

# ПРОТОКОЛ CANopen ДАЛЬНЕЙШЕЕ РАЗВИТИЕ CAN

В феврале 1987 года Роберт Бош (Robert Bosch) на конгрессе Ассоциации инженеров автомобилестроения представил протокол последовательной связи CAN (Controller Area Network), который позволял заменить дорогостоящую электропроводку автомобиля более дешевым сетевым кабелем. Протокол обеспечивал высокую надежность и поддерживал быстрое действие, требуемое для управления электронными устройствами автомобиля, в частности противоблокировочной тормозной системой и воздушными подушками безопасности\*. Получив широкое распространение, особенно в Европе, и не только в электронных системах автомобилей, протокол CAN послужил основой для разработки в 1987–1991 годы открытого стандарта связи высокого уровня CANopen для встроенных сетей с гибкими конфигурационными возможностями. CANopen, первоначально предназначенный для применения в сетях управления движущимися механизмами (в первую очередь в промышленных системах автоматизации), сегодня используется в самых разнообразных областях – сетевом медицинском оборудовании, внедорожных транспортных средствах (автопогрузчиках, карах и т.п.), бортовых корабельных системах, общественном транспорте.

## ИСТОРИЧЕСКАЯ СПРАВКА

Предложенная Робертом Бошем в 1987 году система CAN оперировала короткими, длиной до 8 байт, сообщениями, поддерживала доступ в сеть многочисленных устройств (конфликты разрешались с помощью приоритетов) и отличалась высокой надежностью: каждое сообщение контролировалось 15-бит циклическим избыточным кодом. Уже в середине 1987 года фирма Intel выпустила первую CAN-микросхему модели 82526, выполнявшую все функции, необходимые для поддержки последовательной связи (передачу и прием сообщений, их фильтрацию, поиск передающего объекта и прерывания) при минимальном взаимодействии с хост-микроконтроллером или процессором. Вскоре микросхему CAN-контроллера 82C200 выпустила и компания Philips Semiconductor. Европейское отделение ISO приняло CAN в качестве сетевого протокола высокоскоростной передачи, и

\* Лапин А. Интерфейс CAN. Слагаемые успеха. – ЭЛЕКТРОНИКА: НТБ, 2005, №2, с.40–43.



В.Майская

в 1991 году CAN-интерфейс был реализован в электронных системах автомобилей марки Mercedes Benz класса S. Сегодня микросхемы с CAN-интерфейсом выпускают более 20 фирм, и почти каждый легковой автомобиль, производимый в Европе, имеет по крайней мере одну CAN-сеть.

Хотя протокол CAN первоначально предназначался для применения в легковых автомобилях, он получил широкий спрос в самых разнообразных секторах рынка, особенно европейского. Уже в начале 90-х потребовалась стандартизация различных появившихся к тому времени решений по его реализации. В 1992 году была сформирована международная Ассоциация пользователей и производителей CAN-систем – CAN in Automation (CiA). Одной из первых задач CiA стала разработка стандарта прикладного уровня CAN (CAL), поскольку CAN-протокол, касающийся реализации уровня канала передачи данных, не содержал стандартов, описывающих процессы обмена объектами данных на прикладном уровне. Но, хотя протокол CAL и нашел применение в промышленных системах, каждому пользователю приходилось разрабатывать свой коммуникационный профиль в соответствии с конкретным приложением.

С 1993 года в рамках программы Европейского Сообщества Esprit, направленной на расширение исследований и ускорение внедрения информационных технологий, был образован консорциум во главе с компанией Bosch, целью которого стала разработка сетевого протокола на основе CAL, предназначенного для построения сетей производственных участков. Протокол получил название CANopen. После завершения проекта спецификация CANopen была передана CiA для дальнейшей доработки и технического обслуживания, и в 1995 году был выпущен полностью отредактированный и дополненный протокол CANopen. Окончательный вариант CANopen определял не только прикладной уровень и коммуникационный профиль (CiA DS 301), но и структуру программируемых систем и приборов (CiA 302 – структуру приборов и CiA 303-1 – рекомендации по выбору кабелей и соединителей), интерфейсные и прикладные профили (CiA 303-2). Благодаря этому всего за пять лет протокол CANopen стал ведущей стандартизированной сетью в Европе, а в конце 90-х годов его начали использовать во многих отраслях промышленности (производители печатающих устройств, корабельного, медицинского оборудования и другие).

Параллельно стандарту ЕС разработка CAN-протоколов высокого уровня велась и в США. Эти работы закончились появлением двух похожих, по крайней мере нижними коммуникационными уровнями, протоколов – DeviceNet и Smart Distributed System (SDS). Протокол DeviceNet, продвигаемый компанией Allen-Bradley (теперь входит в фирму Rockwell Automation), стал основным протоколом шинных систем средств промышленного производства в США, тогда как SDS практически остался внутренним протоколом компании-разработчика – Honeywell Microswitch. В результате сейчас существуют два стандартизованных, описываемых международным стандартом



EN 50325-4 (на базе CiA DS 301) прикладных уровня – DeviceNet, оптимизированный для заводских систем автоматизации, и CANopen, предназначенный в основном для встроенных сетей управления машинами и механизмами всех видов.

### CANopen – МЕХАНИЗМЫ СВЯЗИ И ПРОТОКОЛЫ

CANopen – сетевая технология, оптимизированная для применения в промышленных системах управления, и встроенных системах и сетях. Основа нижнего уровня CANopen – протокол CAN, реализуемый многими представленными на рынке дешевыми устройствами с низкой и средней производительностью. Поэтому CANopen удовлетворяет требованиям, предъявляемым к сетевым средствам связи оборудованием на основе 8- и 16-бит микроконтроллеров и системами с низкими энергозатратами.

Функциональные возможности протокола CANopen позволяют ведущему сетевому устройству конфигурировать каждый узел сети. При этом конфигурационные параметры задают режим работы прибора и определяют область записи данных сообщения (поле данных), конкретное сообщение (идентификатор сообщения) и время его запуска.

CANopen-устройство – устройство широкого применения, соединяющее CAN-интерфейс со специализированным портом ввода-вывода с помощью словаря объектов (Object Directory, OD). Каждое CANopen-устройство имеет свой OD, определяющий средства реализации приложения в терминах данных и конфигурации. Словарь разделен на две секции. Первая содержит информацию об устройстве (идентификатор устройства, имя изготовителя, параметры связи и т.п.), вторая описывает функциональные возможности устройств (рис. 1). Сетевой доступ к прикладным объектам устройства (входным и выходным сигналам, его параметрам и функци-

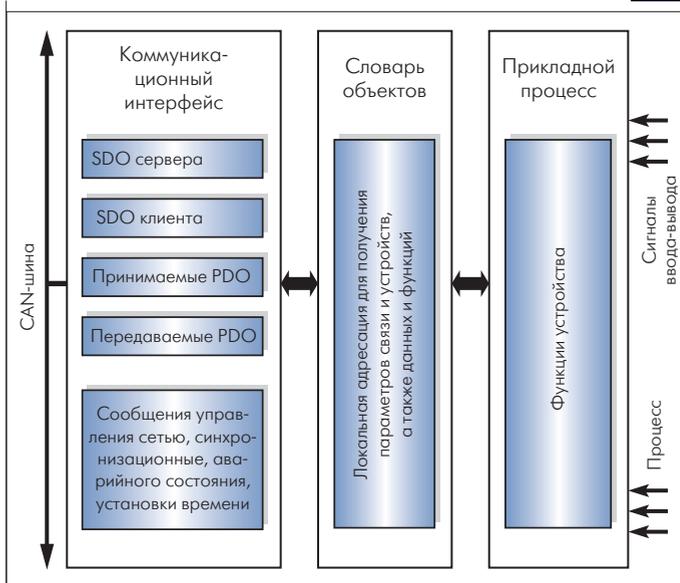


Рис. 1. Структура протокола CANopen

ям, переменным сети) предоставляют точки входа ("объект") в словаре объектов через так называемые объекты данных обслуживания (Service Data Objects, SDO). Точку входа ко всем коммуникационным объектам идентифицирует 16-бит индекс и 8-бит субиндекс (для матричных и структурных объектов). SDO используются прежде всего для передачи параметров в процессе конфигурирования устройства, а также для передачи сообщений. Протокол передачи SDO допускает любой размер объекта. Первый байт первого сегмента содержит необходимую информацию об управлении потоком данных, следующие три байта первого сегмента данных – индекс и субиндекс счи-

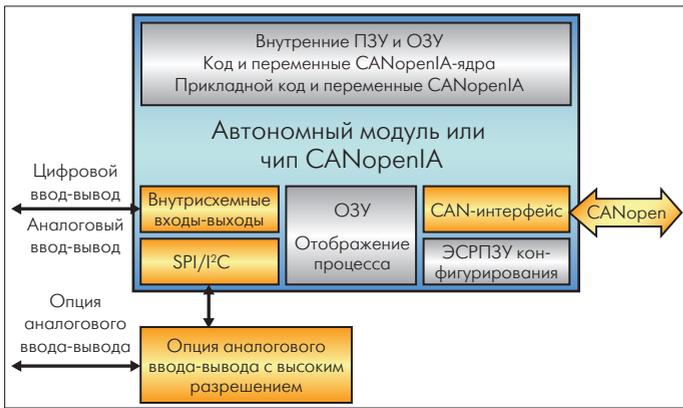


Рис.2. Блок-схема автономной системы CANopenIA

тываемого или записываемого входа словаря объектов. Последние четыре байта первого сегмента предназначены для данных пользователя. Второй и последующие сегменты (с тем же CAN-идентификатором) содержат управляющий байт и до семи байтов данных пользователя.

Помимо механизма передачи данных большой длины через SDO, протокол CANopen, подобно другим полевым шинам, определяет второй основной механизм передачи – высокоскоростной обмен сообщениями малой длины (максимум 8 байт) в реальном времени через так называемые объекты данных процесса (Process Data Objects, PDO), имеющих свой уникальный CAN-идентификатор (заголовок) сообщения. Идентификатор определяет приоритет сообщения, что гарантирует малое время отклика сетевых устройств (1,16–1,48 мс для одного ведомого устройства и сообщения длиной 1 байт). Передавать PDO может только один узел сети, принимать – несколько узлов (изготовителя/потребителя). Как правило, запуск передачи вызывает изменение состояния данных процесса, регулярно повторяющаяся команда таймера (например, передавать каждые 100 мс), запрос удаленного устройства или поступившее синхронизационное сообщение. Передача через PDO выполняется широкоэмитально без дополнительных издержек протокола.

Для сообщения о сбоях устройства стандарт CANopen резервирует инициируемые событиями высокоприоритетные аварийные сообщения (CAN-фрейм с 8-байт полем данных), способные вызвать прерывание. Аварийное сообщение пересылается только один раз при выявлении "факта ошибки". Если дальнейших сбоев прибора не фиксируется, дополнительные аварийные сообщения не передаются.

Центральное сообщение установки времени определяет общесистемное для прикладных устройств время вида "время-дата". Размер поля данных этого сообщения 6 байт. Функциональные возможности, требуемые для подготовки и старта распределенной системы автоматизации, задаются определенным в протоколе CANopen механизмом управления сетью (Network Management, NMT). В NMT входят протоколы начальной загрузки, циклической защиты узлов и готовности связи (Node/Life-guarding и Heartbeat protocols), а также собственно протокол NMT. Ведомая NMT машина состояний системы CANopen задает исходное состояние, предпусковой, рабочий статусы и статус останова. При включении питания каждое устройство CANopen-системы находится в исходном состоянии и автоматически переходит в предпусковое состояние, в котором разрешается передача SDO. Если для одного или более узлов ведущим NMT-узлом устанавливается рабочее состояние, они могут передавать и получать PDO. В состоянии останова могут передаваться только NMT-сообщения.

Скорость передачи задается протоколом CANopen в диапазоне от 10 Кбод до 1 Мбод при максимальной дальности от 5000 до 40 м, соответственно. Технические условия CiA DS 301 определяют специ-

альные функции передачи для 2048 возможных CAN-сообщений. Схема установления связи основана на принципе ведущего/ведомого и позволяет устанавливать соединение типа точка-точка между одним ведущим и 127 устройствами узла. Поддерживается и синхронизированная передача данных типа ведомый-ведомый.

**РАЗВИТИЕ ПРОТОКОЛА CANopen**

CANopen-сети находят самое широкое применение: встраиваемые системы управления станками, транспортные средства, медицинская аппаратура, информационные табло и даже игровые автоматы и профессиональные кофеварки. Сети CANopen можно найти в системах управления лифтами и эскалаторами, в морозильных установках, тепло- и электростанциях. Популярность стандарта CANopen объясняется простотой интеграции совместимых с ним устройств, а также возможностью включения CANopen-устройств в уже существующие CAN-сети. К тому же, на рынке широко представлены готовые средства проектирования, тестирования, конфигурирования и технического обслуживания CANopen-систем. CiA постоянно совершенствует протокол CANopen, выпуская новые спецификации для конкретных приложений: CiA 400 для систем многоуровневых сетей, CiA 415 для систем датчиков управляемых механизмов, используемых при дорожном строительстве. Но не только CiA продвигает этот стандарт.

Большое внимание развитию CANopen уделяет Академия встроенных систем (Embedded Systems Academy, ESAcademy) – организация, изучающая микроконтроллерные архитектуры и средства их проектирования, а также такие вопросы, как своевременный выход на рынок, улучшение качества и реализация CAN, CANopen и построение встроенных Интернет-сетей. С целью минимизации сроков проектирования при включении в CANopen-сеть новых, а также существующих датчиков или исполнительных механизмов ESAcademy предложена концепция реализации CANopenIA, обеспечивающая мгновенный доступ ("Instant Access") к CANopen-сети благодаря упрощению процесса построения ее узлов. Предусмотрены два решения CANopenIA: автономный, непосредственно реализующий полный CANopen приборный профиль без применения дополнительного контроллера (рис.2), и сопроцессорный, реализующий приборный профиль для конкретного приложения и преду-

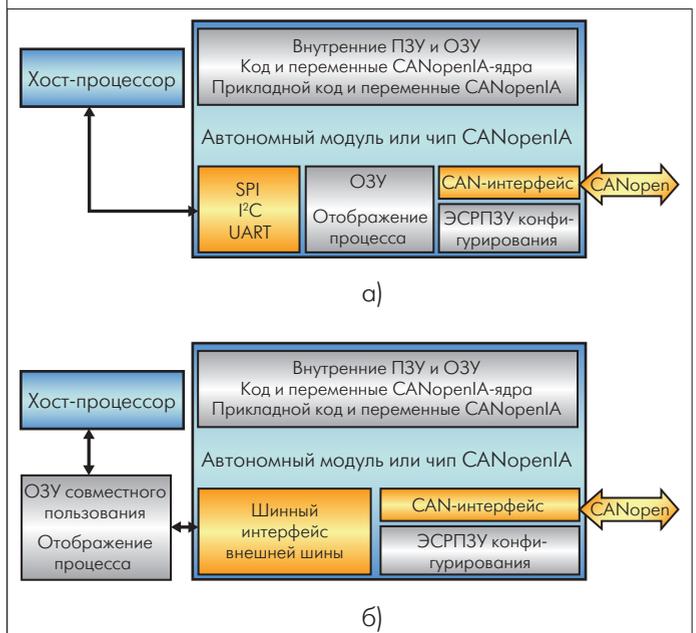


Рис.3. Блок-схема сопроцессорной системы CANopenIA, реализованной с помощью канала последовательной связи (а) или двухпортовой памяти совместного пользования (б)

сма­три­ва­ю­щий при­ме­не­ние хост-кон­трол­ле­ра с до­ступом ко всем пе­ре­мен­ным, пе­ре­да­ва­е­мым в се­ти (рис.3а,б). Авто­ном­ное ре­ше­ние CANopenIA обес­пе­чи­ва­ет са­мый ко­рот­кий ци­кл раз­ра­бот­ки, по­сколь­ку узлы се­ти не­по­сред­ствен­но ре­а­ли­зуют мо­ду­ли или чи­пы. Ци­ф­ро­вые и ана­ло­го­вые вхо­ды-вы­хо­ды вы­пол­не­ны в ис­поль­зу­е­мом ми­кро­кон­трол­ле­ре. SPI- и I<sup>2</sup>C-ин­тер­фей­сы обес­пе­чи­ва­ют до­ступ к до­пол­ни­тель­ным ана­ло­го­вым ком­по­не­нтам, ха­рак­те­ри­зу­е­мым вы­со­ким раз­ре­ше­нием. ЭСРПЗУ или дру­гая эн­ер­го­не­зави­си­мая па­мять со­дер­жит ин­фор­ма­цию о ком­по­нов­ке хра­ни­мых в ОЗУ дан­ных ото­бра­же­ния про­цес­са и кон­фи­гу­ра­ции се­те­вой сис­те­мы, а так­же по­ря­док до­сту­па к дан­ным про­цес­са че­рез SDO и PDO.

В со­про­цес­сор­ном ре­жи­ме мо­дуль или чи­п CANopenIA пред­став­ля­ет со­бой со­про­цес­сор хост-про­цес­со­ра или ми­кро­кон­трол­ле­ра. Связь ме­жду CANopenIA мо­ду­лем/чи­пом и хо­стом обес­пе­чи­ва­ют по­сле­до­ва­тель­ные ка­на­лы (че­рез UART, I<sup>2</sup>C или SPI-ин­тер­фейс).

В слу­чае бо­ль­шо­го объ­ема дан­ных или не­об­хо­ди­мо­сти об­нов­ле­ния дан­ных с вы­со­кой ча­сто­той в со­про­цес­сор­ном ре­жи­ме мож­но при­ме­нять со­вмес­тно ис­поль­зу­е­мое двух­пор­то­вое ОЗУ (рис.3б).

Для мно­гих встро­ен­ных се­те­вых при­ло­же­ний CANopen яв­ля­ет­ся пред­поч­ти­тель­ным ре­ше­нием. Од­на­ко для при­ло­же­ний, в ко­то­рых ис­поль­зу­ет­ся ли­шь не­сколь­ко уз­лов и со­об­ще­ний, за­час­тую при­хо­дит­ся раз­ра­ба­ты­вать соб­ствен­ный про­то­кол вы­со­ко­го уров­ня, и пол­ный стек про­то­ко­лов CANopen в та­ких слу­ча­ях из­бы­то­чен. Это по­бу­ди­ло ESACademy пред­ло­жить спе­ци­фи­ка­цию MicroCANopen, по­зво­ля­ю­щую ми­ни­ми­зи­ро­вать на­бор стан­дар­ти­зи­ро­ван­ных средств, не­об­хо­ди­мых для обес­пе­че­ния свя­зи, а так­же су­ще­ствую­щих средств про­ек­ти­ро­ва­ния ус­тройств се­ти CANopen, та­ких как кон­фи­гу­ра­то­ры и ана­ли­за­то­ры. Воз­мож­но на­ра­щи­ва­ние се­те­вой сис­те­мы MicroCANopen до "пол­но­го" CANopen без из­ме­не­ния ка­на­лов свя­зи. Для ре­а­ли­за­ции про­то­ко­ла MicroCANopen ми­кросхе­ма на ба­зе кон­трол­ле­ра 8051 долж­на рас­по­ла­гать 4-Кбит па­мя­тью ко­да и 170-байт ОЗУ (в пол­но­фор­мат­ных CANopen сис­те­мах объ­ем па­мя­ти про­грамм со­став­ля­ет 40–60 Кбит, ОЗУ – бо­лее 600 байт). Сис­те­ма MicroCANopen тре­бу­ет два ин­тер­фей­са: с ап­па­рат­ны­ми сред­ства­ми (CAN-кон­трол­ле­ром) и при­клад­ны­ми про­грам­ма­ми.

#### ЭЛЕМЕНТНАЯ БАЗА

Од­ни из но­вей­ших ми­кросхе­м с рас­ши­рен­ны­ми CAN воз­мож­но­стя­ми – вы­пу­щен­ные в кон­це 2005 го­да ком­па­ни­ей Atmel ма­ло­мо­щ­ные 8-бит КМОП-ми­кро­кон­трол­ле­ры на ос­но­ве ус­о­вер­шен­ство­ван­ной AVR RISC-ар­хи­тек­ту­ры ти­па AT90CAN32 и AT90CAN64. Бла­го­да­ря вы­пол­не­нию ко­ман­д за один так­то­вый ци­кл про­из­во­ди­тель­ность AT90CAN32/64 дос­ти­га­ет 1 Mips/MГц, со­став­ляя 16 Mips на ча­сто­те 16 МГц. Та­ким об­разом раз­ра­бот­чик мож­ет оп­ти­ми­зи­ро­вать по­треб­-

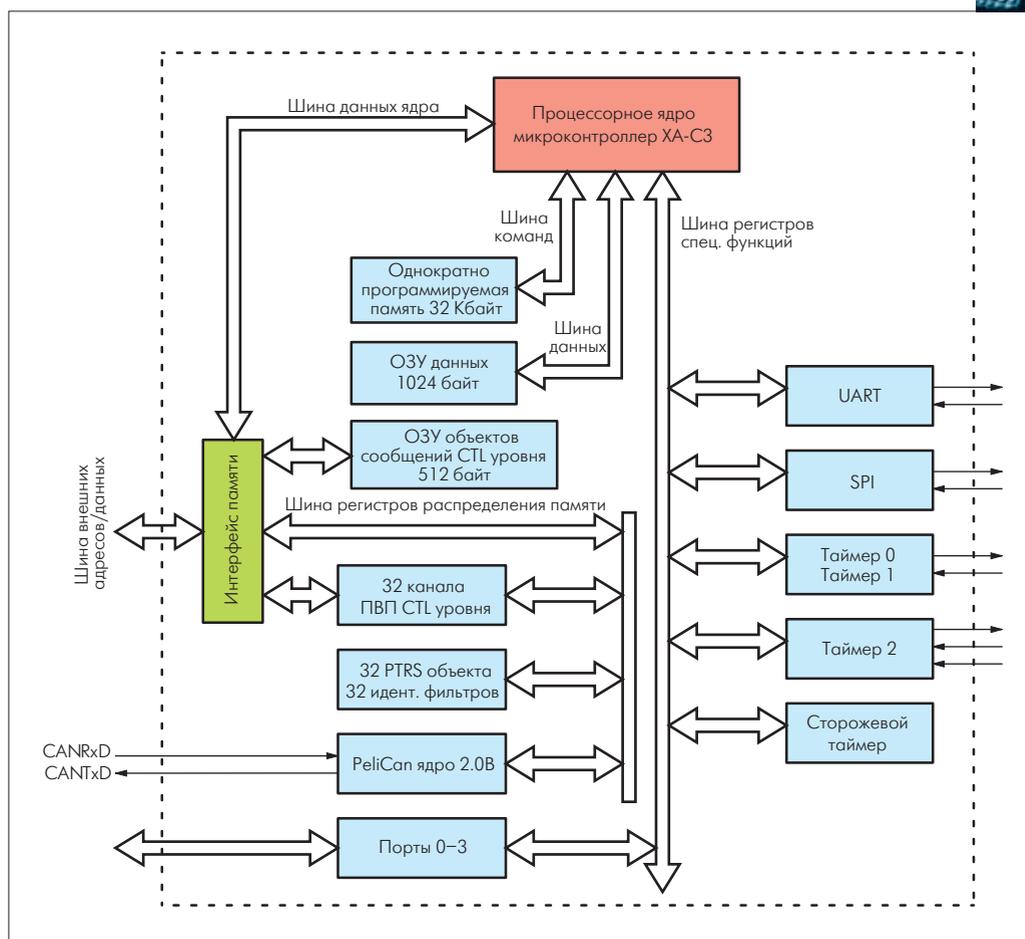


Рис.4. Блок-схема микроконтроллера HA-C3

ляемую мощ­ность се­те­вой сис­те­мы за счет вы­бо­ра ско­ро­сти об­ра­бот­ки. AVR-ядро со­че­та­ет бо­га­тый на­бор ко­ман­д с 32 уни­вер­саль­ны­ми ре­гис­тра­ми, ко­то­рые не­по­сред­ствен­но под­клю­че­ны к АЛУ, что по­зво­ля­ет при вы­пол­не­нии од­ной ко­ман­ды по­лу­чать до­ступ к двум не­зави­си­мым ре­гис­трам. Особен­но­стью но­вых ми­кросхе­м яв­ля­ет­ся воз­мож­ность про­грам­ми­ро­ва­ния в хо­де вы­пол­не­ния при­клад­ной за­да­чи (In-Application Programming, IAP). Бла­го­да­ря это­му лю­бой CAN-узел мож­но пе­ре­про­грам­ми­ро­вать че­рез CAN-ши­ну, не на­ру­ша­я опе­ра­ции об­ме­на или без прерыва­ния вы­пол­ня­е­мых опе­ра­ций. В со­став ми­кросхе­м вхо­дят: 32-/64-Кбайт про­грам­ми­ру­е­мая в сис­те­ме флэш-па­мять; 1-/2-Кбайт ЭСРПЗУ; 2-/4- Кбайт СОЗУ; 53 об­щих ли­ний вво­да-вы­во­да; 32 уни­вер­саль­ных ре­гис­тра; CAN-кон­трол­ле­р, от­ве­ча­ю­щий тре­бо­ва­ни­ям стан­дар­та CAN v2.0A/v.2.0B; ча­сы ре­аль­но­го вре­ме­ни; че­ты­ре гиб­ких тай­ме­ра/сче­т­чи­ка с воз­мож­но­стью срав­не­ния и ШИМ;

два USART; двухпроводной последовательный интерфейс; восьмиканальный 10-бит АЦП; программируемый сторожевой таймер; SPI-порт; JTAG-интерфейс. Предусмотрена возможность выбора пяти энергосберегающих режимов работы.

По производительности микросхемы AT90CAN32/64 на основе AVR RISC-процессора сопоставимы с 16-бит микроконтроллером, а по цене – со стандартными 8-бит устройствами. Расположенный на кристалле CAN-контроллер может принять до 15 независимых объектов сообщений, перепрограммироваться "на ходу" и динамически назначить прием, передачу или запись в буферы (при приеме нескольких CAN-фреймах). Его максимальная скорость передачи на частоте 8 МГц составляет 1 Мбит/с. Маскирование, фильтрация и буферизация CAN-фреймов снижает нагрузку хост-процессора. При выполнении протоколов высокого уровня (CANopen, DeviceNet) производительность и объем памяти микросхем большей частью доступны для приложения. Ускоритель прерываний оповещает процессор о критических сообщениях без выполнения подпрограммы сканирования программных средств, что минимизирует непроизводительные затраты приложений реального времени.

Напряжение питания микросхем 2,7–5,5 В, максимальная частота 8 МГц при напряжении 2,7 В и 16 МГц при 4,5 В. Поставляются в 64-выводных корпусах TQFP- и QFN-типов. Основное назначение микросхем – CAN-сети, в том числе системы управления лифтами, эскалаторами, дверными модулями транспортных средств, системы промышленной автоматизации.

В качестве узла ввода-вывода цифровых и аналоговых данных, отвечающего требованиям спецификации CiA DS401, можно отметить микросхему семейства XA-C3 компании Philips Semiconductor (рис.4). Микросхема относится к семейству микроконтроллеров с расширенной архитектурой, объединяя стандартные периферийные устройст-

ва с PelICAN 2.0В машиной (Philips' Extended Library стандарта CAN 2.0В), поддерживающей 11- и 29-бит идентификаторы, производительность до 1 Mips и CAN-диагностику. Микросхема поддерживает протоколы DeviceNet и CANopen аппаратными средствами, а не отдает их на откуп разработчику для реализации программными средствами. Это позволяет существенно (до 90%) сократить непроизводительные издержки процессора и использовать менее 2% его ресурсов для обработки сообщений, оставляя почти 100% ресурсов процессорного ядра для прогона прикладных программ. При этом используются 32 объекта сообщений CAN транспортного уровня (CTL), которые можно конфигурировать для передачи или приема, а также 32 отдельных фильтра, каждый из которых способен выполнить распознавание 30-бит идентификатора и полное 29-бит маскирование для представления уникальной адресной группы. Для каждого объекта сообщения имеется 512-байт буфер сообщений, не зависящий от внутрисхемного ОЗУ данных. Объем буфера за счет внешней памяти может быть увеличен до 8 Кбайт. В состав микросхемы также входят 16-бит процессорное ядро с 24-бит адресуемыми пространствами программ и данных; 21 16-бит регистр ядра, каждый из которых способен выполнять арифметические и логические операции, будучи при этом регистром указателя памяти; три стандартные счетчика/таймера с возможностями автоматической перезагрузки и ШИМ-выходами; сторожевой таймер; 32-Кбайт однократно программируемое ЭСППЗУ программ и 1-Кбайт СОЗУ данных, UART и SPI-порт, обеспечивающие интеграцию различных последовательных интерфейсов в CAN-сеть; четыре 8-бит порта ввода-вывода с четырьмя программируемыми выходными конфигурациями на вывод.

Микросхема работает на частоте 32 МГц при напряжении питания 4,5–5,5 В. Монтируется в 44-выводной корпус типа PLCC или LQFP. ○