

Введение в CAN 2.0В интерфейс

Статья основывается на технической документации DS00713а
компании Microchip Technology Incorporated, USA.

**© ООО «Микро-Чип»
Москва - 2001**

Распространяется бесплатно.
Полное или частичное воспроизведение материала допускается только с письменного разрешения
ООО «Микро-Чип»
тел. (095) 737-7545
www.microchip.ru

Введение в CAN 2.0B интерфейс

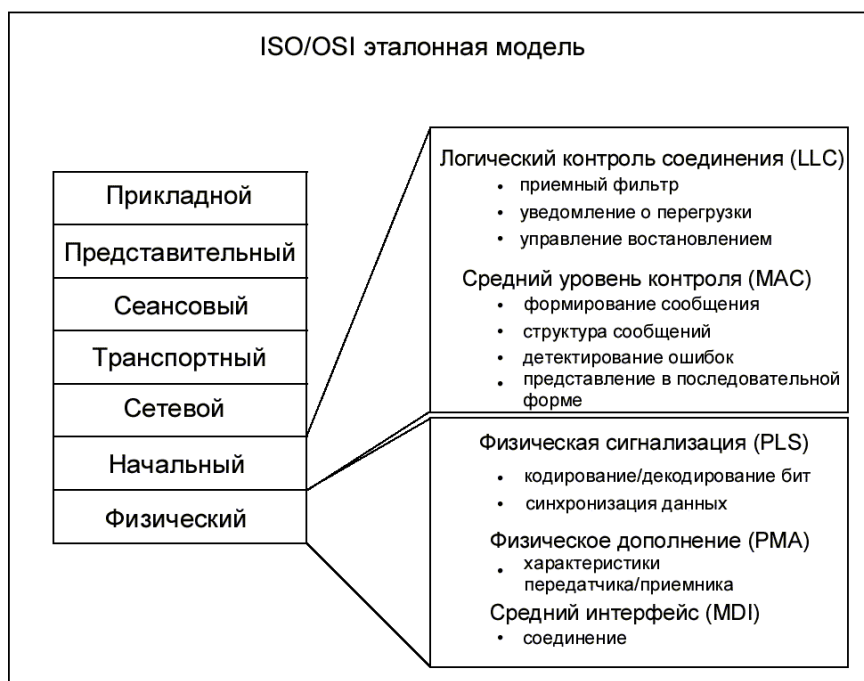
Статья основывается на технической документации DS00713a компании Microchip Technology Incorporated, USA.

Контроллер локальной сети (CAN) был разработан немецкой автомобильной фирмой Robert Bosch в 1980 году для автомобильных приложений. Целью разработки нового интерфейса было повышение надежности передачи информации по последовательному интерфейсу от различных узлов автомобиля, с сокращением общего числа проводников. CAN интерфейс нашел широкое применение в системах автоматизации, автомобильной технике, медицинском оборудовании и др.

Цель статьи состоит в том, чтобы объяснить основы CAN интерфейса для сетевых приложений.

Краткий обзор CAN

Большинство сетевых интерфейсов имеют многоуровневую структуру, позволяющую взаимодействовать изделиям (программам) различных производителей. Стандарт CAN интерфейса был разработан Международной Организацией по Стандартизации (ISO) как шаблон. CAN соответствует эталонной модели иерархического представления сети открытых систем (OSI).



CAN интерфейс поддерживает два нижних уровня эталонной модели. Физический уровень CAN дает возможность оптимизировать протокол связи для различных сред передачи данных (витая пара, однопроводная линия, оптический кабель, радиоканал, инфракрасное излучение).

Международная Организация Стандартов (ISO) и Общество Автомобильных Инженеров (SAE) определили некоторые протоколы связи CAN интерфейса для двух нижних уровней эталонной модели иерархического представления сети.

ISO11898 – стандарт для высокоскоростных приложений

ISO11519 – стандарт для низкоскоростных приложений

J1939 (SAE) – использование CAN в автомобильных приложениях

Все три протокола определяют 5В дифференциальную физическую линию связи.

Верхние уровни сетевой модели должны быть поддержаны программным обеспечением.

5 протоколов верхнего уровня выполняют следующие функции:

- Стандартизировать процедуры инициализации, включая различные скорости передачи информации.
- Распределить адреса среди узлов, участвующих в обработке данных или типов сообщений.
- Определить структуру сообщений.
- Обеспечить программную обработку ошибок на системном уровне.

Это далеко не полный список функций протоколов высокого уровня.

Основы CAN протокола

Множественный доступ с опросом несущей и разрешением конфликтов

Каждый узел сети должен контролировать бездействие шины в течение некоторого времени, прежде чем послать сообщение. Как только обнаружено бездействие шины, все узлы сети имеют равную возможность передать данные.

В CAN используется неразрушающий поразрядный арбитраж на шине, сохраняющий сообщение не поврежденным при потере инициативы. Арбитраж позволяет передавать сообщения с высоким приоритетом, без каких-либо задержек и искажений.

Логические уровни на шине определяются как 'dominant' и 'recessive'. Узел передачи должен контролировать логическое состояние шины для начала передачи данных. На шине CAN логический бит 0 определяется как 'dominant', а бит – 1 как 'recessive'.

Бит 'dominant' всегда будет выигрывать арбитраж у 'recessive' бита, поскольку имеет низкий уровень сигнала и более высокий приоритет. Каждый узел, передающий данные на шины, должен контролировать наличие передаваемого бита на шине. Сообщение с более низким приоритетом будет формировать на шине бит 'recessive', а при проверке обнаружит бит 'dominant', в этом случае узел, формирующий сообщение более низкого приоритета, теряет арбитраж. Узел, потерявший арбитраж, должен дождаться отсутствия активности на шине и повторить попытку передать данные.

Сообщения

В CAN протоколе сообщения не являются адресными, т.е. сообщения не адресуются от одного узла к другому. Сообщение содержит идентификатор источника и собственно данные. Все узлы CAN сети могут принять каждое сообщение на шине и самостоятельно определить: данное сообщение должно быть отвергнуто или обработано.

Например, в автомобиле датчик напряжения бортсети подключен только к центральному маршрутизатору. Маршрутизатор передает данные, полученные с датчика напряжения, другим узлам автомобиля. При использовании CAN интерфейса все «заинтересованные» узлы получили бы самую «свежую» информацию с датчика напряжения минуя маршрутизатор.

Другой полезной особенностью CAN протокола является возможность удаленного запроса данных (RTR). В отличие от предыдущего случая, требуемые данные не ожидаются, когда появятся на шине, а запрашиваются у конкретного узла. Проектировщик может использовать эту особенность для снижения трафика шины при сохранении целостности сети.

Добавление нового узла в систему не требует перенастройки остальных устройств сети. Новый узел начинает принимать сообщения из сети на основе их идентификаторов.

Описание сообщений CAN интерфейса

В CAN протоколе используется 4 типа сообщений.

Первый тип (наиболее распространенный) – стандартное сообщение с 11-разрядным идентификатором, используется для передачи информации к любому или всем узлам сети.

Второй тип сообщений имеет 29-разрядный идентификатор, также используется для удаленного запроса данных (RTR).

Других два типа сообщений используются для обработки ошибок.

Сообщение об ошибке формируются узлами, которые обнаруживают одну из возможных ошибок протокола CAN. Сообщение перезагрузки формируется узлами, требующие большого количества времени для обработки уже полученных данных.

Стандартное сообщение состоит из полей (см. рисунок 2):

- Арбитраж (Идентификатор)
- Управляющее поле
- Данные
- CRC
- 2-х разрядное подтверждение
- Конец сообщения

Поле арбитража используется для указания приоритета и типа сообщения. Сообщение с меньшим идентификационным номером имеет больший приоритет на шине CAN. В стандартном сообщении поле арбитража - 12 бит (11 бит – идентификационный номер, 1 бит – RTR).

В сообщении удаленного запроса поле арбитража состоит из 32 бит (см. рисунок 3):

29 бит – идентификатор

1 бит – для определения, что сообщение удаленного запроса

1 бит – SRR, не используется

1 бит – RTR

Сообщение удаленного запроса является расширенным сообщением (см. рисунок 3).

Бит RTR указывает на назначение сообщения: передача данных или удаленный запрос. Для передачи данных этот бит должен быть 'dominant'.

Поле управления состоит из 6 бит. Первый бит IDE, указывает на тип сообщения: стандартный или расширенный. Для стандартных сообщений этот бит должен быть 'dominant'. Биты RB0, RB1 зарезервированы для дальнейших разработок.

Длина передаваемых данных в байтах определяется четырьмя битами DLC3:DLC0 управляющего поля. Сообщение удаленного запроса не имеет поля данных в не зависимости от значения битов DLC.

Поле данных состоит из числа байтов, описанных в управляющем поле.

Поле CRC состоит из 15 бит, используется для контроля достоверности принятых данных.

Поле подтверждения используется для указания, что сообщение было получено правильно. Любой узел, который правильно принял сообщение, не зависимо от того обрабатывает он его или отказывается, формирует 'dominant' бит на шине.

Последние два типа сообщений об ошибках, будут рассмотрены ниже.

Быстрая и надежная передача данных

С появлением новой версии CAN 2.0B, скорость связи увеличилась в 8 раз, по сравнению с первым вариантом, и составила 1Мбит/с. Поскольку CAN интерфейс первоначально разрабатывался для использования в автомобилях, в его протокол был заложен эффективный механизм обработки ошибок. Протокол позволяет обнаружить большинство ошибок, возникающих во время передачи данных, что сохраняет целостность сообщения с классификацией ошибок как кратковременные и постоянные.

CAN узлы могут находиться в нескольких состояниях: активном – при нормальной работе узла, и пассивном – при обнаружении большого количества ошибок. Такой подход гарантирует некоторую пропускную способность шины для критической информации. Неисправный узел не сможет монополизировать всю шину.

Обнаружение ошибок

В CAN протоколе существует 5 различных типов ошибок.

Ошибка CRC

15-разрядное значение избыточного кода CRC рассчитывается узлом передачи во время формирования сообщения. Все узлы сети принимают это сообщение, вычисляют CRC и сравнивают его с принятым значением. Если значения CRC не совпадают, возникает ошибка, и формируется сообщение об ошибке. Если хотя бы один узел не примет данные должным образом, будет сформировано сообщение об ошибке.

Ошибка подтверждения

В поле подтверждения, передающий узел, проверяет наличие 'dominant' бита. Если хотя бы один узел принял сообщение правильно, то 'dominant' бит будет сформирован в подтверждающем поле. Если был получен бит 'recessive', значит ни одно устройство не приняло сообщение, требуется повторить его через некоторое время.

Ошибка формы

Ошибка формы обнаруживается, если битовое поле, фиксированного формата, содержит один или более запрещенных битов, кроме обнаружения бита 'dominant' в конце сообщения. Первоначальное сообщение должно быть повторно передано.

Ошибка бита

Ошибка бита возникает при передаче 'recessive' на шину, а во время проверки обнаруживается бит 'dominant'. Исключением является арбитраж и подтверждение приема, в этом случае ошибка не возникает. Первоначальное сообщение должно быть повторно передано.

Ошибка заполнения

CAN интерфейс может использовать NZR метод передачи данных. Прием данных узлами синхронизируется от потока данных при переходе от 'recessive' к 'dominant' биту. Если обнаружено подряд 6 битой одной полярности, возникает ошибка заполнения. После возникновения ошибки, узлы приема используют переходы для синхронизации, а не для приема данных. Формируется сообщение об ошибке.

Статус узла

Узел может находиться в одном из трех состояний:

- Активной ошибки
- Пассивной ошибки
- Отключен от шины

Состояние активной ошибки

Узел может принимать активное участие по передаче данных по шине CAN, включая формирование активного сообщения об ошибке, состоящего из 6 последовательных 'dominant' битов. Сформированное сообщение об ошибке, заставляет и другие активные узлы выдавать это же сообщение. Поэтому на шине может возникнуть до 12 'dominant' битов, в зависимости от того, в какой момент другие узлы обнаружат ошибку. Узел находится в состоянии активной ошибки тогда, когда его счетчик ошибок передатчика (TEC) и счетчик ошибок приемника имеют значение меньше 128. Состояние активной ошибки – нормальный режим работы узла, с возможностью принимать и передавать данные без ограничений.

Состояние пассивной ошибки

Узел переходит в состояние пассивной ошибки, когда его счетчик TEC и/или REC превышает значение 127. Узлу, в этом состоянии, запрещается передавать активное сообщение об ошибке, но вместо этого, он может передать пассивное сообщение об ошибке, состоящее из 6-ти 'recessive' битов. Если пассивный узел единственный передатчик на шине, то пассивное сообщение об ошибке создаст условие ошибки заполнения. Если пассивный узел не единственный передатчик на шине, то он не будет иметь никакого влияния на передачу данных, т.к. сообщение об ошибке формирует 'recessive' битами.

При обнаружении 'dominant' битов во время формирования пассивного сообщения, узел должен дождаться освобождения шины и повторить передачу.

Состояние - отключен от шины

Узел переходит в состояние - отключен от шины, когда значение счетчика TEC больше чем 255. В этом состоянии узел не может передавать, принимать сообщения, выдавать подтверждение и формировать сообщения об ошибке. Имеется порядок действий в протоколе CAN, с помощью которого возможно восстановить статус устройства активной ошибки, после устранения условий неисправности.

Заключение

Протокол CAN оптимизирован для систем, в которых должны передаваться относительно небольшое количество информации (по сравнению с Ethernet или USB разработанные специально для больших объемов данных) к любому или всем узлам сети. Множественный доступ с опросом состояния шины позволяет каждому узлу получить доступ к шине с учетом приоритетов.

Не адресатная структура сообщений позволяет организовать многоабонентскую доставку данных с сокращением трафика шины.

Быстрая устойчивая передача информации с системой контроля ошибок позволяет отключать неисправные узлы от шины, что гарантирует доставку критических по времени сообщений.

Преимущества CAN протокола позволили использовать его в автомобилестроении, автоматизации технологических процессов, медицине и других приложениях.

Статья основывается на технической документации DS00713a
компании Microchip Technology Incorporated, USA.

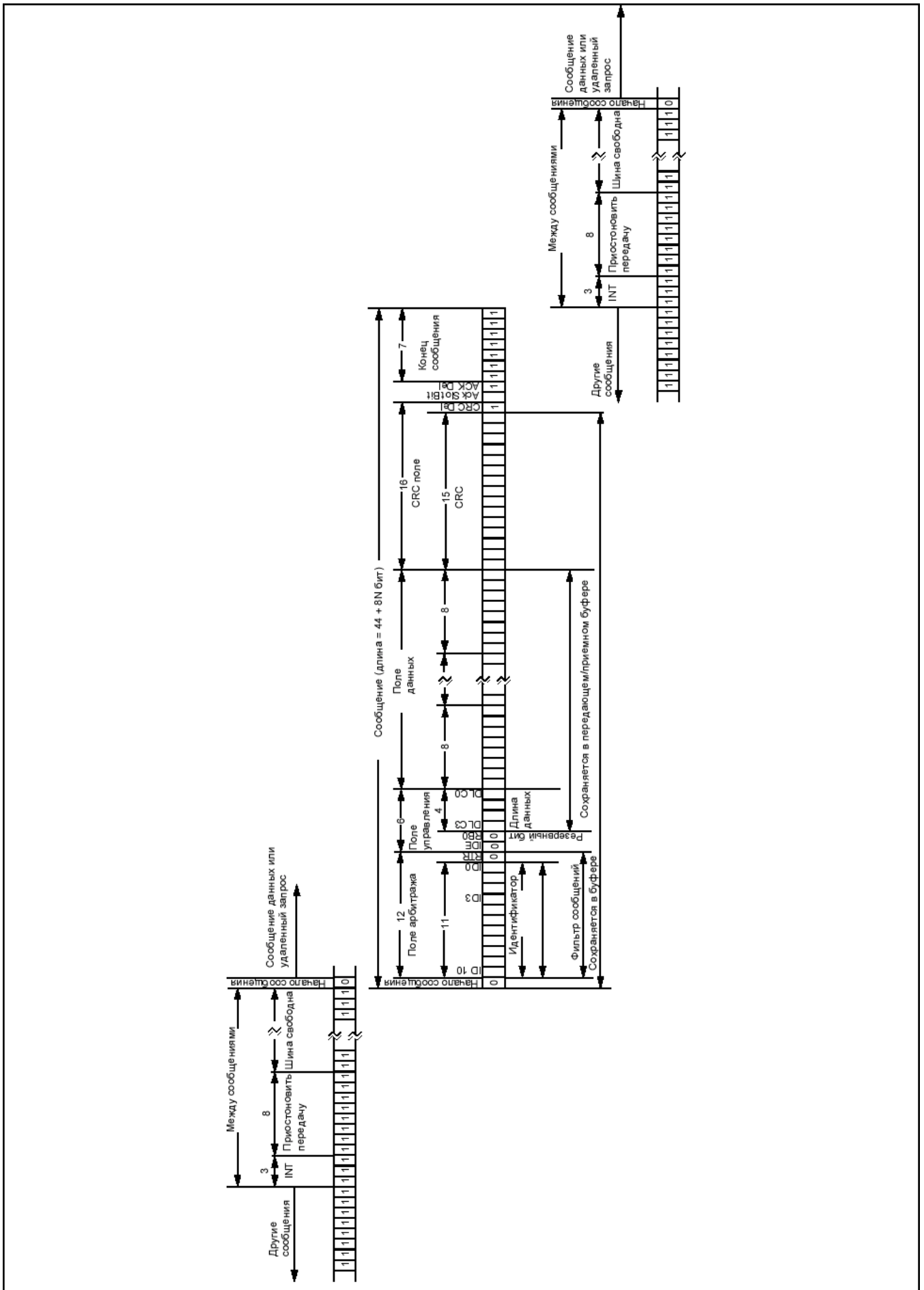


Рис. 2 Стандартное сообщение

Уважаемые господа!

ООО «Микро-Чип» поставляет полную номенклатуру комплектующих фирмы **Microchip Technology Inc** и осуществляет качественную техническую поддержку на русском языке.

С техническими вопросами Вы можете обращаться по адресу support@microchip.ru

По вопросам поставок комплектующих Вы можете обращаться к нам по телефонам:
(095) 963-9601
(095) 737-7545
и адресу sales@microchip.ru

На сайте
www.microchip.ru

Вы можете узнать последние новости нашей фирмы, найти техническую документацию и информацию по наличию комплектующих на складе.