

Протоколы прикладного уровня CAN-сетей

Андрей Щербаков

Описываются наиболее распространенные сетевые протоколы верхнего уровня, базирующиеся на интерфейсе CAN.

Введение

В середине восьмидесятых, когда фирма Robert Bosch GmbH предложила собственный вариант сети контроллеров для решения проблемы угрожающе разрастающейся тогда проводки автомобилей, вероятно, никто, включая самих разработчиков, не предполагал, что спустя несколько лет короткая и звучная аббревиатура CAN (Controller Area Network) станет широко известна далеко за пределами автомобильной отрасли. Сегодня CAN-сети активно применяются в самых, казалось бы, неожиданных устройствах и механизмах — от стиральных машин до томографов и ракет: аттракционы, штамповочное, фрезерное и типографское оборудование, морские суда, промышленные роботы. Одно лишь перечисление областей человеческой деятельности, где сегодня успешно трудится Controller Area Network, способно занять целую журнальную страницу. Можно припомнить хорошо известные в России телескопы Carl Zeiss, упаковщики TetraPak, томографы Siemens, не говоря уже о множестве марок европейских грузовых и легковых автомобилей: BMW, Mercedes-Benz, Renault, Fiat, Volvo, Saab, Audi, в которых CAN-сеть является нервной системой, центром управления жизненно важными узлами.

Ряд оригинальных и эффективных технических решений, положенных в основу CAN-протокола фирмой Bosch, а также последующие годы

«проверки на прочность» CAN-сетей в самых разных, как правило, очень непростых условиях эксплуатации — поистине, во всех трех стихиях: на земле, в небесах и на море — обеспечили CAN мировое признание, закрепленное в 1993 году в международном стандарте ISO 11898. На сегодняшний день стандарт ISO 11898 наряду с современной спецификацией Bosch CAN 2.0A/B является базовым документом разработчиков CAN-устройств — от трансиверов до модулей и сетей. Координацию усилий производителей, разработчиков и пользователей CAN-систем и технологий осуществляет международная некоммерческая организация CiA (CAN in Automation), объединяющая более 300 компаний во всем мире. Среди многочисленных достоинств CAN-сетей можно выделить следующие.

- Невысокая стоимость как самой сети, так и ее разработки. На рынке существует большой выбор CAN-контроллеров по цене до \$10, а простейшие устройства ввода-вывода — CAN SLIO (CAN 2.0A) стоят менее доллара. Следует отметить доступность и широкий выбор готовых CAN-модулей и недорогих инструментальных средств.
- Высокая степень надежности и «живучести» сети, благодаря развитым механизмам обнаружения ошибок (одна незамеченная ошибка за более чем триста лет круглосуточной работы сети на скорости 500 кбит/с), повтору ошибочных

сообщений, самоизоляции неисправных узлов, иммунитета к электромагнитным помехам.

- Простота конфигурирования и масштабирования сети, отсутствие теоретических ограничений на количество узлов.
- Поддержка разнотипных физических сред передачи данных, от витой пары до оптоволокна и радиоканала.
- Эффективность реализации режима реального времени, благодаря мультимастерности, широковещанию, побитовому арбитражу и высокой скорости передачи данных (до 1 Мбит/с).
- Промышленный стандарт — десятки производителей CAN-компонентов и оборудования, включая практически всех электронных гигантов: Intel, Philips, Siemens, Motorola. Гарантированная доступность элементной базы в течение, как минимум, 10 лет.

Однако действующий стандарт CAN ограничивается спецификацией только двух самых нижних уровней эталонной семиуровневой модели взаимодействия открытых систем OSI/ISO — физического и канального (рис. 1). Описываются физические параметры среды передачи данных (только в ISO 11898), форматы сообщений, процессы передачи данных длиной до 8 байт, механизмы обнаружения ошибок и др. Но за рамками стандарта остаются решения таких важных при разработке вопросов, как

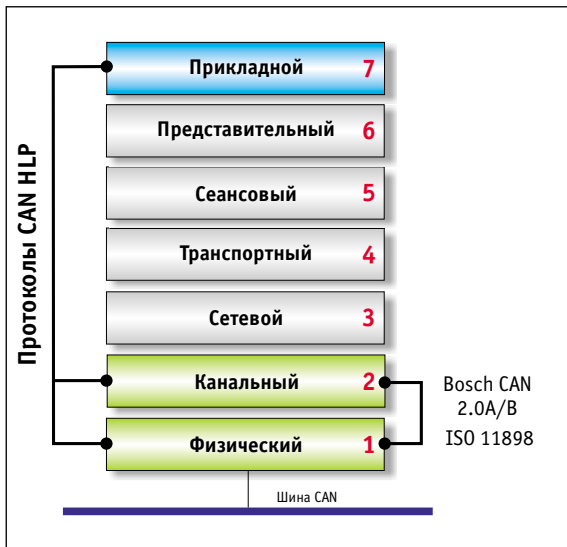


Рис. 1. Соотношение эталонной модели OSI/ISO и CAN-протоколов

адресация узлов, распределение между ними CAN-идентификаторов, интерпретация содержимого фрейма данных, передача данных длиной более 8 байтов и др., то есть все то, что обычно рассматривается на более высоких уровнях, вплоть до 7-го прикладного. Разумеется, сервисов двух нижних уровней может оказаться вполне достаточно, когда речь идет о разработке сравнительно простой сети, не планируемой к расширению и вдобавок состоящей из созданных под нее узлов-модулей. Или, к примеру, стоит задача создать «закрытую» сеть на основе оригинального протокола. Но в подавляющем большинстве случаев практических CAN-разработок двух «стандартных» уровней оказывается явно мало, а изобретение «велосипеда протоколов» для конкретной задачи — слишком дорогое, долгое и, следовательно, малоэффективное занятие. Поэтому с самого начала опубликования CAN-специфика-

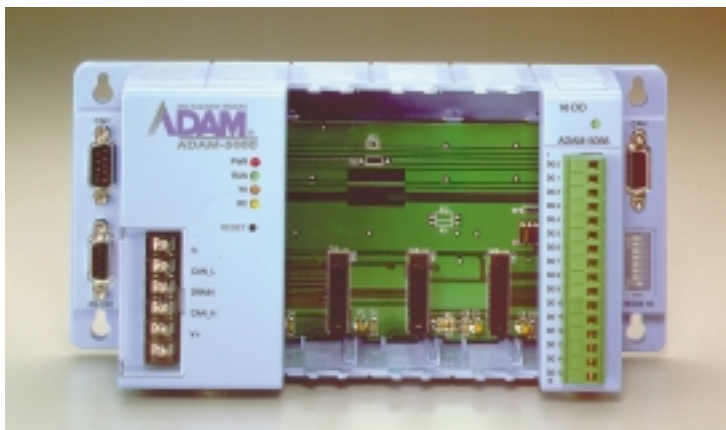
ций и выпуска первых CAN-компонентов как независимыми компаниями, так и ассоциациями по промышленной автоматизации непрерывно велась и продолжается до сих пор работа над созданием спецификаций протоколов верхнего уровня — HLP (Higher Level Protocol) для CAN-сетей.

Уже разработанные и существующие в настоящее время спецификации протоколов CAN HLP, как правило, имеют сжатую трехуровневую архитектуру (рис. 1), включающую в себя два базовых уровня CAN-протокола, иногда дополняемых спецификациями физического уровня (соединители,

кабели и т. п.), и прикладной уровень. Сервисные функции промежуточных уровней либо отсутствуют, либо включены в прикладной. Соблюдение полной иерархии уровней эталонной модели OSI/ISO в системах управления не требуется, кроме того, наличие дополнительных изолирующих межуровневых интерфейсов привело бы к потере производительности системы в режиме реального времени и сделало бы существенно менее предсказуемыми задержки прохождения сообщений в сети.

Преимущества использования стандартных HLP при разработке CAN-сетей очевидны и немалочисленны. Во-первых, в отличие от использования только сервисов ISO 11898 или Bosch 2.0A/B, вместе с тем или иным HLP разработчик получает в руки уже готовые механизмы передачи данных любой длины, процедуры начальной инициализации, распределения идентификаторов и т. п., а кроме этого, часто впри-

дачу и конкретную спецификацию физической среды: длина и топология шины, скорости передачи, типы кабелей, соединителей и т. п. — для своей области применения (например гидравлика, общественный транспорт), на подготовку



Устройство CAN интерфейса ADAM-5000/CAN фирмы Advantech, поддерживающее протоколы CANopen и DeviceNet



Контроллер DacNet фирмы Grayhill, поддерживающий протокол DeviceNet

и тестирование которой в реальных условиях уже потрачены силы большого числа разработчиков и экспертов. Во-вторых, появляется возможность интегрирования модулей сторонних производителей и простого наращивания сети в будущем, применения широкого спектра имеющихся на рынке инструментальных средств для того или иного HLP, что значительно снижает время и стоимость разработки и положительно сказывается на показателях надежности. В-третьих, протоколы HLP позволяют максимально эффективно задействовать многие преимущества CAN, особенно при работе в режиме реального времени. И, наконец, немалое число всевозможных групп пользователей и производителей оборудования для тех или иных HLP способны если не решить за разработчика его задачу, то уж, во всяком случае, значительно облегчить ему жизнь.

А многочисленность существующих CAN-протоколов прикладного уровня — на сегодня их уже более четырех десятков — наряду с наличием метапротоколов (например CAN Kingdom) в известной мере снимает проблему, связанную с обратной стороной любой стандартизации и заключающуюся в ограничении свободы системного разработчика.

Среди многообразия CAN HLP, представленных на современном рынке CAN-технологий, особого внимания заслуживают четыре поддерживаемых ассоциацией CiA и получивших наибольшее распространение в последнее время. Это CAL/CANopen, CAN Kingdom, DeviceNet и SDS (Smart Distributed System).

CAL (CAN Application Layer)

Одной из главных целей создания организации CiA в 1992 году была разработка и последующая поддержка открытого протокола прикладного уровня (7-й уровень модели OSI), предназначенного для CAN-сетей в сфере промышленной автоматизации. В качестве прототипа при разработке такого протокола был взят уже существовавший

в то время и положительно зарекомендовавший себя HLP, разработанный фирмой Philips. Результатом его апробации и последующего усовершенствования специальной рабочей группой CiA явилось опубликование в 1993 году спецификаций CAL — CAN Application Layer (CiA DS 20x). Фундаментом CAL служит канальный уровень CAN. CAL не является ориентированным на конкретные приложения стандартом протокола, не содержит каких-либо профилей, привязанных к конкретным устройствам или задачам, и не определяет содержание передаваемых данных, но предлагает стандартизованные элементы сетевого сервиса прикладного уровня. Решение же вопроса, какую часть из них использовать, находится в ведении разработчика. CAL включает в себя четыре составные части:

- спецификация CAN-сообщений (CMS — CAN Message Specification),
- сетевое управление (NMT - Network Management),
- распределение идентификаторов (DBT — Identifier Distributor),
- управление уровнем (LMT — Layer Management).

Спецификация CMS описывает типы объектов взаимодействия в рамках объектно-ориентированного подхода, правила передачи данных раз-

ных типов посредством CAN-фреймов, взаимодействие между модулями в терминах модели клиент-сервер, механизмы передачи данных, включая передачу пакетов длиной более 8 байтов. Сетевое управление построено на взаимодействии типа master-slave. Один модуль сети является NMT-мастером, все остальные — NMT-ведомые. Посредством сервисов управления NMT-мастер инициализирует, управляет NMT-ведомыми, которые желают принять участие во взаимодействии, и позволяет им общаться между собой посредством CMS-сервисов. Также в задачи сетевого управления входят контроль ошибок и конфигурирование устройств. Благодаря DBT-сервисам происходит бесконфликтное распределение идентификаторов среди модулей под контролем DBT-мастера. Посредством LMT-сервисов возможны запрос и изменение текущих параметров (значений идентификаторов, скорости передачи, битового квантования и т. п.) в модулях непосредственно из CAN-сети.

Сетевые CAN-приложения, основанные на прикладном уровне CAL, в настоящее время успешно работают в медицинской электронике, системах контроля дорожного движения, на транспорте, в промышленном оборудовании.

CANopen

Результатом дополнения CAL (точнее, некоторого его подмножества) системой профилей (устройств, интерфейсов, приложений и т. д.) и спецификациями физического уровня (типы соединителей, правила битового квантования, определяющие, на

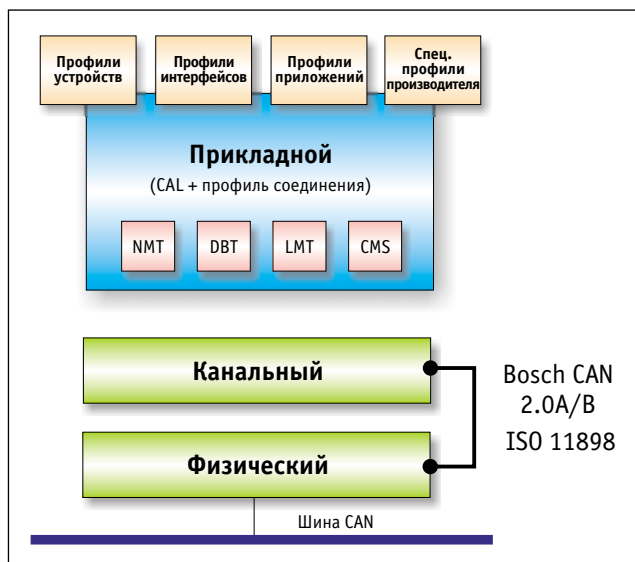


Рис. 2. Архитектура протокола CANopen

сколько квантов разделять бит и в каком месте бита считывать его значение, и т.д.) явилось появление более «конкретного» стандарта протокола CANopen. По существу, CANopen является одним из приложений прикладного уровня CAN, но единственным приложением подобного рода, поддерживаемым ассоциацией CiA. Профили устройств (CiA DS 40x) упрощают интеграцию модулей разных производителей в единую сеть, а определение минимального обязательного (mandatory) набора свойств модулей гарантирует работоспособность системы на базовом уровне.

Первоначально CANopen предназначался для сетей управления движущимися механизмами в системах промышленной автоматизации. Однако впоследствии он нашел применение в медицине, морской электронике, на транспорте и в системах автоматизации зданий.

Структура CANopen в соответствии с моделью OSI приведена на рис. 2. Два нижних уровня соответствуют стандарту CAN (ISO 11898, CAN Specification 2.0 A/B). В дополнение к спецификациям физического уровня ISO 11898 (среда передачи данных — экранированная или неэкранированная двухпроводная дифференциальная линия) CANopen содержит собственные правила битового квантования, а также определяет три рекомендуемых типа соединителей:

- 1) 9-контактный D-Sub (DIN 41652),
- 2) 5-контактный круглый Mini (ANSI/B93.55M-1981),
- 3) 5-контактное открытое клеммное соединение.

Рекомендуемой разводкой контактов для всех типов соединителей предусмотрена возможность подачи питания (положительной полярности) на трансиверы узлов, имеющих гальваническую развязку. В сети CANopen определены восемь градаций скоростей передачи данных: 1 Мбит/с, 800, 500, 250, 125, 50, 20 и 10 кбит/с. Поддержка скорости 20 кбит/с является обязательной для всех модулей.

Прикладной уровень представляет собой некоторое подмножество CAN и базируется на четырех его основных сервисных элементах: CMS, NMT, DBT и LMT, дополненных профилем соединения (CiA DS 301), определяющим базовые правила обмена данными и структуру словаря объектов. Более развитые механизмы сетевого взаимодействия для интеллектуальных устройств (человеко-машинные интерфейсы — HMI, PC-контроллеры,

PLC, инструментальные средства и т.п.) описаны в дополнении к коммуникационному профилю (CiA DS 302).

В сети CANopen на прикладном уровне модули обмениваются между собой объектами-сообщениями — COB (Communication Object), включающими в себя один или более CAN-фреймов. Всего существует четыре типа таких объектов:

- объекты данных процесса — Process Data Objects (PDO),
- объекты сервисных данных — Service Data Object (SDO),
- объекты специальных функций — Special Function Objects,
- объекты сетевого управления — Network Management Objects.

Собственно для целей передачи данных используются два различных механизма — с использованием PDO и на основе SDO. SDO позволяют модулям обмениваться данными любого объема (при последовательностях более 8 байтов — благодаря использованию нескольких CAN-фреймов) в ациклическом низкоприоритетном режиме. Как правило, этот тип обмена используется для конфигурирования устройств или настройки формата

PDO. Любое устройство, интегрируемое в сеть CANopen, должно обязательно поддерживать SDO-обмен. В противоположность SDO-типу, обмен на основе PDO используется для синхронной (циклической или ациклической) или асинхронной (инициируемой внешними прерываниями) скоростной передачи не более 8 байтов (длина поля данных фрейма CAN), имеет более высокий приоритет, чем SDO, и применяется для пересылки данных в режиме реального времени. Различия между этими двумя типами передачи данных подобны разнице между тяжелым грузовиком и быстрым, легким спортивным автомобилем.

Для выполнения специальных задач, в том числе диктуемых спецификой режима реального времени, служат объекты специальных функций:

- синхронизации — Synchronization Object (SYNC) — служит для запуска синхронных процессов,
- временных маркеров — Time Stamp Object — содержит значение абсолютного времени,
- аварийный — Emergency Object (EMCY) — служит для передачи кодов ошибок модулей.

Объекты сетевого управления включают сообщения сервисов NMT, LMT и DBT. Администрированием сети занимается NMT-мастер, который инициализирует устройства, обеспечивает контроль ошибок, а также производит их периодическую «переключку» (Life Guarding) с помощью PDO-сообщений (Node Guarding Object) для выявления узлов, находящихся в нерабочем состоянии ввиду физического отсутствия или отключения от шины (bus off) по счетчику ошибок.

Устройство в сети CANopen включает в себя три основные логические части:

- интерфейс связи и ПО протокола,
- словарь объектов,
- интерфейс ввода-вывода и прикладное ПО.

Первая часть обеспечивает прием-передачу объектов по сети. Словарь объектов описывает типы данных, объектов связи (COB) и прикладных объектов, используемых в данном устройстве. Третья часть обеспечивает внутреннюю функциональность устройства и взаимодействие с его аппаратным интерфейсом.

В целях максимального упрощения процесса интеграции модулей независимых производителей в единую сеть CANopen использует концепцию профилей устройств.

К настоящему времени завершено формирование следующих профилей:

- модули ввода-вывода (аналоговые и цифровые DSP-401),
- приводы и модули управления перемещением (DSP-402),
- элементы человеко-машинного интерфейса (DSP-403),
- измерительные устройства и регуляторы (WD-404),
- кодеры (DSP-406).

В процессе разработки находятся профили для модулей управления гидравлическими механизмами, дизельными двигателями и железнодорожным транспортом. Кроме этого, существует единственный пока профиль интерфейса — IEC 1131 (DSP-405). Отдельного упоминания заслуживает профиль приложения WD-407 (IBIS-CAN) CAN-сетей в области управления электроникой на общественном транспорте (где CAN-сети вообще используются довольно интенсивно по всей Европе): билетный контроль, подсчет пассажиров, информационные панели и т. п.

Другим менее известным протоколом-приложением прикладного уровня CAL (и, в отличие от CANopen, требующим лицензирования) является

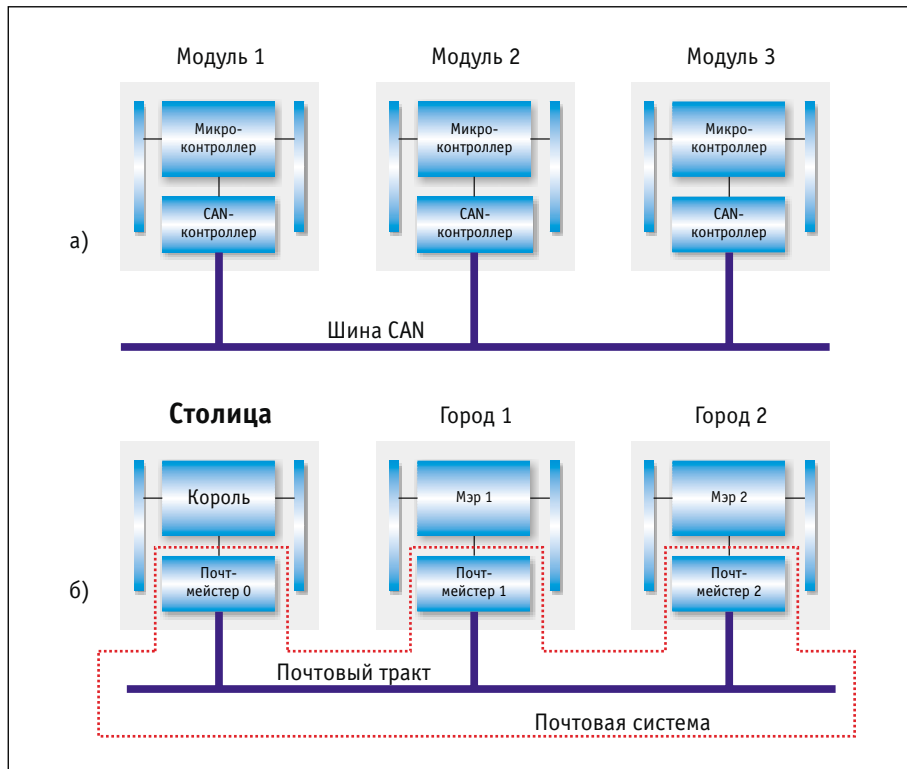


Рис. 3. Два взгляда на структуру CAN-сети:

а) традиционный,

б) с точки зрения протокола CAN Kingdom

протокол P-CAL (Portable CAN Application Layer), разработанный Университетом Вооруженных сил Германии.

CAN Kingdom

За довольно романтичным (CAN-королевство) названием протокола шведской компании KVASER-AB скрывается не менее красивая и оригинальная концепция сетевого взаимодействия устройств, выделяющая его на общем фоне других протоколов высокого уровня. Началу работ над первой версией (текущая — третья) протокола CAN Kingdom в 1990 году предшествовал многолетний опыт компании в области создания систем распределенного управления. Протокол был специально разработан для управления машинами и механизмами: промышленными роботами, текстильными станками, мобильными гидравлическими устройствами — и позволяет удовлетворить такие свои-ственные подобным приложениям требования, как

- эффективность функционирования в режиме реального времени,
- жесткие требования безопасности,
- высокая общая производительность.

CAN Kingdom является также основой американского военного стан-

дарта CDA 101 и широко используется в военной технике, от надувных лодок и систем наведения на цели до сверхзвуковых ракет и истребителей.

Основной целью создания протокола было предоставление системному разработчику максимальной свободы в реализации своих идей при построении сети, сохранив при этом возможность использования стандартных модулей независимых производителей. CAN Kingdom не является «готовым» протоколом в том смысле, в каком это справедливо, например, по отношению к стандартам типа CANopen или DeviceNet. Это скорее набор примитивов — метапротокол, с помощью которых можно «собрать» протокол для конкретной сети модулей, что позволяет достичь уникального сочетания простоты интеграции готовых модулей с высокой степенью «закрытости», защищенности оригинального протокола.

При разработке спецификации CAN Kingdom авторы отказались от принятого в подобных случаях и широко распространенного следования правилам взаимосвязи открытых систем OSI. Причина этого проста: семиуровневая модель OSI/ISO создавалась изначально для описания традиционных компьютерных сетей, телекоммуникационных, корпоративных, офис-



Семейство IBM PC совместимых контроллеров (CIF — Communication InterFace) фирмы Hilscher для построения сетей CANopen, DeviceNet и SDS

ных, которые предназначены не для работы в реальном масштабе времени, а для обслуживания пользователей, требования которых заранее (на этапе построения такой сети) неизвестны и непредсказуемы, и в процессе работы подвержены частым изменениям. (Справедливости ради следует отметить, что большинство протоколов компьютерных сетей также редко в точности следуют этой абстрактной модели, особенно в плане обособления и полной изоляции различных уровней сетевого сервиса.) В системах же управления реального времени ситуация прямо противоположная: на стадии разработки все коммуникационные потребности модулей должны быть известны. И готовая сеть должна функционировать точно так, как задумал системный разработчик. Краеугольным камнем концепции сетевого взаимодействия CAN Kingdom является принцип «Модули обслуживают сеть» (MSN — Modules Serves the Network), в отличие от принципа «Сеть обслуживает пользователей» (NSM — Network Serves the Modules), свойственного компьютерным сетям.

На этапе разработки сеть CAN Kingdom приспособляется к нуждам системы, что становится возможным, благодаря априорным знаниям о потребностях системы, где детерминизм функционирования модулей является условием обеспечения требований режима реального времени (необходимо, к примеру, знать, как долго сообщение может следовать от одного узла к другому).

Следствием принципа MSN также является и то, что в сети CAN Kingdom всегда должен существовать один модуль-супервизор, содержащий всю информацию о системе и ответственный за ее инициализацию.

Представление CAN-сети в терминах CAN Kingdom (в сравнении с традиционным взглядом) дано на рис. 3. В CAN Kingdom сеть CAN — это страна (королевство) со своей столицей (центральным контролирующим узлом) и провинциальными городами (это остальные узлы). Король (управляющая программа-супервизор) управляет всем королевством и отвечает за соблюдение закона и порядка в нем, а за местное управление (в пределах своего узла) отвечают мэры городов, то есть управляющие программы узлов. Каждый город экспортирует или импортирует продукцию-информацию посредством почты, которая циркулирует по почтовому тракту (CAN-шина) и проходит через почтмейстеров (CAN-контроллеры). Типы почтовой корреспонденции (информация, передаваемая по сети) и ее соответствие CAN-понятиям таковы:

ПИСЬМО	CAN-фрейм (данных или удаленного запроса)
КОНВЕРТ	CAN-идентификатор
СТРАНИЦА	Поле данных CAN-фрейма
СТРОКА	Байт данных
ЭЛЕМЕНТ СТРОКИ	Бит данных

Для организации и хранения входящей и исходящей «корреспонденции» применяются понятия форм, документов, папок и листов.

Столь неформальный язык описания протокола отнюдь не является праздным — он позволяет любому специалисту, далекому от вычислительной техники или электроники, например биологу, химику или врачу, благодаря интуитивно-понятному описанию сети (как должны функционировать общество или страна, примерно представляют себе все), сознательно формулировать технические условия и иметь представление о принципах ее функционирования. Вероятно, любой российский разработчик способен припомнить случаи, когда представители заказчика, иногда даже из близких к вычислительной технике областей, испытывали серьезные трудности при формулировании ТЗ на разработку.

Перечислим некоторые особенности CAN-системы на базе протокола CAN Kingdom.

- Распределение CAN-идентификаторов находится под полным контролем разработчика. Возможно динамическое распределение идентификаторов. Допускается использо-

вание как стандартного, так и расширенного формата CAN-фрейма.

- Максимальное время прохождения любого сообщения в сети предсказуемо.
 - Во время начальной инициализации системы происходит обязательный этап настройки (setup) протокола, включая построение форматов данных, начиная с битового уровня, методов управления шиной, распределение идентификаторов и т. д.
 - В системе всегда должен присутствовать (как минимум, до завершения настройки протокола) супервизор (король), производящий инициализацию системы, контроль подключенных узлов и т. д. Ни один модуль не может принимать участие в сетевом обмене без разрешения короля.
 - Перед инициализацией сети каждый модуль (город) должен иметь свой номер (CAN Kingdom не описывает конкретный способ установки номера модуля — это может быть DIP-переключатель, энергонезависимая память или конфигурация соединителя) и знать идентификатор сообщения инициализации (королевское письмо) и скорость передачи данных в сети.
 - В сеть CAN Kingdom возможна интеграция любых CAN-модулей (включая разработанные для других протоколов, например DeviceNet или SDS), удовлетворяющих стандарту ISO 11898.
 - Не существует каких-либо рекомендуемых скоростей передачи данных. Но в первые 200 мс после подачи питания узел обязан настроиться на прослушивание шины на скорости 125 кбит/с. Допустимы отличающиеся от ISO 11898 спецификации физического уровня.
- Наличие одного центра-короля, который содержит всю информацию о системе, избавляет от использования профилей устройств, часто применяемых в других HLP.
- Правила идентификации модулей основаны на использовании международного кода EAN/UPC, включающего код производителя и продукта. Система распознает только авторизованные системным разработчиком модули. Неавторизованный модуль не получит в свое распоряжение CAN-идентификаторов от короля при инициализации сети. Для поддержки режима plug&play король хранит информацию о том, какие модули и при каких обстоятельствах могут быть добавлены в систему.

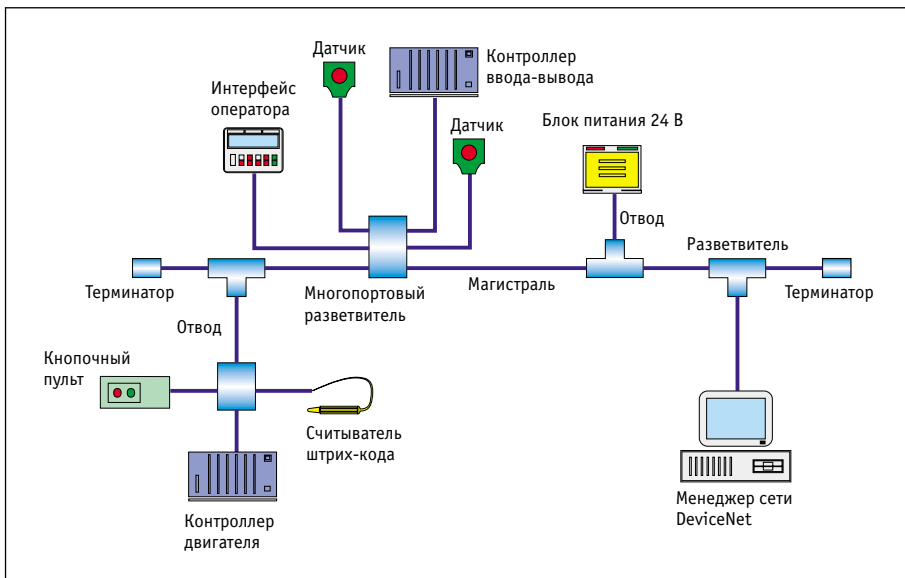


Рис. 4. Пример сети DeviceNet

Среди возможностей CAN Kingdom, способствующих повышению эффективности реализации режима реального времени, можно отметить гибкость режимов передачи и упаковки данных, включая использование поля арбитража для передачи данных, объединение узлов в группы, поддержку часов реального времени, различных режимов доступа к шине.

DeviceNet

DeviceNet — протокол, разработанный и опубликованный в 1994 году компанией Allen-Bradley и впоследствии переданный в ведение специальной организации для его поддержки ассоциации ODVA (Open DeviceNet Vendor Association Inc.). DeviceNet — недорогое, простое и эффективное решение для объединения в единую систему разнообразных устройств промышленной автоматизации независимых производителей (фото-, термодатчики, стартеры, считыватели штриховых кодов, элементы человеко-машинного интерфейса: клавиатуры, дисплейные панели, — наряду с управляющими устройствами: PLC, компьютерами и т. д. — рис. 4). Первые устройства, удовлетворяющие спецификации DeviceNet, появились на рынке в начале 1995 года. Помимо снижения стоимости, при разработке протокола также стояла задача упрощения и унификации диагностики подобных устройств, часто либо физически недоступных, либо допускающих такую диагностику посредством своих собственных, весьма отличающихся между собой интерфейсов. Как и все CAN HLP, протокол DeviceNet построен на двух нижних

уровнях стандарта CAN, дополненных более детальными, чем в других HLP, спецификациями физического уровня.

Сеть DeviceNet имеет шинную топологию с отводами. Физической средой передачи является 4-проводной кабель (CAN_H, CAN_L, Vcc, Ground), причем возможны две его разновидности: толстый (внешний диаметр 12,2 мм) и тонкий (6,9 мм). Оба варианта кабеля могут использоваться как для основной магистрали (транка), так и для отводов или комбинироваться. Определены лишь три значения скорости передачи данных — 125, 250 и 500 кбит/с. Максимальные длины центральной магистрали и отводов в зависимости от скорости передачи и типа кабеля приведены в табл. 1.

Важной особенностью сети DeviceNet является возможность питания модулей непосредственно от сетевого кабеля, причем стандартизованы как напряжение питания (24 В), так и максимальная токовая нагрузка (8 А на толстом кабеле, 3 А на тонком), а также допускается применение нескольких (в отличие от других стандартов на базе CAN, которые вообще предусматривают питание от шины) источников питания, например с целью резервирования, в любой точке шины. Все это дает возможность построения автономной сети, не зависящей от наличия или качества внешне-

го питания, а при необходимости позволяет легко демонтировать и снова развернуть систему на новом месте. Сеть DeviceNet допускает «горячее» (без обесточивания сети) подключение и отключение модулей. При наличии оптоэлектронной развязки сигнальных цепей в модулях их питание может осуществляться от внешнего источника.

Спецификацией DeviceNet предусмотрены и такие нюансы, как типы, цвет и количество индикаторов состояния модуля (включения, работоспособности, подключения к сети), хотя само по себе наличие таких индикаторов не является обязательным. Стандарт DeviceNet содержит также подробное описание многочисленных типов переходников, разветвителей (одиночных и многопортовых), соединителей (mini, micro), сетевых отводов и т. п.

В целях еще большего снижения стоимости системы на базе сети DeviceNet не так давно фирмой Allen-Bradley был предложен новый тип кабельной разводки на основе 4-проводного плоского кабеля — KwikLink.

При описании организации типов данных, сетевого поведения модулей в DeviceNet используется объектно-ориентированная модель. Обязательные классы объектов включают в себя следующие:

- объект удостоверения (Identity object) содержит информацию о модуле (код производителя, продукта, версия и т. п.);
- объект соединения (Connection object) — логический порт ввода-вывода устройства;
- объект DeviceNet включает MAC ID (идентификатор модуля), скорость передачи, состояние модуля и т. п.;
- объект сообщения (Message router object) перенаправляет явное сообщение получателю.

При передаче данных в сети DeviceNet эффективно используется принцип адресации CAN-протокола с ориентацией на потребителя, и узлы выбирают «свои» передаваемые в сети данные по их идентификаторам. Всего определены два типа сообщений:

- сообщения ввода-вывода (I/O messages) предназначены для целей управления устройствами и передачи данных в реальном времени между

Таблица 1. Соотношения предельной длины и скоростей передачи данных сети DeviceNet

Скорость передачи, кбит/с	Длина магистрали, м		Длина отводов, м	
	толстый кабель	тонкий кабель	одиночных	суммарная
125	500	100	6	156
250	250	100	6	78
500	100	100	6	39

узлами в широковещательном режиме или в режиме «точка-точка». Используют идентификаторы с высоким приоритетом, которые и определяют содержание сообщения;

- явные сообщения (Explicit messages) предназначены для многоцелевого обмена данными в режиме «точка-точка» и обеспечивают типичный сервис «запрос-ответ», используют идентификаторы с низким приоритетом и применяются обычно для конфигурирования устройств и целей диагностики. Значение сообщения содержится в поле данных.

При необходимости передачи данных длиной более восьми байтов применяется механизм фрагментации. В зависимости от потребностей обмена и возможностей модулей возможны мастер-ведомый (master-slave), мультимастерный (multi-master) или равноправный (peer-to-peer) способы взаимодействия устройств. Пересылки данных могут инициироваться путем опроса, циклически или при изменении их значения (change of state). Максимальное число узлов в сети DeviceNet — 64. Такое ограничение связано с 6-разрядным двоичным форматом идентификатора модуля MAC ID (он является частью CAN ID, причем в DeviceNet используется только стандартный тип CAN-фрейма с 11-разрядным ID). Однако общее число устройств ввода-вывода может достигать 2048 (по 32 на узел).

Модули в сети могут быть как UCMM-типа (UnConnected Message Manager), способные выставлять равноправные (peer-to-peer) соединения с другими модулями, так и Predefined Master/Slave типа, которые не могут произвольно выбирать путь соединения, и их объекты соединения конфигурируются при включении устройства. Реализация последнего типа модуля требует меньшей длины кода и производительности управляющего микроконтроллера, что снижает общую стоимость устройства.

В сети DeviceNet не всегда устройство с меньшим значением идентификатора модуля — MAC ID (он составляет лишь часть CAN-идентификатора) выигрывает арбитраж. Это зависит и от того, к какой группе принадлежит сообщение. Всего таких групп четыре (в порядке убывания приоритета):

- 1) наиболее критичные ко времени доставки сообщения,
- 2) явные и сообщения ввода-вывода для соединения типа Predefined Master/Slave,

Таблица 2. Предельные значения длин магистрали и отводов сети SDS

Максимальная длина магистрали, м	Скорость передачи, кбит/с	Максимальная длина отводов, м	Максимальное количество узлов
22,8	1000	0,3	32
91,4	500	0,9	64
182,8	250	1,8	64
457,2	125	3,6	64

3) несрочные сообщения, используемые для диагностики и мониторинга,

4) сообщения для off-line подключения используются на этапе инсталляции модулей.

Для обеспечения стыкуемости устройств разных производителей и их взаимодействия в рамках единой сети стандарт DeviceNet, подобно некоторым другим HLP, определяет ряд профилей устройств. Формированием и стандартизацией библиотек профилей занимаются специальные группы (Special Interest Groups) ассоциации ODVA.

Более 285 производителей-членов ассоциации ODVA занимаются разработкой и производством устройств, инструментальных средств и программного обеспечения для сетей DeviceNet.

SDS (Smart Distributed System)

SDS — детище компании Honeywell Inc. (Micro Switch Division). Наряду со стандартом DeviceNet, SDS представляет собой еще одно недорогое и законченное решение для сетевого управления интеллектуальными датчиками и актуаторами от центрального контроллера (PLC, компьютера) в системах промышленной автоматизации.

По степени завершенности — от спецификаций физической среды до прикладного уровня — и по ориентировке на снижение стоимости системы SDS-стандарт напоминает DeviceNet, а функционирование сети походит на работу сети DeviceNet в режиме Predefined Master/Slave.

Архитектура протокола SDS включает в себя три уровня модели

OSI/ISO — физический, канальный и прикладной. Шинная топология представляет собой линейную шину (магистраль или транк) с короткими отводами (рис. 5). Определены два базовых типа кабельной разводки: Mini (применяемый при сборке транка сети) — 4-проводной кабель с максимальной токовой нагрузкой 8 А, 5-контактный разъем, и Micro (для подключения физических устройств к сети) — 4-проводной кабель, 3 А, 4-контактный разъем без отдельного контакта для экрана кабеля.

В сети SDS допускается и обычная проводная разводка с использованием открытых клеммных соединителей. Всеми типами кабельной разводки и соединителей, так же как и в сети DeviceNet, предусмотрено подведение питающего напряжения (диапазон 11-25 В на стороне устройства) к узлам. Предельные значения длин магистрали и отводов сети SDS в зависимости от скорости передачи приведены в табл. 2. Дробные представления длин в метрах связаны с прямым пересчетом их величин, выраженных в футах.

Сообщения, циркулирующие в сети SDS, носят название APDU (Application layer Protocol Data Unit) — блоки данных протокола прикладного уровня. APDU представляет собой CAN-фрейм стандартного формата (расширенный формат фрейма в SDS-сети не применяется), элементы которого имеют свое собственное назначение в SDS (рис. 6). В поле арбитража (ID3-ID9) расположен 7-разрядный адрес устройства (максимально допустимое количество устройств в сети SDS — 126). Тип APDU (3-разрядное поле) определяет тип

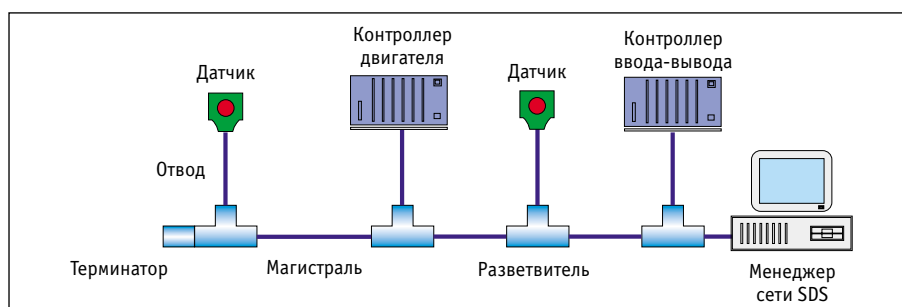
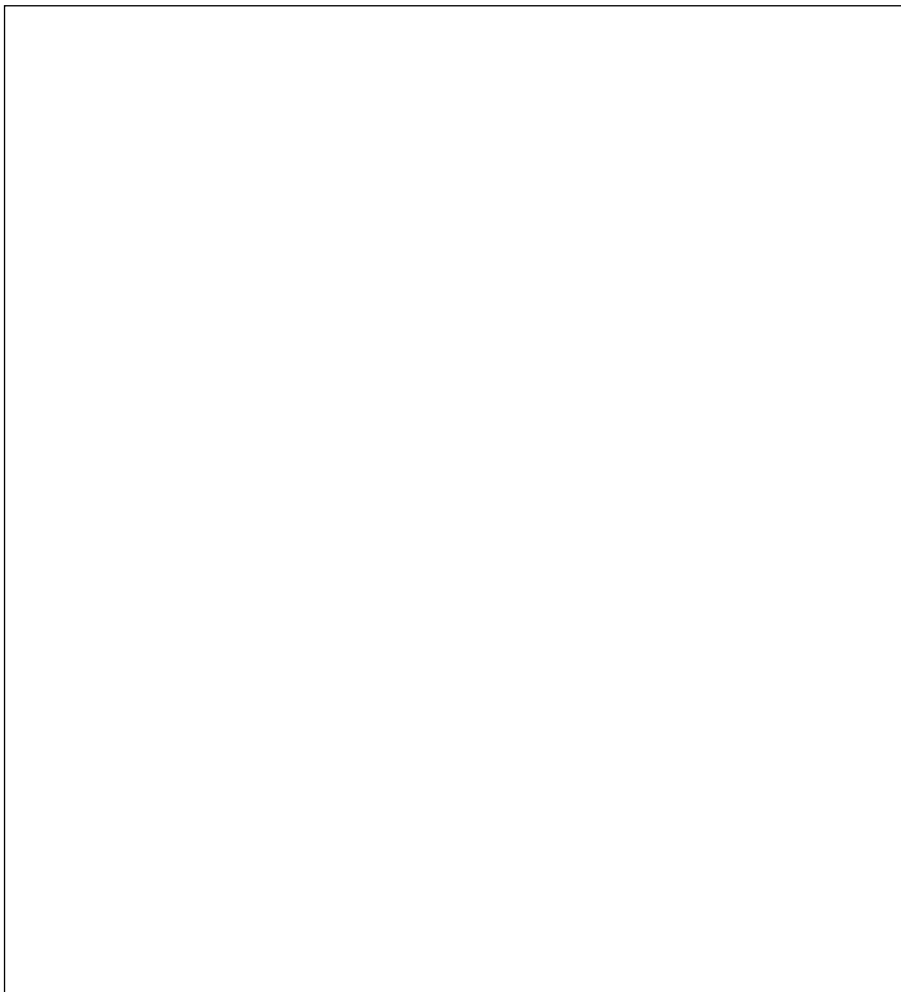


Рис. 5. Пример сети SDS



сервиса (0...7) прикладного уровня, которому соответствует данный APDU. Нулевое значение бита ID10 (DIR) поля арбитража указывает, что адрес устройства (device address) является адресом назначения, а единичное — адресом источника. Чем ниже значения логического адреса, тем выше приоритет сообщения. Бит RTR в SDS CAN-фреймах всегда имеет нулевое значение (удаленный CAN-фрейм в SDS-спецификации не применяется). Блок APDU имеет две формы — укороченную и длинную. Укороченная форма APDU содержит в поле DLC все нули и для передачи данных не используется. В поле данных длинной формы APDU содержится код длины (2...8) поля данных CAN-фрейма (2), два первых байта которого содержат спецификатор сервиса (Service Specifier), идентификатор встроенного объекта (EOID) и дополнительные параметры сервиса, а оставшиеся шесть предназначены для передачи собственно данных. При необходимости передачи последовательностей данных более шести байтов используется фрагментированный формат (до 64 фрагментов по 4 байта) длинной формы APDU.

Укороченная форма APDU используется в следующих сервисах прикладного уровня:

- Change of State (Off, On, Off ACK, On ACK) — обнаружение изменения состояния логического устройства,
- Write (On State, Off State, On State ACK, Off State ACK) — управление состояниями логического устройства.

К сервисам, использующим длинную форму APDU, относятся следующие:

- Channel — обеспечение как широковещательного (multicast), так и равноправного (peer-to-peer) каналов соединения,
- Connection — открытие/закрытие индивидуальных типов соединения,
- Write — чтение атрибутов объектов устройства,

- Read — изменение атрибутов объектов устройства,
- Action — команда объекту устройства выполнить действие,
- Event — сигнализация объектов устройства о событии.

При инициализации взаимодействия модулей сети SDS используются 4 сервисные функции-примитива:

- Запрос (Request) — генерация APDU устройством-инициатором соединения,
- Ответ (Response) — ответный APDU устройства-ответчика,
- Индикация (Indication) — фиксация факта приема APDU устройством-ответчиком,
- Подтверждение (Confirm) — подтверждение приема APDU устройством-инициатором.

Сеть SDS всегда требует наличия единственного мастера-менеджера сети, как минимум, на этапе включения для выполнения автонастройки скорости передачи модулей.

В процессе работы сети допускаются наличие нескольких мастеров на шине, но они должны функционировать в пределах своих адресных доменов, а при включении сети только один из них может брать на себя функцию сетевого менеджера для автонастройки скорости устройств.

Модули с внешним питанием (не от SDS-шины) должны иметь механизм обнаружения пропадания питания шины для блокировки своей активности и выполнения автонастройки скорости после повторного включения сети. В сети SDS возможны четыре скорости передачи данных: 1 Мбит/с, 500, 250 и 125 кбит/с.

7	6	5	4	3	2	1	0	Позиция
ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	Байт 1
ID2	ID1	ID0	RTR	DLC (Data Length Code)				Байт 2

Заголовок стандартного CAN-фрейма

7	6	5	4	3	2	1	0	Позиция
DIR	Device Address							Байт 1
APDU Type		0	DLC (Data Length Code)					Байт 2

Заголовок APDU сети SDS

Рис. 6. Заголовок APDU сети SDS

Сравнение протоколов. Прочие HLP

Несмотря на все разнообразие представленных на рынке протоколов верхнего уровня, включая не рассмотренные в данной статье, все они решают в целом ряд очень похожих между собой задач, описанных в начале статьи, — распределение идентификаторов, передача данных более 8 байтов и т. п. Задачи эти возникают в связи с функциональной незавершенностью CAN-спецификаций, ограниченных описанием лишь двух нижних уровней сетевого взаимодействия. Тем не менее, различия в способах их решения в тех или иных HLP приводят, в конечном счете, к различиям, порой весьма существенным, в стоимостных и функциональных характеристиках сетей на их основе, что необходимо учитывать при выборе HLP для конкретного приложения. Далеко не последнюю роль играет и поддержка того или иного HLP со стороны производителей CAN-оборудования и инструментальных средств.

Самым простым и компактным вариантом объединения несложных промышленных устройств под управлением одного мастера является стандарт SDS. Несколько более развитые сервисы предоставляет спецификация DeviceNet. Наибольшей гибкостью и возможностью максимально эффективной реализации режима реального времени обладает протокол CAN Kingdom. В отличие от трех других рассмотренных протоколов, CAN Kingdom не касается каких-либо аспектов физического уровня (среда, соединители и т. п.), выходящих за рамки стандарта ISO 11898, и представляет собой высокоуровневую надстройку над канальным уровнем CAN. В таблицу 3 сведены некоторые характеристики четырех рассмотренных в статье HLP.



Контроллер WAGO I/O System. Для перехода на тот или иной CAN-протокол (CANopen, CAL или DeviceNet) достаточно установить соответствующий сетевой адаптер

Среди других прикладных CAN-протоколов, получивших признание в последнее время, можно выделить стандарт SAE J1939 (SAE — Society of Automotive Engineers), пришедший на смену более старому J1708/J1587 и предназначенный для управления в режиме реального времени узлами транспортных средств (грузовики, автобусы), реализующий plug&play режим для модулей и использующий расширенный формат (29-битовый идентификатор) CAN-фрейма. Ряд специализированных групп (например, HUG — Hydraulic Users Group) в области промышленной автоматизации работают над собственными дополнениями уже существующих CAN HLP в целях адаптации их параметров к своим областям применения.

Следует отметить, что большинство существующих на рынке HLP, включая рассмотренные в данной статье, находятся в процессе развития и далеки от завершения, особенно в плане формирования библиотек профилей (для тех HLP, в которых они определены), в связи с непрерывным расширением областей применения CAN-сетей.

Заключение

В последние годы во всем мире наблюдается стремительный рост числа разработок CAN-сетей и расширение

спектра областей применения CAN-технологий.

По информации ассоциации CiA, если в 1996 году в мире было установлено 11 млн. CAN-узлов, в 1997 — 25 млн., то в 1998 — уже более 59 миллионов. Прогнозируемое число на 1999 год — около 83 млн., а на 2000 год — более 125 млн. узлов. Эти прогнозы не учитывают все возрастающий интерес к сетям CAN со стороны североамериканских производителей, а также крупнейших юго-восточных корпораций. Непрерывно расширяется и предложение готовых модулей, а также инструментальных программных и аппаратных средств для тех или иных стандартов прикладных протоколов.

В подобной ситуации вопрос — использовать или не использовать стандартный CAN HLP — переходит в иную плоскость: какой из существующих HLP предпочесть для решения той или иной задачи, поскольку только на основе стандартного и правильно выбранного HLP зачастую становятся возможными создание конкурентоспособной продукции, интеграция в одной сети готовых модулей, экономия средств и времени на разработку самой сети и ряд других, уже упомянутых ранее преимуществ. ●

Таблица 3. Сравнительные характеристики четырех CAN HLP

	CANopen	Can Kingdom	DeviceNet	SDS
Допустимые скорости передачи данных, кбит/с	10, 20 (обязательная), 50, 125, 250, 500, 800, 1000	Любые до 1000, инициализация на 125	125, 250, 500	125, 250, 500, 1000
Защита от некорректной установки скорости передачи модулей	Нет	Да	Нет	Да
Автонастройка скорости передачи	Нет	Возможна, но не определена	Возможна, но не определена	Да
Допустимые номера узлов	0-127	0-255	0-63	0-125
Поддержка расширенного CAN-фрейма	Нет	Да	Нет	Нет
Наличие профилей устройств	Да (CiA SIG)	Нет	Да (ODVA SIG)	Да (Honeywell Inc.)
Поддержка протокола	CiA (CAN in Automation)	KVASER AB	ODVA (Open DeviceNet Vendor Assoc.)	Honeywell Inc.
Спецификация соединителей	Да	Нет	Да	Да