

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

ИНСТИТУТ МАШИНОСТРОЕНИЯ

(наименование института полностью)

Кафедра «Промышленная электроника»

(наименование)

11.04.04 «Электроника и нанoeлектроника»

(код и наименование направления подготовки)

Электронные приборы и устройства

(направленность (профиль))

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)**

на тему «Разработка CAN-шины для автономного транспортного средства»

Обучающийся

С.А. Кривов

(И.О. Фамилия)

(личная подпись)

Научный

к.т.н., доцент, А.А. Шевцов

руководитель

(ученая степень, звание, И.О. Фамилия)

Тольятти 2022

Содержание

Введение	3
1.Описание протокола CAN и CANopen.....	5
1.1. Физический уровень CAN	5
1.2. Организация логического уровня CAN.....	17
1.3. Адресация CAN	24
1.4. Организация синхронизации CAN-шины.	26
1.5. Протокол CANopen	29
2.Приемопередатчики для сетей CAN	39
3.Контроллеры CAN-bus.....	49
3.1 Контроллеры CAN-bus в структуре микроконтроллеров	49
3.2 ПЛК с CAN-интерфейсами	60
4.Структура оборудования автономного транспортного средства	61
5.Разработка системы оборудования на основе шины CAN.....	63
Заключение	82
Список используемой литературы и используемых источников	83

Введение

Актуальность и научная значимость настоящего исследования. В современной промышленности активно внедряются и активно используются автономные транспортные средства. Прогресс в области электроники и алгоритмических наработок приводит к постоянному усложнению конструкции транспортных средств, а следовательно и к усложнению архитектуры системы управления. Особое значение приобретает организация связи между различными узлами транспортного средства. Соответственно разработка различных технических решений повышающих эффективность, надежность и универсальность, сокращающих время, затрачиваемое на разработку новых продуктов имеет актуальность.

Объект исследования.

Объектом исследования является организация внутренней шины данных для передачи информации от датчиков в систему управления и исполнительных команд от системы управления в исполняющие устройства, включая устройства индикации, двигатели и т.п.

Цель исследования: Целью исследования является оптимизация процесса передачи данных между различными узлами автономного транспортного средства.

Для достижения поставленной цели необходимо решить следующие **задачи:**

1. Собрать данные по используемой в управляемом человеком автомобильном транспорте и промышленных сетях CAN-шине.
2. Исследовать имеющиеся протоколы верхнего уровня на примере протокола CANopen.
3. Выделить типовую структуру автономного транспортного средства.
4. Спроектировать схему электрическую принципиальную управляющего узла сети CANbus.

5. Спроектировать схему электрическую принципиальную моста между CANшиной и исполнительными устройствами.

В области автономных транспортных средств на настоящее время можно выделить две больших сферы применения – это транспортные средства для перевозки людей (пассажирские транспортные средства) и технологические или промышленные транспортные средства. Как было отмечено выше, в данной работе будут рассматриваться аспекты развития технологических транспортных средств в контексте разработки внутренней сети передачи информационных потоков в соответствии с протоколом CANbus.

Стандарты, описывающие эту шину декларируют ее состав на физическом и алгоритмическом (программном или верхнем) уровнях.

Соответственно при решении поставленных на работу задач нам необходимо будет собрать основную информацию об уровнях реализации сетей CANbus, включая верхний уровень, вплоть до протокола CANopen.

Итогом данной работы должна стать схема электрическая принципиальная ведущего устройства этой сети, являющегося системой управления для технологических транспортных средств, которые используются для решения логистических задач на современном производстве.

Замена набора жгутов проводов на промышленно стандартизованную CANbus должно упростить монтаж, эксплуатацию и последующую модернизацию таких транспортных средств, за счет значительного увеличения степени модульности конструкции электронной части устройств.

В целях использования в подобных конструкциях узлов, изначально не приспособленных для эксплуатации в сети CANbus в работе также будет разработана схема электрическая принципиальная моста на основе микроконтроллера и трансивера между шиной CANbus и устройствами с наиболее распространенными интерфейсами обмена информацией.

1. Описание протокола CAN и CANopen

1.1. Физический уровень CAN

Современный технологический уклад привел к внедрению электронных блоков и модулей практически во все части сложных технических механизмов и систем. Внутри сложных изделий циркулируют разнообразные потоки данных: информация о параметрах окружающей среды, состоянии частей и узлов, параметрах производственных и технологических процессов собирается системами управления с датчиков информации, исполнительные устройства получают команды на выполнение определенных действий от систем управления, обработанные данные и информация, адаптированная для восприятия человеком, подается на устройства человеко-машинного интерфейса (HMI – human-machine interface) в виде сигнальных индикаторов, специализированных панелей и экранов, управляющие команды от человека поступают через клавиатуры или различного типа манипуляторы.

Всё это множество устройств обычно имеет свои собственные схемы и протоколы соединений проводного или беспроводного типов. В итоге, проводные, кабельные соединения разрастаются до значительных размеров, при этом информационные кабельные линии и жгуты связи различных объектов могут достигать в длину километры и даже десятки километров. Подобные размеры приводят к чрезмерному усложнению кабельного хозяйства объектов, что повышает трудозатраты при монтаже, сборке, эксплуатации и ремонте изделий. Значительная масса металлов с высокой стоимостью (медные и алюминиевые проводные системы) приводит к заметному удорожанию продукции. Большое количество токопроводящих проводов приводит к усложнению картины электромагнитного поля в замкнутых пространствах, возрастанию его интенсивности и, как следствие, к росту интенсивности электромагнитных помех различного типа. В таких

условиях возрастают как индуктивные, так и кондуктивные помехи, что приводит к необходимости принятия дополнительных мер по их нивелированию за счет экранирования проводов, принятия дополнительных технических мер. Это, в свою очередь, дополнительно удорожает всю конструкцию.

Таким образом, насыщение устройств датчиками, проводами, системами сбора данных, исполнительными устройствами, устройствами ввода и вывода информации для человека приводит к заметному удорожанию механизмов и их систем.

Надо сказать, что с подобными проблемами сталкиваются при проектировании и изготовлении промышленных систем: автоматических, робототехнических комплексов, станочного парка, термических систем и т.п.

Однако, наиболее остро проблема насыщенного информационно-управляющего кабельного хозяйства стоит в транспорте.

Технологический прогресс, прогресс в электронике, информационных технологиях, разработка и использование разнообразных алгоритмов управления привели к тому, что транспортные средства: легковые и грузовые автомобили, локомотивы и поезда на их основе, суда и корабли, авиационная и космическая техника, насыщаются информационными системами, системами сбора и обработки информации, исполнительными устройствами с гибкими и разнообразными алгоритмами управления.

И именно в мобильных системах важное значение приобретают сложность, длина и масса кабельного хозяйства. С его усложнением растет стоимость проектирования, изготовления и эксплуатации транспортных средств. Растет вероятность возникновения отказов, которые могут привести, в том числе, и к возникновению аварийных ситуаций, в том числе и с неприемлемым исходом.

Отдельным аспектом является топливная или энергетическая эффективность транспортных средств: увеличение массы тех или иных узлов

или подсистем, приводит к повышению расхода топлива или электрической энергии.

Преодолеть такое количество негативных факторов позволяет использование шинного подхода при конструировании информационного и управляющего кабельного хозяйства.

Шинный подход характеризуется тем, что вместо прокладывания отдельного жгута проводов или кабеля от системы управления к каждому датчику или каждому исполнительному устройству, что приводит к катастрофическому увеличению отдельных проводов, используют ограниченное количество проводов, организованных в физический канал связи, непосредственно к которым подключаются источники или потребители данных.

Как правило, количество проводов в таких шинах уже сильно ограничено (на практике – от двух до пяти-девяти) и приводит к заметно ограниченному использованию кабелей и соединительных жгутов проводов.

Изначально на транспорте, а теперь и в промышленность нашел применение протокол, включающий в себя физический и программный уровень и называемый «Controller Area Network» или CAN-шиной.

Этот протокол разработан компанией Robert Bosch GmbH в середине 80-х годов 20 века.

Протокол CAN описывается стандартом ISO 11898, включающим в себя группу документов, регламентирующих различные уровни построения сети CAN, актуальные в настоящее время:

- ISO 11898-1:2015 «Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical signaling»
- ISO 11898-2:2003 Road vehicles — Controller area network (CAN) — Part 2: High-speed medium access unit
- ISO 11898-3:2006 Road vehicles — Controller area network (CAN) — Part 3: Low-speed, fault-tolerant, medium-dependent interface

- ISO 11898-4:2004 Road vehicles — Controller area network (CAN) — Part 4: Time-triggered communication

ISO 11898-5:2007 Road vehicles — Controller area network (CAN) — Part 5: High-speed medium access unit with low-power mode

- ISO 11898-6:2013 Road vehicles — Controller area network (CAN) — Part 6: High-speed medium access unit with selective wake-up functionality

На территории Российской Федерации действуют следующие стандарты, описывающие протокол CAN-шины:

- ГОСТ Р ИСО 11898-1-2015 Транспорт дорожный МЕСТНАЯ КОНТРОЛЛЕРНАЯ СЕТЬ (CAN) Часть 1 Канальный уровень и передача сигналов

- ГОСТ Р ИСО 11898-2-2015 Транспорт дорожный МЕСТНАЯ КОНТРОЛЛЕРНАЯ СЕТЬ (CAN) Часть 2 Устройство доступа к высокоскоростной среде

Рассмотрим основные принципы функционирования CAN-шины [23,29].

CAN-шина является сетью, в которой среда передачи данных является общей. Это значит, что в единую сеть объединены контроллеры, исполнительные устройства и датчики.

Режим передачи в сети CAN- последовательный, широковещательный с пакетной передачей информации.

Как правило, практической реализацией CAN – шины является проводная организация на основе дифференциальной «витой пары».

Подключение устройств производится так, как показано на рисунке 1.

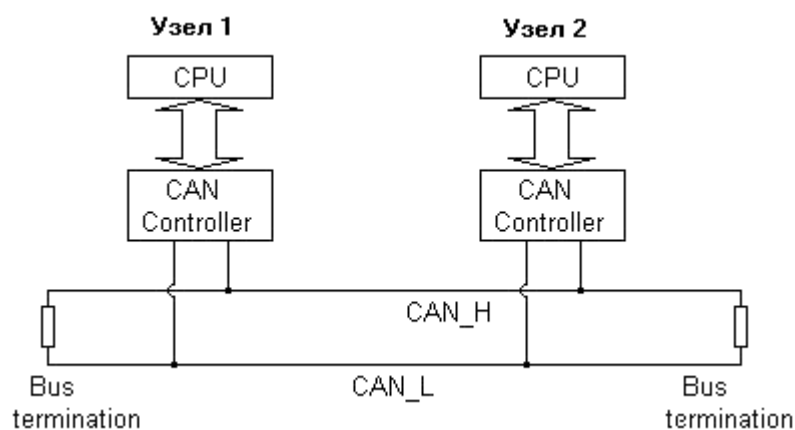


Рисунок 1 – Подключение устройств в сети CAN

Для понимания принципа генерации электрических сигналов в сет CAN на физическом уровне, надо представлять , что подключение устройств реализуется по принципу «проводное И» и его можно изобразить, как показано на рисунке 2.

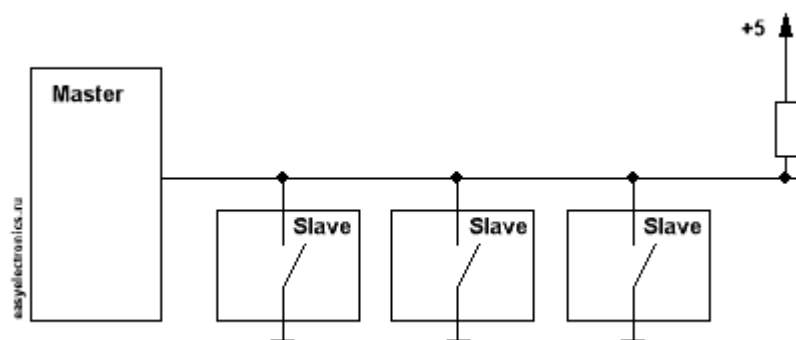


Рисунок 2 – Подключение устройств по принципу «Проводное И»

Для описания физической структуры сети используют понятие – узел. Каждое устройство – контроллер или датчик, или исполнительный модуль, считаются узлами сети. В сети CAN невозможно послать сообщение какому-то конкретному узлу – сообщения принимают все узлы. В соответствии со

стандартом, устройства, которые являются контроллерами сети, могут на аппаратном уровне проводить фильтрацию CAN-сообщений.

В общем случае, как показано на рисунке 1.1, каждый узел состоит из CAN-контроллера, обеспечивающего взаимодействие собственно с CAN-шиной и некоего CPU или контроллера, который реализует алгоритмы верхнего уровня – или надстройку над CAN, или же, в дополнение, еще и реализацию технологических алгоритмов управления, сбора данных и т.п.

Физически, CAN-шина состоит из дифференциальной пары проводников, один из которых называют доминантной линией, а другой – рецессивной линией. При этом стараются не использовать описание битов как «0» или «1», а переходят к понятиям «доминантного» и «рецессивного» битов. При этом, поскольку физическое подключение устройств к сети рассматривают, как «проводное или», то при выставлении одним устройством доминантного бита, а другим – рецессивного, на линии будет удерживаться доминантный бит. Если рассматриваем проводное соединение, то доминантному состоянию будет соответствовать напряжение близкое к нулю вольт, а рецессивному – к максимальному напряжению. Другими словами, с использованием жаргонизмов, доминантный бит «подтягивает» шину к «земле», а рецессивный – к напряжению питания.

Т.е. в соответствии со стандартом CAN-шины, имеется требование, что доминантный бит должен подавлять рецессивный. Если же рассматривать не проводные каналы связи, а, например, оптический, то доминантному биту будет соответствовать наличие света, а рецессивному – его отсутствие. При рассмотрении связи по радиоканалу отсутствие сигнала будет рассматриваться как рецессивный бит, а наличие – как доминантный.

В случае, когда линия связи находится в рецессивном состоянии, то перевести в рецессивное состояние ее сможет любой узел – подав световой сигнал в оптоволокне или закоротив линию на землю в проводной связи, а если линия находится в доминантном состоянии, то выйти из него она сможет только после того, как узел, выставивший доминантное состояние

снимет его – или выключит свет, или уберет «подтягивание» линии к «земле».

На физическом уровне протокола CAN стандартами ISO 11898, ISO 11519, SAE J2411 определяются характеристики кабеля, уровни электрических сигналов и так далее.

Как правило, в транспортных приложениях, на физическом уровне пользуются стандартом ISO 11898. При этом в качестве среды передачи используется двухпроводная дифференциальная пара проводов с волновым сопротивлением 120 Ом. Эту линию с двух сторон необходимо ограничивать терминаторами сопротивлением по 120 Ом каждый. При этом допускаются колебания импеданса от 108 до 132 Ом. Терминаторы убирают отражения сигнала на концах шины и ограничивают уровни постоянного тока между линиями.

Существует второй тип шин CAN, описанный в третьей части стандарта ISO 11898–3. Это тип шин описывается как менее скоростная но также двухпроводная сбалансированная сигнальная схема (дифференциальная пара). Эта реализация шины более устойчива к сбоям и она будет сохранять функционирование даже при обрыве одной из линий дифференциальной пары. Кроме этого, если в результате аварии, потенциал одной из линий будет равняться потенциалу «земли» или потенциалу напряжения питания, эта версия шины всё равно сохранит работоспособность.

Существует даже однопроводный вариант CAN-шины, когда нет дифференциальной пары, а есть общий провод и один информационный. Примером стандарта, описывающего такую шину является SAE J2411. Этот протокол версии CAN-шины используется, в основном в автомобилях, как пример – GM-LAN.

CAN устройства, реализованные на физическом уровне по разным стандартам, как правило не могут работать друг с другом в одной сети. Например, стыковка в рамках одной шины устройств протоколов high-speed и low-speed может быть работоспособной только в редких случаях.

Для работы с проводной частью шины используются специализированные микросхемы – CAN приемо-передатчики (трансиверы), преобразующие уровни сигналов TTL в уровни сигналов на шинах CAN. Кроме этого трансиверы могут содержать дополнительный уровень блоков, которые выполняют функции контроллера шины CAN, обеспечивающего работу с ошибками, подсчет контрольных сумм, управление выставлением необходимых уровней сигналов на линиях шины при наступлении тех или иных событий.

На следующем уровне физического управления уже могут находиться различного типа микроконтроллеры, ПЛК, промышленные компьютеры, а также устройства, выполняющие функции либо исполнительных устройств, либо устройств сбора данных.

Совокупность физических параметров среды передачи, а также производительность контроллеров, обеспечивающих реализацию протокольных вычислений и управляющих функций, задает скорость передачи информации в шине CAN. Максимальная скорость, которая может быть в сети, задается протоколом и он равняется 1Мбит/с. Однако, в соответствии с регламентом, максимальная длина кабеля при такой скорости ограничивается 40 м.

Надо учитывать, что одним из основных ограничителей скорости передачи информации в сети является механизм побитового арбитража, который будет описан ниже. Во время процедуры арбитража все узлы сети должны получать актуальный бит одновременно. Соответственно, линия должна обладать распределенными параметрами (волновым сопротивлением), позволяющим завершиться переходным процессам распространения и установления этого сигнала по всему кабелю за квант времени в CAN-шине.

Между скоростью передачи информации и длиной сети установлены соотношения, сведенные в таблицу 1.

Таблица 1 – Соотношение между длиной сети и скоростью передачи в ней.

Скорость передачи, Кбит/сек	Максимальная длина сети, м
1000	40
500	100
250	200
125	500
10	6000

Подключение различных устройств к проводной части шины осуществляется с помощью разъемных соединений.

При этом стандартом вид разъема не регламентируется. Однако, сложившаяся практика показывает, что наиболее распространенными разъемами являются

9–контактный DSUB, предложенный ассоциацией CAN In Automation;

5–контактный Mini–C и/или Micro–C, используется DeviceNet и SDS;

6–контактный Deutsch разъем, предложенный CANHUG для транспортных гидравлических систем.

Назначение контактов, рекомендованное ассоциацией CAN In Automation показано в таблице 2.

Таблица 2 - Рекомендованное CAN In Automation назначение выводов разъемов

Номер вывода	Обозначение линии	Функции
1	-	Резерв
2	CAN_L	Линия шины CAN_L (доминантная низкая)
3	CAN_GND	Заземление CAN
4	-	Резерв
5	(CAN_SHLD)	Опционально: экран CAN
6	(GND)	Опционально: заземление CAN
7	CAN_H	Линия шины CAN_H (доминантная высокая)
8	-	Резерв (линия ошибок)
9	CAN_V+	Опционально: питание

Подаваемое на разъем питание должно выдерживаться в диапазоне +7...+13В и обеспечивать ток нагрузки до 100мА. На разъемах штыревой части («папа») контакты 3 и 6 должны быть соединены между собой внутри прибора.

Распределение контактов в разъеме DSUB показано на рисунке 3

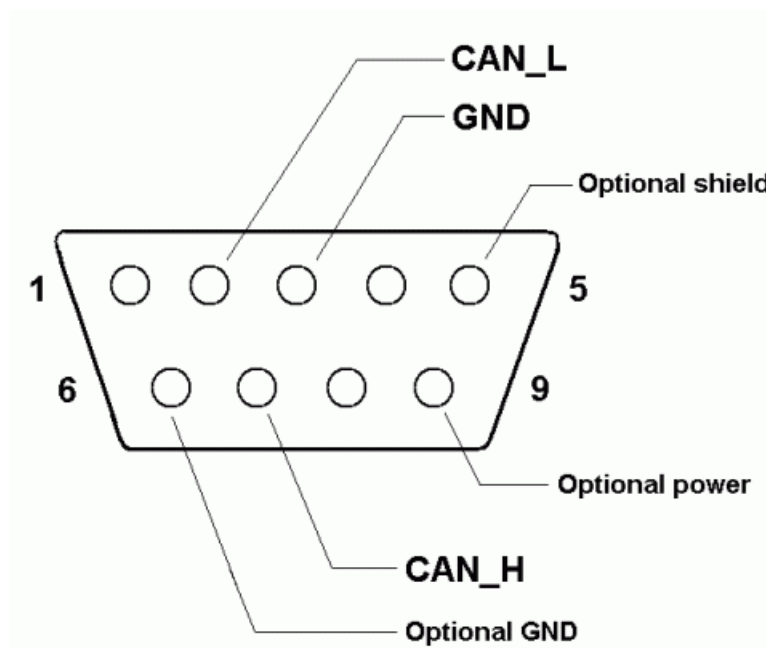


Рисунок 3 – Распределение контактов в разъеме типа DSUB для CAN шины, вид с стороны контактов

Следующий распространенный тип разъемов регламентируется протоколами DeviceNet и SDS. Внешний вид штыревого разъема со стороны контактов показан на рисунке 4.

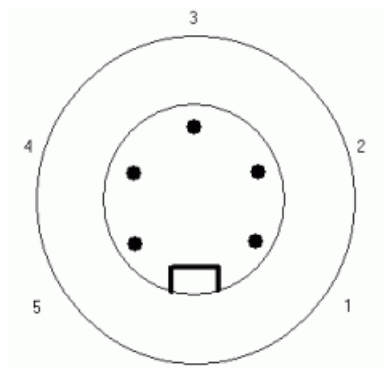


Рисунок 4 – Внешний вид и обозначение выводов 5-ти контактного разъема CAN, вид со стороны контактов

Назначение контактов, рекомендованное протоколами DeviceNet и SDS показано в таблице 3.

Таблица 3 - Рекомендованное DeviceNet и SDS назначение выводов разъемов

Номер вывода	Обозначение линии	Функции
1	CAN_SHLD	Экран CAN
2	V+	Питание
3	V-	Заземление CAN
4	CAN_H	Линия шины CAN_H (доминантная высокая)
5	CAN_L	Линия шины CAN_L (доминантная низкая)

6-контактный Deutsch DT04-6P

Рекомендован протоколом CANHUG для использования в гидравлических транспортных системах. На изделиях используется разъем штыревого типа («папа»), на проводах шины – гнездового типа («мама»). Внешний вид разъема показан на рисунке 5.



Рисунок 5 – Внешний вид разъема типа Deutsch DT04-6P

Назначение контактов, рекомендованное протоколом CANHUG показано в таблице 4.

Таблица 4 - Назначение выводов в разъемы Deutsch DT04-6P в соответствии с протоколом CANHUG

Номер вывода	Обозначение линии	Функции
1	V-	«минус» питания
2	CAN_H	Линия шины CAN_H (доминантная высокая)
3	Опционально: заземление	Заземление CAN
4	Опционально: запуск	
5	V+	Питание
6	CAN_L	Линия шины CAN_L (доминантная низкая)

Мы рассмотрели организацию воздействия на линию связи узлами на самом низком, физическом уровне. Теперь перейдем к рассмотрению организации следующего уровня в соответствии со стандартом – организацию логического уровня передачи информации.

1.2. Организация логического уровня CAN

Передача данных в CAN осуществляется кадрами. Это означает, что все сообщения на нижнем логическом уровне представляются в виде стандартизованных наборов битов.

Разделяют четыре типа кадров [15]:

- Data Frame;
- Remote Frame;
- Error Frame;
- Overload Frame.

Рассмотрим каждый из типов кадров.

Data Frame.

Этот тип кадров предназначен, в первую очередь, для передачи собственно данных. Его структура, в соответствии со стандартом, следующая:

- arbitration field (поле арбитража) – указывает уровень приоритета сообщения в системе, используется для принятия решения о порядке передачи сообщений, на случай, если несколько узлов будут одновременно пытаться передавать данные. Поле арбитража включает в себя идентификатор, который может состоять из 11-битного слова для стандарта CAN-2.0A и из 29 битного слова для стандарта CAN-2.0B, после чего располагается 1 бит RTR;

- data field (поле данных) включает в себя до 8 байт данных;

- CRC field (поле контрольной суммы) включает в себя 15 разрядов контрольной суммы сообщения для обнаружения на стороне приемника ошибок, возникающих в шине;

- acknowledgement slot – слот подтверждения размером 1 бит. Устанавливается в лог «1» контроллером, в случае, если контроллер верно принял сообщение. Этот бит отслеживается контроллером-источником информации и по его состоянию принимается решение об успешном или неуспешном принятии информации. Факт установки «1» в этот слот показывает, что хотя бы один узел в сети принял это сообщение.

Графически структуру кадра Data Frame можно изобразить так, как показано на рисунке 6.

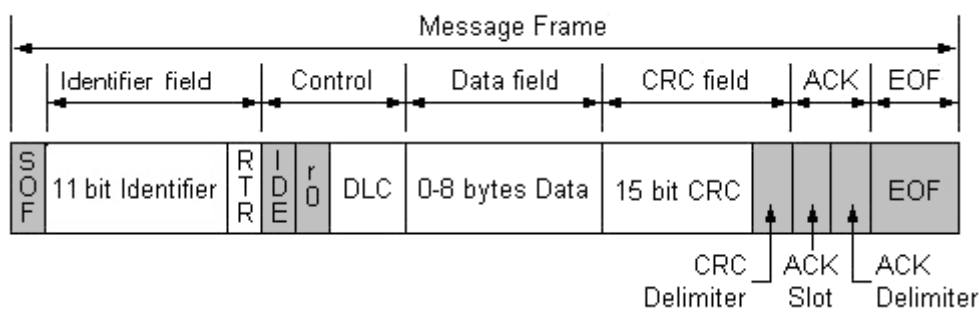


Рисунок 6 – Структура кадра Data Frame

Remote Frame.

«Кадр Remote Frame предназначен для указания одним из узлов сети того, что передачу будет осуществлять другой узел. Этот кадр аналогичен кадру Data Frame, только в нем отсутствует поле передачи данных. В кадре Remote Frame бит RTR выставлен в «1».» [15] Структура этого кадра показана на рисунке 7.

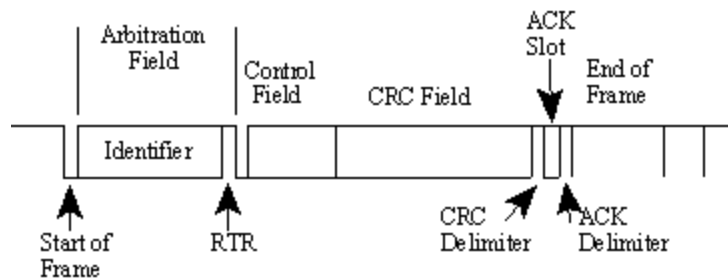


Рисунок 7 – Структура кадра Remote Frame

Error Frame.

Кадр Error Frame предназначен для асинхронного уведомления контроллера сети о необходимости повторной передачи информации. Если какой-то из узлов передает в сеть Error Frame, то все остальные узлы регистрируют ошибку и также начинают транслировать в сеть Error Frame. В результате, передающий узел начинает повторную передачу своего сообщения в сеть. Структура этого кадра показана на рисунке 8.

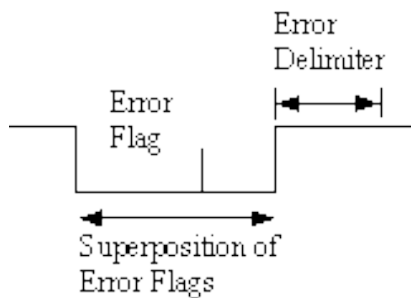


Рисунок 8 – Структура кадра Error Frame

В состав Error Frame входит поле Error Flag, состоящее из 6 бит, каждый из которых имеет одинаковые значения, что принудительно нарушает правила проверки Bit Stuffing. Также, в состав Error Frame входит поле Error Delimiter, состоящее из 8 рецессивных битов. Это поле позволяет

остальным узлам сети при обнаружении кадра Error Frame отправлять в сеть свои поля Error Flag.

Overload Frame.

Кадр Overload Frame повторяет кадр Error Frame и его выставляет перегруженный задачами узел, который в текущий интервал времени не может обрабатывать новые задачи. Выставляя кадр Overload Frame, перегруженный узел просит повторить адресованное ему сообщение.

Реалии нынешнего этапа развития электроники и ее производительности привели к тому, что кадр Overload Frame фактически не используется.

Рассмотрим как работает процедура арбитража, задаваемая значениями идентификаторов в поле arbitration field. Каждый узел, который собирается начать передачу данных в сеть сравнивает текущий бит, который он собирается передавать и выставляет на линию, с текущим состоянием сети. Если значения передаваемого бита и бита состояния сети совпадают, то анализируется-передается следующий бит. Эта процедура повторяется до тех пор, пока контроллер узла не столкнется с отличающимся состоянием сети. В случае, если контроллер передает доминантный бит (логический «0»), а состояние сети – рецессивное, то контроллер продолжит передачу, если же контроллер выставляет на линии рецессивный бит, а состояние сети - доминантное, то контроллер прекращает передачу.

Этот процесс следует понимать так, что состояние сети определяется и другими контроллерами, и если состояние битов их полей арбитража совпадает, то они продолжают процедуру арбитража, при этом, если у какого-то контроллера появится рецессивный бит, в то время, когда остальные передают доминантные, то контроллер с рецессивным битом прекратит передачу и конкурентный арбитраж продолжится между оставшимися контроллерами узлов.

Такую процедуру называют недеструктивным арбитражом. Он производится, как видно, уже на следующем уровне абстракции, по

сравнению с уровнем оценки состояния сети на стадии «занято-свободно», когда контроллер просто ждет появления рецессивного состояния сети.

Пример проведения недеструктивного арбитража для случая трех контроллеров в сети показан на рисунке 9.

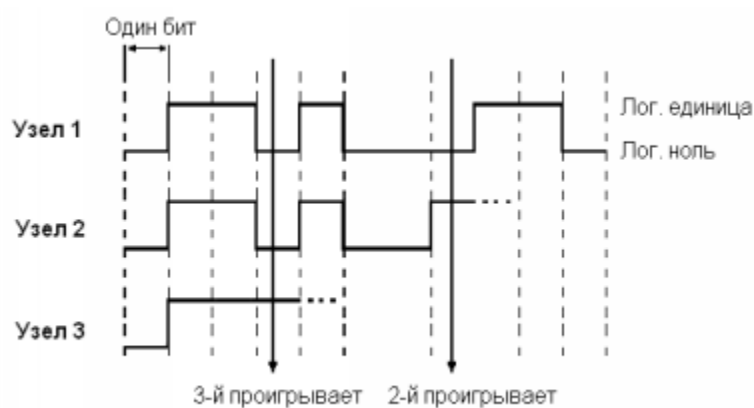


Рисунок 9 – Пример проведения процедуры недеструктивного арбитража для трех контроллеров в сети CAN

На этом рисунке видно, что когда у узлов 1 и 2 в поле арбитража установлен доминантный бит, а у узла 3 появляется рецессивный, то третий узел выходит из процедуры арбитража и прекращает попытку передать данные.

Когда у узла 1 выставлен доминантный бит, а у узла 2 – рецессивный, то из процедуры недеструктивного арбитража выходит узел 2 и он прекращает передачу. Таким образом арбитраж «выигрывает» узел 1 и он проводит процедуру передачи в сеть своего сообщения.

«Стандарт предоставляет пять способов обнаружения ошибок.

Это:

- Bit monitoring
- Bit stuffing
- Frame check

- ACKnowledgement Check

- CRC Check

Bit monitoring – каждый CAN-контроллер проверяет значение состояния шины с тем, которое он передает в настоящий момент. Если значения не совпадают, то контроллер выставляет бит Bit Error. Этот способ отключается, когда узлом в сеть передается поле арбитража arbitration field.

Bit stuffing – передающий узел передает в шину последовательно 5 бит с одинаковым значением, а шестой – с противоположным им значением. На принимающей стороне контроллер шестой бит удаляет. При обнаружении принимающим контроллером 6 бит с одинаковыми значениями, контроллер генерирует ошибку Stuff Error.» [15]

Frame check – контроллер производит контроль значений в стандартных полях CAN-сообщений, одинаковых для любых случаев, и если обнаруживает расхождение, то формирует ошибку Form Error.

ACKnowledgement Check – узел, который получил сообщение и проверил его, выставляет этот бит в доминантное значение. Если передающий узел не обнаруживает этот бит, то его контроллер фиксирует ошибку Acknowledgement Error.

CRC Check – по результатам приема каждого сообщения принимающий узел вычисляет контрольную сумму и сравнивает ее с принятым значением. Если принятое и вычисленное значения расходятся, то контроллер-приемник выставляет ошибку CRC Error.

Для того, чтобы ограничить количество ошибок, возникающих в сети, стандартом определен Механизм ограничения ошибок (Error confinement). Работает он следующим образом.

В процессе работы в CAN-сети, каждый контроллер узла отслеживает каждую из описанных выше ошибок. Если ошибка обнаружена, то контроллер не только генерирует соответствующую ошибку, которую обнаруживают устройства и алгоритмы верхнего уровня, но также генерирует в саму сеть кадр Error Frame. «Этот кадр разрушает текущий

трафик сети, останавливая процессы передачи и приема сообщений. Остальные узлы в сети, при обнаружении Error Frame, сбрасывают принятые сообщения. При этом каждый узел ведет свои счетчики ошибок Transmit Error Counter и Receive Error Counter.» [15] Первый ведет подсчет возникших ошибок передачи, второй - ошибок приема.

Для управления счетчиками существуют свои правила.

Ошибка передачи увеличивает счетчик Transmit Error Counter на 8. Ошибка приема увеличивает значение счетчика Receive Error Counter на 1. Случай правильного приема или передачи уменьшают счетчики на 1. Таким образом, значение Transmit Error Counter увеличивается быстрее, чем Receive Error Counter, т.е. предполагается, что вероятность появления ошибок на передающем узле выше, чем на приемном.

«После запуска работы сети, каждый узел находится в состоянии Error Active. Если значение одного из двух счетчиков контроллера узла превысит значение 127, то узел переходит в состояние Error Passive. Когда значение превышает 255, то в состояние Bus Off.

Если находящийся в состоянии Error Active узел обнаруживает в сети ошибку, то он передает в сеть поле Active Error Flags.» [15] Это поле состоит из 6 доминантных бит, а значит все узлы его зарегистрируют и проведут действия по соответствующему алгоритму. Если ошибку обнаруживает узел в состоянии Passive Error, то он передает в сеть поле Passive Error Flags. Это поле состоит из 6 рецессивных бит и остальные узлы его игнорируют, что в итоге только увеличивает счетчик ошибок самого узла. Если узел находится в состоянии Bus Off, то он фактически отключается от сети и не передает никакие кадры.

Этот механизм позволяет снижать влияние на сеть узлов с большим количеством ошибок передачи, увеличивая время использования сети другими узлами, которые можно считать менее аварийными.

Стандартом ISO 11898 предусмотрено несколько видов событий, которые являются аварийными. «Это:

1. CAN_H прерван
2. CAN_L прерван
3. CAN_H короткозамкнутый на напряжение батареи
4. CAN_L короткозамкнутый на землю
5. CAN_H короткозамкнутый на землю
6. CAN_L короткозамкнутый на напряжение батареи
7. CAN_L короткозамкнутый на провод
8. CAN_H и CAN_L прерваны в одном и том же месте
9. Потеря соединения с оконечной нагрузкой сети»[15]

В ситуациях 1-6 и 9 шина должна оставаться работоспособной. Для этого снижается допустимое соотношение сигнал/шум. Также, за счет снижения допустимого значения соотношения сигнал/шум, возможно сохранять работоспособность системы и в ситуации 7.

В практических ситуациях сохранение шиной работоспособности зависит от используемого трансивера CAN-шины. Одни приемопередатчики, особенно ранних разработок, как правило гарантированно не сохраняют работоспособность в ситуациях 1-7 и непредсказуемо сохраняют работоспособность в ситуациях 8 и 9.

1.3. Адресация CAN

Стандарты CAN из группы стандартов ISO 11898 не описывают требований к адресации узлов или индексации (адресации) сообщений. Имеющееся поле Identification field используется в механизме арбитража и описано как адресное. Тем самым стандарт разрешает располагать идентификаторы узлов и сообщений в произвольных местах кадра Data Frame: или в поле арбитража или в поле данных или и там и там.

Обязанность по детерминированию назначения фрагментов этих полей возлагаются на стандарты верхнего уровня по сравнению с протоколом CAN в стандарте ISO 11898 – протоколами Higher Layer Protocols (HLP).

Соотношение описаний CAN и HLP в соответствии с эталонной сетевой моделью ISO/OSI показано на рисунке 10.

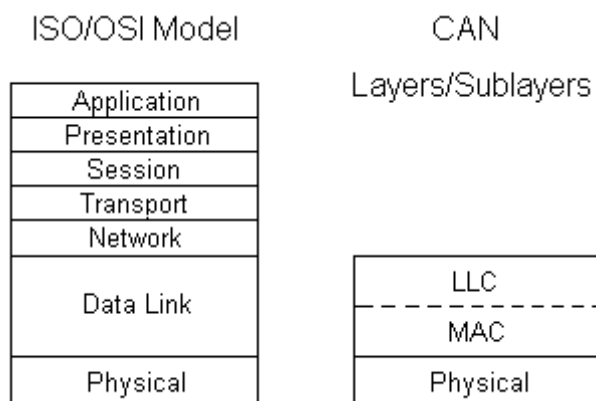


Рисунок 10 – Соотношение уровней, описываемых стандартом CAN ISO 11898

На сегодняшний день среди протоколов высшего уровня, функционирующих в среде CAN можно перечислить:

- CANopen
- DeviceNet
- CAN Kingdom
- J1939
- SDS
- NMEA-2000 (морской транспорт)
- ARINC-825 (авиация) (нем.)
- UAVCAN (робототехника и летательные аппараты)

Эти протоколы имеют достаточно локальное с позиции специализации применение. Они регламентируются своими стандартизованными в определенных сообществах протоколами и могут быть защищены

определенным набором патентных решений, или же, наоборот имеют открытую структуру, позволяющую использовать ее или при соединении к сообществу, или же на условиях ограничения теми или иными рамками коммерческого использования

1.4. Организация синхронизации CAN-шины.

Синхронизация шины необходима для того, чтобы все узлы в шине работали с одной и той же информацией в одни и те же промежутки времени (такты). Для организации тактовой синхронизации каждый бит разделяется не менее чем на 4 части:

- сегмент синхронизации;
- сегмент воспроизведения;
- сегмент фазы 1;
- сегмент фазы 2.

Схема разделения бита на кванты показана на рисунке 11.

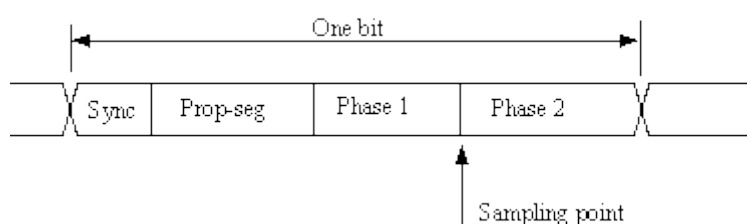


Рисунок 11 – Структура разделения бита шины CAN на кванты

Фронт сегмента синхронизации появляется каждый раз при смене данных на шине. Он имеет длину 1 квант.

Сегмент «Prop-seg» или сегмент воспроизведения служит для компенсации изменяющихся задержек на линии шины.

Сегменты фаз могут быть или укорочены или удлинены по мере необходимости для синхронизации тактовых частот.

Контроллер замеряет уровни шины на границе между сегментом фазы 1 и сегментом фазы 2.

Кроме того, есть физические реализации контроллеров шины, которые производят замеры уровней троекратно в рамках одного бита: на границах двух квантов перед точкой замера, а результат потом делится на три.

Регулирование генератора тактовых частот производится контроллером CAN за счет сокращения или удлинения одного бита на целое число квантов. Параметром Synchronization Jump Width, SJW («ширина скачка синхронизации») задается максимальное число временных поправок бита.

Так называемая жесткая синхронизация осуществляется при изменении стартового бита от рецессивного к доминантному уровню и с этой границы, как раз, и производится отсчет времени прохождения бита.

В случае, если край бита не попадает в сегмент синхронизации сообщения происходит повторная синхронизация. В этом случае, если какой-то из сегментов укорачивается или удлиняется в соответствии со значением параметра Synchronization Jump Width.

Как правило, разработчик может настраивать тактовую синхронизацию в контроллере шины программными средствами. Типовые контроллеры предоставляют для этого следующие параметры:

- Значение предварительного делителя тактовой частоты
- Количество квантов перед точкой замера
- Количество квантов после точки замера

- Количество квантов в параметре «ширина скачка синхронизации» (Synchronization Jump Width)

Типовым образом в контроллерах для управления этими параметрами имеются два регистра btr0 и btr1.

Структура этих регистров, например, в контроллерах NXP показана в таблице 5:

Таблица 5 - Структура регистров btr0 и btr1

разряд	7	6	5	4	3	2	1	0
btr0	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
btr1	SAM	TSEG22	TSEG21	TSEG20	TSEG13	TSEG12	TSEG11	TSEG10

- BRP0..BRP5 – биты значения предварительного делителя тактовой частоты;

- SJW0..SJW1 биты длину SJW

- TSEG10..TSEG13 устанавливают количество квантов перед точкой замера (стартовый бит не включен)

- TSEG20..TSEG22 разряды количество квантов после точки замера

- SAM задает количество замеров: при 1 - производится три замера, при 0 – один замер.

Например, при тактовой частоте генератора 16 МГц и желаемой скорости передачи в шине 250 кбит/сек, расположением точки замера на уровне 62% и значению SJW=2 кванта, устанавливаем BRP = 4 и при этом продолжительность кванта будет равняться $2 \times 4 / 16000000 \text{ с} = 500 \text{ нс}$, устанавливая TSEG1 = 5 получаем 5 квантов перед точкой замера, задавая TSEG2 = 3 получаем 3 кванта после точки замера. В итоге каждый бит будет содержать 5+3=8 квантов. При этом результирующая скорость передачи $1 / (8 \times 500 \text{ нс}) = 250 \text{ кбит/с}$, что совпадает с желаемым значением в начале примера.

1.5. Протокол CANopen

Мы рассмотрели физическую организацию, и организацию на нижнем протокольном уровне шины CAN, однако для эффективной работы разработчиков аппаратной и программной частей различного оборудования, необходим следующий уровень надстройки над протоколом, который позволял бы оперировать с шиной в более упрощенной и определенной форме.

Такой надстройкой является протокол CANopen, обеспечивающий прикладной уровень в модели OSI. Рассмотрим его более детально.

CANopen — открытый сетевой протокол верхнего уровня для подключения встраиваемых устройств в бортовых транспортных и промышленных сетях [9]. Он получил наибольшее распространение в промышленности на оборудовании выполняющем различное перемещение грузов, роботах-манипуляторах, сварочных линиях, транспортерных и конвейерных системах. В этих системах протокол обеспечивает связь между главным управляющим устройством (как правило ПЛК или промышленным компьютером) и блоками датчиков, различными интеллектуальными датчиками, пневматическими и гидравлическими вентилями и прочими устройствами ввода данных и информации.

Как показала практика, CANopen, по сравнению с другими версиями протоколов высокого уровня, наиболее приспособлен для использующихся в промышленности и на транспорте систем управления перемещениями и в контурах регулирования с обратными связями.

Основой протокола CANopen является серия документов, регламентирующая его устройство и работу с ним. К этим документам относятся протоколы от CiA 201 до CiA 207 под единым названием «CAN Application Layer (CAL) for industrial applications» [8,11].

Функциональная схема связи двух узлов в CAN-шине с помощью протокола CANopen показана на рисунке 12 [35].

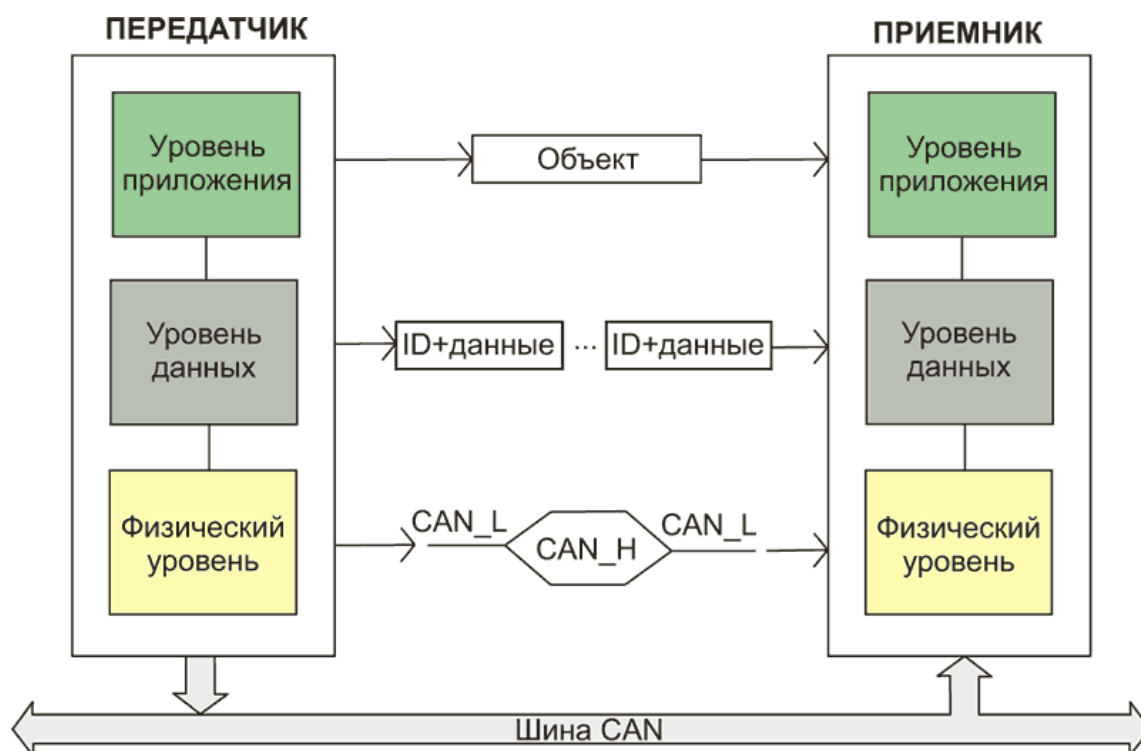


Рисунок 12 - Функциональная схема связи двух узлов с помощью шины CAN и протокола CANopen

Протокол описывает свою основную единицу как объект. Объектом могут быть данные, конфигурационные параметры сети или узла, возникающие ошибки и т.п. [38]

Понятием следующего уровня является словарь. Словарь – это сгруппированные особым образом объекты. Словарь объектов является связью между приложением и передаваемыми на физическом уровне данными. Структура модели устройства с интерфейсом CANopen показана на рисунке 13.

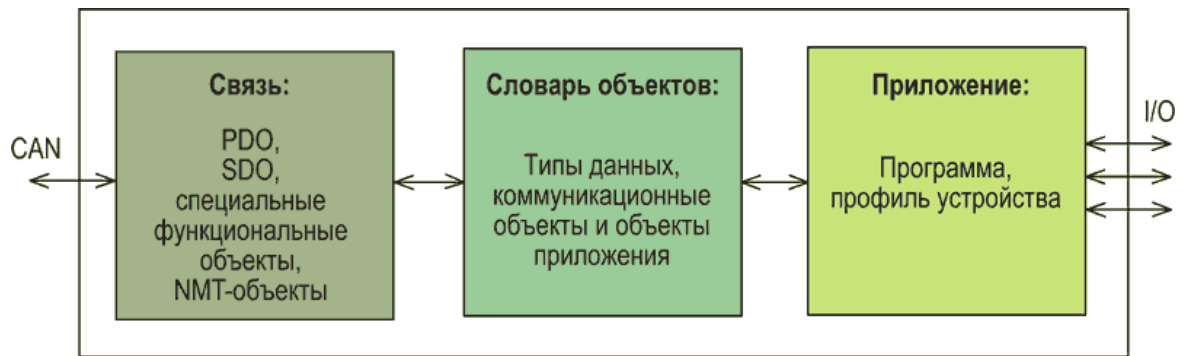


Рисунок 13 - Модель устройства с интерфейсом CANopen

Если производитель какого-нибудь узла или устройства выпускает его для использования в сети CAN по протоколу CANopen, то он должен также подготовить его профиль в виде файла *.eds (Electronic DataSheet).

CANopen-устройство имеет три условных составляющих: программный модуль обработки протокола и пакетов интерфейса, словарь объектов и программное обеспечение на уровне приложения.

CANopen различает следующие типы сообщений:

- Layer Management (LMT) и Network Management (NMT) – сообщения управления сетью;

- Service Data Objects (SDO) – сообщения, связанные контролем корректной работы, конфигурированием узлов и прочими сервисными моментами;

- Process Data Objects (PDO) – сообщения, связанные непосредственно с технологическими, процессными данными, которые связаны с физическими величинами, измеряемыми датчиками, командами от контроллеров на исполнительные устройства и т.п.

- группа predetermined сообщений являющимися техническими и называемыми: Sync Object (SYNC), Time Stamp Object, Emergency Object (EMCY).

Работа шины по протоколу CANopen начинается с процедуры инициализации, включающей в себя распределение идентификаторов CAN

для SDO/PDO соединений и сервиса обработки ошибок [31,2222]. При включении устройство проходит этап внутренней инициализации, и после ее успешного завершения переходит в состояние готовности. В этом состоянии уже возможно осуществить настройку CANopen-узла с помощью SDO.

Протоколом описывается также процедура управления сетью:

- распределение идентификаторов CAN SDO/PDO соединений и сервиса обработки ошибок в динамическом или статическом режимах;
- управление состоянием работы устройств;
- контроль режимов соединений в устройствах;
- периодический опрос устройств для определения сбоев в устройствах;
- обработка сообщений устройств об их нормальном функционировании.

Сообщения типа SDO обеспечивают доступ к элементам словаря объектов. Для осуществления связи методом SDO между связываемыми узлами устанавливается соединение типа «точка-точка» с реализацией «клиент-серверной» модели взаимодействия, схема которого показана на рисунке 14.

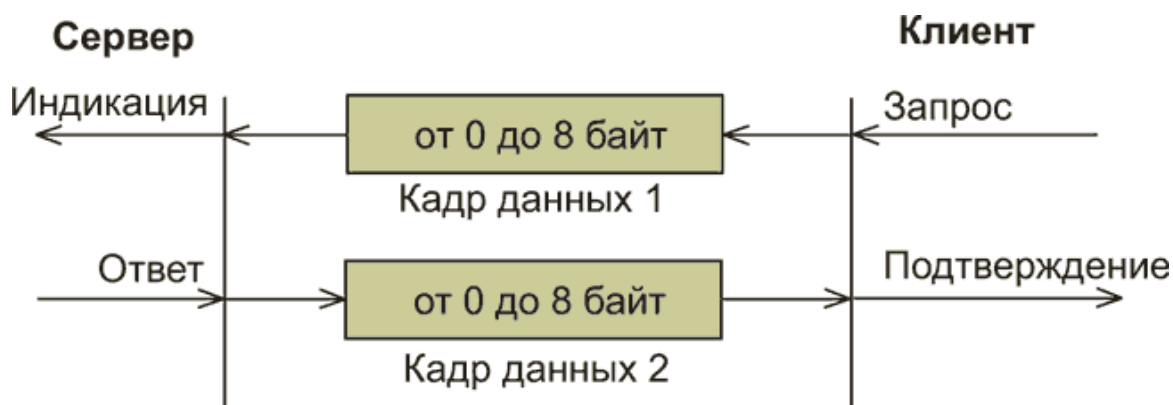


Рисунок 14 – Модель взаимодействия методе SDO

Клиент, используя индекс и подиндекс объектного словаря, предоставляет серверу набор данных для передачи. SDO передаются как

последовательность сегментов. До передачи выполняется подготовка клиента и сервера к обмену сообщениями [7]. Существует несколько типов передачи:

- expedited transfer – ускоренная передача, производимая на стадии инициализации узла, применяется в случаях, когда передаваемые данные занимают меньше 4 байт;

- block transfer – передача набора блоков, каждый состоит из 127 сегментов, в одном сегменте располагаются данные и их порядковый номер сегмента в блоке.

Данные технологического процесса (скорость, угол поворота, ускорение, температура...) передаются объектами PDO. Их передача происходит широковещательно с использованием модели «производитель-потребитель» (producer-consumer). Если узел выдает PDO на шину, то он называется производителем, а узел, принимающий это PDO – потребителем. Модель такого взаимодействия приведена на рисунке 15.

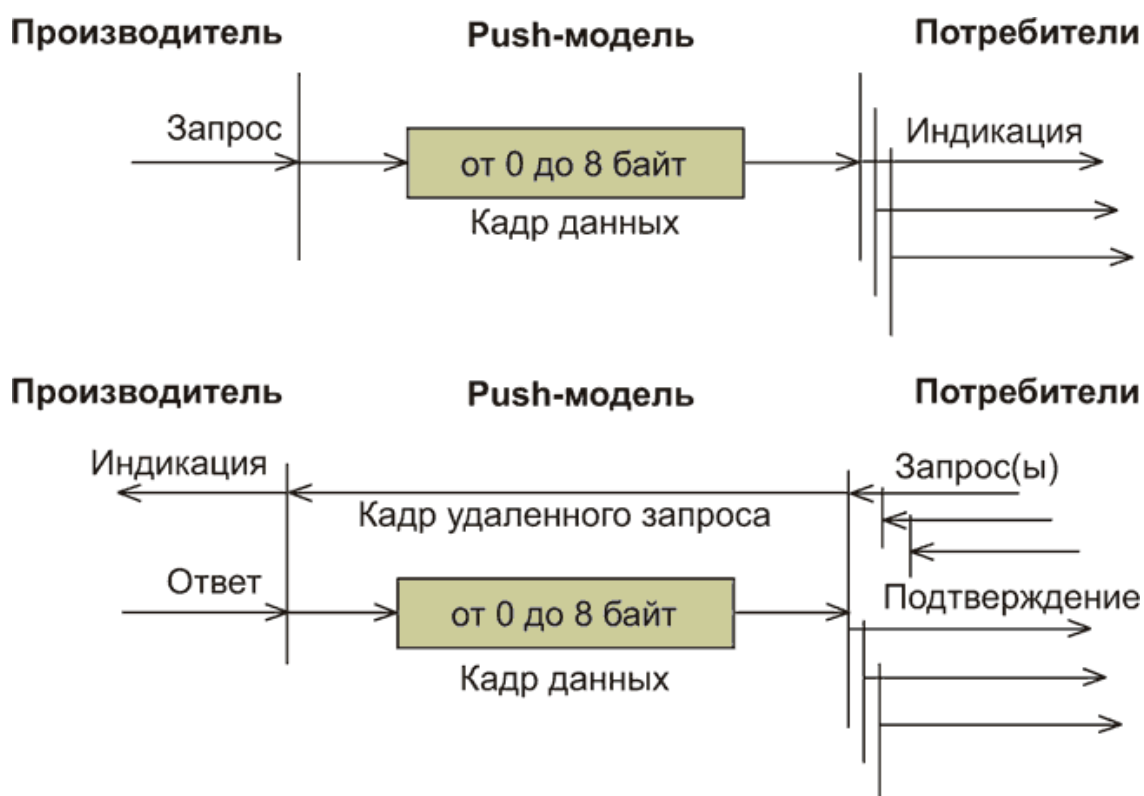


Рисунок 15 – Модель «производитель-потребитель» при передаче объектов PDO

При этом различают два подкласса объектов PDO: TPDO для передачи данных и RPDO - для приема данных, что учитывается при описывании в объектном словаре устройства.

В одном PDO можно передать от 1 до 8 байт данных. Одна сеть CANopen может содержать до 512 TPDO и до 512 RPDO.

Сеть CANopen имеет синхронизирующую последовательность битов, которую называют объектом SYNC. Относительно этого объекта сообщения в сети могут передаваться или синхронно, или асинхронно. Синхронная передача производится в течение определенного промежутка времени после прохождения объекта SYNC. Также в сети может производиться асинхронная передача сообщений. Это происходит, если «производитель» не привязывается к факту появления в сети объекта SYNC. Если представить последовательность битов объекта SYNC в виде периодических синхроимпульсов, то синхронные и асинхронные PDO можно представить, как показано на рисунке 16.

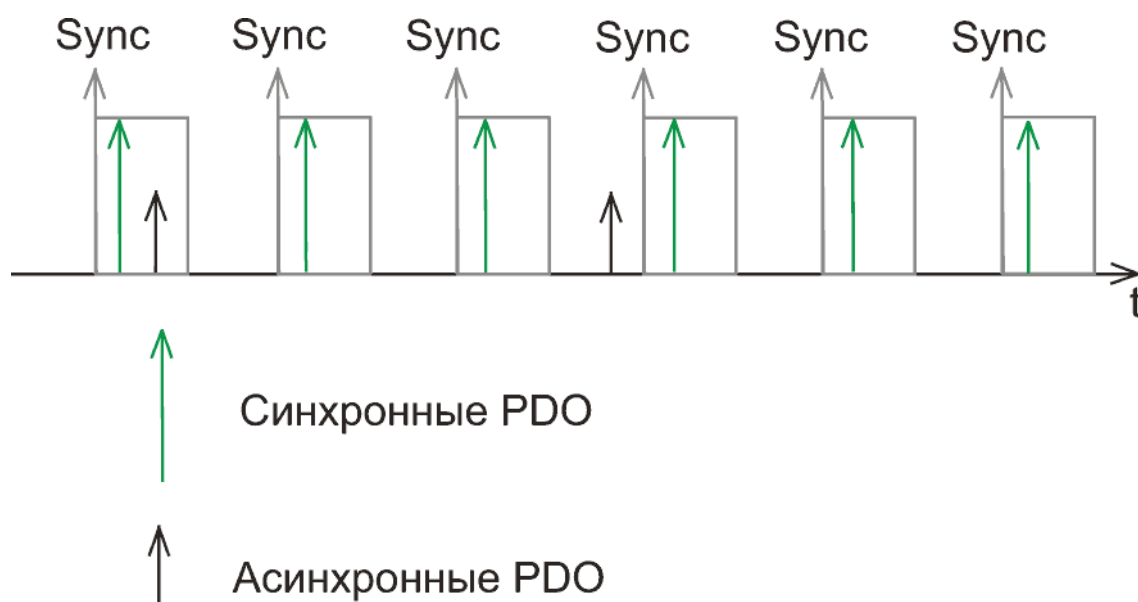


Рисунок 16 – Синхронные и асинхронные PDO

Протоколом описаны три режима вызова PDO:

- по событию или по таймеру: передача PDO производится при возникновении внутреннего события или по срабатыванию таймера устройства;

- по удаленному запросу: устройство начинает передачу PDO после получения кадра удаленного запроса (Remote Frame) от другого устройства на шине;

- синхронная передача (циклическая или ациклическая): производится при появлении SYNC-объекта.

Схемы режимов вызова PDO показаны на рисунке 17.

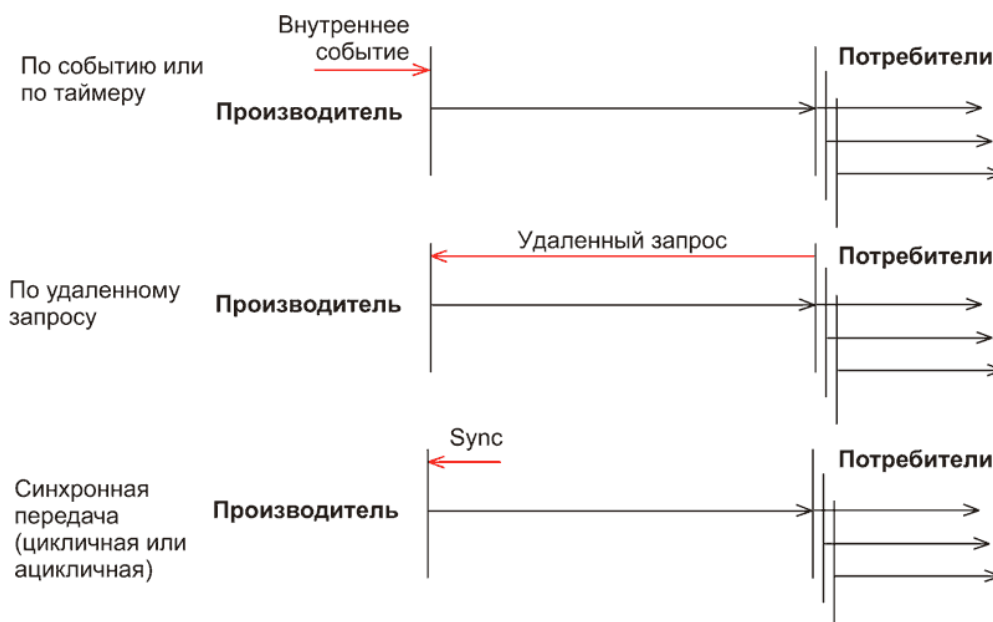


Рисунок 17 – Режимы вызова PDO

Синхронные PDO могут передаваться либо в циклическом, либо в ациклическом режимах. Циклический режим подразумевает выставление данных на шину через фиксированные промежутки времени – периоды. Ациклический режим – выставление в произвольные моменты времени, но с обязательной привязкой к синхрособытиям SYNC.

«Производитель» и «потребитель» должны иметь одинаковое представление о правилах интерпретации PDO-сообщения. Для этого служит PDO-отображение. Оно описывает какие данные и каким образом должны передаваться и упорядочиваться, их тип и длину. Словари объектов и на стороне «производителя» и на стороне «потребителя» должны содержать записи PDO-отображения о содержании и значениях поля данных каждого конкретного PDO. Принцип отображения PDO показан на рисунке 18.

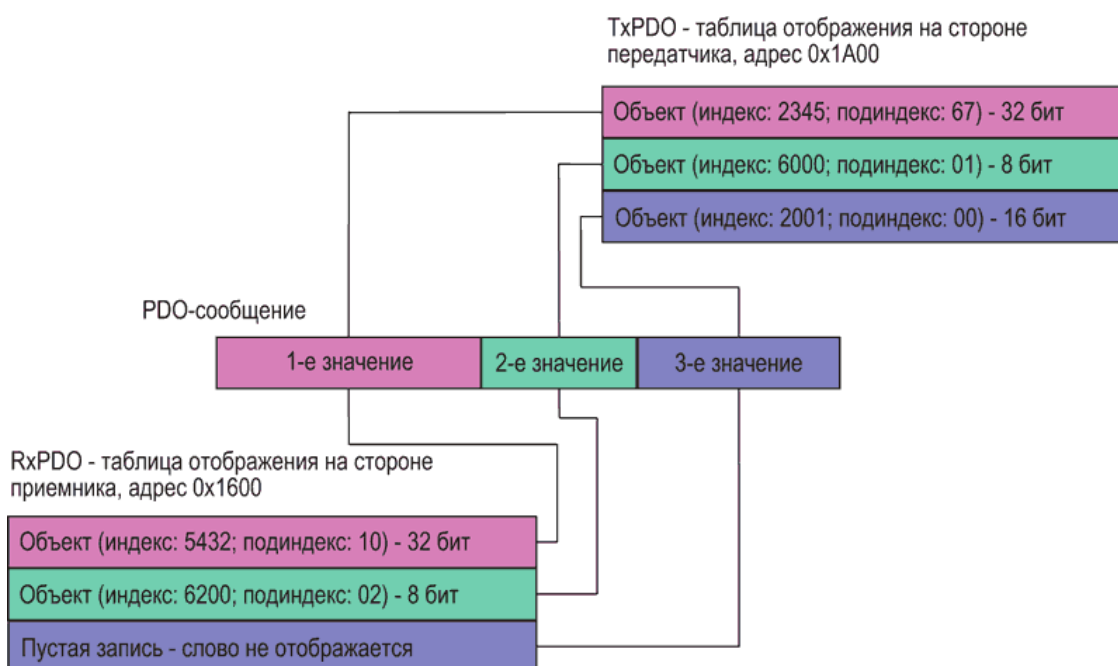


Рисунок 18 – Принцип отображения PDO

Управление работой сети по протоколу CANopen производится с помощью NMT-объектов по модели «Master-Slave».

В каждый момент времени узел в CAN-сети может находиться в одном из четырех состояний, протоколом CANopen определяемых, как [10]:

- Initialization (инициализация);
- Pre-operational (готовность);
- Operational (работа);
- Stopped (остановка).

После включения происходит процедура инициализации. При успешном завершении инициализации узел переходит в состояние готовности. В этом состоянии производится настройка параметров узла с помощью SDO. После настройки параметров и получения от сетевого Master широковещательного сообщения Start_remote_node, узел переходит в рабочее состояние.

NMT-сообщения имеют самый высокий приоритет в сети, поэтому их ID=0. Описание NMT сообщений показано в таблице 6.

Таблица 6 - Описание NMT-сообщений

Сообщение	Обозначение	Команда в составе формата сообщения
Запуск удаленного узла	Start_remote_node	1
Остановка удаленного узла	Stop_remote_node	2
Вход в состояние готовности	Enter_pre_operational	128
Сброс узла	Reset_node	129
Сброс связи	Reset_communication	130

В состав NMT-сообщения также входит ID узла-адресата. Сообщение вида 0/1/0 запускает все узлы на шине, а сообщение 0/2/9 останавливает узел с ID=9.

Объекты срочных сообщений (Emergency Object или EMCY) передаются при возникновении на каком-то узле внутренних ошибок. EMCY передается в сеть один раз сразу после возникновения какой-либо ошибки и не повторяется в дальнейшем. Новое сообщение EMCY от того же самого узла передается если в нем возникла новая ошибка, с другим ID. Механизм передачи сообщений класса EMCY позволяет при эксплуатации системы вовремя определять и устранять ошибки и неполадки в узле.

Кроме этого, протоколом CANopen предусмотрены процедуры контроля функционирования шины: процедура Node guarding protocol

(протокол караула узлов) и Heartbeat protocol (протокол контрольного тактирования).

При реализации процедуры Node guarding protocol NMT-мастер через равные промежутки времени посылает запросы Slave-устройствам, которые в ответ шлют свой статус. Время ожидания ответа настраивается для каждого узла в зависимости от его характеристик и загруженности в системе. Если через эти промежутки времени к узлу не пришел запрос от ведущего (Master-узел), то возникает ошибка сторожевого запроса Life Guarding Event. Если же от Slave-устройства не пришел ответ о его сетевом статусе или пришедший ответ не соответствует установленному для Slave-устройства статусу, то со стороны Master'a выставляется ошибка караула узла - Node guarding event.

В процедуре Heartbeat-протокола нет необходимости рассылать запросы Slave-устройствам. Узел, настроенный для широковещательной трансляции Heartbeat-сообщения, производит так называемые «контрольные такты». Прочие узлы настраиваются на прием Heartbeat-сообщения и называются «потребителями контрольных тактов». Они генерируют ошибку контрольного тактирования в случае, если за время тактового промежутка не получают Heartbeat – сообщения.

Следует отметить что протоколы Node guarding protocol и Heartbeat protocol являются взаимоисключающими и на практике по умолчанию используется Heartbeat protocol.

Таким образом мы рассмотрели основные моменты протокола верхнего уровня CANopen, который является надстройкой над CAN и предназначен для построения реальных систем, использующих CAN-шину.

2. Приемопередатчики для сетей CAN

Рассмотрим, как же на физическом уровне, с использованием электронной элементной базы реализуется работа устройств с шиной CAN.

Как было показано выше, непосредственно шину организуют провода CAN_L, CAN_H, которые образуют между собой дифференциальную или сбалансированную пару. При этом должны существовать устройства, которые будут на физическом уровне обеспечивать подачу напряжения на эти линии в соответствии с теми битами, которые необходимо передать по шине.

Эти устройства называют трансиверами и главная их задача - преобразование сигналов из логического TTL уровня в уровни напряжения, необходимые для функционирования CAN шины.

Рассмотрим типовую структуру приемо-передающего устройства – трансивера.

Физически трансиверы выполнены в виде отдельной микросхемы, на входные контакты которой подаются логические TTL уровни сигналов, а выходные контакты подключаются к линиям CAN_L, CAN_H.

Внутренняя структура трансивера показана на рисунке 19 [20].

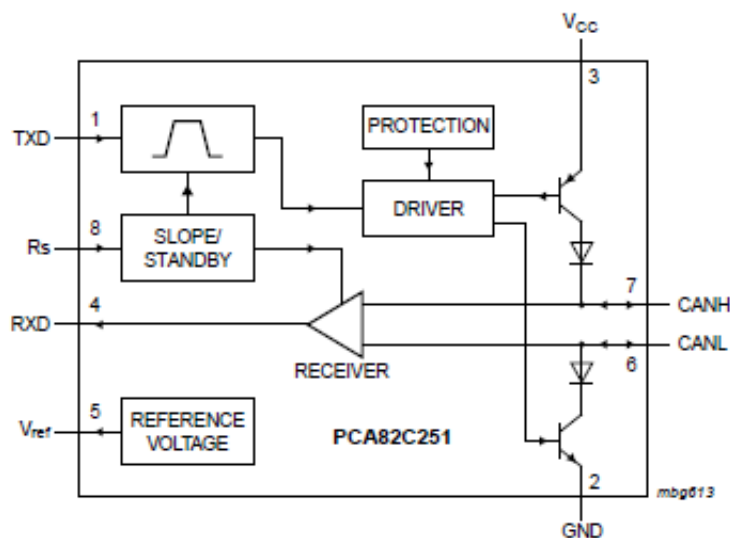


Рисунок 19 – Внутренняя структура трансивера CAN-шины

Внутри трансивера располагается тракт приема, построенный на усилителе синфазного сигнала на основе операционного усилителя. Он обеспечивает обнаружение рецессивного и доминантного сигналов на шине и формирует цифровой выходной сигнал на выводе Rx.

Передача сигнала построена на основе буфера цифрового сигнала, поступающего на вывод Tx, который управляет работой формирователя фронта сигнала. Формирователь фронта ограничивает скорость нарастания фронта сигнала, снижая тем самым уровень импульсных составляющих на линиях шины, что снижает уровень электромагнитного излучения. Тем самым появляется возможность не подключать к шине дополнительные сглаживающие дроссели, а следовательно снизить стоимость и уменьшить габариты всей системы. С формирователя фронта управляющий сигнал поступает на драйвер, управляющей работой выходных транзисторов, верхний подключает линию CAN_H через диод к положительному полюсу источника напряжения шины, а нижний – к общей точке. В доминантном состоянии открываются оба транзистора и к линии прикладывается напряжение питания шины.

Таймер задает границы временных интервалов логических состояний на линиях шины. Т.е. он выполняет роль генератора «синхроимпульсов». Ограничение длительности доминантного состояния на аппаратном уровне, в этом случае, является одной из защитных функций шины, не позволяющей вышедшему из строя контроллеру или при возникновении программного сбоя одному узлу парализовать работу всей шины.

Это можно рассматривать как одно из защитных мероприятий физического уровня в категории Bit stuffing.

Узел температурной защиты закрывает транзисторы, когда их температура превысит определенных порог. На практике, как правило, у различных производителей это около 160С. Закрытие транзисторов остановит выделение мощности на них и позволит остыть всему кристаллу

трансивера. При этом, как правило, остальные функции микросхемы остаются функционирующими. Аварийная ситуация, от последствий которой защищает термоконтроль, возникает в случае короткого замыкания между линиями, когда токоограничивающие терминаторы на концах линии оказываются шунтированными.

Приемопередатчики могут быть предназначены для работы с уровнями логических сигналов как +5В, так и +3В и +3.3В. Соответственно, в состав микросхемы могут входить и узлы стабилизации напряжения, согласующие входы трансивера с сигналами различного уровня.

Существуют так называемые энергоэффективные трансиверы, которые имеют отдельный вход для введения их в состояние пониженного энергопотребления. В этом режиме выключаются все узлы трансивера, а контроль состояний линий шины производится отдельным узлом маломощного операционного усилителя.

Кроме этой функции, некоторые трансиверы предоставляют возможность контроля скорости работы шины, что позволяет контроллеру шины реализовывать функцию автоподстройки скорости. Работает это следующим образом.

На вход AUTB (Autobaud) подается высокий логический уровень, который переключает вход TxD от передатчика на выход RxD. При этом происходит логическое объединение («И») с данными, поступающими от приемника. Таким образом, контроллер получает из сети данные и сравнивает с собственной текущей скоростью работы. Если принимается пакет на скорости отличной от собственной, то формируется Error Frame, который по образованной в трансивере петле возвращается назад в контроллер. Контроллер может перебирать разрешенные стандартом скорости и находить такую, на которой кадр ошибки формироваться не будет, что означает нахождение скорости работы сети.

При этом внутреннее в трансивере отключение от сети необходимо для того, чтобы кадр ошибки не проходил в сеть, нарушая ее работу.

Мы рассмотрели обобщенную структуру трансиверов, преобразующих биты логического уровня TTL напряжений 5 или 3 В в доминантные или рецессивные состояния CAN-шины. Такие трансиверы обладают минимальной отказоустойчивой, достаточной, тем не менее для большинства приложений. Рассмотрим теперь более сложные структуры, в которых вводятся элементы отказоустойчивости от различных событий в сети.

В качестве примера такого усложненного трансивера рассмотрим микросхему AMIS-41682 производства On-semiconductor [37]. Внутренняя структура этой микросхемы показана на рисунке 2.2.

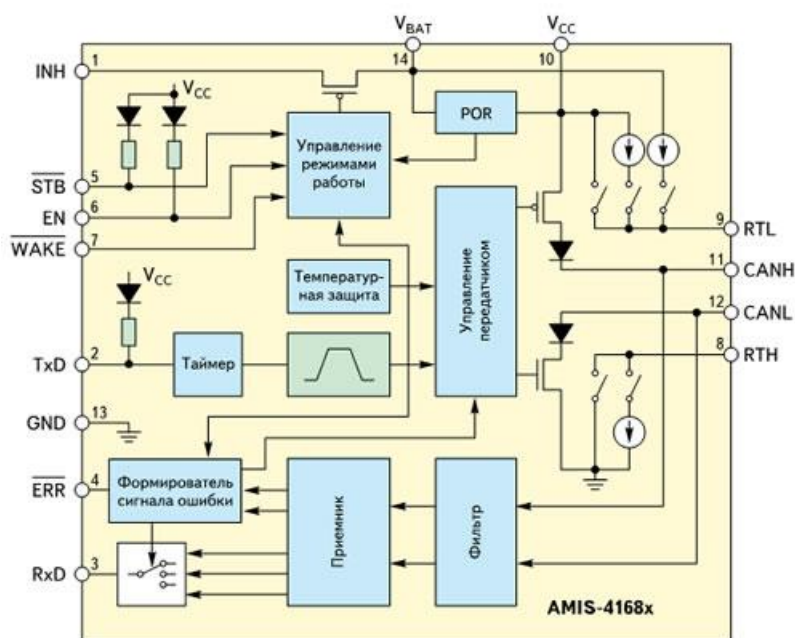


Рисунок 20 – Структура ИС трансивер AMIS-41682

Относительно универсальных приемопередатчиков появился ряд дополнительных выводов и внутренних блоков.

К выводам RTL и RTH можно подключать внешние резисторы для автоматических коммутаций, направленных на восстановление функционирования приемопередатчиков после устранения ошибочных или аварийных состояний непосредственно на шине.

Ко входу VBAT можно подключать внешний источник питания шины с напряжениями 7-36В. Обычно в роли такого источника выступает или автомобильный аккумулятор, или непосредственно бортовая электрическая сеть. При этом напряжение питания Vcc формируется встроенным стабилизатором из напряжения VBAT.

На вывод INH можно подавать сигналы отключения стабилизатора напряжения от Vcc. С помощью выводов INH и VBAT можно осуществлять коммутации, устраняющие последствия от замыкания CAN-H или CAN-L на питающую бортовую сеть.

Выходы STB, EN (Enable) и WAKE задают режимы работы и энергосбережения передатчика.

В состав схемы трансивер включен также узел POR, производящий процедуру сброса после появления напряжения питания для установления трансивера в начальное состояние.

Выход ERR – это вывод признака наличия ошибки или нештатного режима работы трансивера.

Таким образом, дополнительные отказоустойчивые функции трансивера позволяют выявлять нештатные ситуации на CAN-шине, выполнять необходимые коммутации, защищающие трансивер от выхода из строя и позволяющие самой шине продолжать функционирование.

В итоге, с такими трансиверами, можно продолжать эксплуатировать CAN-шину при некотором снижении помехоустойчивости, если останется подключенной хотя бы одна линия CAN-H или CAN-L. Для этого на стороне приемника соответствующий трансивер выполняет необходимые коммутации, используя вместо неисправной линии фиксированное напряжение.

Рассмотрим пример, проиллюстрированный рисунком 2.3.

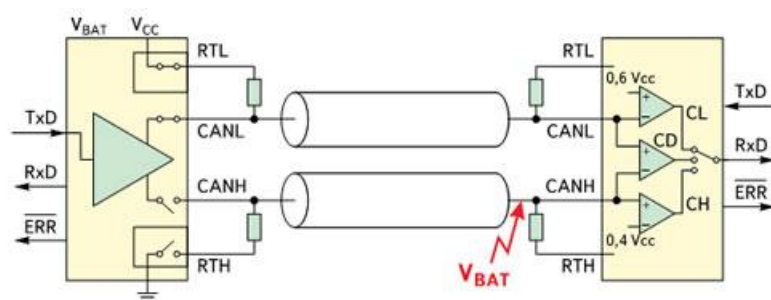


Рисунок 21 – Пример неисправности на линии

Ситуация следующая: на линию CAN-H попал потенциал бортовой сети (аккумулятора), для того, чтобы сохранить функционирование шины на стороне передатчика выполняется размыкание неисправной линии CAN-H от выхода и «земли», а на стороне приемника –используется внутреннее напряжение $0,6 V_{re}$ в качестве замены. Всё это адекватно работает, когда узлы сети выполняются на однотипных схемотехнических решениях с одинаковыми скоростными установками.

Рассмотрим некоторые примеры наиболее распространенных приемопередатчиков.

В настоящее время на рынке РФ наиболее распространены изделия марок NXP, Texas Instruments, Maxim, ON Semiconductor, а также ряда других производителей.

Типовыми решениями, давно присутствующими на рынке можно назвать микросхемы PCA82C250, производства NXP, ведущую свою историю еще от компании Philips Semiconductors. Однако развитие электроники, усложнение сетей предъявляют к трансиверам новые требования.

Приведем в таблице 7 краткие параметры и характеристики микросхем NXP [32].

Таблица 7 - Краткие параметры и характеристики микросхем CAN-трансиверов NXP

Наименование	Диапазон напряжения питания, В	Максимально допустимое напряжение на шине CAN, В	Максимальная скорость передачи, кбит/с	Ток потребления в экономичном режиме, мкА	Диапазон рабочей температуры, °С	Корпус
PCA82C250T/N4	4,50...5,50	-36...+36	1000	250	-40...125	SOIC-8
PCA82C251/N3	4,50...5,50	-36...+36	1000	250		SOT97-1
PCA82C251T/N3	4,50...5,50	-36...+36	1000	250		SOIC-8
TJA1040T/N1	4,75...5,25	-27...+40	1000	15		SOIC-8
TJA1040T/V	4,75...5,25	-27...+40	1000	15		SOIC-8
TJA1041AT	4,75...5,25	-27...+40	1000	10		SOIC-14
TJA1041T/N1	4,75...5,25	-27...+40	1000	10		SOIC-14
TJA1050T/N1	4,75...5,25	-27...+40	1000	–		SOIC-8
TJA1050U/N1	4,75...5,25	-27...+40	1000	–		SOIC-8
TJA1054T/N3	4,75...5,25	-40...+40	125	10		SOIC-14
TJA1054AT/N4	4,75...5,25	27...+40	125	10		SOIC-14
TJA1055T/N3	4,75...5,25	-58...+58	125	5		SOIC-14

NXP выпускает также специализированные решения, которые могут работать с однопроводными вариантами CAN - это AU5790. Этот трансивер был разработан для приложений с требованиями минимальной стоимости изделия. Он имеет скорость передачи данных до 40 кбит/с; скорость приема данных до 100 кбит/с; режимы энергосбережения, возможность взаимодействия между двумя устройствами без запуска всей шины, спящий режим с потреблением тока 70мкА и поддержку до 32 узлов в сети.

CAN-трансиверы Texas Instruments являются улучшенными образцами распространенных трансиверов серий PCA82C25x, TJA10xx от PHILIPS [33]. Их параметры сведены в таблице 8.

Таблица 8 - Параметры микросхем CAN-трансиверов Texas Instruments

Наименование	Диапазон напряжения питания, В	Максимально допустимое напряжение на шине CAN, В	Максимальная скорость передачи, кбит/с	Ток потребления в экономичном режиме, мкА	Диапазон рабочей температуры, °С	Корпус
SN65HVD230	3,0...3,6	-4...+16	1000	370	-40...85	SOIC-8
SN65HVD231				100		
SN65HVD232				–		
SN65HVD230M-EP	3,0...3,6	-4...+16	1000	370	-50...125	SOIC-8
SN65HVD230Q	3,0...3,6	-4...+16	1000	370	-40...125	SOIC-8
SN65HVD231Q				100		
SN65HVD232Q				–		
SN65HVD1040	4,75...5,25	-27...+40	1000	5	-40...125	SOIC-8
SN65HVD1050				–	-40...125	
SN65HVD1050-EP				–	-55...125	
SN65LBC031	4,50...5,50	-5...+20	500	–	-40...85	SOIC-8 DIP-8
SN65LBC031Q					-40...125	
SN75LBC031						

Приемопередатчики SN65HD23x/SN65HD23xQ полностью совместимы по корпусу с изделием NXP – PCA82C250 при этом обладают усовершенствованными характеристиками: увеличенной экономичностью, напряжением питания 3,3 В, расширенным диапазоном рабочей температуры, спящим режим со сверхнизким током потребления 0,1 мкА (SN65HD231 /SN65HD231Q).

SN65HVD1040/1050/1050-EP являются аналогами трансиверов NXP – TJA1040/1050, но с улучшенными параметрами: защита до 12 кВ по выводам шины CAN относительно земли (по сравнению с 4 кВ в TJA1050); уменьшенный с 30 мкА (TJA1050) до 5 мкА ток потребления в ждущем режиме; автоматическое пробуждение из ждущего режима по факту приема данных по шине; расширенный диапазон рабочей температуры в отрицательной области (-55°С у SN65HVD1050-EP против -40°С у TJA1040/1050).

«CAN-трансиверы Maxim (табл. 9) отличаются от аналогичных микросхем других компаний более высокими скоростями работы – до 2

Мбит/с; повышенным максимально допустимым напряжением на шине (до +/-80 В); повышенной экономичностью, т.е. сверхнизким собственным током потребления (35 мкА при доминантном уровне на выходе шины и 2 мкА при рецессивном); дополнительной защитой по току выходных транзисторов. Все микросхемы CAN-трансиверов совместимы с изделиями конкурентов как по выводам, так и по корпусам, в которых она размещены.» [39] Основная область их применения – промышленная в системах автоматических линий и роботах.

Таблица 9 - Параметры микросхем CAN-трансиверов Maxim

Наименование	Диапазон напряжения питания, В	Максимально допустимое напряжение на шине CAN, В	Максимальная скорость передачи, кбит/с	Ток потребления в экономичном режиме, мкА	Диапазон рабочей температуры, °С	Корпус
MAX3050	4,50...5,50	-80...+80	2000	15	-40...125	SOIC-8
MAX3057						
MAX3051	3,10...3,45	-7,5...+12,5	1000	1	-40...85	SOIC-8 SOT23-8
MAX3053	4,50...5,50	-80...+80	2000	15	-40...85 (E) -40...125 (A)	SOIC-8
MAX3054	4,75...5,25	-80...+80	250	3	-40...125	SOIC-14
MAX3055						
MAX3056						
MAX3058	4,50...5,50	-7,5...+12,5	1000	15	-40...125	SOIC-8
MAX3059				10		
MAX13050	4,75...5,25	-80...+80	1000	11	-40...85 (E) -40...125 (A)	SOIC-8
MAX13052						
MAX13053						
MAX13054	4,75...5,25 и 3,00...3,60					
MAX13051	4,75...5,25	-80...+80	1000	15	-40...85 (E) -40...125 (A)	SOIC-8

MAX3054/55/56 – это улучшенные версии трансиверов TJA1054 (NXP): они имеют высокое допустимое напряжение на шине CAN (до +/-80 В), из-за чего заметно увеличивается надежность; встроенные фильтры позволяют снизить уровень высокочастотных помех; встроенный узел управления скоростью нарастания напряжения на шине CAN увеличивает устойчивость к

внешним электромагнитным полям. Они максимально подходят для работы на скоростях 40 кБит/с (МАХ3056), 125 кБит/с (МАХ3055) и 250 кБит/с (МАХ3054).

3. Контроллеры CAN-bus

3.1 Контроллеры CAN-bus в структуре микроконтроллеров

Мы рассмотрели трансиверы CAN-шины, которые занимаются преобразованием TTL-уровней входных логических сигналов в коммутационные воздействия на шину CAN, а также реализуют некоторые меры безопасности и снижения ошибок.

Однако, трансиверы не решают задачи контроля потока кадров в кан-шине. Этим занимаются другие устройства, называемые контроллерами CAN-шины.

Такие контроллеры могут быть выполнены на основе микроконтроллеров различной архитектуры, в состав которых входят дополнительные модули для реализации функций работы с кадрами различных типов, описанных стандартами CAN-шины. Промышленностью выпускаются контроллеры CAN-шины с различными ядрами: STM выпускает микроконтроллеры серии STM32 с ядрами Cortex различных серий, Philips Semiconductors выпускает микроконтроллеры для CAN-шины на ядрах 80C51.

Фирма Microchip выпускает продукты с CAN-модулями самых разных типов: 8-ми разрядные микроконтроллеры PIC и AVR, 16 разрядные PIC24 микроконтроллеры и dsPIC33 цифровые сигнальные процессоры, 32 разрядные PIC32 и микроконтроллеры SAM.

В качестве обзорного примера кратко рассмотрим микроконтроллер Microchip AT90CAN32-16AU [4].

Этот микроконтроллер в дополнение к ядру AVR содержит в своем составе модуль контроллера CAN-шины. С позиции программиста он является обычным AVR микроконтроллером, похожим на ATMEGA32, только с дополнительными инструкциями, управляющими дополнительным блоком.

Блок CAN в этом микроконтроллере имеет выходы TXCAN и RXCAN, подключаемые к трансиверу.

Рассмотрим основные параметры и функции AVR ядра микроконтроллера.

Микроконтроллер AT90CAN32-16AU имеет расширенную RISC архитектуру, включающую в себя 133 инструкции, 32 8-ми разрядных регистра, производительность до 16MIPS при частоте 16МГц.

Память 32Кбайт перепрограммируемый флэш, загрузочная область переменным объемом от 1 до 8 Кбайт, возможность перепрограммирования через интерфейсы CAN или UART с использованием загрузчиков в загрузочной области;

- 1 К EEPROM;
- 2К SRAM;
- интерфейс для программирования и отладки JTAG;
- два 8-ми разрядных таймера-счетчика;
- один двойной 16-ти разрядный таймер счетчик;
- 8-ми канальный 10-ти разрядный АЦП;
- аналоговый компаратор;
- двухпроводной последовательный интерфейс SPI;
- двойной программируемый последовательный интерфейс UART;
- имеет 5 режимов сна;
- 53 программируемых линий ввода-вывода;
- 64 контакта;
- рабочее напряжение 2.7 – 5.5В;
- максимальная рабочая частота 8 МГц при питании 2.7В, 16 МГц при напряжении 4.5В.

Корпус TQFP или QFN.

Кроме ядра AVR в состав рассматриваемого микроконтроллера входит также контроллер CAN Controller 2.0A & 2.0B - ISO 16845.

Контроллер CAN обладает следующими характеристиками:

- предоставляет функции, необходимые для реализации ядра протокола шины CAN в соответствии с эталонной моделью ISO/OSI:

- канальный уровень данных;
- подуровень Logical Link Control (LLC);
- подуровень управления доступом к среде (MAC);
- физический уровень;
- подуровень физической сигнализации (PLS);
- не поддерживается - подключение к физическому носителю (PMA);
- не поддерживается - зависимый от среды интерфейс (MDI).

Контроллер CAN может обрабатывать все типы кадров (данные, удаленные, ошибки и перегрузка) и достигает скорости передачи данных 1 Мбит/с.

19.1 Особенности

- контроллер полной банки;
- полностью соответствует стандарту CAN версии 2.0 А и версии 2.0 В;
- 15 MOB (объект сообщений) со своими собственными:
 - 11 бит тега идентификатора (версия 2.0 А), 29 бит тега идентификатора (версия 2.0 В);
 - 11 бит маски идентификатора (версия 2.0 А), 29 бит маски идентификатора (версия 2.0 В);
 - 8-байтовый буфер данных (статическое размещение);
 - Tx, Rx, буфер кадров или конфигурация автоматического ответа;
- отметка времени;
- максимальная скорость передачи 1 Мбит/с на частоте 8 МГц;
- таймер TTC;
- режим прослушивания.

Блок CAN в этом микроконтроллере имеет структуру, показанную на рисунке 3.1.

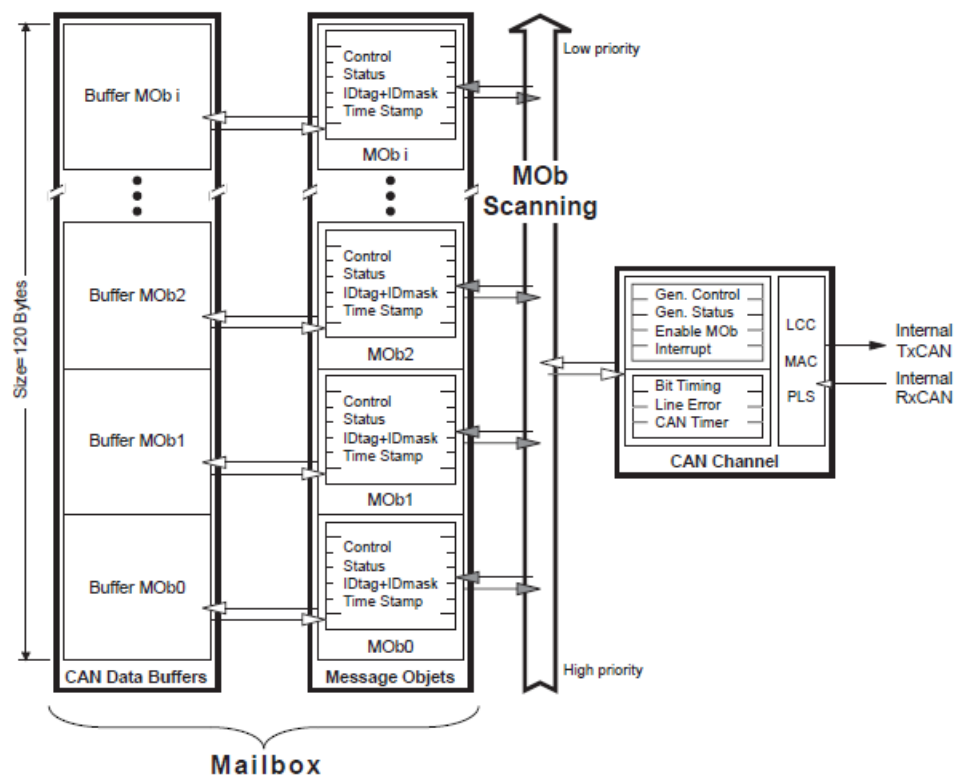


Рисунок 22 – Структура модуля CAN микроконтроллера AT90CAN32-16AU

Модуль CAN имеет аппаратную фильтрацию приема сообщений и широкое управление сообщениями. При инициализации периферийного устройства модуль определяет, какие сообщения должны быть переданы и получены. Принимаемые сообщения проверяются на совпадение идентификатора сообщения. Контроллер сохраняет в буферной памяти только те сообщения, идентификаторы которых совпадают с наперед запрограммированными и вырабатывает прерывание, которое уведомляет исполняемый программный код.

Еще одним достоинством встроенного CAN контроллера является отправка в ответ на входящий Remote frame автоматического ответа соответствующими для это данными.

Встроенный контроллер позволяет удерживать на шине высокие скорости обмена данными и позволяет обрабатывать сообщения на шине с высокой степенью интенсивности.

CAN-контроллер может находиться в следующих режимах [4] :

- Включенный режим.

В этом режиме:

– канал CAN (внутренний TxCAN и RxCAN) включен,

– входные часы включены.

- Режим ожидания.

В режиме ожидания:

– передатчик постоянно выдает рецессивный уровень (по внутреннему TxCAN) и приемник отключен,

– входные часы включены,

– регистры и страницы остаются доступными.

- Режим прослушивания

Этот режим прозрачен для CAN-канала:

– включает аппаратную петлю, внутренний TxCAN на внутреннем RxCAN;

– обеспечивает рецессивный уровень на выходном контакте TXCAN

- не отключает входной контакт RXCAN;

- замораживает счетчики ошибок TEC и REC.

В режиме прослушивания аппаратная структура цепей выглядит, как показано на рисунке 23.

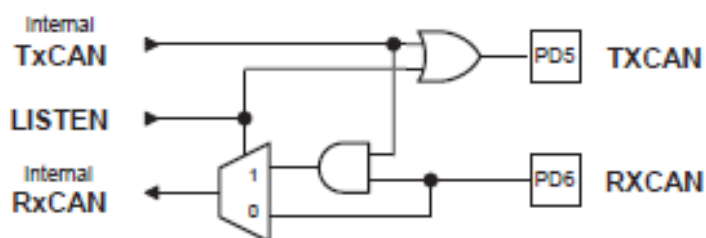


Рисунок 23 - Аппаратная структура цепей в режиме прослушивания

Встроенный CAN-контроллер обеспечивает арбитраж сообщений на шине в соответствии с концепцией, называемой «множественный контроль несущей» и описанной в первом разделе работы.

Иллюстрация процедуры арбитража в микроконтроллере иллюстрируется рисунком 24.

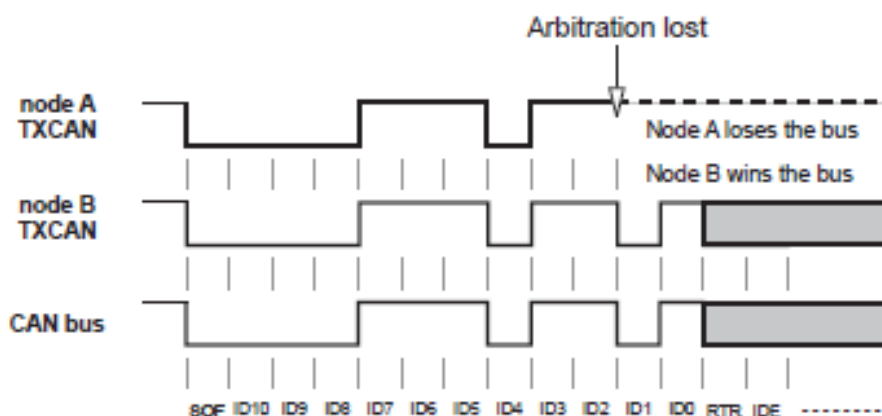


Рисунок 24 – Иллюстрация процедуры арбитража сообщений на CAN шине в микроконтроллере

Контроллер на аппаратном уровне поддерживает декларированный в стандарте механизм контроля ошибок.

Канал CAN может находиться в одном из трех следующих состояний:

- Error active (начальное состояние по умолчанию):

Канал CAN принимает участие в обмене данными по шине и может отправлять активный кадр ошибки Error frame, когда макрос CAN обнаруживает ошибку.

- Error passive:

Канал CAN не может отправить активный кадр ошибки. Принимает участие в шинном общении, но при обнаружении ошибки отправляется

пассивный кадр ошибки Error frame. Так же после передачи кадра пассивной ошибки будет ждать, прежде чем начать дальнейшую передачу.

- Bus off:

Каналу CAN не разрешается оказывать какое-либо влияние на шину.

Для локализации сбоев используются счетчик ошибок передачи (TEC) и счетчик ошибок приема (REC). Биты BOFF и ERRP предоставляют информацию о состоянии канала CAN. Установка BOFF в уровень логической единицы генерирует прерывание.

Граф перемещения между статусами ошибок показан на рисунке 25.

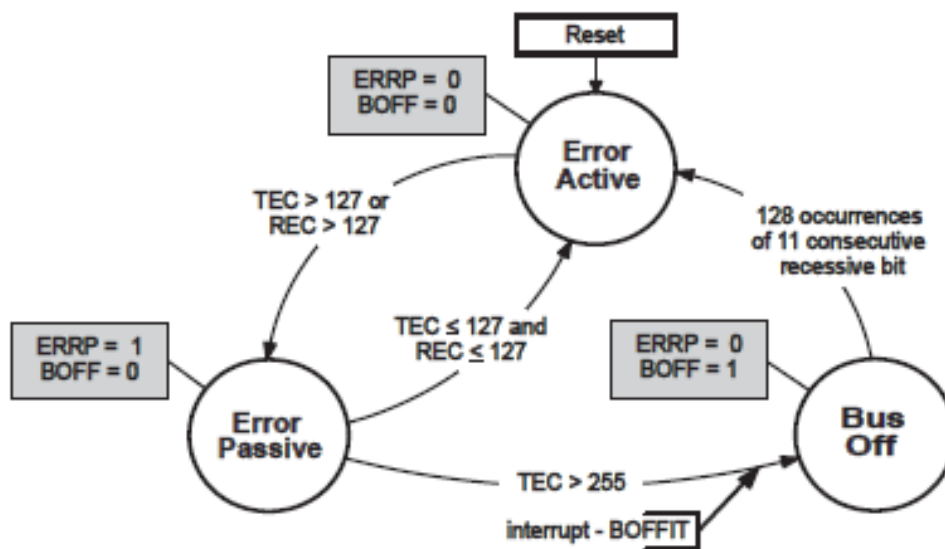


Рисунок 25 - Граф перемещения между статусами ошибок

CAN-контроллер идентифицирует следующие ошибки:

- BERR: битовая ошибка. Отслеживаемое значение бита отличается от отправленного значения бита. Исключения:

- отправляемый рецессивный бит отслеживается как доминантный во время поля арбитража и слота подтверждения.

- обнаружение доминирующего бита при отправке кадра ошибки.

- SERR: материальная ошибка. Обнаружение более пяти последовательных битов с одинаковой полярностью.

- CERR: ошибка CRC (только Rx). Получатель выполняет проверку CRC для каждого распакованного полученного сообщения от начала кадра до поля данных. Если эта проверка не совпадает с удаленным полем CRC, устанавливается ошибка CRC.

- FERR: Ошибка формы. Ошибка формы возникает в результате одного (или нескольких) нарушений фиксированной формы следующих битовых полей:

- CRC delimiter (разделитель CRC),
- acknowledgement delimiter (разделитель подтверждения),
- end-of-frame (конец кадра)
- error delimiter (разделитель ошибок),
- overload delimiter (разделитель перегрузки).

- AERR: Ошибка подтверждения (только Tx). Нет обнаружения доминантного бита в слоте подтверждения.

Процедура обнаружения ошибки в кадре данных DATA FRAME показана на рисунке 3.5.

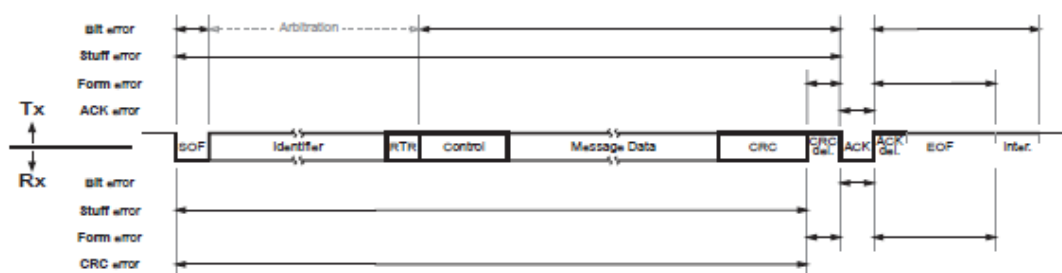


Рисунок 26 - Процедура обнаружения ошибки в кадре данных DATA FRAME

Настройка ошибок в контроллере CAN Канал CAN может обнаруживать некоторые ошибки в сети CAN.

- В передаче:

Ошибка устанавливается на уровне MOB.

- В приемной:

- Идентифицированное совпало:

Ошибка устанавливается на уровне MOB.

- Идентифицированное не совпало или еще не совпало:

Ошибка устанавливается на общем уровне.

После обнаружения ошибки канал CAN отправляет кадр ошибки в сеть. Если канал CAN обнаруживает кадр ошибки в сети, он отправляет свой собственный кадр ошибки.

Рассмотрим, как организован механизм прерываний, связанный с событиями, относящимися к CAN.

Различают прерывания:

- прерывание при «Прием завершен ОК»,

- прерывание при «Передача завершена ОК»,

- прерывание при ошибке (битовая ошибка, ошибка заполнения, ошибка CRC, ошибка формы, ошибка подтверждения),

- прерывание при заполнении кадрового буфера,

- прерывание при настройке «Bus Off»,

- прерывание по превышению таймера CAN.

Разрешение общего прерывания обеспечивается битом ENIT, а конкретное разрешение прерывания для переполнения таймера CAN обеспечивается битом ENORVT.

Поведение при возникновении прерываний следующее.

Когда происходит прерывание, бит флага прерывания устанавливается в соответствующем регистре MOB-CANSTMOB или в общем регистре CANGIT. Если в регистре CANIE установлен бит ENRX/ENTX/ENERR, то в регистре CANSITn устанавливается соответствующий бит MOB.

Для подтверждения прерывания MOB соответствующие биты регистра CANSETMOB (RXOK, TXOK,...) должны быть очищены программным приложением. Для этой операции требуется программная процедура чтения-изменения-записи. Для подтверждения общего прерывания соответствующие биты регистра CANGIT (BXOK, BOFFIT,...) должны быть очищены программным приложением. Эта операция выполняется путем записи логической единицы в эти флаги прерывания (запись логического нуля не изменяет значение флага прерывания).

Флаг прерывания OVRTIM сбрасывается, как и другие источники прерываний регистра CANGIT, а также сбрасывается при входе в его специальный обработчик прерываний.

Когда узел CAN находится в режиме передачи и обнаруживает ошибку формы в своем фрейме, также возникает битовая ошибка. Следовательно, могут возникнуть два последовательных прерывания, оба из-за одной и той же ошибки.

Когда ошибка MOB возникает и устанавливается в собственном регистре CANSTMOB, в регистре CANGIT не устанавливается общая ошибка.

На рисунке 27 показана структура прерываний, генерируемых контроллером CAN в микроконтроллер.

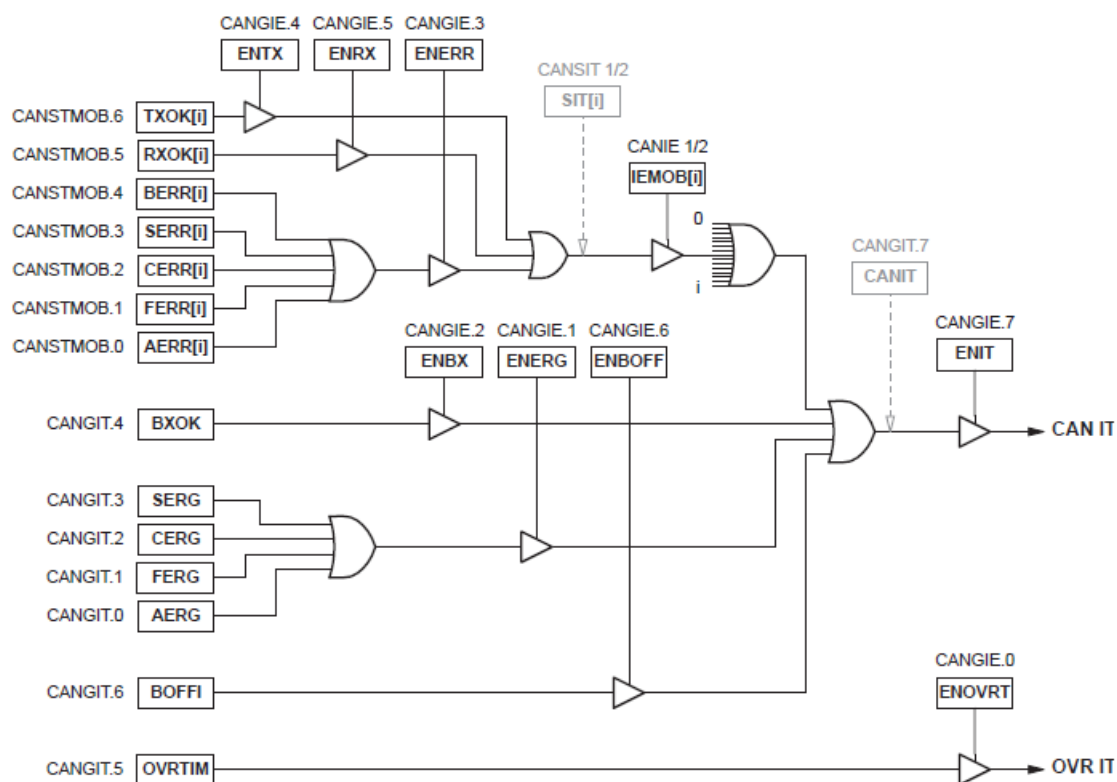


Рисунок 27 - Структура прерываний, генерируемых контроллером CAN

Таким образом, контроллер CAN, встроенный в микроконтроллер AT90CAN32-16AU, реализует на аппаратном уровне функции, необходимые для реализации функционирования узла CAN-шины, которым управляет этот контроллер, в соответствии с требованиями стандартов серии ISO 11898. Аналогичные контроллеры CAN-шины встраиваются и в микроконтроллеры других серий производства фирмы Microchip с ядрами AVR и PIC.

Это позволяет организовывать реализацию на этих микроконтроллерах узлов сети CAN различного назначения: контроллеры для сбора данных и управления, исполнительные устройства, различного типа датчики и устройства ввода информации.

Обобщая, можно говорить о том, что первым вариантом реализации узлов различного назначения в CAN-шине будет построение электронных устройств с использованием микроконтроллеров со встроенным контроллером CAN шины. Такие микроконтроллеры выпускаются

различными производителями с различными ядрами: STM32 от STM, AVR и PIC от Microchip, 80C51 от Philips и NXP.

3.2 ПЛК с CAN-интерфейсами

Другим вариантом организации CAN-сети, по сравнению с микроконтроллерным, является использование в качестве управляющих узлов программируемых логических контроллеров (ПЛК). ПЛК в качестве основных устройств управления, обработки информации используются в промышленных системах, хотя их использование на транспорте и в автономных транспортных средствах набирает всё большие обороты.

Работа сети на основе ПЛК с CAN-интерфейсом принципиально не отличается от сетей на основе микроконтроллеров с транзиттерами. С физической стороны ПЛК или уже содержат в себе CAN-контроллер и трансивер, в виде микросхем распаянные на плате ПЛК, или имеют в номенклатуре модулей расширения интерфейсные модули для CAN-шины.

Программные средства работы с CAN-шиной, как правило, встроены в системы программирования ПЛК и основаны на протоколе CANopen.

В настоящее время на рынке РФ представлены ПЛК с подсистемами CAN-шины марок GCAN (ПЛК GCAN-PLC-400/510/511 [14] и модули ввода/вывода GCAN-IO-8000), Delta (модель DVP32ES200TC [27]), segnetics (модель Matrix-1021-90-0) [28] и некоторые другие [6].

4. Структура оборудования автономного транспортного средства

Автономные транспортные средства – движущиеся объекты, не управляемые человеком непосредственно. Они, как правило, используются в настоящее время на производственных площадках, выполняя те или иные логистические функции: перевозку сырья и заготовок из складских помещений в производственную зону, перемещают изделия между промежуточными точками обработки, осуществляют перевозки тары между корпусами, обслуживают складские помещения.

Традиционно автономное транспортное средство с позиции электроники и электромеханики состоит из следующих узлов [21, 30, 18, 36]:

- система питания - аккумуляторная батарея с контроллером состояния элементарных ячеек батареи и контроллером заряда;
- система управления, состоящая из контроллера в виде типового ПЛК или проприетарной микроконтроллерной системы;
- электродвигателей, приводящих во вращение колеса;
- рулевого сервопривода, если кинематическая схема транспортного средства использует выделенный поворотно-рулевой механизм;
- драйверов электродвигателей, формирующих необходимые кривые питающего напряжения или тока для них;
- навигационной системы, которая может состоять из видеокамеры с подсистемой машинного зрения, лидара (лазерного дальномера), датчиков линии (магнитных или электронно-оптических);
- набора ультразвуковых датчиков или датчиков касания, если периметр безопасности транспортного средства подразумевает их наличие;

- интерфейса пользователя HMI (Human-Machine Interface), включающего в себя экран (символьный или растровый), необходимые клавиши/кнопки и, опционально, другие манипуляторы.

Зачастую, на практике, все эти элементы системы объединяются между собой различными интерфейсами: RS485, UART/USART, 1wire и т.п.

Рассмотрим вариант объединения этих узлов автономного транспортного средства с целью упрощения информационной кабельной проводки, посредством шины CAN. Структура электрической подсистемы в этом случае будет выглядеть, как показано на рисунке 28.

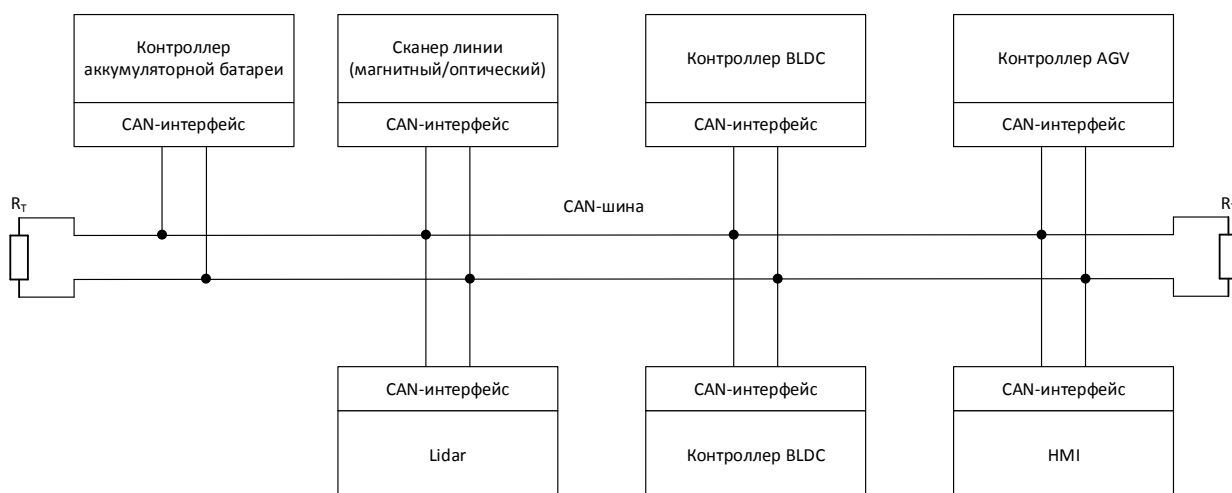


Рисунок 28 – Структурная схема электрического оборудования автономного транспортного средства.

На схеме показаны основные узлы транспортного средства, а также терминаторные резисторы на шине CAN.

Изображенная схема показывает обобщенный случай контроллера и системы HMI без детализации относительно используемой архитектуры: ПЛК или на основе микроконтроллера.

5. Разработка системы оборудования на основе шины CAN.

Спроектируем первый вариант построения CAN-системы для автономного транспортного средства с центральным управляющим устройством на основе микроконтроллера и оборудованием с произвольными интерфейсами, для которых выберем имеющиеся в продаже или спроектируем собственные мосты в шину CAN.

Структура системы управления на основе микроконтроллера будет включать в себя собственно сам микроконтроллер, контроллер шины CAN, а также трансивер выходных управляющих сигналов контроллера CAN в собственно сигналы шины CAN. Кроме этого в схему системы управления по традиционной архитектуре микроконтроллерных систем управления включим жидкокристаллический экран модуль из кнопок с поворотной кнопочным энкодером и джойстиком для удобства управления транспортным средством оператору.

В качестве центрального микроконтроллера выберем микросхему ATmega64M1 производства Microchip [5].

В качестве устройства отображения системы HMI выберем дисплей с интерфейсом SPI, который имеется в выбранном в качестве центрального устройства микроконтроллере. Интерфейс SPI обеспечивает скорость передачи данных до 20 Мбит/сек, что позволяет формировать силами микроконтроллера не только статическую графическую картинку на жидкокристаллических дисплеях, но и вводить в нее незначительные элементы динамики.

Подключение дисплея по этому интерфейсу осуществляется следующим образом.

В состав интерфейса SPI микроконтроллера входят линии:

- MOSI (англ. Master Out Slave In)— выход ведущего, вход ведомого;
- MISO (англ. Master In Slave Out)— вход ведущего, выход ведомого;

- SCK (англ. Serial Clock) — последовательный тактовый сигнал для ведомых устройств;

- SS (англ. Slave Select) — выбор ведомого.

Для отображения минимально необходимой информации выбираем дисплей TFT 2,2 дюйма с интерфейсом SPI на контроллере ILI9341. Жидкокристаллические дисплеи, построенные на контроллере ILI9341 имеют выводы [36], сведенные в таблицу:

Таблица 10 - Назначение выводов дисплеев с интерфейсом SPI.

Обозначение	Описание функции
VCC	3.3~5В источник напряжения
GND	Общий
MOSI	Входные данные
MISO	Выходные данные
SCK	Тактовые импульсы
CS	Выбор микросхемы
D/C	Вывод выбора данных/команд. Высокий уровень (подключен к напряжению питания) - входной байт воспринимается, как данные, когда приходит низкий уровень – входной байт воспринимается, как команда и передается в регистр команд.
RES	Сигнал сброса. Активный уровень – низкий.

Этот дисплей имеет следующие характеристики [25]:

- Контроллер: ILI9341;

- Разрешение: 240x320;

- Напряжение питания дисплея VCC: 3.3 - 5 В, на плате присутствует понижающий DC-DC преобразователь на 3.3 В;

- Напряжение логики SPI: 3.3 В;

- Напряжение питания подсветки LED: 3.3 В;

Внешний вид экрана показан на рисунке 29.



Рисунок 29 – Внешний вид экрана для модуля управления

Таким образом, линии от микроконтроллера к интерфейсу ЖК дисплея подключаются следующим образом (таблица 11):

Таблица 11 - Подключение дисплея

Выводы дисплея	Выводы микроконтроллера, сигналы в системе
VCC	VCC (+5В)
GND	GND
MOSI	MOSI (PB1)
MISO	MISO (PB0)
CLK	SCK (PB7)
CS	SS (PD3)
D/C	PD2
RES	PC1

Для тактирования микроконтроллера будем использовать вариант подключения конденсаторов и кварцевого резонатора, показанный на рисунке 30.

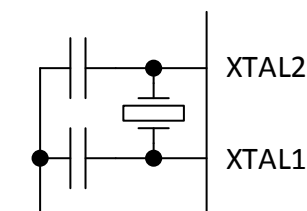


Рисунок 30 – Схема подключения элементов тактирования

Выбираем емкость конденсаторов в соответствии с рекомендациями производителя [5] 22 пкФ, а частоту кварцевого резонатора -16 МГц.

В связи с тем, что питание дисплея осуществляется от напряжения 3.3В, то и питание микроконтроллера будем проводить от напряжения питания 3.3В.

Для управления оператором объектами на экране в рамках реализации НМІ будем использовать энкодер с вращающейся рукояткой, имеющей функцию кнопки.

В качестве энкодера выберем модель КУ-40 [24], внешний вид которой показан на рисунке 31.

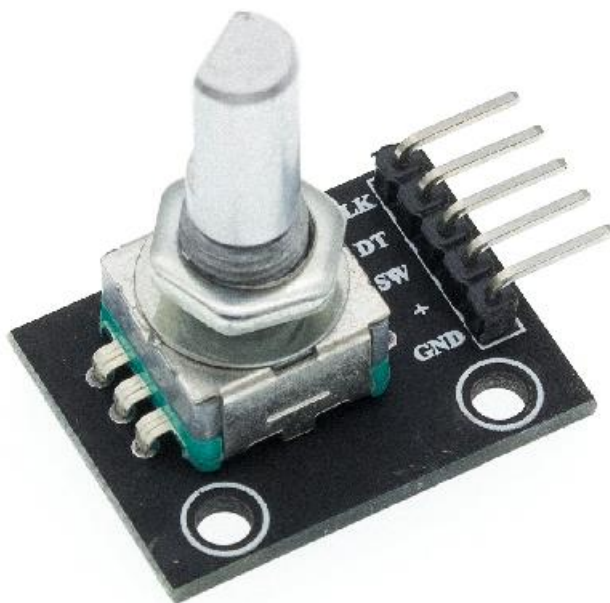


Рисунок 31 – Внешний вид энкодера KY-40

Энкодер имеет выводы:

CLK – тактовый вывод;

DT – вывод данных;

SW – вывод кнопки энкодера;

+ - напряжение питания;

GND - общий провод.

Импульсы на выводах DT и CLK появляются при вращении рукоятки энкодера. Причем, в зависимости от направления вращения меняется последовательность появления импульсов на этих выводах.

Состояние вывода SW меняется при нажатии кнопки. Необходимо включать в микроконтроллере выходные подтягивающие резисторы, если не предполагается использовать внешнюю обвязку. Схема устройства энкодера показана на рисунке 32.

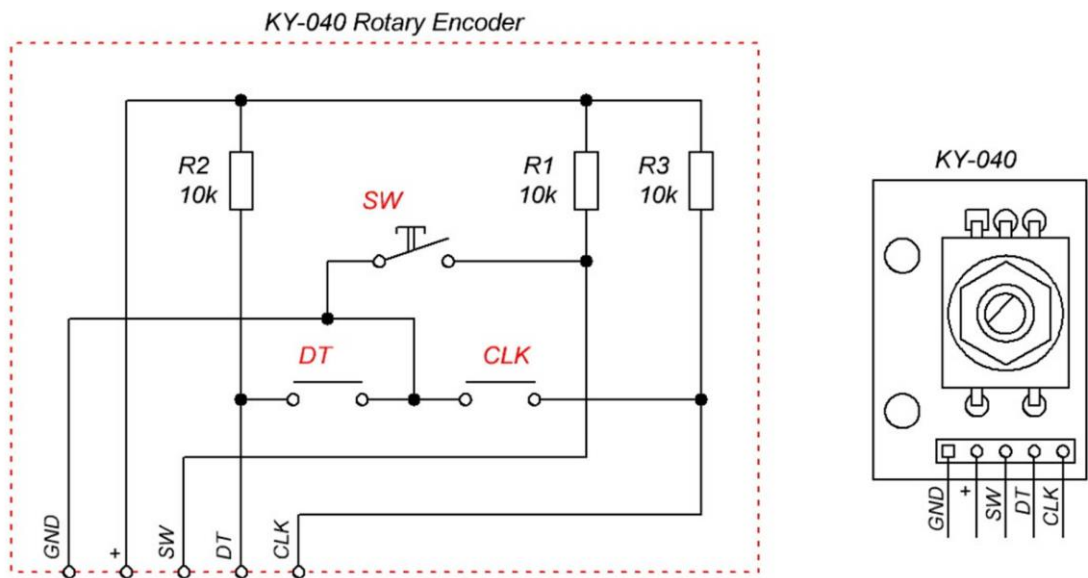


Рисунок 32 – Устройство энкодера KY-40

Для возможности расширить взаимодействие оператора с системой управления автономным транспортным средством в рамках реализации человеко-машинного интерфейса предусмотрели три кнопки с внешними подтягивающими резисторами, которые будут подключать так, как показано на рисунке 33.

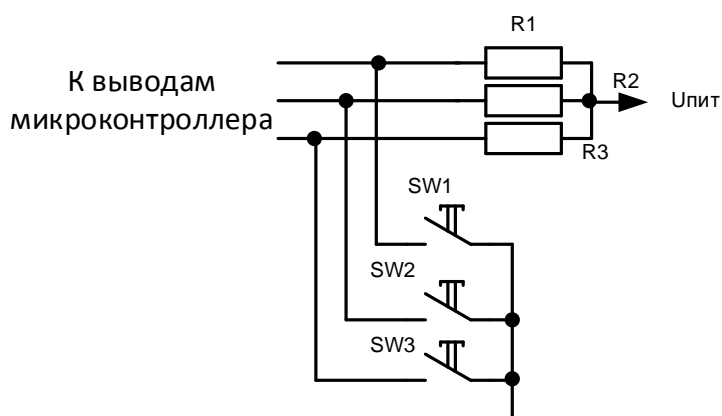


Рисунок 33 – Подключение кнопок управления к микроконтроллеру

Сопровитления резисторов примем равным 1 кОм с мощностью рассеивания 0,125 Вт.

В качестве трансивера для связи с CAN-шиной будем использовать микросхему MCP2551 производства Microchip [19].

«Его характеристики:

- Поддерживает работу со скоростью 1 Мбит/с
- Реализует требования стандарта ISO-11898 к физическому уровню.
- Подходит для систем 12 В и 24 В.
- Управляемый извне наклон для снижения радиопомех выбросы
- Обнаружение замыкания на землю (постоянная доминанта) на входе TXD
- Сброс при включении питания и защита от понижения напряжения
- Обесточенный узел или падение напряжения не будут нарушать работу CAN-шины.
- Работа в режиме ожидания с низким током
- Защита от повреждения из-за короткого замыкания, попадания положительного или отрицательного напряжения батареи на линии шины
- Защита от высоковольтных переходных процессов
- Автоматическая защита от перегрева
- Можно подключить до 112 узлов
- Высокая помехоустойчивость благодаря дифференциальной шине
- Температурные диапазоны:
 - Промышленный (I): от -40°C до +85°C
 - Расширенный (E): от -40°C до +125°C» [19]

Составим таблицу соединений периферии с выводами микроконтроллера.

Внутренняя структура этой микросхемы приведена на рисунке 34.

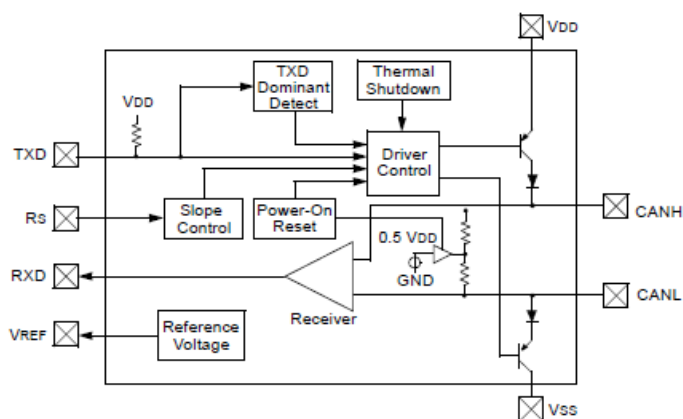


Рисунок 34 – Внутренняя структура трансивера MCP2551

Разводка выводов корпусов PDIP и SOIC показана на рисунке 35.

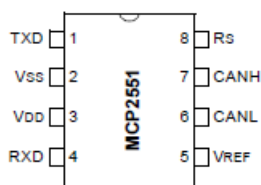


Рисунок 35 – Распределение выводов трансивера MCP2551

Подключение выводов трансивера к выводам микроконтроллера производится в соответствии с таблицей 12.

Таблица 12 - Соединение выводов MCP2551 и микроконтроллера ATmega64M1

Выводы MCP2551	Выводы ATmega64M1
TXD	PC3 RXCAN (CAN Rx Data)
RXD	PC2 TXCAN (CAN Tx Data)

Для взаимодействия системы управления автономного транспортного средства с системами верхнего уровня (системы диспетчеризации, системы сбора данных, порталы систем безопасности и т.д) введем в систему управления модуль WiFi E12 на базе микросхемы ESP8266.

Внешний вид модуля показан на рисунке 36 [12, 13].

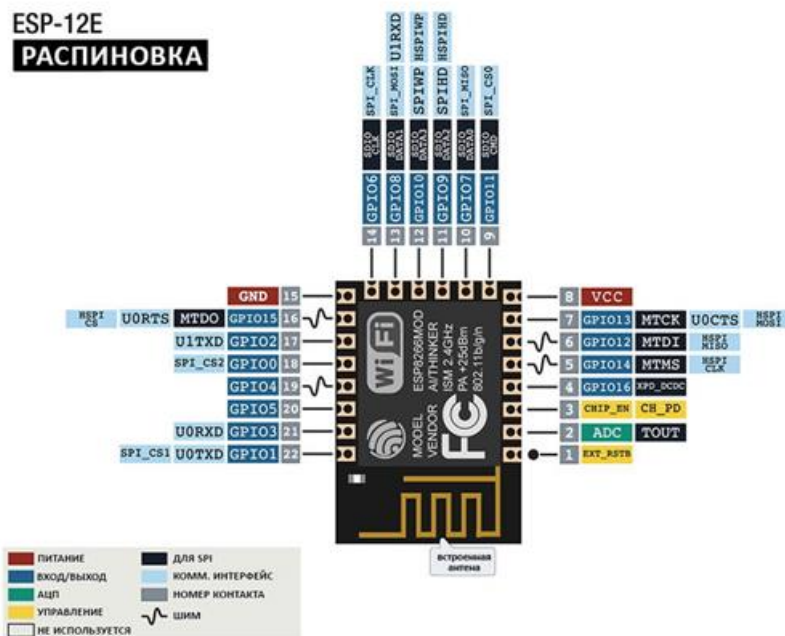


Рисунок 36 – Внешний вид модуля ESP E12

Подключение модуля к микросхеме микроконтроллера производится по аппаратному каналу UART, имеющемуся в микроконтроллере. Интерфейс UART подключен в микроконтроллере к выводу 2 PD3 (TXD (UART Tx data)) линией TXD и к выводу 12 PD4 (RXD (UART Rx data)) линией RXD. По этому интерфейсу в модуль передаются команды и данные от микроконтроллера и осуществляется остальной обмен информацией. Поскольку данный модуль не имеет в своем составе преобразователя интерфейса USB-UART, то его программирование осуществляется предварительно перед монтажом на основную печатную плату.

Схема организации соединений между выводами модуля E12 для соединения с микроконтроллером показана на рисунке 37.

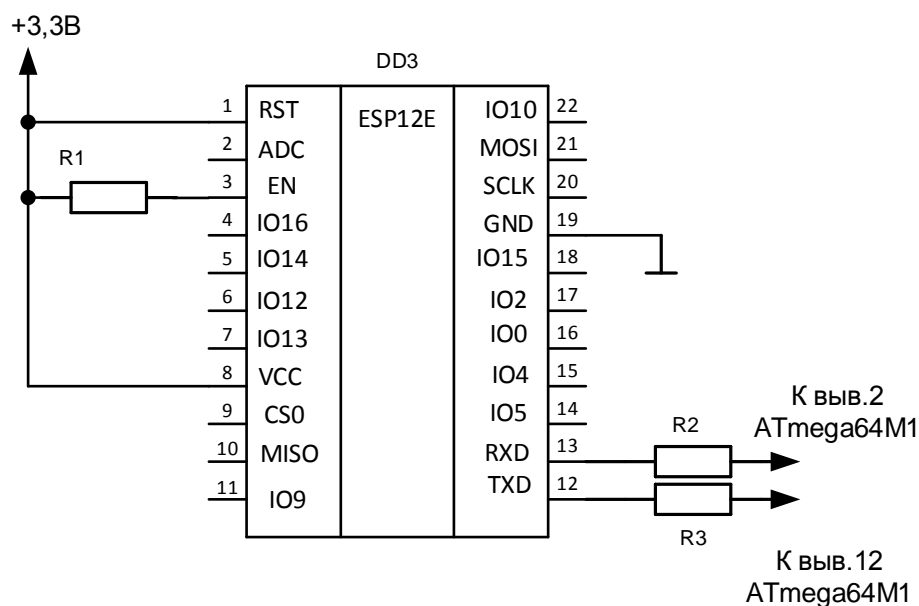


Рисунок 37 – Схема организации соединений модуля E12

Питание модуля осуществляется от источника напряжения 3.3 В, в рабочем режиме потребляется ток 80мА, при этом имеются пиковые импульсы потребления 150мА.

Для длительного хранения собираемых служебных данных, данных о маршруте движения и т.п. в системе управления и как один из вариантов хранения и обмена информации будем использовать считыватель SD-карт формата TF. Он имеет интерфейс SPI и подключается к микроконтроллеру в соответствии с таблицей 13.

Таблица 13 - Подключение считывателя SD-карт .

Выходы считывателя SD-карт	Выходы микроконтроллера, сигналы в системе
SD_CS	SS2 (PB7)
SD_MOSI	MOSI (PB1)
SD_MISO	MISO (PB0)
SD_CLK	SCK (PB7)

В качестве считывателя SD карт воспользуемся встроенным в выбранный дисплей устройством, добиваясь тем самым высокой степени интеграции устройств.

На данном этапе подобраны все периферийные устройства нашей системы управления и теперь можно сформировать таблицу соединений микроконтроллера для облегчения формирования итоговой схемы устройства и выполнения прочих работ. Описанное подключение периферийных устройств приведено в таблице 14.

Таблица 14 - Подключение периферии к выводам микроконтроллера ATmega64M1

Вывод	Обозначение	Функция	Периферия
5	GND	Общий	
20	AGND	Аналоговый общий	
4	VCC	Напряжение питания	
19	AVCC	Аналоговое напряжение питания	
21	AREF	Опорное напряжение	
8	PB0	MISO (SPI Master In Slave Out)	Подключение SPI дисплея. Подключение чипа картридера по интерфейсу SPI

Продолжение таблицы 14

Вывод	Обозначение	Функция	Периферия
9	PB1	MOSI (SPI Master Out Slave In)	
16	PB2	SW	Подключение энкодера
23	PB3		
24	PB4		
26	PB5		
27	PB6	SD_CS	Подключение чипа картридера по интерфейсу SPI
28	PB7	SCK (SPI Bus Serial Clock)	Подключение SPI дисплея. Подключение чипа картридера по интерфейсу SPI
30	PC0	SW2	Кнопка 2
3	PC1	RES	Подключение SPI дисплея
6	PC2	TXCAN (CAN Tx Data)	Подключение CAN-трансивера
7	PC3	RXCAN (CAN Rx Data)	Подключение CAN-трансивера
17	PC4		
18	PC5		
22	PC6		
25	PC7		
29	PD0	SW3	Кнопка 3
32	PD1	SW1	Кнопка 1
1	PD2	D/C	Подключение SPI дисплея
2	PD3	TXD (UART Tx data)	Подключение WiFi модуля
12	PD4	RXD (UART Rx data)	Подключение WiFi модуля
13	PD5	SS (SPI Slave Select)	Подключение SPI дисплея
14	PD6	CLK	Подключение энкодера
15	PD7	DT	Подключение энкодера

Продолжение таблицы 14

Вывод	Обозначение	Функция	Периферия
31	PE0	RESET# (Reset Input)	Внешняя схема сброса
10	PE1	XTAL1 (XTAL Input)	
11	PE2	XTAL2 (XTAL Output)	

Питание схемы обеспечивается источником питания , преобразующее бортовое напряжение с аккумуляторов транспортного средства в напряжение питания с уровне 3.3В и 5 В.

Входное напряжение – напряжение в бортовой сети может быть или 24 или 36В. Соответственно выбираем схему понижающего импульсного преобразователя постоянного напряжения.

Построим двухстадийную схему, когда сначала понизим импульсным преобразователем напряжение до 5 В, а затем линейным преобразователем – до уровня 3.3В и уже этим напряжением будем питать элементы системы управления. Подобная организации структуры преобразователя напряжения позволяет понизить на значительную величину входное напряжение ИППН с высоким КПД, а затем с помощью линейного стабилизатора добиться высококачественной кривой питающего напряжения с минимальным уровнем пульсаций.

В качестве преобразователя первой ступени возьмем микросхему стабилизатора напряжения LM2596-5 [17]. Эта микросхема преобразует входное напряжение в диапазоне от 2,4 до 40В в выходное напряжение 5В.

Типовая схема включения:

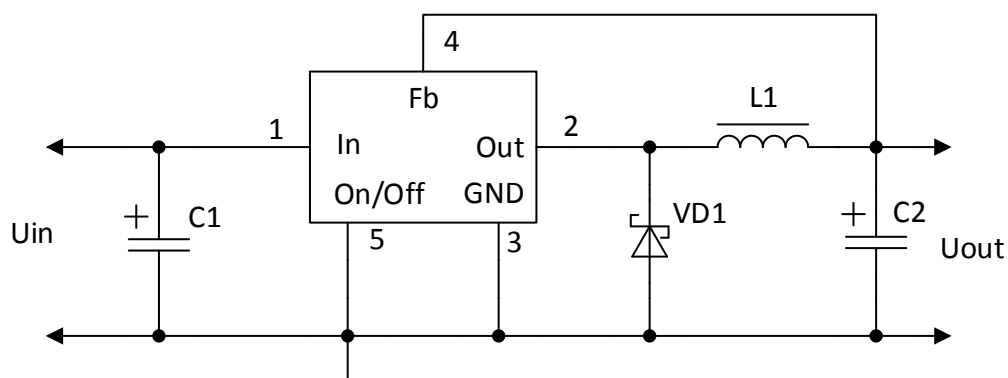


Рисунок 38 - Схема включения импульсного стабилизатора LM2596-5

На основании рекомендаций принимаем значения номиналов элементов:

$C1=400\mu\text{кФ}$, $C2=220\mu\text{кФ}$, $L1=33\mu\text{кГн}$, диод - 1N5824.

Стабилизированное напряжение 3.3В получаем с помощью стабилизатора LM1117-3,3, который подключаем к выходу первого узла на основе LM2596-5. Схема включения стабилизатора LM1117-3,3 показана на рисунке 39.

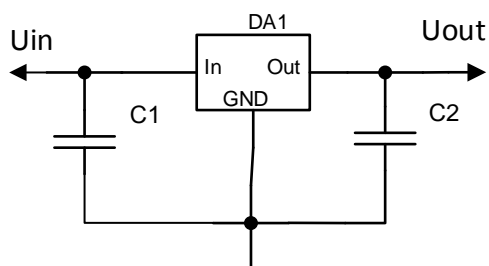


Рисунок 39 - Включение стабилизатора LM1117-3,3

Параметры конденсаторов: $C1=10\mu\text{кФ}$, а $C2=100\mu\text{кФ}$.

Результирующий двухстадийный узел питания выглядит следующим образом:

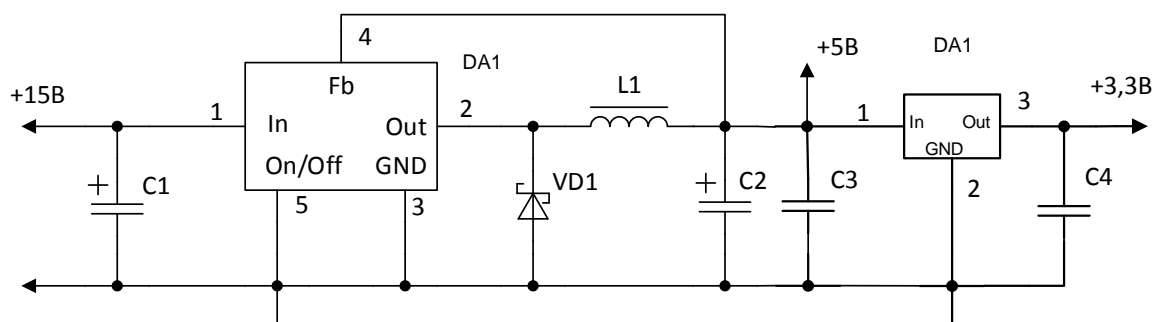


Рисунок 40 – Схема узла питания системы управления

Таким образом, мы можем сформировать результирующую схему системы основного узла шины CAN автономного транспортного средства - системы управления.

Схема системы управления показана на рисунке 41.

Рассмотрим построение схемотехники узлов шины CAN для подключения прочих устройств, рассмотренных в разделе 4.

Если устройство уже имеет в своем составе интерфейс CAN, то его интеграция в состав всей CAN-системы на аппаратном уровне производится просто подключением соответствующим образом к проводной паре CANH и CANL. Если же устройства не имеют в своем составе интерфейса CAN, но по каким-то условиям являются необходимыми для внедрения в состав системы и имеется экономическая целесообразность для этого, то можно использовать мост между интерфейсом у устройства и шиной CAN.

В этих целях можно использовать в качестве шаблона уже имеющееся схемное решение на базе микроконтроллера ATmega64M1. Составим это схемотехническое решение.

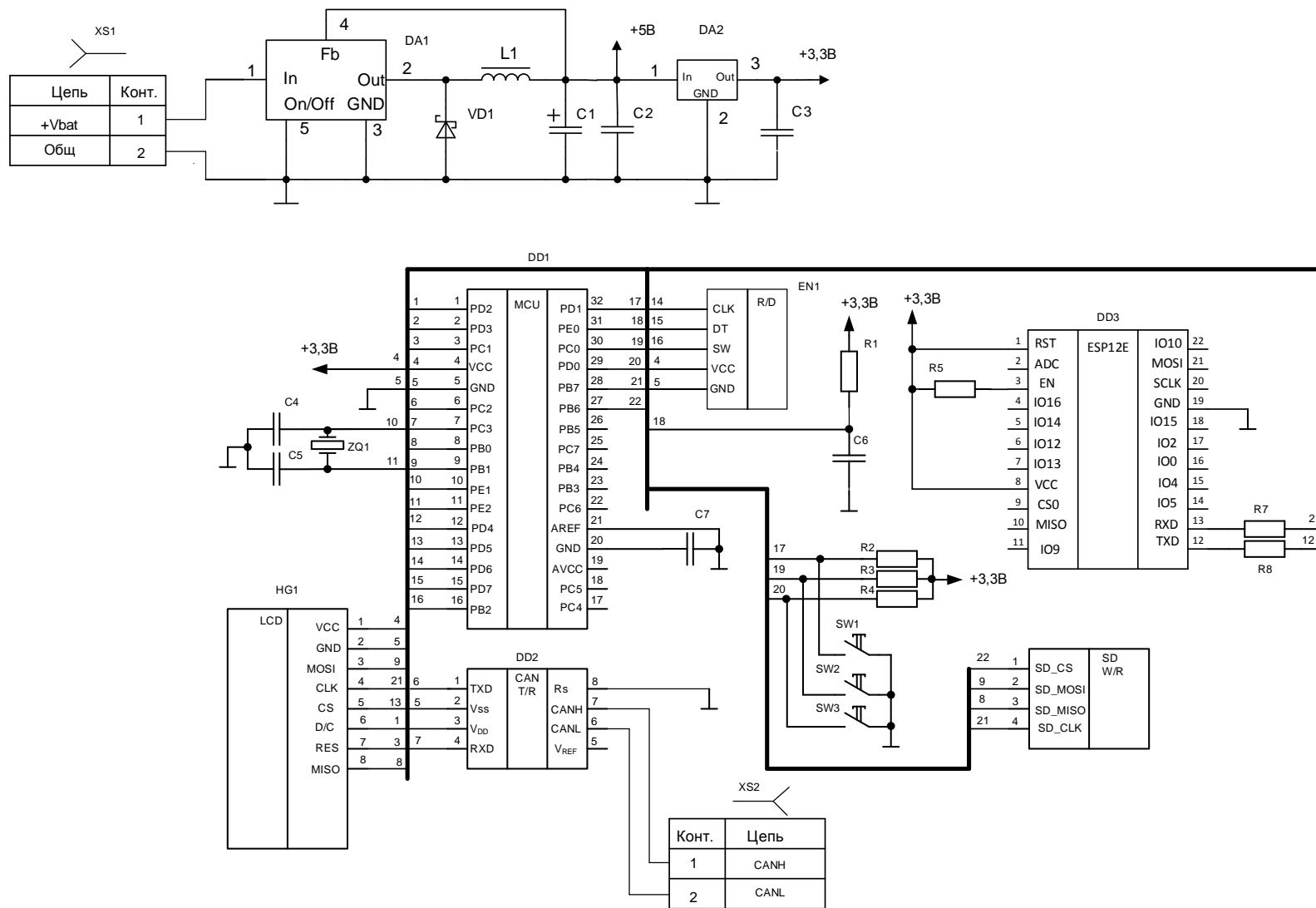


Рисунок 41 – Схема системы управления автономным транспортным средством с подключением к шине CAN

Для построения моста периферия - CAN-шина, нам необходим собственно микроконтроллер ATmega64M1, в который уже встроен контроллер CAN, а также трансивер MCP2551. Кроме этого, необходим преобразователь постоянного напряжения для питания моста от бортовой сети автономного транспортного средства. Сделаем два варианта моста – один специализированный под интерфейс UART и второй универсальный, задействующий максимальное количество интерфейсов ATmega64M1. Упрощенный вариант моста показан на рисунке 42.

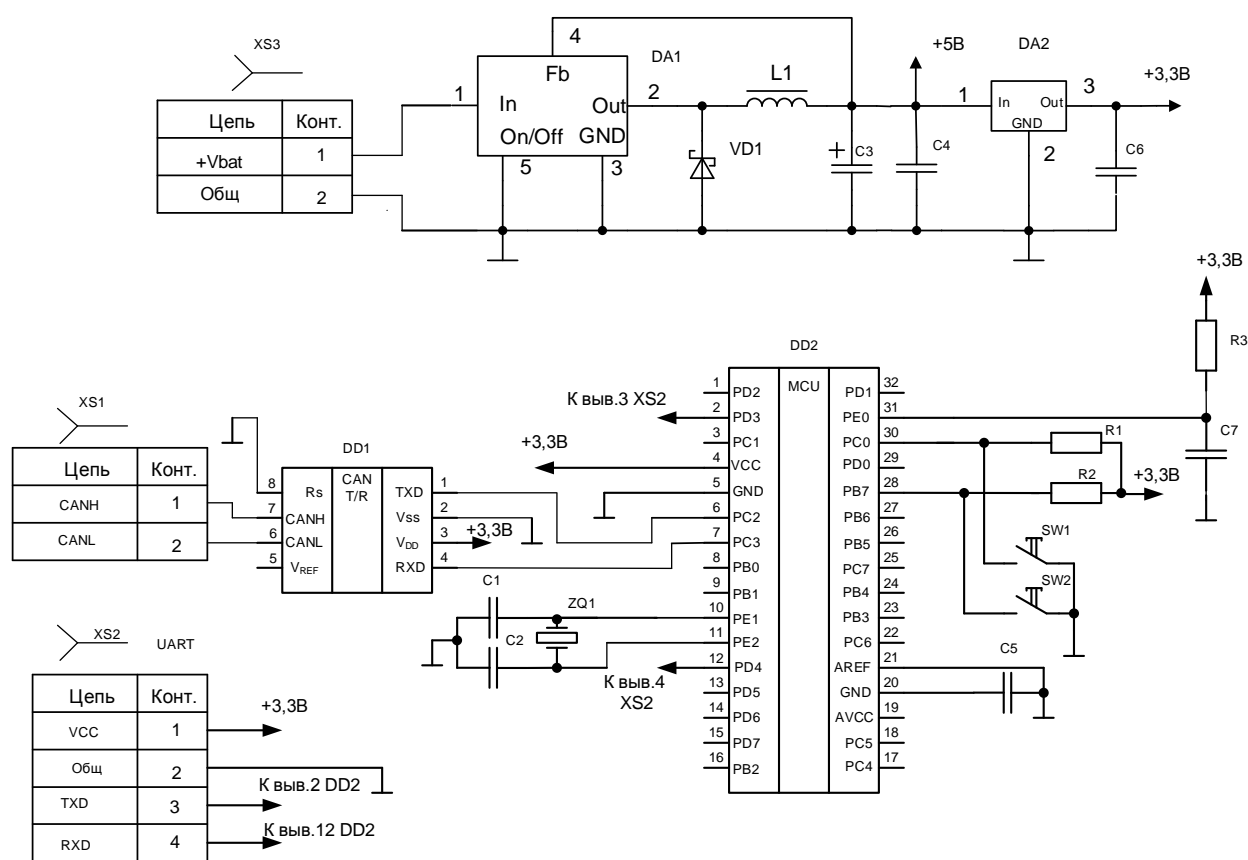


Рисунок 42 - Принципиальная схема моста CAN-шина – устройство UART

Будем считать, что внешние устройства могут подключаться к аппаратному UART микроконтроллера, к интерфейсу SPI, а также к входным интерфейсам АЦП, выходным интерфейсам ЦАП, к выходам PSC (power

stage controller), обеспечивающим работу управляемых ШИМ узлов (например, драйверы моторов) с возможностями аварийной остановки.

Тогда схема моста будет выглядеть, как показано на рисунке 43.

Таким образом, этот вариант моста содержит в своем составе следующие интерфейсы:

- два интерфейса ISP (ISP0 и ISP1);
- интерфейс UART;
- интерфейс дифференциального усилителя AMP;
- интерфейс ЦАП DAC;
- два интерфейса PSC (PSC0, PSC1);
- интерфейс с двумя входами ADC.

И включает в свой состав наиболее распространенные варианты интерфейсов периферийных устройств для микроконтроллеров: датчиков и драйверов управления электродвигателей.

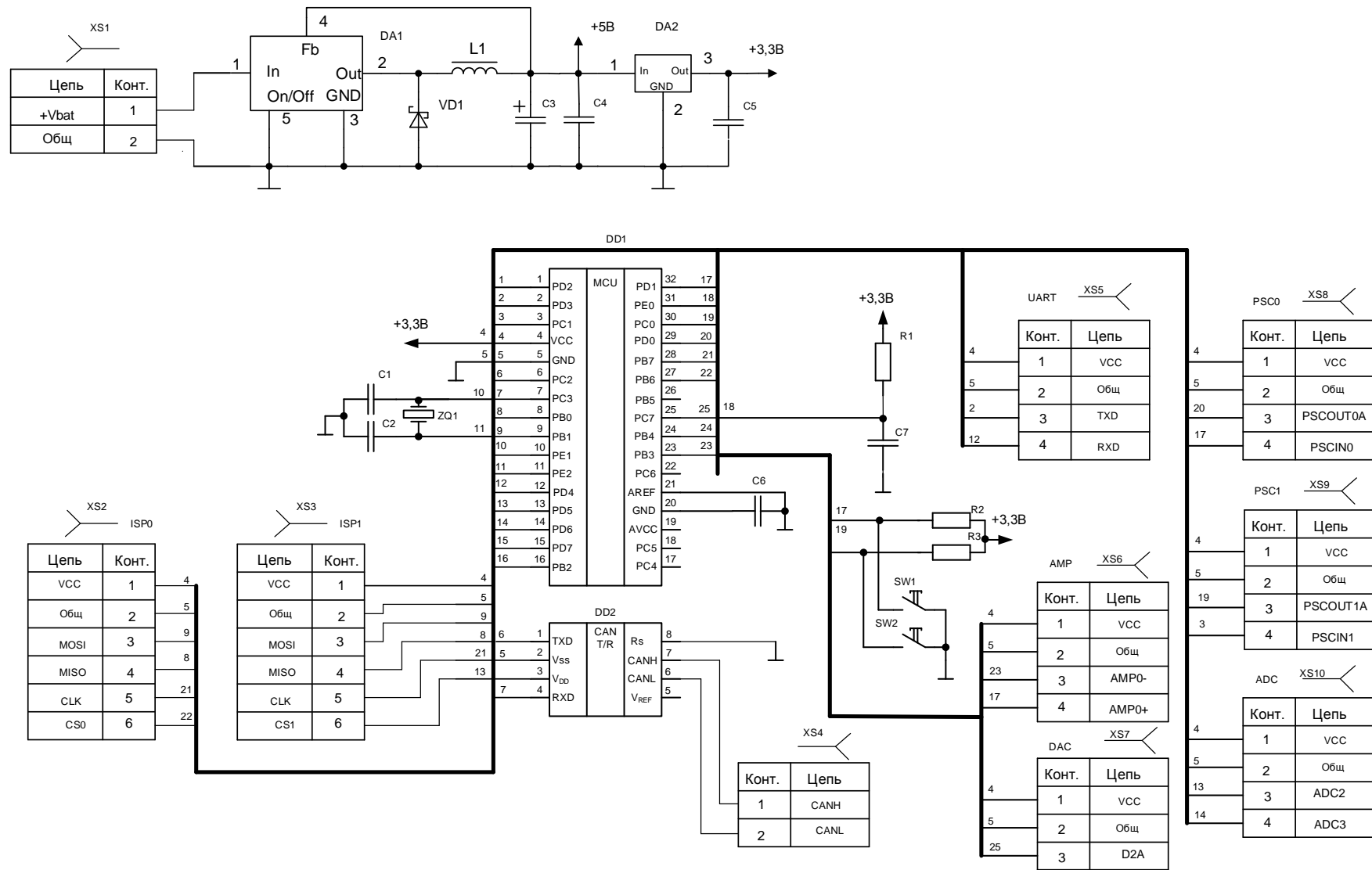


Рисунок 43 - Принципиальная схема моста CAN-шина – устройство

Заключение

В результате проведенной работы собрана основная информация о работе CANbus на физическом уровне, разобрана работа различных вариантов организации построения сети на физическом уровне.

Рассмотрена структура протокола верхнего уровня CANopen. Определен порядок формирования сообщений различного типа, основные события в сети и принципы обнаружения и исправления ошибок передачи на физическом уровне с помощью механизмов протокола верхнего уровня.

Составлена структура типового технологического автономного транспортного средства, выполняющего логистические задачи производственных процессов. Определена необходимость формирования узлов сети при переходе от кабельных линий и жгутов проводов от датчиков к центральной системе управления. Установлена необходимость разработки схемотехнических решений центральной системы управления и мостов между периферийными устройствами и CAN-шиной.

Разработана электрическая принципиальная системы управления автономным транспортным средством, включающей в свой состав контроллер CANbus и трансивер сигналов TTL уровня в CANbus.

Разработаны два варианта схем электрических принципиальных моста между шиной CANbus и интерфейсами реальных устройств:

- упрощенная версия, реализующая трансляцию потоков информации между шиной CANbus и интерфейсом UART,

- универсальная версия, реализующая трансляцию потоков информации между шиной CANbus и интерфейсами ISP, UART, интерфейсом дифференциального усилителя AMP, интерфейсом DAC, интерфейсом PSC, интерфейсом с двумя входами ADC.

Таким образом, задачи поставленные в работе выполнены полностью.

Список используемой литературы и используемых источников

1. AMIS-41682, AMIS-41683. Fault Tolerant CAN-Transceiver. Data sheet компании ON Semiconductor
2. AMIS-42700 Dual High-Speed CAN Transceiver. Feedback Supression. Документ AND8358.pdf компании ON Semiconductor
3. AMIS-42700. Multiple CAN Bus Network. Документ AND8360.pdf ком-пании ON Semiconductor.
4. AT90CAN32, AT90CAN64, AT90CAN128 8-bit Microcontroller with 32K/64K/128K Bytes of ISP Flash and CAN Controller. Rev. 7679H–CAN–08/08. Atmel Corporation.
5. ATmega16M1/ATmega32M1/ATmega64M1/ATmega32C1/ATmega 64C1 Automotive. 8-bit AVR Microcontroller with 16K/32K/64Kbytes In-system. DATASHEET 7647O-AVR-01/15 // 2015 Atmel Corporation
6. CAN // https://www.bookasutp.ru/Chapter2_6.aspx
7. CAN + CANOpen + CANfestival + STM32. Часть первая // <http://we.easyelectronics.ru/STM32/can-canopen-canfestival-stm32-chast-pervaya.html>
8. CAN Application Layer (CAL) for industrial applications. CiA 201 to CiA 207. Version 1.1.0. Public Available Specification // CAN in Automation (CiA) e. V.
9. CANopen // <https://ru.wikipedia.org/wiki/CANopen>
10. CANopen. Application Layer and Communication Profile. CiA Draft Standard 301. Version 4.02, 2005.
11. Device and network design. Part 1: CANopen physical layer. / CiA 303-1. Technical Report. Version 2.0.0 // CAN in Automation (CiA) e. V.
12. ESP-12E WiFi модуль (ESP8266) // <https://radioprogram.ru/shop/merch/33>
13. ESP8266EX. Datasheet. Version 6.0 // Espressif Systems Copyright © 2018

14. GCAN-PLC-400/510/511 — с возможностью связи по шине CAN // <https://www.gcanbus.com/products/plc-con-ru/?etext=2202.Y7y3FPnPf0wX2N6-byU8XUklh3KrI-mi8kP8kmeFudhqbmpvaWthb3huYWJud3dx.83d4ea1d3a95b9303ce5ea40929476dc67b9e38f&yclid=3276520126135198044>
15. ISO 11898-1:2015 Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical signaling // <https://www.iso.org/standard/63648.html>
16. LM1117 800-mA, Low-Dropout Linear Regulator / SNOS412O – FEBRUARY 2000–REVISED JUNE 2020 // Texas Instruments Incorporated
17. LM2596 SIMPLE SWITCHER® Power Converter 150-kHz 3-A Step-Down Voltage Regulator / SNVS124F – NOVEMBER 1999 – REVISED APRIL 2021 // Texas Instruments Incorporated
18. LS LiDAR AGV И Роботизированная Навигация И Приложения Для Предотвращения Столкновений // <https://www.leishenlidar.com/ru/%d0%bf%d1%80%d0%b8%d0%bb%d0%be%d0%b6%d0%b5%d0%bd%d0%b8%d1%8f/%d0%bb%d0%b8%d0%b4%d0%b0%d1%80-%d0%b4%d0%bb%d1%8f-agv-%d1%80%d0%be%d0%b1%d0%be%d1%82-%d0%bd%d0%b0%d0%b2%d0%b8%d0%b3%d0%b0%d1%86%d0%b8%d1%8f-%d0%b2-%d0%bf%d0%be%d0%bc%d0%b5%d1%89%d0%b5%d0%bd%d0%b8/>
19. MCP2551. High-Speed CAN Transceiver. DS21667D // Datasheet. - Microchip Technology Inc.
20. PCA82C251. CAN transceiver for 24 V systems. Rev. 04 — 25 August 2011. Document identifier: PCA82C251. Product data sheet // NXP B.V. 2011
21. В. Сементин Использование лидаров в качестве сенсоров автоматизированных погрузчиков // https://laserportal.ru/content_1294
22. Введение в CANopen // <http://can.marathon.ru/page/can-protocols/canopen/canopenintro>

23. Введение в протокол CAN <http://can.marathon.ru/page/can-protocols/canbus/canintro> , Краткий обзор протокола CAN. Часть I / https://micromax.ru/about/articles/2160/?sphrase_id=26822347
24. Датчик вращения (энкодер) KY-040 // https://3drob.ru/datchiki_i_sensory/datchiki_vibracii_ugla_naklona/datchik_vrashe_niya_ekoder_ky_040
25. Дисплей TFT 2,2 дюйма SPI (ILI9341) // <https://mcustore.ru/store/displei/displej-tft-22-dyujma-spi-ili9341/>
26. Карпенко Е. Возможности CAN-протокола// СТА. 1998. № 4.
27. Контроллер DVP32ES200TC // <https://deltronics.ru/catalog/programmiruemyie-kontrolleryi/dvp-es2-ex2/dvp32es200tc/>
28. Контроллеры segnetics matrix // https://aesf.ru/catalog/segnetics_matrix/
29. Краткий обзор протокола CAN. Часть II / https://micromax.ru/about/articles/2161/?sphrase_id=26822347
30. ЛИДАРЫ ДЛЯ AGV / МОБИЛЬНЫХ РОБОТОВ: ЛИДАР ДЛЯ AGV / МОБИЛЬНЫХ РОБОТОВ WLR-716 // https://sensotek.ru/catalog/vanjee/section_6656/lidar-dlya-agv-mobilnykh-robotov-wlr-716/
31. Мельников С. Использование сети CAN и стека CANopen // <https://habr.com/en/post/144435/>
32. Приемопередатчики для сетей CAN производства NXP, Texas Instruments и Maxim <https://www.compel.ru/lib/54030>
33. Приемопередатчики для сетей CAN производства NXP, Texas Instruments и Maxim // <https://www.compel.ru/lib/54030>
34. ПРОГРАММИРУЕМЫЕ ЛОГИЧЕСКИЕ КОНТРОЛЛЕРЫ // <https://delta-kip.ru/catalog/programmiruemye-kontrollery/>

35. Русских М. Протокол высокого уровня CANopen. Часть 1 // Журнал РАДИОЛОЦМАН, январь 2013.
<https://www.rlocman.ru/review/article.html?di=147763>
36. Система управления транспортными тележками AGV или AMR из доступных промышленных компонентов.//
<https://zen.yandex.ru/media/adventory/sistema-upravleniia-transportnymi-telejkami-agv-ili-amr-iz-dostupnyh-promyshlennyh-komponentov-5fcbb26a788eda75c7ee94a8>
37. Современные приемопередатчики сети CAN компании ON Semiconductor . Ромадина Ирина // <https://kit-e.ru/interface/sovremennye-priemoperedatchiki-seti-can-kompanii-on-semiconductor/>
38. Третьяков С.А. CAN на пороге нового столетия. Мир компьютерной автоматизации №2, 1999 // www.datamicro.ru
39. Цветков Д. Приемопередатчики для сетей CAN производства NXP, Texas Instruments и Maxim // <https://www.rlocman.ru/review/article.html?di=55664>