

*Все идеи и алгоритмы, описываемые в данной статье, являются результатом моей независимой и полностью самостоятельной интеллектуальной деятельности. Как автор, разрешаю свободно использовать, изменять, дополнять все идеи и алгоритмы любому человеку или организации в любых типах проектов при обязательном указании моего авторства.*

© Балыбердин Андрей Леонидович 2019 Rutel@Mail.ru

**Синхронная Символьная Иерархия  
системы промышленной автоматизации**

Автор : Балыбердин А.Л.

Новосибирск, 2023 г.

## **Введение.**

В настоящее время для систем промышленной автоматизации применяются операционные системы реального времени, позволяющие реагировать на получаемые от датчиков сигналы в пределах заданного интервала времени. Для построения сложных АСУ ТП необходимо объединять несколько управляющих систем в единую сеть. Сеть АСУ ТП тоже должна соответствовать некоторому набору требований, таких как гарантия времени доставки, гарантия скорости, возможность построения резервных каналов и прочее.

В настоящее время абсолютное большинство сетей являются сетями, построенными на принципах пакетной коммутации, а пакетная коммутация не может предоставлять никаких гарантий по скорости, потерям и доставке. Можно сказать, что системы реального времени существуют, а вот сетей реального времени нет.

Сеть ССИ построена на принципах коммутации плезиохронных потоков данных и полностью соответствует самым жестким требованиям систем реального времени.

## **Какие плюсы можно получить при внедрении сети ССИ в современные системы промышленной автоматизации?**

Используя в качестве коммутаторов устройства с сетевыми интерфейсами ССИ, можно создавать полностью синхронные распределенные системы передачи данных с гарантированной пропускной способностью и латентностью, не зависящими от загруженности сети данными от других устройств. Иметь практически мгновенное оповещение о выходе из строя линий связи или управляющих контроллеров.

Можно добавить небольшое число «лишних» линий связи, так что была возможность построить несколько независимых соединений между взаимодействующими устройствами. Что позволит не терять данные в локальных сбоях в процессе передачи и строить механизмы реакции на такие события.

Если вспомнить историю развития электронных систем, то раньше электроника была существенно менее надежной и для получения требуемого качества работы применяли полное многократное дублирование с мажоритарными элементами для выявления ошибок. Современные контроллеры стали более надежными, компактными, быстрыми и из-за дороговизны от использования этой практики в большинстве систем отказались. Возможность новой сети позволяют ее возобновить без излишних финансовых затрат, получив существенное повышение надежности управляющих систем.

Большинство контроллеров обладают многократно большей, чем это необходимо для решения конкретной задачи производительностью и ее можно использовать для дублирования управляющих программ. Сеть ССИ позволяет одновременно передавать данные нескольким адресатам. Эти два свойства,

позволяют запустить копию одной и той же функции на нескольких физических контроллерах и контролировать не только одинаковость данных пришедших различными маршрутами, но и идентичность результатов работы ПО более нижнего уровня. В реальном масштабе времени можно производить дублирование критичных частей управляющего ПО и при выявлении сбоев в работе аппаратных средств, реализовывать различные виды корректирующих реакций. Процесс дублирования можно производить на всех уровнях системы управления. Реализовывать данный механизм нужно в виде стандартной библиотеки автоматизирующей большую часть работы разработчика системы. Кроме того, появляются достаточно простые механизмы для отладки и сохранения параметров работы системы в целом с привязкой к шкале реального времени, а не для отдельных контроллеров как сейчас.

Использование данных механизмов позволит эффективно создавать управляющие системы крайне высокой надежности без использования слишком большого объема дополнительных аппаратных средств.

### **Пути внедрения ССИ в современные системы промышленной автоматизации.**

Идеальным решением для систем реального времени был бы прямой интерфейс к сети ССИ (сетевая карта или модуль с данным интерфейсом), но пока таких промышленных контроллеров нет. Для того что бы данная проблема не тормозила внедрение сети ССИ, необходимо найти такое решение когда интерфейс ССИ имеет только коммутатор, а вся остальное оборудование только стандартные для микроконтроллеров и промышленных компьютеров интерфейсы (Ethernet, RS-485, CAN и тд).

### **Для примера опишем подключение микроконтроллера через интерфейс RS485 (half-duplex).**

Поскольку свойством гарантии доставки в реальном времени обладает только устройство с реализованным интерфейсом в сеть ССИ, то выбираем тип связи Master-Slave. Мастером является коммутатор, вернее модули RS-485 реализованные в структуре коммутатора. Подчиненными, все устройства, подключаемые к этим каналам. Алгоритм доступа к каналу получается не сильно отличным от канонического, подчиненное устройство получает доступ к каналу только после разрешения от мастера. Виртуальные каналы ССИ имеют свойство передавать символы даже если нет передачи реальных данных, что гарантирует периодическую отправку символов в сторону подчиненных устройств. Для единообразия добавляем обязательный ответ от подчиненного устройства мастеру, содержащий данные для передачи. Если в течении установленного срока ответного пакета нет, то считается что данное конкретное устройство отсутствует или вышло из строя. Остальные правила обмена будут сформулированы в процессе создания протокола.

### **Примерный алгоритм обмена данными.**

1. Модуль RS-485, в составе коммутатора ССИ, постоянно мониторит состояние FIFO виртуальных каналов данных, предназначенных для передачи подчиненным устройствам.
2. В момент появления данных происходит формирование информационного пакета с адресом конкретного устройства на шине RS485 и принятыми из сети ССИ данными, с дальнейшей его передачей через интерфейс RS485 подчиненному устройству.
3. Блок RS485 на подчиненном устройстве, принимает некоторое число данных (заголовок пакета) и инициирует прерывание работы микроконтроллера для декодирования заголовка пакета.
4. Обработчик прерывания проверяет принадлежность пакета данному устройству и если адрес соответствует, то начинает готовить пакет с ответными данными (данные будут отправлены в сеть ССИ).
5. После завершения приема входящего пакета, подчиненное устройство отправляет ответ.
6. В свою очередь коммутатор ССИ, после передачи пакета ожидает ответ. Если в течении некоторого времени его нет, то в соответствующий виртуальный канал (закрепленный за данным адресом RS-485) помещается символ, сигнализирующий об ошибке связи.
7. Если пакет принимается, то данные из этого пакета извлекаются и помещаются в буфер виртуального канала.