

Сеть CAN и протокол верхнего уровня CANopen в современном электроприводе*

CAN and Higher Layer Protocol CANopen in Modern Electric Drive

Анучин Алексей Сергеевич

Аннотация

В статье описываются основные возможности CAN-сетей для организации обмена данными в электроприводах и задачах локальной промышленной автоматизации. Приведены характеристики физического уровня передачи данных. Рассмотрены вопросы арбитража и способы обеспечения гарантированной доставки сообщений. Предложен способ повышения скорости передачи данных в CAN-сети. Определено назначение протоколов высокого уровня, рассмотрен протокол CANopen, проанализированы его основные недостатки и перечислены меры по их устранению. В качестве примера подробно рассмотрен вариант адаптации сервиса наблюдения за качеством связи.

Ключевые слова: электропривод, промышленная информационная сеть, CANopen, протокол верхнего уровня.

The article deals with the main abilities of CAN for data exchange in electric drives and local industrial automation. The characteristics of physical layer are presented. The problems of arbitration and data delivery are considered. The method of data rate increasing in CAN is offered. The purpose of higher layer protocols is determined and CANopen protocol is considered; main disadvantages are

* Исследования проводились при финансовой поддержке по гранту Президента Российской Федерации для поддержки молодых российских ученых-кандидатов наук МК-3595.2005.8

analyzed and the ways of their elimination are mentioned. As an example a possible adaptation of heartbeat protocol is described.

Keywords: electric drive, industrial network, CANopen, higher layer protocol.

Введение

С усложнением технологического процесса, увеличением числа контролируемых величин растет число линий связи между системой верхнего уровня и исполнительными устройствами. Например, система питания вагонного кондиционера, устанавливаемая в поезда дальнего следования, имеет более 20 линий дискретного ввода/вывода. Более 20 экранированных витых пар проложено от управляющего компьютера до силового преобразователя, установленного в ящике под вагоном. Передаются команды включения и задания режима работы, возвращается информация о нештатных ситуациях. Сам преобразователь системы питания может различать более 100 различных типов неисправностей[†], но передать по существующим дискретным каналам подробный отчет не представляется возможным. Более того, если в каком-либо дискретном канале произошел обрыв, определить это система верхнего уровня не сможет.

Чтобы уменьшить количество проводных соединений при увеличении сложности проекта, связь между устройствами организуют по последовательному каналу связи. Подключение устройства в локальную промышлен-

[†] Данные указаны для преобразователя БПЭДК-24, разработанного ОАО «Ижевский радиозавод» в сотрудничестве с ООО «НПП Цикл+» и ООО «НПФ Вектор».

ную сеть производится по последовательной двух или трехпроводной (иногда и больше) линии передачи данных. Управляющие воздействия, обратные связи и информация режимах работы передаются между устройствами в цифровом виде. Применение промышленных сетей имеет несколько основных преимуществ перед традиционными параллельными дискретными и аналоговыми способами связи устройств входящих в технологический процесс. Во-первых, существенно сокращается количество проводных соединений, что снижает стоимость и повышает надежность системы в целом. Во-вторых, в промышленных сетях постоянно ведется проверка наличия и качества соединения, а передаваемые данные не подвержены помехам, так как производится проверка их целостности и вероятность приема недостоверной информации приближается к нулю. Механизмы определения обрывов аналоговых или дискретных связей сложнее и увеличивают стоимость проекта, к тому же эти сигналы подвержены влиянию помех. В-третьих, в сеть можно относительно легко добавлять при необходимости новые устройства без перепрокладки информационных кабельных соединений, так как подключение будет проводиться к уже существующей сети.

В настоящее время существует несколько стандартов для построения промышленных сетей. К ним относятся Profibus, Modbus, CANopen, DeviceNet, промышленный Ethernet и другие. Существует множество статей [1, 2, 3, 4], описывающих преимущества того или иного сетевого стандарта, однако очень малое внимание в них уделяется способам реализации интерфейсов в конечных устройствах. То есть, имеющиеся материалы определяют об-

ласть применения конкретной сетевой технологии, но не дают разработчикам рекомендаций по реализации стандарта в своих устройствах. В условиях существенного ограничения ресурсов у большинства разработчиков электротехнической продукции выбор падает не на подходящий сетевой стандарт, а на самые простые интерфейсы типа Modbus RTU, который легко реализуем на любом современном микроконтроллере. В итоге разработанное устройство, имея пусть даже лучшие характеристики в своем классе, теряет конкурентоспособность, так как не может встраиваться в автоматизированный технологический процесс и работать в нем в режиме полноценного реального времени, оно требует дорогостоящего преобразователя сетевого интерфейса, обеспечивающего совместимость с другими более современными устройствами объекта автоматизации.

К сложностям реализации устройств на базе Profibus следует отнести закрытость протокола и необходимость приобретения для каждого выпускаемого устройства микросхемы поддержки этого интерфейса. Modbus прост в реализации, но морально и технически устарел, он не может работать в режиме реального времени, а из типов данных поддерживает только бит и целое слово. В настоящее время широкую популярность приобрели CAN-сети, так как CAN-интерфейс встраивается по умолчанию в большинство микроконтроллеров. CAN используют как в качестве бортовых сетей автомобилей и кораблей, так и на производстве. Сотни фирм предлагают преобразователи частоты, датчики, контроллеры с выходом именно на этот интерфейс. На базе CAN-сети реализуются протоколы верхнего уровня CANopen и DeviceNet. К

сожалению материалов по реализации этих протоколов, кроме официальных спецификаций очень мало, поэтому цель данной статьи определить необходимые шаги для адаптации протокола верхнего уровня к задачам электропривода и локальной автоматизации и привести пример модернизации сервиса протокола CANopen.

Особенности CAN-сети

CAN-сеть в обычной своей реализации использует провод типа «витая пара» для передачи дифференциального сигнала. Информация передается пакетами, состоящими из 11-разрядного (29-разрядного в версии протокола CAN2.0B) заголовка и кадра данных длиной от 0 до 8 байт. Малый размер пакета – это первый признак CAN-сети, как сети реального времени. То есть каждая посылка не отнимает большого количества времени и между низкоприоритетными посылками могут вклиниваться высокоприоритетные сообщения, латентность доставки которых должна быть строго регламентирована для систем реального времени.

Все устройства в CAN-сети соединяются последовательно. Скорость передачи зависит от длины сети. Рекомендованная область скоростей передачи показана на рис.1.

Если сравнивать скорость CAN-сети со скоростью, например, Profibus-DP (до 12 Мбод), то первая очень сильно проигрывает. Проигрыш в скорости связан с физическим принципом передачи информации в CAN-сети. Передаваемые на шину данные представлены «нулями» и «единицами». В терминологии CAN-сети «ноль» является доминантным (преобладающим) состоянием

ем, а «единица» – рецессивным (исчезающим). То есть, если два устройства сети одновременно выдают на линию разные данные (одно устройство – «ноль», другое – «единицу»), то на линии оказывается «ноль» – рецессивная «единица» потеряна.

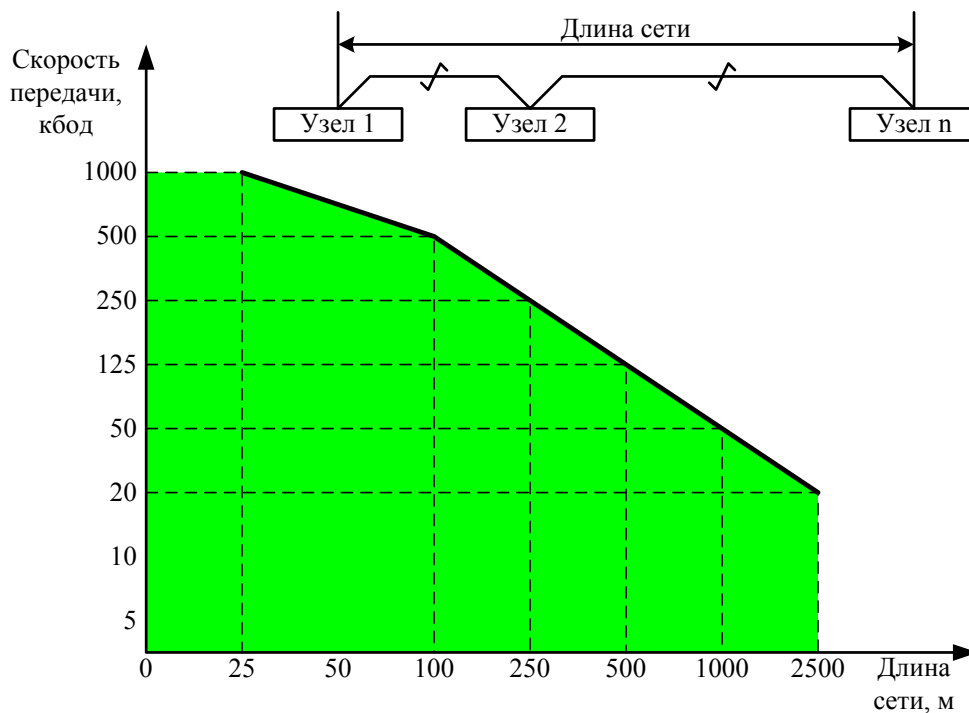


Рис.1. Рекомендованная область скоростей CAN-сети

Пользуясь этим принципом, CAN-сеть разрешает коллизии (или одновременное включение на передачу двух или нескольких устройств сети) методом недеструктивного арбитража (метод определения самого приоритетного из передающих устройств без потери времени на разрешение коллизии). В качестве арбитражного поля используется заголовок CAN-сообщения. Предположим, что одновременно три узла стали передавать сообщения (рис.2).

После первого стартового бита, который был одинаков для всех трех устройств, передается первый бит заголовка. У третьего узла этот бит оказался равен «единице», в то время как остальные узлы выдали на шину «ноль».

«Ноль» является доминантным состоянием шины, поэтому третий узел, читая состояние шины, видит, что на ней оказалась не та информация, которую он хотел передать. Таким образом третий узел проигрывает арбитраж и начинает прием данных. Его сообщение откладывается до тех пор, пока шина не освободится для передачи, когда снова должна будет повториться фаза арбитража.

Узел 1	Узел 2	Узел 3	Шина	
0	0	0	0	Стартовый бит
0	0	1	0	Узел 3 проиграл арбитраж.
0	0	Принимает	0	
1	1		1	
1	0		0	Узел 1 проиграл арбитраж
Принимает	x		x	
	x		x	
	x		x	

Узел 2 продолжает передачу

Рис.2. Арбитраж в CAN-сети

Следующие два бита оказываются одинаковыми для первого и второго узлов сети. Различие начинается дальше, когда первый узел передает рецессивную «единицу», а второй – доминантный «ноль». Первый узел видит несоответствие данных на шине и проигрывает арбитраж, переключаясь в режим приема сообщения. На шине остается только второй узел, приоритет которого оказался наивысшим. Данный способ арбитража требует выполнения очень важного условия.

Сигнал, передаваемый любым из узлов сети, должен распространяться по всей длине сети с малой фазовой задержкой. То есть различие во времени от отправки бита до его получения любым другим устройством не должно превышать малой доли времени передачи одного бита.

Это условие существенно ограничивает возможную скорость передачи информации в CAN-сети. Если в других сетях можно успеть переслать много разрядов сообщения прежде чем первый его разряд, распространяясь со скоростью света, достигнет получателя, то в CAN-сети все узлы работают на «досветовой» скорости. Данное свойство CAN-сети, являясь на первый взгляд существенным недостатком, открывает принципиально новые возможности.

В CAN сети допускается, чтобы несколько устройств начинали передачу сообщений одновременно, но это абсолютно неприемлемо в высокоскоростных сетях, поэтому там применяют технологию «мастер-подчиненный» (или «клиент-сервер»). В сети Profibus-DP «мастер» сети должен опрашивать каждое устройство по очереди, чтобы получить с него данные и определить его состояние.

В CAN-сети любой из узлов может инициировать связь. Например, если пульт управления опрашивает состояние каких-либо параметров привода, то этот процесс не мешает выполнению технологической задачи. А возникшая аварийная ситуация должна вызывать немедленную реакцию. Тогда аварийные сообщения получают заголовки вида «001 xxxx xxxx», заголовки вида «01x xxxx xxxx» получают сообщения технологической задачи, а запросы

пульта управления будут иметь заголовки «1xx xxxx xxxx» (подобная схема используется в протоколах верхнего уровня DeviceNet и CANopen). Тогда аварийные сообщения всегда будут передаваться первыми сразу после завершения передачи текущего передаваемого сообщения, так как они будут выигрывать арбитраж на шине. Если все протекает нормально и аварийных сообщений нет, то шина CAN-сети занята сообщениями технологического характера. Перерывы между технологическими сообщениями занимают запросы пультов управления и средств диспетчеризации.

Возможность инициации сообщения любым из устройств и отсутствие явного ведущего мастера достигается именно благодаря неdestructивному арбитражу на шине, потере в скорости передачи. Взамен для высокоприоритетных сообщений мы получаем максимальную задержку равную T – времени пересылки одного сообщения (рис.3).

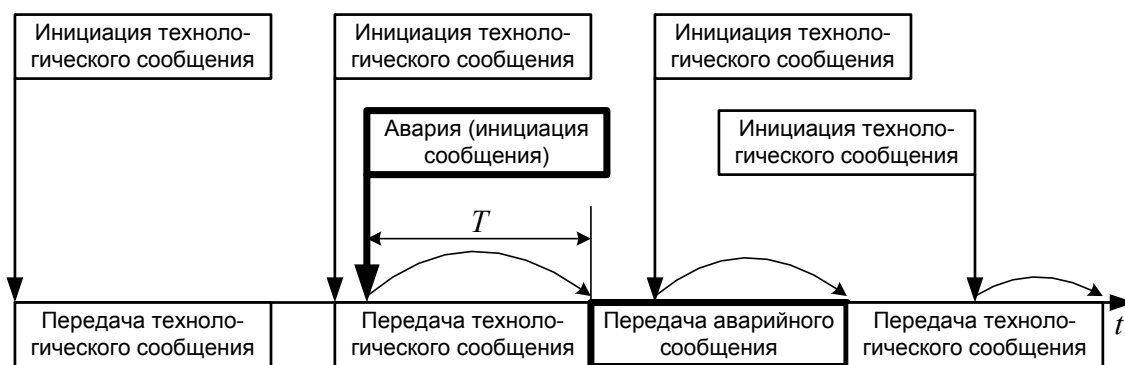


Рис.3. Задержки в передаче сообщений по CAN-сети

Все сообщения в CAN-сети широковещательные, то есть могут быть услышаны любым из устройств. Идентификатор сообщения не указывает получателя и используется для определения содержимого сообщения. Косвенно по идентификатору можно судить об отправителе, так как одну и ту же ин-

формацию разные узлы отсылать в сеть не могут. Каждый узел может решать принимать ему сообщения с этим идентификатором или не принимать, что делать с принимаемой информацией, как ее интерпретировать.

Широковещательность позволяет экономить трафик сети, так как в одной посылке можно упаковывать информацию для разных потребителей. Например, плата дискретного ввода рассылает состояние своих дискретных входов в одной посылке в сеть, включающую с себя несколько преобразователей частоты. Тогда первый преобразователь, получая посылку с состоянием дискретного ввода, может воспринимать первый бит информации как кнопку «Пуск», второй как кнопку «Стоп», а второй преобразователь аналогичным образом может воспринимать третий и четвертый биты передаваемой информации. В сети Profibus-DP мастер выполняет ту же самую операцию другим образом. Он запрашивает состояние платы дискретного ввода, а затем направляет полученную информацию сначала одному ПЧ, затем другому. Только на данном примере количество сообщений сократилось в три раза.

Способ повышения скорости передачи данных по CAN-сети

Скорость CAN-сети ограничена тем обстоятельством, что каждый бит передаваемой информации должен быть одновременно услышан всеми устройствами сети. В некоторых случаях разработчики отказываются от стандартной реализации CAN и предлагают свои варианты с увеличенной скоростью передачи, которые теряют совместимость и лишаются преимуществ, изначально присущих CAN-сетям. Существует, например протокол TTCAN,

который работает на скоростях до 10 Мбод, но в нем все сообщения отсылаются по временным меткам, и отсутствует фаза арбитража. В таком случае гарантировать своевременную надежную доставку сообщений уже нельзя.

Еще один способ увеличения скорости передачи предлагает автор. Суть метода заключается в разделении посылки на разные типы полей, так как в сообщении всегда есть поля (рис.4), требующие одновременного восприятия всеми узлами сети (выделены на рисунке серым цветом), а есть не требующие. Действительно, если передается заголовок сообщения, то поле арбитража следует передавать на стандартной скорости для существующих CAN-сетей. Поле данных сообщения передается тогда, когда уже выигран арбитраж и единовременное восприятие всеми узлами сети каждого бита не требуется. Во время этой фазы скорость можно поднять в $5 \div 10$ раз, так как приемопередатчики рассчитаны на эту скорость. Затем наступает фаза подтверждения правильности приема сообщения получателями, где опять необходимо снизить скорость передачи.

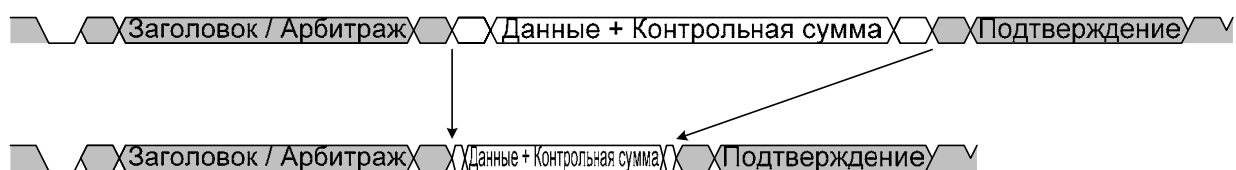


Рис.4. Метод передачи данных с переменной частотой

Данный метод позволяет увеличить пропускную способность сети приблизительно в $2 \div 2,5$ раза, так как данные с контрольной суммой, которые можно передавать на повышенной скорости, составляют 96 из 128 бит сообщения CAN-сети. Недостатком метода является необходимость аппаратной

доработки CAN-модулей вновь производимых контроллеров и несовместимость с существующими CAN-сетями.

Протоколы верхнего уровня для CAN-сети

В CAN-сети данные передаются пакетами по 8 байт с гарантированной доставкой сообщений и с широкими возможностями по расстановке приоритетов сообщений. Наполнение пакетов данными определяется протоколами верхнего уровня или протоколами работы с приложением, то есть с программой, обеспечивающей главную функцию электротехнического устройства (например, с системой управления преобразователем частоты).

Существует большое количество протоколов верхнего уровня, предназначенных для решения разных задач. Выбирая между протоколами следует анализировать следующие важные факторы:

- совместимость с другими узлами сети;
- реализуемость протокола на выбранном оборудовании;
- стоимость спецификации протокола;
- стоимость драйвера протокола.

Когда организуется какая-либо промышленная сеть, необходимо, чтобы внутри нее устройства поддерживали выбранный протокол верхнего уровня, а если протоколы разных устройств все-таки различаются, то они не должны мешать друг другу внутри одного информационного пространства. Чтобы избежать проблем, требование следует формулировать более жестко,

запретив внутри сети разные протоколы верхнего уровня, даже если это не мешает основной задаче, чтобы исключить вероятность сбоев.

Вторым ключевым моментом является возможность реализации протокола на используемом контроллере. Дело в том, что для работы современного совместимого протокола требуется большой объем памяти данных и программ и процессорного времени, которое так необходимо для решения прямых технологических задач. Если рассмотреть пример с протоколом CANopen, то оказывается, что его драйвер для микроконтроллеров TMS320LF2406A от Texas Instruments (семейство специализированных микроконтроллеров со встроенной периферией для управления двигателями и полупроводниковыми преобразователями) займет практически весь объем памяти программ и почти весь объем ОЗУ данных и не оставит каких-либо ресурсов на выполнение основных задач при том, что данный контроллер нельзя отнести к низкопроизводительным. Он является флагманом своего семейства и не многие фирмы, производящие микроконтроллеры, выпускают устройства с такими техническими характеристиками.

Спецификации протокола доступны членам ассоциации поддержки и разработки протокола. При создании устройства с поддержкой какого-либо протокола следует приобрести уникальный код производителя (VendorID), чтобы устройство имело гарантированно уникальный адрес в сети и могло быть идентифицировано сторонними производителями.

После выбора протокола требуется реализовать его стек (программный драйвер протокола) на конкретном процессоре. Стоимость готового драйвера

протокола CANopen составляет несколько тысяч долларов и варьируется в зависимости от типа процессора. Драйвер можно писать и самостоятельно при наличии спецификации, но это, в случае с протоколом CANopen, может потребовать до полугода труда программиста при отсутствии гарантии полной совместимости со всем необходимым оборудованием. Преимущество разработки собственного драйвера заключается в возможности адаптации его под конкретную задачу и сокращения используемых ресурсов процессора.

В настоящее время основными используемыми протоколами являются CANopen и DeviceNet, они похожи по набору сервисов передачи данных и принципу работы. В своей практике автор использует протокол CANopen. Зимой 2004 года CiA (CAN in Automation – организация по стандартизации CAN в промышленности) предоставила свободный доступ к спецификации CANopen [5], что позволило оценить ресурсоемкость протокола и начать написание собственного драйвера. Вторым обстоятельством, повлиявшим на выбор, стало наличие в глобальной сети ресурса www.microcanopen.com [6], на котором рассматривались вопросы создания устройств ограниченной совместимости, реализующих только основные жизненно важные функции протокола. После анализа полной спецификации и предложений по ограничению функций автором был разработан адаптированный CANopen-совместимый протокол, пригодный для решения задач локальной автоматизации и управления двигателями, и требующий менее одной трети ресурсов памяти микроконтроллеров TMS320LF2406A и TMS320F2810.

Адаптация CANopen для задач электропривода

Протокол верхнего уровня CANopen разработан организацией CiA и изначально ориентирован для решения задач электропривода. В сети может одновременно находиться до 127 устройств. Протокол обеспечивает постоянный мониторинг устройств в сети, позволяет читать или изменять любой из параметров устройства и обеспечивает регулярную пересылку ограниченного числа параметров технологического процесса (задания и обратные связи).

Так как разработка протокола производилась профессионалами в области коммуникаций, некоторые особенности важные для решения задач электропривода оказались неучтенными. Приведем примеры.

Параметры устройства, такие как задание скорости и значение обратной связи по скорости, описываются в словаре объектов, который хранится непосредственно внутри устройства. Чтобы получить доступ к параметру и прочесть его значение или записать его, требуется указать номер параметра в словаре объектов. Однако средствами CANopen нельзя получить из устройства список его параметров.

Для устройств, таких как преобразователи частоты, платы ввода/вывода информации, датчики величин и т.д. существуют стандартные словари объектов (словари параметров), но преобразователи частоты разных производителей имеют разный формат представления одних и тех же величин, разные схемы встроенных структур систем управления и разные настраиваемые параметры. Наиболее полным в настоящее время является стан-

дарт для плат дискретного ввода/вывода, в то время как стандартизовать профиль на преобразователи частоты не удастся.

Параметры, устройств представляются целыми числами, булевыми числами, числами с плавающей точкой и строковыми переменными и константами. В системах управления электроприводами физические величины удобно представлять в относительных единицах, но CANopen не поддерживает в явном виде этот формат. Числа с плавающей точкой и строковые переменные крайне редко применяются в современных контроллерных системах, так как работа с ними ресурсоемка. Обработку строковых типов и плавающей точки можно исключить из возможностей драйвера для сокращения объема используемой памяти данных и программ.

При разработке драйвера CANopen автором были учтены все перечисленные недочеты стандартного протокола. Адаптация проводилась за счет включения в драйвер надстроек и не ухудшает совместимость с другими CANopen устройствами. Были решены следующие задачи:

1. Реализована поддержка представления данных в относительных единицах. Каждому параметру в относительных единицах ставится в соответствие масштабирующий коэффициент, необходимый для преобразования из относительных величин в абсолютные. Каждый параметр имеет размерность и префикс (к, М, Г, п, н, мк, м) для представления больших и малых величин.
2. Разработан механизм определения перечня параметров устройства, что позволяет подключаться к устройствам CAN-сети и производить

просмотр и настройку параметров, не имея файла описания словаря устройства. Это позволяет подключать пульта управления, универсальные для всех устройств сети.

3. Реализованы режимы автоматического включения устройства в работу без участия «мастера» сети со слежением за наличием в сети обязательных устройств.

Последняя задача обусловлена первоначальной ориентацией CANopen на сети с ведущим «мастером», который частично или полностью контролирует поток данных и отслеживает состояние устройств в сети. Например, в замкнутой структуре «преобразователь частоты с выходом на CANopen – датчик давления с выходом на CANopen» требуется наличие «мастера» сети, для слежения, за исправностью оборудования, иначе преобразователь продолжит работу, оставшись без обратной связи по давлению в случае отказа датчика или его системы питания. Мониторинг состояния устройств в сети осуществляется сервисом «Heartbeat» (от английского «сердцебиение»), адаптацию которого рассмотрим более подробно.

Адаптация сервиса «Heartbeat»

Сервис «Heartbeat» служит для рассылки в сеть периодических сообщений, содержащих информацию о текущем состоянии устройства. Потребитель сообщений в спецификации протокола не оговаривается, как не оговаривается механизм их обработки. Для адаптированного протокола был разработан следующий алгоритм работы сервиса «Heartbeat».

Каждое устройство посылает в сеть сообщение с периодичностью, определенной параметром «Producer Heartbeat Time» (период рассылки сообщений «Heartbeat»). Каждое устройство принимает все сообщения «Heartbeat», рассылаемые в сети. Автором было предложено, что получая сообщение, устройство определяет номер источника сообщения и устанавливает один из 128 флагов в соответствии с номером источника. С заданной параметром «Consumer Heartbeat Time» (период ожидания сообщений «Heartbeat») периодичностью производится проверка соответствия установленных флагов и маски обязательных устройств. Если флаги закрывают всю маску обязательных устройств, то устройство продолжает работать в нормальном режиме, если же выясняется, что одно из обязательных устройств потеряно, устройство рассылает аварийное сообщение и переходит в пассивное состояние. Программа, выполняющая основную функцию устройства, получает предупреждение об ошибке в канале связи.

В адаптированном протоколе CANopen сервис Heartbeat служит не только для слежения за устройствами в рабочем состоянии, но и для введения их в работу в автоматическом режиме (без участия мастера сети) как после подачи питания, так и после восстановления связи. Для этого автором были введены параметры «Heartbeat Auto Start» (признак разрешения автоматического запуска), который определяет возможность проведения самозапуска, и «Heartbeat Auto Recovery» (признак разрешения перезапуска), позволяющий переходить из аварийного состояния в рабочее.

Включаясь устройство проходит состояние «Bootup» (начальная инициализация) и переходит в состояние «Preoperational» (предфункционирование), в котором оно ожидает появления в сети всех обязательных устройств. Если значение «Heartbeat Auto Start» равно булевой истине, и в сети присутствуют все необходимые устройства, происходит переход в состояние «Operational» (функционирование), когда все сервисы обмена данными CANopen включаются в работу. Если какое-либо из обязательных устройств сети перестает рассылать свое сообщение «Heartbeat», то это определяется другими устройствами и они переходят в состояние «Stopped» (остановка), выход из которого назад в предфункционирование возможен, если параметр «Heartbeat Auto Recovery» равен истине, и все обязательные устройства снова появились в сети.

Для определения перечня обязательных устройств сети автором был введен параметр «Heartbeat Mask» (маска обязательных устройств сервиса «Heartbeat»), а наличие устройств сети может определяться по значению параметра «Heartbeat Device Present» (наличие устройств в сети).

Если все же предполагается использование «мастера» сети, то адаптированный сервис Heartbeat остается полностью совместимым с оригинальным сервисом, описанным в спецификации.

Заключение

CAN-сети позволяют решать задачи локальной автоматизации используя простой и дешевый физический интерфейс. CAN-модули встроены практически во все выпускаемые в настоящее время микроконтроллеры и требу-

ют лишь наличия внешней микросхемы физического драйвера. Обмен технологическими данными производится с помощью протоколов верхнего уровня, таких как CANopen. Именно сложность реализации протокола верхнего уровня и не полная приспособленность к задачам электропривода и локальной автоматизации является основной проблемой внедрения программной поддержки CAN-сети в преобразовательную технику, датчики, платы сбора информации и программируемые логические контроллеры, однако при правильном выборе необходимого набора функций и адаптации драйвера можно получить устройства, интегрируемые в сложные технологические процессы с обменом огромным количеством данных всего по двум медным проводам.

Разработанные автором способы адаптации драйвера протокола CANopen реализованы в преобразователях частоты серий «Универсал» и «iCAN». С помощью разработанного CANopen драйвера решена задача автоматизации металлорежущего станка на заводе ООО «Металлическая упаковка» в г.Калининград, создан электропривод ВИП-800 на мощность 800 кВт с согласованной по CAN-сети работой четырех преобразователей мощностью 200 кВт на один двигатель.

Библиографический список

1. С.А.Третьяков, CAN на пороге нового столетия, Мир компьютерной автоматизации №2/99
2. <http://www.aldis.ru/techno/nets/history/>
3. <http://ruwiki.com/article/Profibus>

4. K. Etschberger, CAN-based Higher Layer Protocols and Profiles, Proc. of the 4. International CAN Conference, 1997, Berlin
5. CANopen Application Layer and Communication Profile, CiA Draft Standard 301, Version 4.02, Date: 13 February 2002
6. www.microcanopen.com

Автор: Анучин Алексей Сергеевич в 2001 г. окончил кафедру Автоматизированного электропривода Московского энергетического института (ТУ), где в 2004 г. защитил кандидатскую диссертацию «Разработка системы управления многофазного вентильно-индукторного привода с промежуточным регулируемым звеном постоянного тока». Доцент кафедры Автоматизированного электропривода Московского энергетического института (ТУ). Электронная почта: anuchinas@mpei.ru. Телефон: +7-495-362-7021.