигурации при установке для определения данных, которые следует передавать между сетями.

Мосты также решают все задачи связи между двумя сетями. Сегодня они обычно поддерживают простой обмен данными, но в будущем в них будет закладываться больший набор функций. Для обеспечения Web-доступа к устройству с последовательным интерфейсом и открытым протоколом необходимо использовать шлюз с поддержкой уровня 7 модели OSI для интерпретации реальных данных и их связи со встроенным Web-сервером.



### Заключение

Передача данных между двумя сетями - задача непростая, и для ее решения важно учитывать требования приложений, работающих в подключаемых сетях.



### КОНТАКТЫ:

т. (044) 241-87-39, 241-67-54  
e-mail: info@holit.com.ua  
HMS: www.anybus.com

## Платы

## ВВОДУ-Выводу в стандарті PC/104

- 40°C



### DMM-48-AT

АЦП 16 біт, 200 кГц,  
16SE каналів,  
8хЦАП 12 біт,  
8 вих. реле SPDT  
4 вх. "сухий" контакт  
4 ліній TTL,  
таймер-лічильник 16 біт



### RMM-1612

16хЦАП 12 біт,  
вихід по напрузі,  
24 ліній В/В (82С55)



### OMM-XT

48 ліній В/В (2х82С55)  
3-канальний  
таймер-лічильник (82С54)



### OMM-XT

48 ліній В/В (2х82С55)

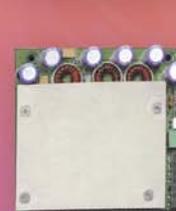


### EMM-8M

8хRS-232 / 422 / 485

### EMM-8232

8хRS-232



### JMM-512

джерело живлення 50Вт,  
+5В/10А, +12В/2А



## PANDORA

безкабельна система корпусів  
висотою 1.7" .. 10.0"

