



Возможности CAN-протокола

Евгений Карпенко

Введение

CAN-протокол был разработан фирмой Robert Bosch GmbH для использования в автомобильной электронике, отличается повышенной помехоустойчивостью, надежностью и обладает следующими возможностями:

- конфигурационная гибкость,
- получение сообщений всеми узлами с синхронизацией по времени,
- неразрушающий арбитраж доступа к шине,
- режим мультимастер,
- обнаружение ошибок и передача сигналов об ошибках,
- автоматическая передача сбойных сообщений при получении возможности повторного доступа к шине,
- различие между случайными ошибками и постоянными отказами узлов с возможностью выключения дефектных узлов,
- работает по витой паре на расстоянии до 1 км.

Естественно, что все эти качества делают CAN-протокол весьма привлекательным для использования в производственных приложениях, тем более что он поддерживается рядом фирм-производителей микросхем, выпускающих недорогие устройства, которые аппаратно реализуют требования CAN-протокола и работают в широком температурном диапазоне.

CAN-протокол распространяется на следующие уровни.

- Объектный уровень обеспечивает фильтрацию сообщений и обработку сообщений и состояний.
- Транспортный уровень представляет собой ядро CAN-протокола. Он отвечает за синхронизацию, арбитраж, доступ к шине, разделение посылок на

фреймы, определение и передачу ошибок и минимизацию неисправностей.

- Физический уровень определяет, как именно будут передаваться сигналы, их электрические уровни и скорость передачи.

Физический уровень

Физический уровень определяется стандартом ISO 11898 и характеризуется следующими возможностями.

Дифференциальное включение приемопередатчиков обеспечивает подавление синфазной помехи, при этом уровень сигналов составляет 1/3 от значения напряжения питания, причем само напряжение питания не определяется жестко. Например, типичные значения при напряжении питания +5 В приведены на рис. 1, причем доминирующим уровнем является нижний уровень, а рецессивным, соответственно, верхний.

- Максимальное расстояние между узлами — до 1 км.
- Скорость обмена до 1 Мбит/с при длине линии 60 м.
- Возможность применения гальванической развязки, причем гальваническая развязка может устанавливаться



Рис. 1. Сигнальные уровни на CAN-шине

либо между приемо-передающим буфером и микросхемой, обеспечивающей функции CAN, либо между микросхемой и остальной системой (рис. 2).

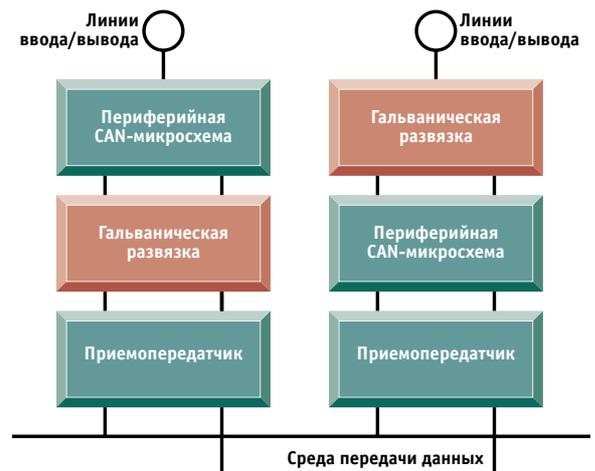


Рис. 2. Два типа гальванической развязки

Типы фреймов в CAN-протоколе

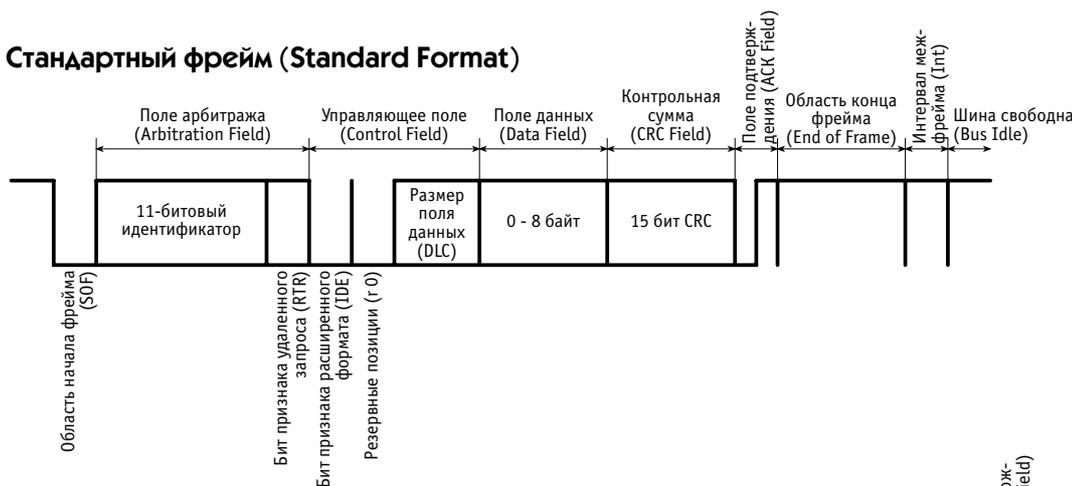
В CAN-протоколе определены следующие типы фреймов:

- фрейм данных перемещает данные с передатчика на приемник (приемники);
- удаленный фрейм запрашивает передачу фрейма данных, связанного с определенным идентификатором;
- фрейм ошибки выражает, какой узел обнаружил ошибку шины/сети;
- фрейм перегрузки обеспечивает задержку между передачей фреймов, чтобы управлять потоком данных.

Рассмотрим подробнее фрейм данных (рис. 3).

Он состоит из стартового поля SOF, поля арбитража Arbitration Field, управляющего поля Control Field, поля данных Data Field, поля контрольной суммы CRC,

Стандартный фрейм (Standard Format)



Расширенный фрейм (Extended Format)

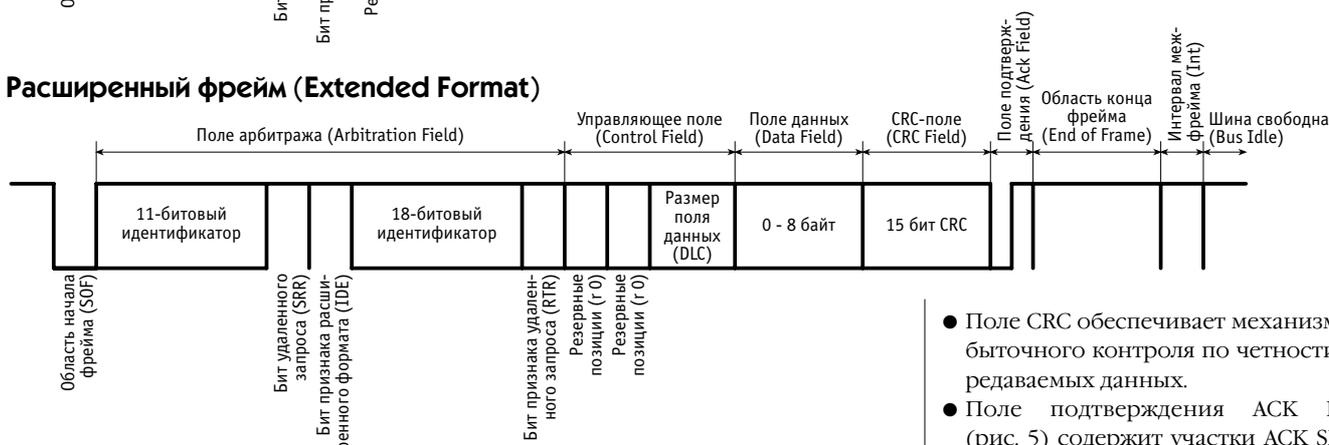


Рис. 3. Фрейм данных

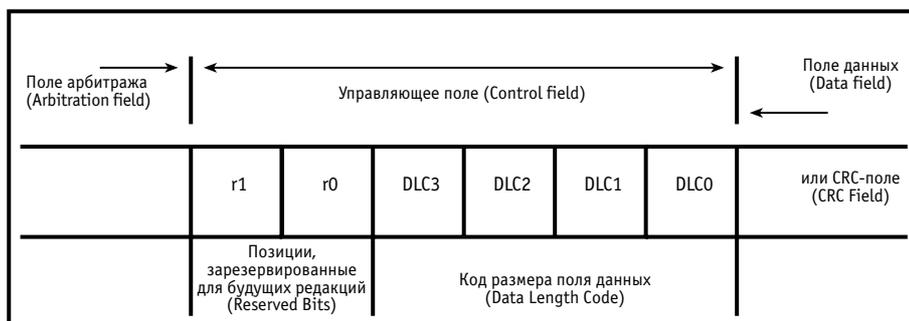


Рис. 4. Управляющее поле фрейма данных

поля подтверждения ACK Field, поля конца фрейма EOF.

- Поле SOF (Start of Frame) находится в начале фрейма данных и удаленного фрейма и содержит один доминирующий бит.
- Поле арбитража Arbitration Field содержит 11-битовый идентификатор и RTR-бит, показывающий, является данный фрейм фреймом данных или удаленным фреймом. Идентификатор предназначен для адресации сообщений и используется механизмом арбитража.
- Управляющее поле Control Field (рис. 4) содержит 6 битов, из которых 4 бита (DLC0-DLC4) составляют поле Data Length Code, показывающее ко-

личество байтов данных, которое будет передаваться в поле данных; два других бита зарезервированы для следующих редакций протокола.

- Поле данных Data Field содержит передаваемые данные, причем количество передаваемых байтов указывается в поле Control Field и не может превышать 8.

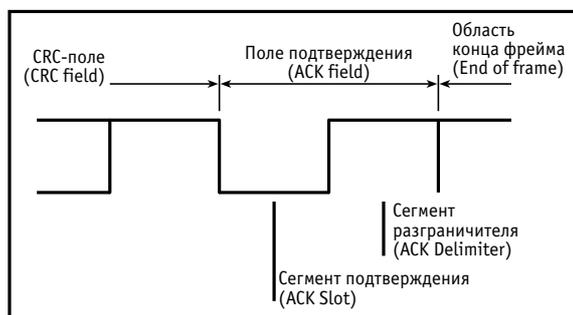


Рис. 5. Поле подтверждения фрейма данных

- Поле CRC обеспечивает механизм избыточного контроля по четности передаваемых данных.
- Поле подтверждения ACK Field (рис. 5) содержит участки ACK Slot и ACK Delimiter и выполняет следующую функцию: передающий узел посылает по одному рецессивному биту на каждом из участков, а приемник, если он принял сообщение без сбоев, устанавливает на линии доминирующий бит в поле ACK Slot. При наложении рецессивного и доминирующего уровней на линии устанавливается доминирующий, и это событие сигнализирует передающему узлу о том, что передача прошла нормально и повтор не требуется.
- Поле конца фрейма EOF содержится в фрейме данных и удаленном фрейме и состоит из семи рецессивных битов. Удаленный фрейм аналогичен по структуре фрейму данных, но не имеет поля данных, а фрейм ошибок и фрейм перегрузки содержат по 2 поля: в первом располагаются флажки ошибок и служебная информация, а второе является полем разграничителя Delimiter, и содержит восемь рецессивных битов.

Средства управления доступом к шине в CAN-протоколе

Передающий узел в CAN-протоколе слышит ВСЕ другие узлы в сети и подтверждают это. Всякий раз, когда шина свободна от передачи, узел может начинать передавать. Если узел передает, эта передача должна быть завершена прежде, чем другой узел может пытаться передать. Если два или больше узла на-

чинают передавать в одно и то же время, конфликт решается при помощи неразрушающего (non-destructive) поразрядного алгоритма арбитража, использующего поле арбитража.

Поле арбитража, включенное во все фреймы данных, состоит из

- 11-битового поля идентификатора,
- RTR-бита.

RTR-бит указывает, является ли фрейм фреймом данных или удаленным фреймом.

11-битовое поле идентификатора передается от старшего к младшему значащему биту. Доминирующий уровень — логический 0. Одновременная передача бита с доминирующим уровнем (логический 0) и бита с рецессивным уровнем (логическая 1) дает в результате уровень логического 0.

В течение передачи поля арбитража каждый передатчик контролирует текущий уровень на шине и сравнивает это с битом, который он должен передавать. Если значения равны, узел способен затем продолжить передачу. Если бит с пассивным уровнем (логический 1) был передан, а активный бит (логический 0) обнаружен на шине, то данный узел теряет право передачи и должен прекратить передачу последующих данных (рис. 6). Узел, который потерял шину, может сделать попытку передачи снова, когда текущая передача завершена.

Важно следующее: идентификатор с самым низким значением выигрывает арбитраж.

Из сказанного можно сделать следующие выводы.

Приоритетным является не передающий или приемный узел, а сообщение, имеющее меньшее значение идентификатора. Если в сети один из узлов (сервер) будет ответственным за принятие решений, то он должен иметь наименьший адрес из задействованных.

Вторая возможность, которую дает механизм арбитража, использована в сети верхнего уровня DeviceNet. В этой сети количество узлов ограничено 64 и для адресации отведены младшие разряды идентификатора, а старшие разряды предназначены для кодирования видов сообщений. Естественно, что сообщение, имеющее 0 в старшем бите, захватит шину первым, независимо от адреса узла приемника. Это, в свою очередь, обеспечивает передачу сообщений первого вида, например об аварии, по сети первыми, независимо от адресов приемных и передающих узлов.

Адресация в CAN-протоколе

CAN — это протокол, ориентированный на использование в условиях помех.



Рис. 6. Пример поразрядного арбитража

Различные сообщения, передающиеся по сети, имеют идентификатор, и каждая станция решает, основываясь на этом идентификаторе, получать или нет это сообщение. Этот идентификатор определен в поле идентификатора CAN-фрейма.

При этом адрес приемника устанавливается в самом приемнике путем настройки входных фильтров соответствующих микросхем.

Входные фильтры представляют собой решетку, или идентификационные экраны. Любое сообщение, которое проходит через входные фильтры, должно быть обработано процессором обслуживания CAN-контроллера. Чем большее количество единиц может быть отфильтровано, тем меньше нагрузка на процессор.

Микросхемы, поддерживающие CAN-протокол, могут иметь одиночный фильтр или многократные фильтры, в зависимости от конкретной реализации.

Существуют следующие два типа входных фильтров:

- фиксированные — фильтры, которые требуют, чтобы биты соответствовали точно один к одному (one-for-one).
- Mask-and-Match (маскируемые) — фильтры, которые применяют маску к полю идентификатора, прежде чем он сравнивается с приемным регистром кода.

Например, на рис. 7 регистр маски сконфигурирован так, что полученные биты 10-6 идентификатора должны со-

ответствовать битам 10-6 в приемном регистре кода. В этом примере биты 10-6 идентификатора должны быть установлены в 11110, а остальные не имеют значения. Если биты 10-6 установлены в 11110, то эти сообщения принимаются независимо от значений битов 5-0.

Управление ошибками

CAN-протокол обеспечивает механизмы обнаружения следующих типов ошибок.

- *Разрядная ошибка* появляется, когда передатчик сравнивает уровень на шине с уровнем, который должен передаваться, и обнаруживает их неравенство. При этом обнаружение активного бита, когда передается пассивный бит, не выдает ошибку в течение передачи поля арбитража, поля ACK Slot или флажка пассивной ошибки.
- *Ошибка подтверждения* возникает, когда передатчик определяет, что сообщение не было подтверждено. Слот подтверждения существует внутри фреймов данных и удаленных фреймов. Внутри этого слота все приемные узлы, независимо от того, являются они пунктом назначения или нет, должны подтвердить получение сообщения.
- *Ошибка заполнения* появляется, когда узел обнаруживает шесть (6) последовательных битов одного и того же значения. В процессе нормальной работы, когда передатчик обнаруживает, что он послал пять (5) последова-

Фильтрация типа Mask-and-Match											
10	9	8	7	6	5	4	3	2	1	0	Поразрядное значение идентификатора
1	1	1	1	0	0	0	0	1	1	1	Принятое значение идентификатора
mm	mm	mm	mm	mm	X	X	X	X	X	X	Регистр маски
1	1	1	1	0	0	0	0	0	0	0	Регистр кода после фильтрации
Принятое сообщение											

mm — код маски
 x — произвольный код («1» или «0»)

Рис. 7. Пример поразрядного маскирования Mask-and-Match

тельных битов одного и того же значения, он заполняет следующий бит противоположным значением (это называется заполнением бита). Все приемники удаляют заполненные биты до вычисления CRC (контрольного кода). Таким образом, когда узел обнаруживает шесть (6) последовательных битов того же значения, возникает ошибка заполнения.

- **CRC-ошибка** появляется, когда CRC-значение (контрольный код) не соответствует значению, сгенерированному передатчиком. Каждый фрейм содержит поле контрольного кода, которое инициализировано передатчиком. Приемники вычисляют CRC и сравнивают его со значением, сгенерированным передатчиком. Если эти два значения не тождественны, то имеет место CRC-ошибка.
- **Ошибка формы** возникает, когда недопустимое разрядное значение обнаружено в области, в которую должно быть передано predetermined значение. В CAN-протоколе существуют некоторые predetermined значения, которые должны быть переданы в определенных местах. Если недопустимое разрядное значение обнаружено в одной из этих областей, имеет место ошибка формы.

CAN позволяет минимизировать негативные последствия наличия дефектного узла в сети при помощи механизма определения состояния узла. Узел может быть в одном из трех состояний ошибки.

- **Ошибка активная** фиксируется, когда активный узел обнаруживает одну из упомянутых ошибок, он передает активный фрейм ошибки, который состоит из шести (6) последовательных доминирующих битов. Эта передача отменит любую другую передачу, проходящую в то же самое время, и заставит все другие узлы обнаружить ошибку наполнения, которая, в свою очередь, заставляет их отбрасывать текущий фрейм.

Когда узел в состоянии активной ошибки обнаруживает проблему с передачей, он предотвращает получение всех других данных из пакета сообщений, передавая фрейм активной ошибки. Этот процесс выполняется независимо от того, был ли узел, обнаруживающий ошибку, получателем данных или нет.

- **Ошибка пассивная** фиксируется, когда пассивный узел обнаруживает одну из упомянутых ошибок, — он передает фрейм пассивной ошибки, который состоит из шести (6) последовательных пассивных битов. Этот

фрейм может быть наложен на передачу, которая ведется в то же самое время, при этом данные из передачи не теряются, если другие узлы не обнаруживают ошибку.

- **Шина выключена** — узел на шине в выключенном состоянии и не откликается на любое воздействие на шине. Это логическое отключение от сети.

Общий краткий обзор действий, имеющих в механизме минимизации неисправностей, приведен далее.

- Узлы следят, передают и получают значения счетчиков ошибок.
- Узел начинает передачу в состоянии активной ошибки со счетчиками ошибок, равными нулю (0). Узел в этом состоянии «понимает», что любая обнаруженная ошибка — не неисправность.
- Типы ошибок и точки, в которых они были обнаружены, имеют различный код, который добавляется к текущему общему количеству, в зависимости от того, является ли ошибка передаваемой или принимаемой. Значимые величины получения и передачи вызывают декремент этих счетчиков, при этом ноль (0) является минимальным значением.

Когда любой из данных счетчиков проходит соответствующий порог, определенный в CAN-протоколе, узел фиксирует пассивное состояние ошибки. В таком состоянии узел полагает, что это — причина ошибки.

- Когда переданное состояние счетчика ошибки в другом узле проходит определенный порог, узел вводит шину в отключенное состояние. Эта спецификация определяет механизмы перехода из состояния отключения шины к состоянию активной ошибки.
- Когда и передающий, и приемный счетчики пассивной ошибки узла декрементируются ниже определенного порога, узел еще раз подтверждает состояние активной ошибки.

Стандартный и расширенный фрейм

CAN-микросхемы поддерживают стандартный или расширенный фрейм.

Стандартный фрейм означает, что CAN-микросхема поддерживает 11-битовое поле идентификатора. Расширен-

ный фрейм означает, что микросхема поддерживает 29-битовое поле идентификатора. Новые CAN-микросхемы могут поддерживать форматы как стандартного фрейма, так и форматы расширенного фрейма.

Прерывания в CAN-протоколе

Проектировщики должны учитывать интервал возможных прерываний их CAN-контроллеров при проектировании своих изделий. Так как фрейм данных в CAN-протоколе короткий (от 0 до 8 байт), скорость поступления прерываний на процессор может быть высокой. В связи с этим следует рассматривать CAN как высокоскоростную сеть.

Рисунок 8 демонстрирует два передаваемых подряд CAN-фрейма данных с минимальным интервалом между фреймами, называемым интервалом межфрейма. Таблица 1 показывает жесткий режим прерывания для случая, если CAN-приемник получает все фреймы во время текущей связи (непрерывные фреймы в режиме back-to-back).

Строка «Число битов в CAN-протоколе» в таблице принимается с условием, что заполнение дополнительными битами отсутствует (естественно, что такое заполнение увеличило бы время между прерываниями).

Из таблицы видно, что трафик прерываний достаточно интенсивен. На скорости 500 кбит/с прерывания могут происходить каждые 94 мкс при отсутствии информации в фреймах данных. Большинство микроконтроллеров нижнего уровня не может поддерживать такую высокую скорость обработки прерываний. Следовательно, нужно найти компромисс между возможностями CAN-контроллера и его стоимостью. Следует выбирать CAN-контроллер, который обеспечивает соответствующий уровень предварительной фильтрации. Контроллер должен иметь достаточное время для обработки прикладной программы и успевать обслуживать запросы от CAN-сети, или необходимо выделять отдельный микроконтроллер для обслуживания CAN-приемника.

Также следует помнить, что некоторые CAN-микросхемы маскируют только восемь наиболее значащих битов по-



Рис. 8. Минимальный интервал межфрейма для режима back-to-back

Таблица 1. Трафик прерываний для случая приема всех фреймов в режиме back-to-back

Количество байтов данных	0	1	2	3	4	5	6	7	8
Количество битов в CAN-фрейме	47	55	63	71	79	87	95	103	111
Интервал между прерываниями на скорости передачи 125 кбит/с (период 1 бита = 8 мс)	376	440	504	568	632	696	760	824	888
Интервал между прерываниями на скорости передачи 250 кбит/с (период 1 бита = 4 с)	188	220	252	284	316	348	380	412	444
Интервал между прерываниями на скорости передачи 500 кбит/с (период 1 бита = 2 мс)	94	110	126	142	158	174	190	206	222

ля идентификатора (не все 11 битов) и имеют один фильтр MAC-КИ/СООТВЕТСТВИЯ.

Микросхемы, поддерживающие CAN-протокол

Микросхемы, которые поддерживают CAN-протокол, выпускаются различными поставщиками, такими как Philips, Motorola, Siemens, National Instruments и Intel.

Существуют следующие два типа микросхем.

Встроенные — микросхемы, которые включают в себя CAN-контроллер и один из видов интегрированного микроконтроллера. Это Intel 80196CA, содержащий в одном кристалле стандартный контроллер 80196 и CAN-контроллер 82527; Philips 82C592 и 82C598, имеющие контроллер 80C51 и CAN-контроллер 82C200; Motorola 68HC05X4, 68HC705X4, 68HC705X32 на основе M6805.

Периферийные — микросхемы, которые содержат только CAN-контроллер. Это Intel 82527 с 14 фиксированными входными фильтрами, одним типа Mask-and-Match и поддержкой стандартного и расширенного фреймов; Philips 82C200 с одним входным фильтром типа Mask-and-Match и поддержкой стандартного фрейма; Siemens SAB 81C90, 81C91 с 16 фиксированными входными фильтрами.

Кроме того, фирмами Philips и Texas Instruments выпускается ряд буферных микросхем, формирующих сигналы CAN-магистрали.

Применение в промышленных приложениях

В настоящее время CAN-протокол активно используется в промышленных сетях. Такие известные фирмы, как Honeywell и Allan-Bradley, разработали и поддерживают сетевые протоколы

верхнего уровня SDS и DeviceNet, причем последний является открытым и на данный момент более 200 фирм выпускают и разрабатывают свои изделия в этом стандарте. Кроме того, достаточно известными в Европе являются стандарты сети верхнего уровня CanOpen, CAL (Германия) и CanKingdom (Швеция). Все эти сети используют CAN-протокол на физическом и транспортном уровнях.

Фирма Advantech выпустила плату PCL-841, имеющую 2 гальванически развязанных CAN-порта на Philips 82C200 (рис. 9), и разрабатывает модули удаленного сбора информации с выходом на CAN; фирмы Grayhill и Opto22 выпустили недорогие периферийные контроллеры, поддерживающие сеть DeviceNet (рис. 10), в комплекте WAGO I/O System также имеется контроллер с выходом на CanOpen, DeviceNet и CAL (рис. 11).

Фирма Hilscher выпускает богатый набор плат с CAN-протоколами (CanOpen, Device Net, SDS) для распространенных системных шин типа ISA и PCI, для мезонинной шины PC/104, а также в виде OEM-модулей (рис. 12) для тех изготовителей контроллеров, которые хотят встроить в свои изделия

совместимость с CAN-протоколами, не затрачивая время и средства на собственные разработки. Ряд отечественных фирм также выпускает изделия с CAN-протоколом, в том числе в популярном формате MicroPC.



Рис. 9. Двухканальная CAN-плата PCL-841 фирмы Advantech



Рис. 10. Периферийный CAN-контроллер DacNet фирмы Grayhill

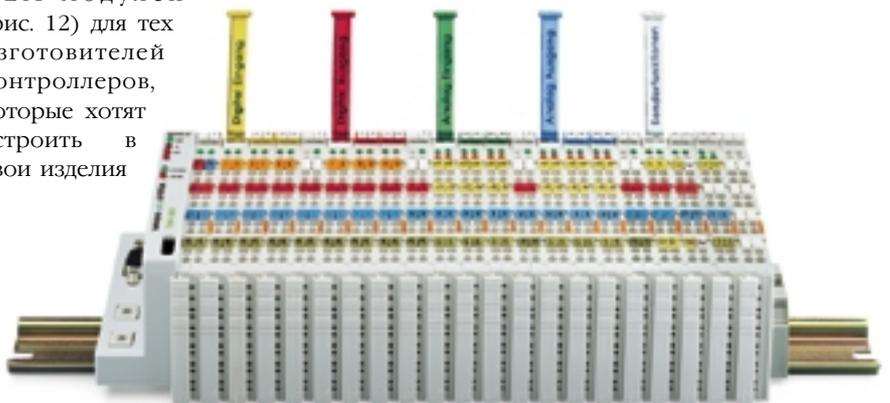


Рис. 11. Контроллер WAGO I/O-System

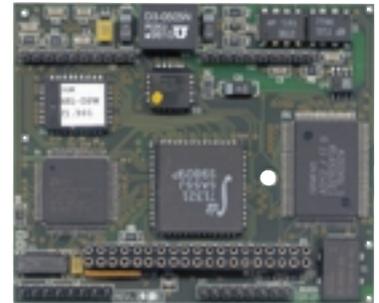


Рис. 12. Малогабаритный модуль фирмы Hilscher с интерфейсом CAN для производителей комплексного оборудования (OEM)

Заключение

Использование CAN-протокола и сетей верхнего уровня на его основе при модернизации отечественных промышленных предприятий позволит разработчикам средств АСУ ТП решить ряд остро стоящих проблем:

- выполнить требования помехоустойчивости;
- обеспечить совместимость с действующими в развитых странах стандартами;
- повысить надежность за счет того, что обеспечивается обязательное подтверждение приема сообщения приемником;
- обеспечить повышение живучести системы при применении режима «мультимастер»;
- снизить стоимость коммуникаций (требуется витая пара). ●