



НИКОЛАЙ ЗАХАРОВ  
 ВЛАДИМИР КЛЕПИКОВ  
 ДМИТРИЙ ПОДХВАТИЛИН  
 ДМИТРИЙ СЕМИКИН  
 АЛЕКСЕЙ ШЕПЕЛЕВ  
 НПП «Дозор» ОАО «Концерн КЭМЗ»

# ПРОТОКОЛ ДЛЯ РАСПРЕДЕЛЕННЫХ СИСТЕМ УПРАВЛЕНИЯ ОТВЕТСТВЕННОГО ПРИМЕНЕНИЯ

Современные тенденции развития авиационной, автомобильной и робототехники диктуют переход на распределенные системы управления, компоненты которых интегрируются с узлами и агрегатами управляемого объекта. Рассмотрен синхронно-временной протокол, обеспечивающий гарантированное стабильное время доставки всех сообщений и обладающий механизмами обеспечения надежности. Предложен контроллер обмена по указанному протоколу, серийный выпуск которого начат в настоящее время.

Для распределенных систем, критичных к надежности и временным характеристикам коммуникационных каналов, применение событийных (event-triggered) сетевых протоколов, таких как, например, CAN, оказывается неприемлемым [1]. Как правило, в таких случаях разработчики переходят от сетевой структуры системы к организации межузловых связей типа «точка–точка», что приводит к повышению сложности и массы системы, снижению ее надежности.

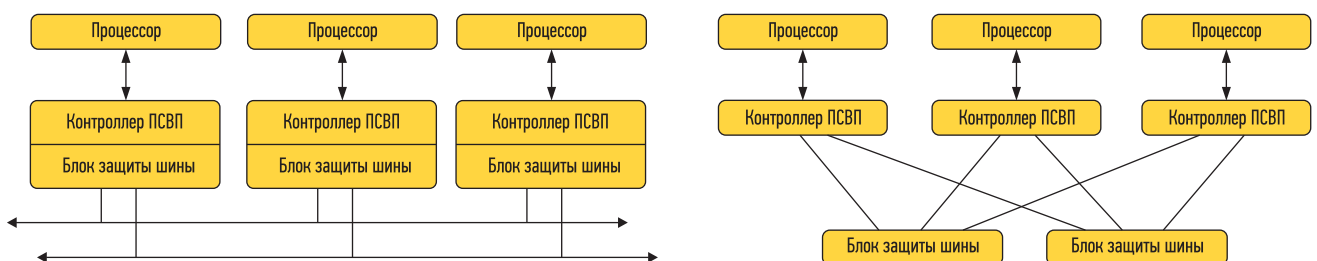
В последнее десятилетие ведущими фирмами активно развиваются так называемые синхронно-временные протоколы (СВП), такие как Time Triggered Protocol (TTP) и FlexRay. Так, консорциумом SAE International введен в действие международный стандарт SAE AS6003 TTP Standard [3].

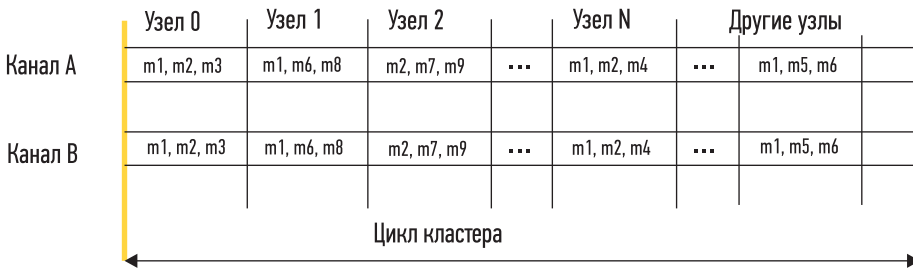
Система на основе СВП строится из сетевых узлов (рис. 1), включенных по схеме «шина» или «звезда».

Узел состоит из процессора с памятью, подсистемы ввода/

вывода, коммуникационного СВП-контроллера, операционной системы и соответствующего прикладного ПО. Все это реализуется в едином модуле, а в идеальном случае — в едином кристалле. Дублированная СВП-шина объединяет узлы в кластер, и вместе с коммуникационными контроллерами узлов они образуют в кластере коммуникационную систему, функционирующую автономно на основе заранее определенного периодического расписания

**РИС. 1.** ▼  
 Архитектура системы с СВП





исключая тем самым монополизацию шины отказавшим узлом. В топологии «шина» БЗШ входит в состав микросхемы-контроллера СВП и получает от него информацию о текущем слоте, но работает от независимого генератора синхронизации. В топологии «звезда» центральный БЗШ выполняется в виде отдельного устройства и независимо реализует алгоритм синхронизации часов кластера.

СВП обеспечивает следующие основные функции:

- передачу сообщений между узлами кластера;
- соответствие заранее установленному расписанию;
- синхронизацию часов всех узлов;
- соблюдение целостности кластера;
- старт и останов узлов.

Обмен данными в кластере организован (рис. 2) в виде циклов фиксированной длительности и структуры. В течение цикла происходит повторяющийся обмен полным набором сообщений. Цикл кластера разделен на слоты. Каждый узел кластера имеет один или несколько своих слотов и должен в каждом цикле выполнять в данных слотах передачу пакетов.

Узлы в кластере могут быть соединены двумя (и более) шинами. При передаче сообщения каждый узел синхронно помещает на обе шины одинаковые копии сообщения. При приеме ожидается получение корректного сообщения хотя бы с одной шины.

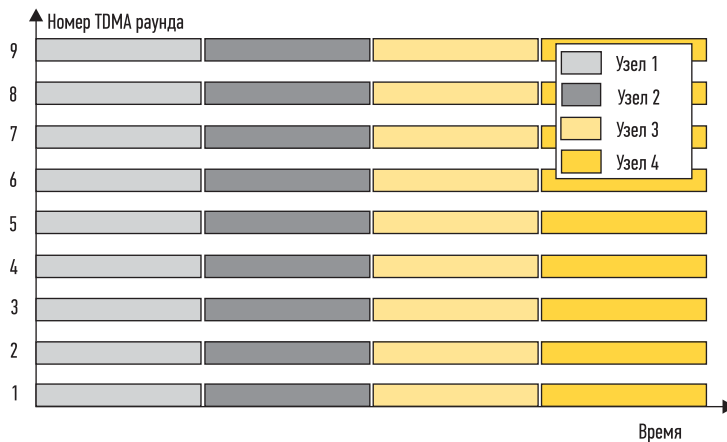
СВП обеспечивает строго синхронный обмен данными (рис. 3), что невозможно в протоколах, основанных на дисциплине доступа по событию (event-triggered), поскольку в них для разрешения конфликтов применяются различные виды арбитража, в результате чего могут возникать задержки в передаче сообщений и цикл оказывается плавающим (волнистая линия на рис. 4).

Как отмечено выше, все узлы используют единое расписание обмена. Для обеспечения всех узлов единой временной базой необходима синхронизация часов. Каждый узел на основе априорно известного ожидаемого времени прихода корректного сообщения и фактического времени его

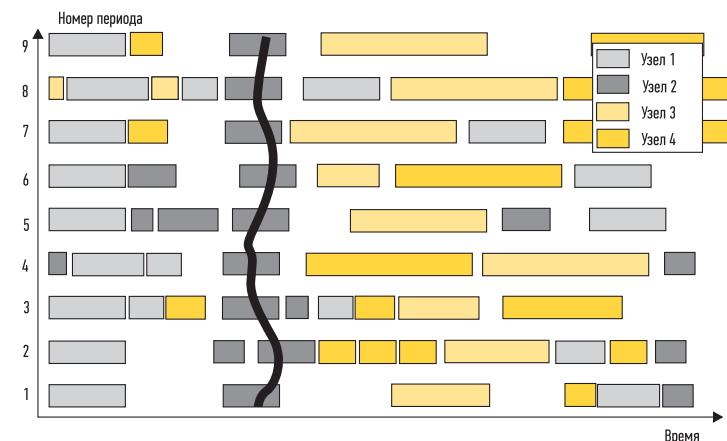
**РИС. 2.** ▲  
Цикл работы кластера системы с СВП

в режиме множественного доступа с разделением времени (Time Division Multiple Access, TDMA). Коммуникационная подсистема читает сообщения (пакеты данных) сетевого коммуникационного интерфейса (Communication Network Interface, CNI) узла в определенные расписанием моменты времени и отправляет их в CNI других узлов, обновляя записанную туда ранее информацию. Моменты времени чтения и записи сообщений содержатся в едином для всех узлов кластера расписании в виде

описателя сообщений (Message Descriptor List, MEDL). Копии MEDL хранятся в каждом узле. Кластер может иметь несколько режимов работы с разными расписаниями и переключаться между ними. Протокол обеспечивает возможность такого переключения. Блок защиты шины (БЗШ) функционирует автономно и защищает каналы передачи данных кластера от временных ошибок отдельных узлов. Он гарантирует, что узел может выполнять передачу только один раз в течение TDMA-раунда,



**РИС. 3.** ▶  
Пример работы системы с СВП



**РИС. 4.** ▶  
Пример работы системы с событийным переключением

прихода вычисляет разницу хода часов передатчика и приемника. Отказоустойчивый усредняющий алгоритм вычисляет коррекцию локальных часов с тем, чтобы они находились в синхронизации со всеми остальными часами кластера. Распределенный алгоритм контроля целостности кластера в случае возникновения отказа выявляет место его возникновения — выходная цепь передатчика или входная цепь приемника. Базовые алгоритмы СВП были формально верифицированы и успешно протестированы в условиях имитации миллионов отказов, в том числе при воздействии радиационного и электромагнитного излучений.

В СВП реализована концепция парирования одиночных сбоев и отказов, заключающаяся в том, что системы на его основе содержат достаточную избыточность, чтобы одиночный сбой или отказ никаким образом не отразились на поведении системы: ни с точки зрения функциональности, ни во временных соотношениях. Данная концепция основана на том, что вероятность одновременного проявления отказов в двух различных компонентах ничтожно мала. При появлении множественных отказов, которые не могут быть парированы самим протоколом, СВП информирует об этом прикладную программу, которая, в свою очередь, может принять решение о прекращении своей работы или о переходе в безопасный режим.

Реализация перечисленных механизмов отказоустойчивости обеспечивается свойствами СВП поддерживать согласованность (consistency) данных. В однопроцессорной системе согласованность гарантируется благодаря возможности всем компонентам ПО пользоваться одной копией данных, хранящихся в ОЗУ. Такой вид согласованности данных не работает в распределенной системе по следующим причинам. Во-первых, из-за задержек при передаче нет гарантии, что переданное сообщение будет принято всеми узлами-приемниками в одно и то же время. Во-вторых, некоторые узлы могут находиться в нерабочем состоянии, или сообщение из-за сбоя в коммуникаци-

онной системе может быть потеряно. Поддержка согласованности данных в СВП обеспечивается на уровне коммуникационного контроллера CNI путем реализации на аппаратном уровне функций контроля целостности кластера (Membership) и подтверждений (Acknowledgment).

Контроль целостности кластера заключается в следующем. Благодаря циклической (round-robin) схеме TDMA-раундов каждый узел ожидает и проверяет список членов кластера для всех узлов данного раунда. Каждый передатчик, не соответствующий списку членов, определяется как неисправный. Это обеспечивает согласованное взаимодействие группы узлов, каждый из которых видит других в своих списках членов кластера.

Подтверждения выполняются следующим образом. Узел А после каждой своей передачи ожидает от других узлов подтверждения того, что его сообщение было принято на коммуникационном уровне. Это достигается проверкой в списке членов кластера узла А первого и, возможно, второго подтвердившего узла. Если эти узлы находят узел А в своих списках членов кластера, они подтверждают, что передача узла А была успешно принята. В противном случае узел А извещает о неудачной передаче. В силу

принципа временного разделения повторная передача выполняется в следующем цикле.

Комбинация этих алгоритмов наряду с общей временной базой, поддерживаемой алгоритмом синхронизации часов, обеспечивает согласованность коммуникационного канала. Это гарантирует, что все корректно работающие узлы получают одинаковую информацию в одинаковые моменты времени.

Упрощенная функциональная схема контроллера СВП показана на рис. 5. Контроллер поддерживает связь с процессором по 16-разрядной параллельной шине данных или по интерфейсу SPI.

Расписание загружается в контроллер СВП программным обеспечением процессора и сохраняется в памяти. После старта контроллера СВП эта память становится доступной только для чтения.

После старта контроллер СВП работает асинхронно по отношению к процессору, получая синхросигналы от отдельного генератора (хотя и может генерировать прерывания для процессора в соответствии с установками в регистрах управления). Процессор обменивается принимаемыми и передаваемыми данными через память данных. Если процессор не изменит выдаваемые данные в некотором цикле, то контроллером СВП будет

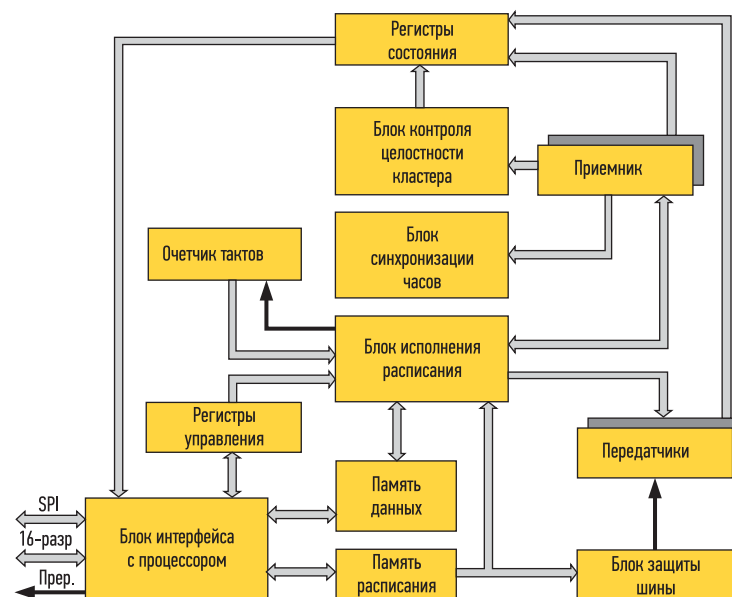


РИС. 5. Упрощенная функциональная схема контроллера СВП

повторно передана на шину старая копия данных.

Данные забираются от приемников и выдаются передатчикам блоком исполнения расписания, который, обращаясь в память расписания, получает описание каждого нового слота и инструкции, какие операции выполнить в этом слоте (выдать данные, принять данные, синхронизировать время и др.).

В выдаваемом сообщении каждый узел также дополнительно перечисляет узлы, от которых он успешно принял сообщения в прошлом цикле. На основе этой информации каждый узел может заключить, слышал ли его предыдущее сообщение некоторый другой абонент шины (и слышал ли кто-либо вообще). Данная информация помещается в регистры статуса и, наряду с признаками состояния приемников, передатчиков и узла в целом, позволяет ПО определить ситуацию потери связи, отказа передатчиков/приемников и пр.

Описанный выше контроллер СВП реализован НПП «Дозор» совместно с ЗАО «ПКК Миландр» в виде микросхемы K5600BG2Y (рис. 6).

Проектирование системы архитектуры с жестким временным разделением доступа к шине ТТА (Time Triggered Architecture) выполняется в два этапа или на двух уровнях — кластерном и узловом. На кластерном уровне проектируется топология сети и интерфейсы узлов. Далее каждый узел проектируется на основе функциональных спецификаций и спецификаций сетевого интерфейса.

Двухуровневый подход к проектированию ТТА позволяет реализовать важное свойство композитности (composability) системы. Система, обладающая таким свойством, позволяет декомпозировать ее на отдельные модули, которые могут быть разработаны и протестированы независимо друг от друга и затем проинтегрированы без учета их взаимного влияния. При интеграции в единый кластер модули не оказывают влияния на работу друг друга, т. к. каждый из них взаимодействует только со своим блоком CNI. Блоки CNI, в свою очередь, работают под управлением статически сформированного расписания, не зависящего от того, какие модули присутствуют в кластере.

Наиболее эффективное использование ТТА достигается при применении операционной системы жесткого реального времени, такой как TTP-OS, разработанной фирмой TTEch, или uOS, разработанной в НПП «Дозор». TTP-OS специально предназначена для приложений, основанных на использовании СВП (TTP). Данные системы занимают исключительно малые вычислительные ресурсы и обеспечивают быстрое переключение задач. TTP-OS разработана в соответствии с требованиями стандарта сертификации авиационных систем DO-178B Level A.

НПП «Дозор» в своих проектах с использованием СВП применяет uOS, которая представляет собой операционную систему реального времени для встроенных применений. Основными преимуществами uOS являются:

- Переносимость. uOS портирована на большое число архитектур процессоров: AVR, MSP430, ARM, ARM Cortex-M (в частности, микроконтроллеры «ПКК Миландр»), ARM Cortex-A, MIPS32 release 1 и 2 (в частности, процессоры фирмы «Элвис»), i386.
- Модульность. Базовый модуль ядра занимает около 2 кбайт ПЗУ и 200 байт ОЗУ, набор заменяемых модулей конфигурируется под конкретную задачу.
- Расширяемость. Состав модулей системы легко может быть расширен пользователем системы.

- Вытесняющая многозадачность.
- Высокая готовность, т. е. малое время инициализации системы (порядка единиц миллисекунд, в зависимости от процессора), малое время задержки обработки прерывания и малые накладные расходы на переключение задач.
- Внутренняя простота. В uOS используется концепция обобщенного мьютекса как единого примитива синхронизации, который работает как мьютекс или семафор с возможностью передачи сигналов (замена условных переменных и сигналов в других операционных системах).

- Поддержка сетевого стека протоколов TCP/IP v4. uOS является системой с открытыми исходными кодами, которые могут быть свободно получены с сайта проекта (<http://code.google.com/p/uos-embedded>). Для сборки проектов под uOS используются кроссплатформные свободные средства разработки, основанные на GCC и Eclipse.

Коммерческие и технологические преимущества архитектуры распределенной системы с жестким временным разделением доступа к шине управления подробно рассмотрены в [2]. В частности, обеспечивается снижение стоимости системы, повышаются стабильность и надежность ее функционирования.

Для разработчиков систем управления НПП «Дозор» поставляется отладочный комплект, содержащий модули на основе микроконтроллера 1986BE91 и СВП-интерфейс, а также комплект отладочного программного обеспечения с примерами реализации СВП-систем. ●

#### ЛИТЕРАТУРА:

1. Захаров Н. А., Клепиков В. И., Подхватилин Д. С. Синхронно-временной протокол для распределенных систем управления // Автоматизация в промышленности. 2013. № 2.
2. Захаров Н. А., Калинин С. В., Клепиков В. И., Подхватилин Д. С. Архитектура распределенных систем управления жесткого реального времени // Радиоэлектронные и компьютерные системы. 2008. № 5.
3. [http://www.tttech.com/fileadmin/content/pdf/AS6003\\_preview.pdf](http://www.tttech.com/fileadmin/content/pdf/AS6003_preview.pdf)

РИС. 6. ▼

Микросхема K5600BG2Y

