УДК 681.518

ИССЛЕДОВАНИЕ И ИСПОЛЬЗОВАНИЕ ИНТЕРФЕЙСА ПЕРЕДАЧИ ДАННЫХ CAN 2.0 В МИКРОКОНТРОЛЛЕРАХ НА БАЗЕ ЯДРА CORTEX M4 КОМПАНИИ STMICROELECTRONICS

Е.Г. Вовк аспирант, ХНАДУ

Аннотация. *Рассмотрена методика настройки и использования интерфейса передачи данныхCAN2.0 в микроконтроллерах на базе ядра CortexM4. На основании рассмотренных теоритических вопросов запрограммирован и исследован интерфейс с применением трансиверов SN65HVD230.*

Ключевые слова: *CAN, трансивер, интерфейс, микроконтроллер.*

ДОСЛІДЖЕННЯ І ВИКОРИСТАННЯ ІНТЕРФЕЙСУ ПЕРЕДАЧІ ДАНИХ CAN 2.0 В МІКРОКОНТРОЛЕРАХ НА БАЗІ ЯДРА CORTEX M4 КОМПАНІЇ STMICROELECTRONICS

Є.Г. Вовк аспірант, ХНАДУ

Анотація. *Розглянута методика настройки та використання інтерфейсу передачі даних CAN2.0 в мікроконтролерах на базі ядра CortexM4. На підставі розглянутих теоретичних питань запрограмований і досліджений інтерфейс із застосуванням трансиверів SN65HVD230.*

Ключові слова: *CAN, трансивер, інтерфейс, мікроконтролер.*

THE STUDY AND USE OF DATA INTERFACE CAN 2.0 IN MICROCONTROLLERS BASED ON THE CORE CORTEX M4 COMPANY STMICROELECTRONICS

Y. Vovk graduate student, KhNAHU

Abstract. *The method of setting up and using communication interface CAN2.0 in microcontrollers based on the core CortexM4. Based on the issues discussed theoretically investigated and programmed interface using transceivers SN65HVD230.*

Keywords: *CAN, transceiver interface microcontroller.*

Введение

Разработка современных систем управления в автомобильной отрасли предполагает наличие и использование интерфейсов передачи данных для межблочной коммуникации. Разработанный компанией Robert Bosch в середине 1980-х CAN интерфейс является одним из самых надежных и широко распространённых [1]. При передаче данных CAN протокол аппаратно обеспечивает формирование сообщения, выполняет передачу данных, осуществляет побитную синхронизацию, выполняет идентификацию сообщения, использует прием вставки нулевого бита (битстаффинг), подтверждает правильность приема данных всеми принимающими и передающими устройствами, а также обнаруживает и исправляет ошибки в процессе передачи и приема сообщений.

Анализ публикаций

Применение CAN интерфейса при разработке электронных систем управления в автомобильной промышленности на сегодняшний день получило наибольшее распространение. Большинство 16-ти и 32-х битных микроконтроллеров поддерживают работу с данным интерфейсом. Однако такие возможности существенно отражаются в стоимости самого микроконтроллера, следовательно, и конечной стоимости блока управления.

Цель и постановка задачи

Целью исследования является рассмотрение аппаратных возможностей CAN интерфейса в микроконтроллерах на базе ядра Cortex M4, реализация и исследование его функционирования при различных неисправностях.

Исследование интерфейса на базе микроконтроллера STM32F4

Микроконтроллеры на базе ядра Cortex M4 (STM32F4) аппаратно поддерживают CAN 2.0 интерфейс. Максимальная скорость передачи данных 1 Мбит/сек (при максимальной длине в 40 метров). При подключении микроконтроллера к шине, необходимо установить дополнительный трансивер (например, SN65HVD230), который обеспечит функциональные возможности физического канала и канала данных [2].

В микроконтроллерах STM32F4 для настройки работы и функционирования интерфейса используются следующие режимы:
1) режим инициализации;
2) режим пониженного энергопотребления;
3) режим Normal;
4) тестовые режимы(Silent mode, Loop back mode, Loop back combined with silent mode).

При разработке программного аппаратного обеспечения с использованием CAN интерфейса для проверки работоспособности устройства необходимо наличие шины с подключенными к ней другими блоками управления согласно установленных условий инициализации. Если отсутствует такая возможность при отладке устройства, то разработчик может воспользоваться 3 тестовыми режимами и проверить работоспособность настраиваемого CAN интерфейса. Передача данных возможна по базовому и расширенному формату кадру данных. При разработке аппаратного программного обеспечения указываются запрос на передачу данных (кадры данных или кадры удаленного запроса), длина данных (от 0 до 8 байт), расширенный или стандартный идентификатор. После установки необходимых данных происходит автоматический подсчет контрольной суммы и осуществляется передача данных. Прием данных осуществляется в несколько буферов, в зависимости от нагрузки и прохождения данных через настроенные фильтры приема сообщений.

В случае ошибки, CAN сигнализирует о возникновении ошибок передачи (реализовано 5 аппаратных механизмов проверки на наличие ошибки):
1) Bit error. Возникает в случае, когда бит передается доминантным уровнем, а принимается рецессивным или наоборот. Рассматривается, как ошибка бита и вызывает передачу кадра ошибки в очередном битовом интервале.
2) Stuff error. При передаче сообщения работает алгоритм битстаффинга (вставка дополнительного бита после пяти подряд идущих бит с одинаковым значением).
3) CRC error. Последовательность контрольной суммы содержит результат контрольной суммы всего сообщения, передаваемого передатчиком. Все приемники (которые присутствуют на шине), принимая сообщение, вычисляют контрольную сумму и сравнивают ее с полученной. При несовпадении контрольных сумм сообщение считается ошибочным.
4) Form error. Все устройства, находящиеся в шине проверяют соответствие структуры принимаемого сообщения его фиксированному формату и размеру. В случае возникновения несоответствия формату – выдаётся ошибка формата.
5) Acknowledgement error. Ошибка подтверждения обнаруживается передатчиком, если хотя бы один приемник не подтвердил получение правильного сообщения [3].

При приеме кадра все узлы, находящиеся на шине, выполняют механизмы обнаружения ошибки. В случае возникновения ошибки, узел генерирует кадр “Error field”, который прерывает передаваемый кадр данных или удаленного запроса. При этом передатчик автоматически аппаратно повторит передачу сообщения.

Каждый CAN узел поддерживает два внутренних счетчика ошибок: счетчик ошибок при приеме и счетчик ошибок при передаче. Организация передачи данных по CAN шине гарантирует 2 события – либо данные правильно приняты всеми, либо данные не принял никто. Использование сложного механизма оценки сбойных ситуаций на узле позволяет распознать неисправность самого узла, и в случае необходимости автоматически отключить его от сети (при переполнении счетчика ошибок при передачи). Находясь в том состоянии, CAN контроллер передает только рецессивные биты.

Для исследования работы функционирования CAN интерфейса на микроконтроллере использовалась схема соединений, представленная на рис.1. Внешний вид разработанного комплекса для проведения исследования представлен на рис.2.

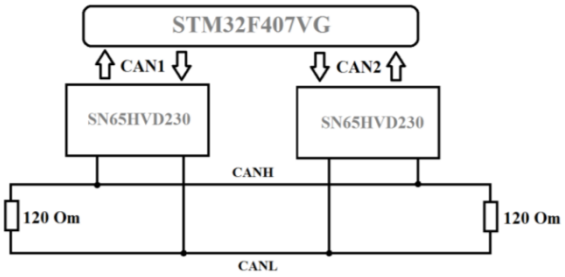

Рис.1. Схема соединения 2-х интерфейсов CAN с использованием одного микроконтроллера

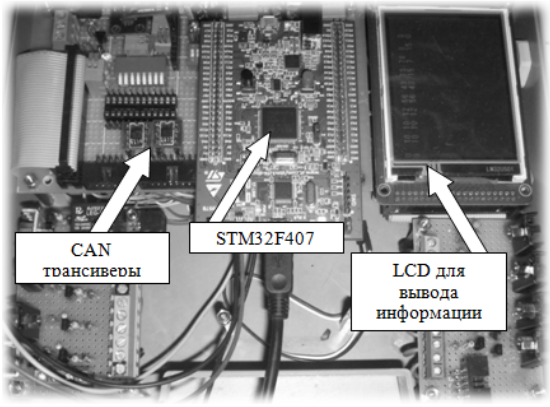

Рис.2. Разработанный программно-аппаратный комплекс для исследования функционирования CAN интерфейса

При разработке аппаратного программного обеспечения, с целью реализации передачи данных по интерфейсу, для отслеживания алгоритмов работы интерфейса и реализации встроенной функции определения ошибок дополнительно создали отдельные массивы для счетчика срабатывания установки флагов событий, прерываний и количества ошибок по каждому событию. В случае возникновления ошибки или события во время передачи или приема данных, микроконтроллер мгновенно выводит все необходимые данные на дисплей в виде описания события и количества его повторений. Дополнительно реализована функция счетчика передачи и приема сообщений с начала инициализации CAN интерфейса.

Процесс инициализации работы CAN интерфейса состоит из следующих пунктов:
1) Инициализация работы шины тактирования выходных портов и CAN интерфейса;
2) Настройка работы выходных и входных портов в режиме альтернативной функции для CAN интерфейса;
3) Настройка работы интерфейса(выбор режима работы, расчет и настройки скорости передачи (Bittiming), настройка дополнительных режимов и функций. “Bittiming” определяется квантом времени, который зависит от частоты опорного генератора и параметров делителя битовой скорости узла.
4) Настройка работы режима функционирования (Mask mode, Identifierlist mode) и размера фильтра(32, 16 бит);
5) Настройка работы прерываний(transmit interrupt, FIFO interrupt, status change error interrupt).

В зависимости от требований на разработку устройства управления с использованием CAN интерфейса разработчик может выбрать и настроить прерывания на все реализованные аппаратные ошибки и работу с FIFO.Данная информация представлена в техническом описании на микроконтроллер.

Во время тестирования программы на разных скоростях передачи данных (от 250 кбит/сек до 1Мбита/сек) интерфейс в течении всего времени работы не обнаружил ни одной ошибки, в том числе при воздействии на CAN шину механических колебаний. При установке разных скоростей передачи данных на CAN2 и CAN1 в результате несовпадения временных интервалов интерфейс, который выполнял первым передачу данных, в результате срабатывания прерываний выдал ошибку “Warning”,

“Error” и “Bit recessive”. В результате попыток передать аппаратно сообщение заново, счетчик ошибок постепенно увеличивался. Прерывания ошибок интерфейса, принимающего данные, сработали 1 раз и выдали кроме ошибок, описанных в передающем устройстве, еще и ошибку “Form”. На основании полученной информации с дисплея, можно сформировать вывод о том, что передатчик пробует передать данные к приемнику, однако в результате ошибок – постепенно передает данные с «нуля» по достижению условия подтверждения приема (которое в результате временной разницы не может произойти). Приемник сформировал разовое прерывание и сообщил о попытке передачи данных и неправильной формы сообщений. В случае выхода из строя трансивера при передаче данных, кроме основных предупреждающих ошибок срабатывает прерывание ошибки “Bus off ”. Однако в процессе возобновления работоспособности приемник сформирует разовую ошибку приема данных и шина полностью возобновит свою работоспособность. Аналогичные действия будут при замыкании шины передачи данных как на источник питания, так и между собой (CANH и CANL).

При разработке электронных систем управления, имеющих одну общую шину передачи данных, но различные логические уровни микроконтроллеров (3В и 5В), одним из оптимальных вариантов передачи информации является использование CAN трансиверов с гальванической развязкой. Компания Texas Instruments разработала трансивер, соответствующий данной модификации и поддерживающий стандарт ISO11898-2[4]. Согласно указанной информации в техническом описании он поддерживает также протоколы CANopen, DeviceNet, NMEA2000, ARINC825, ISO11783, CAN Kingdom, CANaerospace. Функциональная диаграмма такого трансивера представлена на рис.3.

Использование трансивера компании Texas Instruments с микроконтроллером STM32F407 позволяет осуществлять подключение разрабатываемых электронных блоков к существующим и ранее разработанным системам управления с 5В логикой. Данное решение обеспечивает качественную передачу данных и уменьшает размер печатной платы.

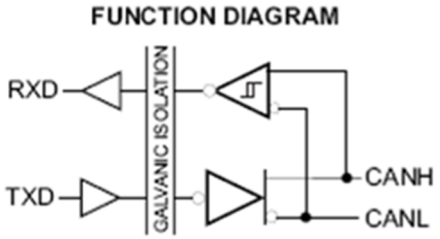

Рис.3. Функциональная диаграмма трансивера ISO1050

Экспериментальным путем было установлено максимальную скорость передачи данных в 2,33 Мбит/сек при длине шины в 0,3 м и оптимальных лабораторных условиях тестирования. При дальнейшем увеличении скорости возникает ошибка формы кадра данных и шина прекращает функционирование.

Выводы

В работе был исследован CAN интерфейс на микроконтроллерах STM32F407 и разработано аппаратное программное обеспечение для микроконтроллера. Рассмотрен алгоритм инициализации, схемы подключения, режимы функционирования, аппаратные средства обнаружения ошибок и функционирование интерфейса при использовании высокоскоростного трансивера.

Литература

1. http://ru.wikipedia.org/wiki/Controller_Area_Network.
2. Texas Instruments. 3.3-V can transceivers SN65HVD230.
3. RM0090 Reference manual for STM32F4.
4. Texas Instruments. Isolated can transceiver ISO1050.