# ­РЕФЕРАТ

РПЗ 31 страница, 18 рисунков, 4 таблицы, 13 источников, 1 приложение.

ПРОТОКОЛ, АВТОМОБИЛЬ, СКАННЕР, ДАТЧИК

Объектом исследования являются данные, получаемые с компьютера автомобиля.

Цель работы – анализ процесса коммуникации с компьютером автомобиля, протоколов и технологий, задействованных в этом. Определить основные функции системы сбора и обработки информации с компьютера автомобиля, выбрать архитектуру разрабатываемого приложения.

В результате работы был проведен анализ процессов передачи и получения информации с компьютера автомобиля, исходя из которого были определены функции разрабатываемого приложения. Помимо этого, была выбрана платформа для разработки системы, язык программирования, а также архитектура приложения.

# СОДЕРЖАНИЕ

[Введение 6](#_Toc183031107)

[1 Анализ процесса общения между компьютером автомобиля и его компонентами 7](#_Toc183031108)

[2 Протоколы связи автомобиля. 10](#_Toc183031109)

[2.1 Протокол LIN. 10](#_Toc183031110)

[2.2 Протокол CAN. 10](#_Toc183031111)

[2.3 Протокол FlexRay. 11](#_Toc183031112)

[2.4 Кадры данных рассматриваемых протоколов. 12](#_Toc183031113)

[2.5 Сравнительный анализ. 13](#_Toc183031114)

[3 Виды данных, получаемых с компьютера автомобиля. 15](#_Toc183031115)

[4 OBD-II. 19](#_Toc183031116)

[4.1 Назначение и структура. 19](#_Toc183031117)

[4.2 Стандарты OBD-II. 20](#_Toc183031118)

[4.3 Структура запросов и ответов OBD-II. 23](#_Toc183031119)

[5 Выбор устройства для коммуникации. ELM327. 27](#_Toc183031120)

[5.1 Сравнительный анализ ELM327 v1.5 и ELM327 V2.1 27](#_Toc183031121)

[5.2 ELM327 v1.5 28](#_Toc183031122)

[Заключение 30](#_Toc183031123)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 31](#_Toc183031124)

Приложение А (Черновик технического задания ВКРБ)……………………..33

# ОБОЗНАЧЕНИЯ, ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ

ОС – операционная система.

ПО – программное обеспечение.

ЭБУ – электронный блок управления.

# Введение

Автомобили в наше время перешли из некогда предмета роскоши во некоторого рода обыденность. Частой проблемой является поддержание этих самых автомобилей в надлежащем состоянии. Несмотря на большое количество автосервисов, у большинства автовладельцев не имеется времени для их регулярного посещения.

Данные, хранящиеся в компьютере автомобиля, использующиеся в том числе для проведения непосредственной его диагностики, как правило не доступны человеку без использования специальных приборов и ПО. Разрабатываемое приложение позволит решить проблему сканирования состояния автомобиля и непосредственного общения с ним, путем предоставления возможности коммуникации с автомобилем пользователю, который может быть как автовладельцем, так и автомехаником.

Исходя из всего вышеперечисленного можно сделать вывод об актуальности и пользе программной системы сбора и обработки информации с компьютера автомобиля.

# 1. Исследовательская часть

## 1.1 Анализ процесса общения между компьютером автомобиля и его компонентами

У большинства современных автомобилей в наше время есть бортовые компьютеры, помощники, рассказывающие базовую информацию об автомобиле. Они также выполняют диагностику автомобиля и обеспечивает обмен информацией с другими компонентами, в том числе через интерфейсы для диагностики и ремонта. Примером таких систем являются электронные блоки управления, которые могут управлять отдельными функциями автомобиля, такими как управление двигателем, тормозами или системой подвески. Каждый ЭБУ может быть специализирован для определённой задачи и обмениваться данными с другими ЭБУ через автомобильную сеть. Например, принцип действия электронного блока управления двигателем показан на рисунке 1.

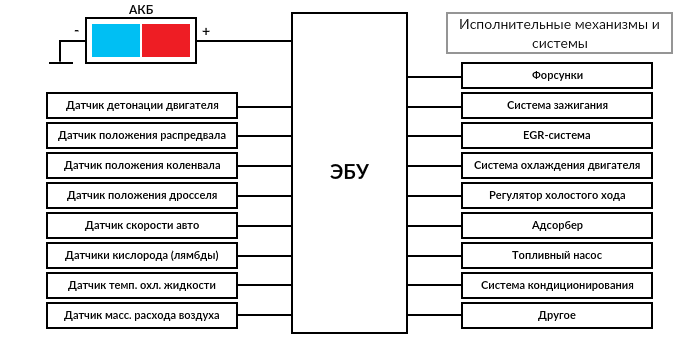


Рисунок 1 – Принцип действия ЭБУ двигателя.

В автомобиле предусмотрен ряд специальных систем, используемых в том числе при сканировании, в число которых входят:

* датчики и сенсоры: являются основными источниками данных для

бортового компьютера, включают в себя датчики температуры, давления, скорости, положения руля, датчики угла наклона и ускорения, они предназначены для сбора информации как о состоянии самого автомобиля, так и для сбора информации об окружающей его среде;

* ЭБУ: в каждом автомобиле их может быть множество, у многих систем

есть свои собственные ЭБУ для непосредственного управления определенными параметрами, к примеру ЭБУ двигателя управляет топливной системой и процессом сгорания, а ЭБУ тормозной системы в свою очередь контролирует работу антиблокировочной системы колес;

* центральный процессор: очень важный элемент автомобиля,

обрабатывающий всю информацию, поступающую с различных ЭБУ и выполняющий все вычисления необходимые для принятия каких-либо решений, этот процессор может быть связан с диагностическими системами, такими, как например OBD-II;

* диагностический интерфейс: элемент, который используется для связи

с какими-либо внешними устройствами, в число которых входят сканнеры, для получения и анализа данных о работе автомобиля.

Хорошими примерами бортовых систем могут послужить

* система управления двигателем: этот ЭБУ контролирует подачу топлива,

углы зажигания и другие параметры, чтобы обеспечить эффективную работу двигателя, снизить выбросы и повысить экономию топлива.

* антиблокировочная система тормозов (ABS): эта система использует

датчики, чтобы предотвратить блокировку колёс при торможении, обеспечивая улучшенное сцепление с дорогой и сокращает вероятность аварий.

* система стабилизации (ESP): система, контролирующая сцепление

колес с дорогой и предотвращающая занос автомобиля, достигает своего результата за счет данных, получаемых от датчиков скорости колес, угла поворота руля и ускорения.

* системы помощи водителю (ADAS): современные автомобили

оснащаются такими системами, как адаптивный круиз-контроль, системы предупреждения о выходе из полосы и автоматического торможения. Эти системы полагаются на датчики и данные с бортовых компьютеров для обеспечения безопасности водителя и пассажиров [1].

## 1.2 Протоколы связи автомобиля

Все данные в зависимости от автомобиля могут передаваться по-разному, самыми распространенными способами передачи данных между главным процессором автомобиля, его ЭБУ, датчиками, сенсорами и другими устройствами, в том числе диагностическими, являются протоколы CAN, LIN, Flex Ray [2].

Рассмотрим для общего понимания основные характеристики и плюсы каждого из них.

Основные особенности LIN:

* скорость подключения до 20 Кбит/с;
* переменная длина пакета данных (2, 4 или 8 байт);
* гибкость настройки;
* контрольная сумма и обнаружение ошибок;
* недорогостоящее;
* один мастер и 15 ведомых нод;
* обнаружение “проблемных” нод.

Протокол LIN создавался как более дешевый вариант протокола, так как уже активно применяющийся протокол CAN, был довольно дорогим способом коммуникации систем, а также в некоторых системах был излишне производителен. В данном протоколе всё взаимообщение между нодами осуществляется по запросу от Master’а, и выбранная им нода начинает общение, благодаря чему в данном протоколе нет необходимости учета коллизий в сети. Мастер и ведомые как правило ­– это микроконтроллеры. Как правило этот протокол используется совместно с протоколом CAN в автомобильной сфере, выстраивая иерархическую сеть в автомобиле [3].

Основные особенности CAN:

* однопоточное подключение до 1 Мбит/с;
* очень высокая надежность и защищенность;
* каждое сообщение имеет свой собственный приоритет;
* механизм обнаружения ошибок;
* средний ценовой сегмент;
* возможность присутствия нескольких мастер нод;
* обнаружение “проблемных/сбойных” нод;
* высокая устойчивость к помехам.

Протокол CAN является самым распространенным протоколом в автомобильной сфере в наше время. Протокол является золотой серединой между Flex Ray и LIN, имея неплохую скорость передачи и среднюю цену. Данным протоколом пользуются большинство автомобильных концернов [4].

Основные особенности Flex Ray:

* при однопоточном подключении скорость до 10 Мбит/с;
* передача с таймерным управлением;
* гарантированная задержка отклика;
* контрольная сумма и обнаружение ошибок;
* дорогой ценовой сегмент;
* возможность двухпоточного подключения;
* высокая устойчивость к помехам.

Протокол Flex Ray в свою очередь был разработан консорциумом автомобильных и электротехнических компаний, с целью создания протокола с большей скоростью передачи, надежностью, возможностью разделения в статические и динамические сегменты для коммуникаций, вызванных действием или по времени. Преимущественно используется в авиационной и автомобильной сфере. Статический сегмент разделен на части, предопределенные для того или иного типа коммуникации, что в свою очередь дает ему большую определенность чем у предшествующего протокола CAN. Динамический сегмент работает более схоже с CAN, ноды в нем берут управление над шиной, при ее доступности, что позволяет применять методику передачи по вызову [5].

Рассмотрим примеры кадров данных, используемых в этих протоколах, изображенные на рисунках 2 – 4.

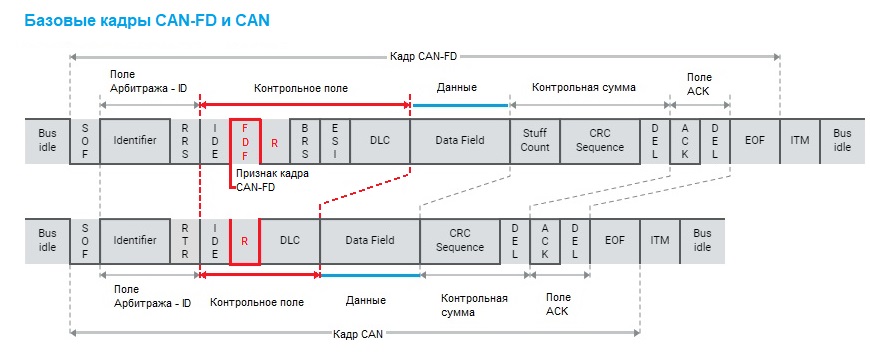


Рисунок 2 – Формат кадра данных CAN

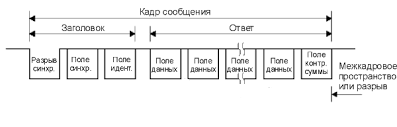


Рисунок 3 – Формат кадра данных LIN

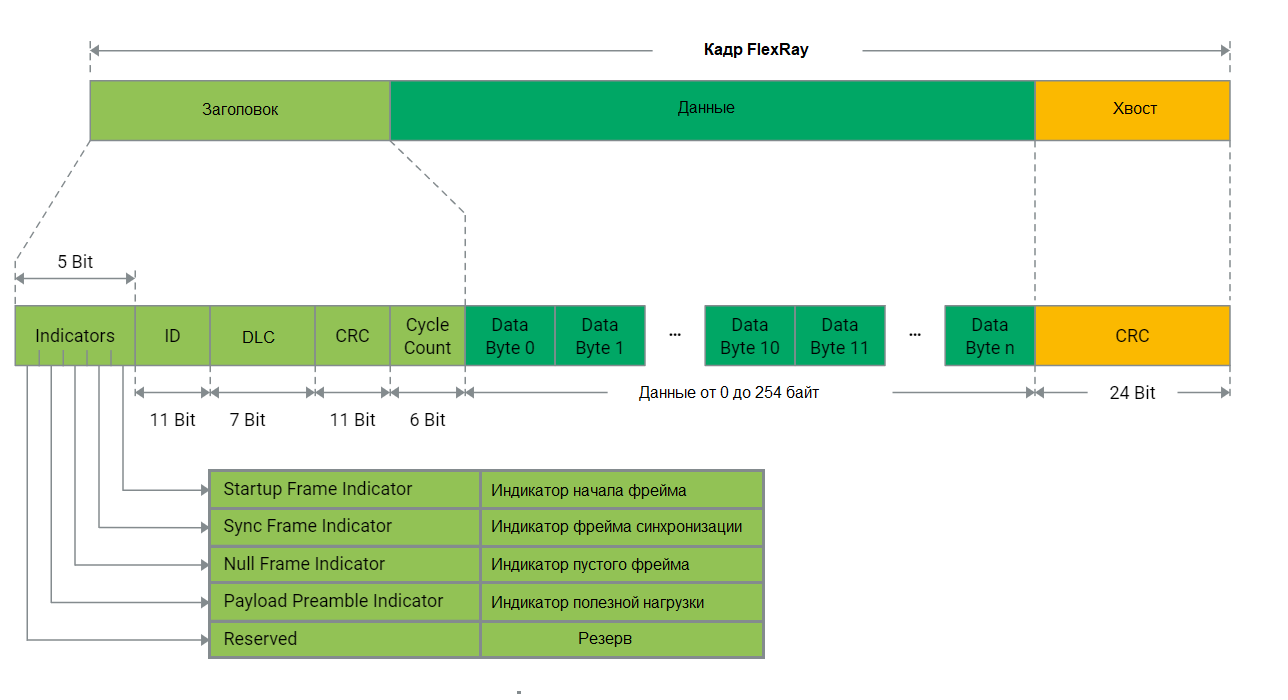


Рисунок 4 – Формат кадра данных Flex Ray

В рассматриваемых кадрах можно заметить, как и многие сходства, такие как поле контрольной суммы, поля хранящие в себе непосредственные данные, поля идентификации в заголовках кадров, биты DLC, так и биты присущие только определенным из протоколов.

Биты контрольной суммы используются в протоколах для проверки целостности пакетов и отсутствия ошибок, в полях данных хранятся сами байты данных, передающиеся в кадрах, поле идентификатора отвечает за идентификацию кадра, биты CRC указывают количество байт, передаваемых в кадре в поле данных, значения остальных полей не столь важны в нашей задаче, и рассмотрены подробнее не будут [6].

Проведем сравнительный анализ перечисленных протоколов, визуализированный в таблице 1.

Таблица 1 – Сравнительный анализ протоколов.

|  |  |  |  |
| --- | --- | --- | --- |
|  | LIN | CAN | FlexRay |
| Максимальная скорость | 20 Кбит/с | 1 Мбит/с | 10 Мбит/с |
| Ценовой сегмент | Экономичный | Средний | Высокий |
| Обнаружение ошибок | Присутствует | Присутствует | Присутствует |

Подводя итоги можно выделить следующие тезисы:

* CAN (Controller Area Network) – является основным протоколом обмена

данными между бортовыми компьютерами, контроллерами, датчиками и устройствами сканирования. Это высокоскоростной канал, используемый для передачи сообщений между компонентами автомобиля.

* LIN (Local Interconnect Network) – более медленный и дешевый протокол.

Используется для передачи информации между компонентами, не сильно зависящими от скорости, к примеру датчики в сиденьях, зеркала и освещение.

* Flex Ray – протокол нашедший свое применение в зависимых от

скорости передачи компонентах и приложениях в современных автомобилях, примером его применения являются системы для управления активной подвеской, динамическим управлением и системами автоматического вождения.

## 1.3 Виды данных, получаемых с компьютера автомобиля

Для лучшего понимания видов данных, получаемых с компьютера автомобиля, начнем анализ с устройств, с которых эти данные приходят на сам бортовой компьютер. Такими устройствами являются датчики и сенсоры, установленные в различных системах автомобиля, для наблюдения за их состоянием и состоянием вокруг них.

Перечислим одни из основных по важности из датчиков, активно использующихся в автомобилях, в том числе и для диагностики, в наше время:

* датчик массового расхода воздуха (ДМРВ);
* датчик положения коленчатого вала (ДПКВ);
* датчик температуры охлаждающей жидкости (ДТОЖ);
* датчик положения дроссельной заслонки (ДПДЗ);
* дегулятор холостого хода (РХХ);
* датчик кислорода (лямбда зонд);
* датчик детонации.

Теперь рассмотрим каждый из этих датчиков поподробнее с целью понимания данных, за которые они ответственны, выяснения параметров, которые в дальнейшем будут доступны для сканирования в разработанном ПО и более точного представления картины работы автомобильных систем.

Внешний вид всех вышеперечисленных датчиков представлен на рисунках 5 – 11.

Рисунок 5 – ДМРВ Рисунок 6 – ДПКВ

Рисунок 7 – ДТОЖ Рисунок 8 – ДПДЗ



Рисунок 9 – РХХ Рисунок 10 – Лямбда зонд



Рисунок 11 – Датчик детонации

Датчик массового расхода воздуха имеет функцию учета количества поступающего воздуха в камеры сгорания смеси топлива, проблемы, связанные с этим датчиком, могут выражаться в том числе в повышенном расходе топлива, падении мощности двигателях [7].

Датчик положения коленчатого вала является уникальным в своем роде датчиком. Если он выйдет из строя, то автомобиль не получиться эксплуатировать до устранения неполадки. Его задача – фиксирование и передача информации о положении коленчатого вала на ЭБУ, который производит все необходимые корректировки, такие как объем смеси топлива, поступающей в двигатель, угол опережения зажигания, угол поворота распредвала и время подачи топлива [7].

Датчик температуры охлаждающей жидкости также отправляет полученные данные на ЭБУ, если датчик неисправен, то данные не будут переданы, ЭБУ не запустит системы охлаждения двигателя, и произойдет перегрев [7].

Датчик положения дроссельной заслонки ответственен за объем подачи топлива, выполняет эту функцию за счет фиксирования положения дроссельной заслонки. В зависимости от угла поворота заслонки меняется напряжение и передается на ЭБУ. Неисправность этого датчика может привести к ухудшению показателей двигателя [7].

Регулятор холостого хода обеспечивает работу двигателя в момент, когда дроссельная заслонка закрыта. В случае его неисправности автомобиль сможет запуститься только при нажатии педали газа [7].

Задачей датчика кислорода является анализ отработанных газов и определение уровня несгоревшего кислорода. Показания служат для корректировки подачи топлива [7].

Датчик детонации необходим для того, чтобы ЭБУ, получивший необходимые значения выставлял корректный уровень опережения зажигания. При неисправности можно заметить потерю мощности автомобиля, увеличенный расход топлива и повышенную температуру двигателя [7].

Исходя из этого анализа можно сделать вывод, что при выходе любого из этих датчиков, как правило, возникает снижение мощности автомобиля, повышенный расход топлива, а также проблемы с запуском. Помимо этого, можно выделить параметры, которые обязательно стоит включить в доступные в разрабатываемом ПО. Такими параметрами являются:

* температура охлаждающей жидкости;
* коррекция времени впрыска;
* давление впуска;
* угол опережения зажигания;
* температура воздуха на впуске;
* положение дроссельной заслонки;
* напряжения датчика О2;
* коррекция времени впрыска;
* напряжение аккумулятора;
* и прочие полезные параметры.

## 1.4 Анализ принципа работы OBD-II

OBD-II – это стандарт который описывает тип разъема для диагностики и его распиновку, указанную на рисунке 12, а так же включающую в себя пин питания от автомобиля, для подключенного к нему сканнера, поддерживает работу протоколов с электронными сигналами, и их формат сообщений. Он предоставляет объемный список параметров автомобиля, которые можно сканировать с помощью него, а также кодировку этих параметров с их значениями.

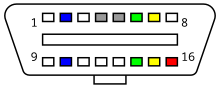
[](https://en.wikipedia.org/wiki/File:OBD-II_type_A_female_connector_pinout.svg)

Рисунок 12 – Распиновка OBD-II

Поясним распиновку, указанную на рисунке 12. На схеме все белые пины, а именно 1, 3, 8, 9, 11, 12, 13 не являются стандартизированными и могут быть использованы в зависимости от производителя. Пины отмеченные синим цветом, а именно 2 и 10 являются положительной и отрицательной шиной соответственно, отвечающие за протоколы SAE-J1850 VPW и SAE-J1850 PWM. Пины отмеченные серым цветом, 4 и 5, отвечают за заземление кузова и сигнальное заземление соответственно. 6 и 14 пины ответственны за линии CAN-High и CAN-Low высокоскоростной шины CAN Highspeed соответственно. Красные пины – это пины K-Line и L-Line протоколов ISO 9141-2 и ISO 14230. 16 пин – питание +12В от АКБ [8].

Стоит также учитывать, что OBD-II предоставляет большой список автомобильных ошибок.

OBD-II по стандарту SAE J1979 и новее, поддерживает 10 режимов работы.

* считывание текущих параметров работы системы управления;
* получение сохраненной копии текущих параметров работы системы

управления на момент возникновение кодов неисправностей;

* считывание хранящихся кодов неисправностей;
* стирание кодов неисправностей, результатов тестовых мониторов;
* считывание и просмотр результатов теста датчиков кислорода, не для

шины CAN;

* считывание результатов тестов, контролирующих работу катализатора,

системы рециркуляции выхлопных газов и так далее;

* запрос результатов диагностики непрерывно действующих тестов;
* управление бортовыми системами;
* запрос информации о диагностируемом автомобиле;
* ошибки, которые были удалены.

Однако стоит понимать, что производители автомобилей не обязаны поддерживать работу всех режимов работы, а также имеют полное право добавления своих режимов работы, идущих после девятого режима.

Так как OBD-II является стандартизованным для всех автомобилей, то с помощью одного сканнера с таким же разъемом можно проводить запросы на любой автомобиль, вышедший после определенного временного промежутка. Однако несмотря на это, разные страны по-разному переработали этот формат, кто-то для более гибкой настройки и использования именно в своей стране, а кто-то как временное решение, до полного введения формата OBD-II. Так появились новые форматы OBD-II, указанные в таблице 2.

Таблица 2 – Форматы OBD-II.

|  |  |  |  |
| --- | --- | --- | --- |
| Название формата | Место, использующее  формат | Год утверждения для дизельных автомобилей | Год утверждения для бензиновых автомобилей |
| EOBD-II | Европа | 2007 | 2002 |
| JOBD | Япония | 2001 | 2001 |
| ADR 79/01 | Австралия | 2007 | 2006 |
| ADR 79/02 | Австралия | 2010 | 2010 |
| EMD | Северная Америка | 2007 | 2007 |
| EMD+ | Северная Америка | 2010 | 2010 |
| EOBD | Европа | 2004 | 2001 |

Несмотря на то что все из вышеперечисленных форматы практически идентичны и имеют небольшое количество различий, не всегда результат полученный с автомобилей разных стандартов OBD-II идентичен, например используя сканнер предназначенный для автомобилей EOBD-II с автомобилем JOBD, можно получить неточные, а порой даже неверные данные, именно поэтому рекомендуется использовать сканнеры поддерживающие тот же формат, что и сканируемый автомобиль.

OBD-II поддерживает несколько протоколов, по которым на него поступают сигналы:

* SAE J1850 PWM
* SAE J1850 VPW
* ISO 9141-2
* ISO 14230 KWP2000
* ISO 15765 CAN

Обязательным для большинства автомобилей и самым распространенным является ISO 15765 CAN.

Каждый из этих протоколов имеет свои особенности, и использует определенные пины. Как правило в автомобиле реализован только один из этих протоколов на определенную систему. Рассмотрим их особенности поподробнее [9].

SAE J1850 PWM:

* контакт 2 – Bus +;
* вывод 10 – Bus -;
* длина сообщения – 12 байт, в том числе 1 байт CRC;
* использует CSMA/NDA.

SAE J1850 VPW:

* контакт 2 – Bus+;
* длина сообщения – 12 байт, в том числе 1 байт CRC;
* использует CSMA/NDA.

ISO 9141-2:

* контакт 7 – K-Line;
* вывод 15 – L-Line (опционально);
* UART сигналы;
* длина сообщения – 12 байт, в том числе 1 байт CRC.

ISO 14230 KWP2000:

* контакт 7 – K-Line;
* вывод 15 – L-Line (опционально);
* физический уровень идентичен ISO9141-2;
* сообщение может содержать до 255 байт в поле данных.

ISO 15765 CAN:

* вывод 6 – CAN High;
* контакт 14 – CAN Low.

Кадры этих протоколов показаны на рисунках 13 – 17.



Рисунок 13 – Формат кадра данных SAE J1850 PWM

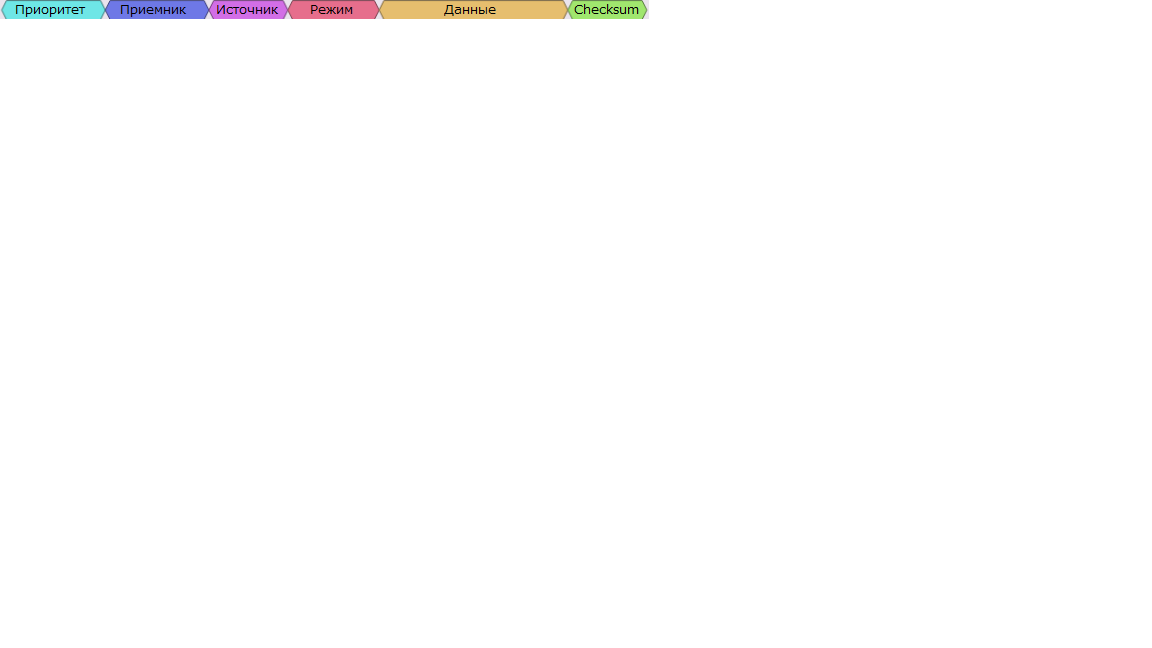


Рисунок 14 – Формат кадра данных SAE J1850 VPW

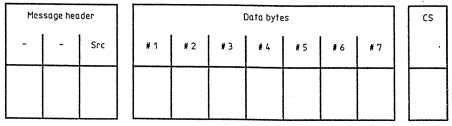


Рисунок 15 – Формат кадра данных ISO 9141-2

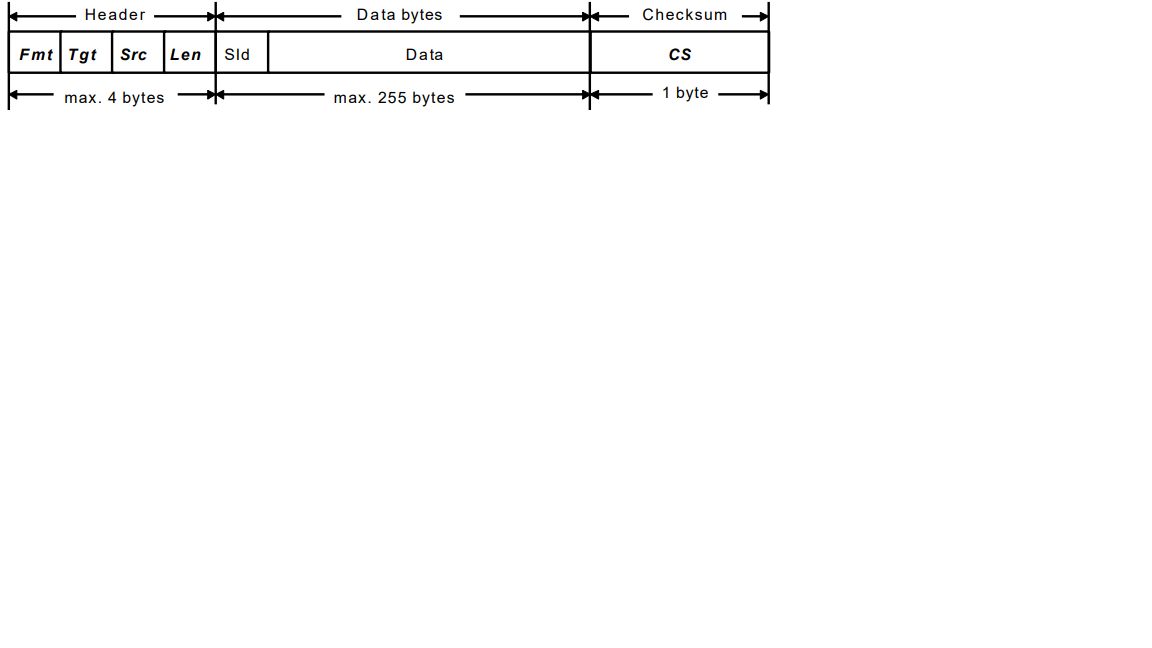


Рисунок 16 – Формат кадра данных ISO 14230 KWP2000

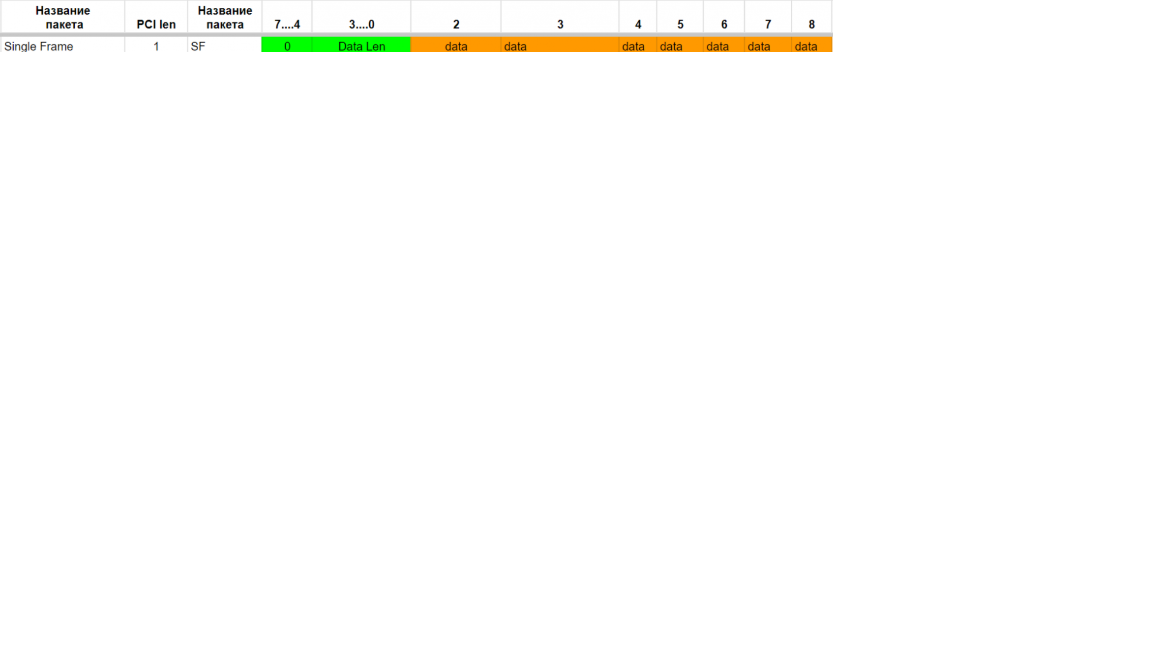


Рисунок 17 – Формат кадра данных ISO 15765 CAN

Таким образом все распиновки протокола OBD-II используют тот же разъем, но разные контакты, в автомобиле как правило используется один из этих протоколов для одной из систем автомобиля.

В зависимости от запрошенного сканером PID, идентификатор параметра, автомобиль отсылает ответ различного формата. Таким образом запросив один параметр, можно получить ответ длиной в 4 байта, а запросив следующий параметр – 2 байта.

Рассмотрим ответы автомобиля на получаемые им запросы на примере параметров, перечисленных в предыдущих пунктах, важных при диагностике автомобиля. Их вид и особенности показаны в таблице 3 [10].

Таблица 3 – Вид и особенности ответов, получаемых от автомобиля.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Режим  + PID | Кол-во  бит | Описание | Min | Max | Мера | Формула |
| 0100 | 4 | Список поддерживаемых PID (0-20) | - | - | - | Кодируется [A7..D0] == [PID 0x01..PID 0x20] |
| 03 | n\*6 | Запрос ошибок | - | - | - | BCD |
| 0104 | 1 | Расчетное значение нагрузки на двигатель | 0 | 100 | % | A \* 100 / 255 |
| 0105 | 1 | Температура охлаждающей жидкости | -40 | 215 | °C | A - 40 |
| 010A | 1 | Давление топлива | 0 | 765 | kPa | A\* 3 |
| 010B | 1 | Давление во впускном коллекторе | 0 | 255 | kPa | A |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 010C | 2 | Обороты двигателя | 0 | 16383,7 | rpm | ((A \* 256) + B) / 4 |
| 010E | 1 | Угол опережения зажигания | -64 | 63.5 | ° | A / 2 - 64 |
| 010F | 1 | Температура всасываемого воздуха | -40 | 215 | °C | A - 40 |
| 0110 | 2 | Массовый расход воздуха | 0 | 655.35 | г/с | ((A\*256) + B) / 100 |
| 0111 | 1 | Положение дроссельной заслонки | 0 | 100 | % | A \* 100 / 255 |
| 011C | 1 | OBD стандарт автомобиля | - | - | - | Data |
| 011F | 2 | Время, прошедшее с запуска двигателя | 0 | 65535 | с | (A \* 256) + B |
| 0140 | 4 | Список поддерживаемых PID’ов (41-60) | - | - | - | Кодируется [A7..D0] == [PID 0x41..PID 0x60] |
| 015D | 2 | Регулирование момента впрыска | -210 | 301.992 | ° | (((A \* 256) + B) – 26,880) / 128 |
| 015C | 1 | Температура масла двигателя | -40 | 210 | °С | A - 40 |

При рассмотрении таблицы становится понятно, что разные ответы от автомобиля, на разные запросы, должны будут расшифровываться уникально по-своему для каждого типа запросов.

Также стоит отметить, что у некоторых марок автомобилей данные PID, могут отличаться, или отсутствовать вовсе в зависимости от модели автомобиля, сканирование которого проводится. Для проверки наличия поддержки тех или иных PID, могут использоваться запросы с PID 00, 20, 40, 60 и прочие, которые помогут узнать, какие PID поддерживает автомобиль.

Особое внимание стоит уделить запросу с PID 1С, ответом на который, автомобиль возвращает OBD стандарт, который поддерживает автомобиль, что достаточно важно при непосредственной установке соединения между сканнером и автомобилем.

Ещё одним важным и выделяющимся запросом является запрос 03, который возвращает ошибки автомобиля, вычисляемые самим автомобилем и хранящиеся в его компьютере. Эти ошибки также нуждаются в расшифровке.

Общая схема расшифровки кодов ошибок OBD-II выглядит примерно так [11]:

Первый символ в коде ошибки означает:

* P – двигатель автомобиля и/или АКПП;
* B – кузовные системы, к примеру стеклоподъемники;
* C – шасси, ходовая часть;
* U – взаимодействия между ЭБУ.

Второй символ:

* 0 – общий индекс;
* 1 и 2 – код изготовителя;
* 3 – резерв.

Третий символ:

* 1 – подача воздуха или системы топлива;
* 2 – подача воздуха или системы топлива;
* 3 – система зажигания;
* 4 – вспомогательное управление;
* 5 – холостой ход;
* 6 – цепи ЭБУ или вся ЭБУ в целом;
* 7 – силовая передача;
* 8 – силовая передача.

Символы 4 и 5 – это порядковый номер ошибки OBD-II.

Таким образом, используя полученную информацию, можно создать алгоритм обработки как ответов автомобиля, так и кодов ошибок, содержащихся в некоторых из этих ответов.

## 1.5 Сравнительный анализ устройств ELM327

Для коммуникации с автомобилем обычно используются устройства, включающие в себя поддержку протоколов OBD-II, сам разъем OBD-II, способ передачи информации на сканнер и прочие полезные функции. По своей сути это приборы, для отправки и получения закодированных сообщений при общении с автомобилем. Основным поставщиком подобных устройств является компания ELM Electronics.

При выборе устройства для коммуникации, выбор по своей сути лежит между двумя устройствами ELM327, отличающихся в том числе микропроцессором, использованным в них. Устройства носят названия ELM327 v1.5 и ELM327 v2.1, сравнение их характеристик показано в таблице 4 [12][13].

Таблица 4 – Сравнение ELM327 v1.5 и Elm327 v2.1

|  |  |  |
| --- | --- | --- |
| Параметр | ELM327 v1.5 | ELM327 v2.1 |
| Поддержка команд OBD - II | Большой список поддерживаемых команд | Урезанная поддержка команд |
| Совместимость с автомобилями | Широкая совместимость, включающая в себя автомобили до 2010г | Ограниченная совместимость, лучше работает с новыми авто |
| Протоколы связи | I SO9141, ISO14230, CAN, J1850 PWM, J1850 VPW | ISO9141, ISO14230, CAN; ограниченно с J1850 |
| Стабильность связи | Высокая стабильность, минимум разрывов связи | Встречаются разрывы связи |
| Скорость работы | Быстрая и отзывчивая передача данных | Может быть медленнее, особенно при рассчетах в реальном времени |
| Цена | Дороже | Дешевле |
| Область применения | Подходит для профессионального использования | Подходит для базовой диагностики |

Подводя итоги можно с уверенностью сказать, что ELM327 v1.5 лучше подходит для задач сканирования автомобилей, а также является более хорошим вариантам в соотношении цена-качество.

Рассмотрим поподробнее устройство ELM327 v1.5, которое было выбрано нами в прошлом пункте.

Схема ELM327 v1.5, представлена на рисунке 18.

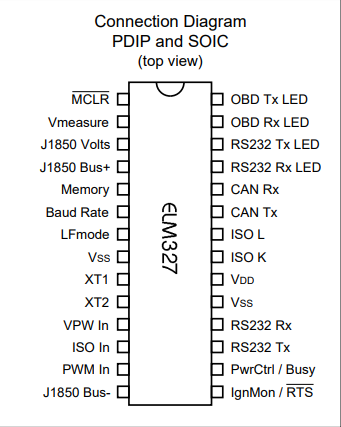


Рисунок 18 – Схема ELM327 v1.5

Стоит обратить внимание что в ELM327 v1.5 предусмотрена передача информации, получаемой от автомобиля и отправляемой автомобилю по Bluetooth, что дает этому устройству большую гибкость и позволяет использовать его как на мобильных устройствах, так и на ноутбуках и ПК.

Подводя итоги, ELM327 v1.5 – самое подходящее устройство для коммуникации с автомобилем среди доступных, оно имеет как поддержку Bluetooth, так и все необходимые функции для работы с OBD-II. Приятным плюсом выступает факт того, что устройство способно само определять используемый автомобилем протокол, и настраиваться на передачу и прием именно по нему.

# 2. Конструкторская часть

Перед началом создания системы необходимо понять принцип, по которому она должна будет работать и функционал, которым она будет обладать.

Первым и очевидным условием для системы выступает возможность подключения к адаптеру OBD2 и поддержания соединения для дальнейшей передачи данных автомобилю и в обратную сторону.

Так же необходима логика кодирования отправляемых запросов и декодирования ответов, получаемых от автомобиля, для чего была изучена специфика передачи и получения данных адаптерами OBD2. Так же для этой цели был изучен процесс передачи данных самим автомобилем на адаптер OBD2, систему кодирования которую он использует и возможные проблемы.

Из других ключевых функций стоит выделить проверку поддержки PID (On-Board Diagnostic Parameter Identification) автомобилем, для определения возможностей в общении программной системы с компьютером автомобиля и поддерживаемые команды.

Еще одной не менее важной функцией является получение и отправка запросов в реальном времени, с целью более точной диагностики или проверки систем автомобиля.

Необходимо предусмотреть опцию сканирования автомобиля по всем доступным PID, для удобства и достаточного количества информации, получаемой с компьютера автомобиля.

Логичным решением будет предусмотреть режим отправки сообщений компьютеру автомобиля вручную, с получением ответа в закодированном виде, что в свою очередь позволит отправить автомобилю сообщения, не предусмотренные системой изначально.

И наконец, возможность работы с кодами ошибками, хранящимися в компьютере автомобиля, возможность их просмотра с расшифровкой, а также возможность их удаления.

Функцию считывания идентификационной информации об автомобиле, такую как идентификационный номер ТС.

## 2.1 Че-то типа того что у Пантелеева

## 2.2 Разработка схемы структурной

Структурная схема помогает подробнее рассмотреть структуру разрабатываемого программного обеспечения, позволит увидеть взаимодействия между его модулями.

При начале работы программы, первым необходимым действием является подключение адаптера OBD2 для связи с автомобилем, за что отвечает модуль соединения с адаптером.

Следом за ним пользователю предоставляется возможность выбора режима работы программного обеспечения, далее в зависимости от выбора пользователя, в работу вступает один из пяти функциональных модулей.

Режим сканирования доступных PID, позволяющий быстро собрать информацию об автомобиле по имеющемуся списку PID. Режим работы с ошибками, дающий возможность получить информацию об ошибках, обнаруженных автомобилем и стереть их. Режим ручного ввода, с помощью которого можно вести общение напрямую с автомобилем. Режим обнаружения доступных PID, позволяющий выявить возможности автомобиля, в области общения с устройствами OBD2. Режим реального времени, дающий возможность повторяющейся отправки и получения информации автомобилю с определенным интервалом.

Во всех случаях после работы того или иного режима происходит визуализация полученных данных, в случае с режимом сканирования доступных PID, эти данные так же формируются в виде отчета и пользователю дается возможность скачать сформированный отчет.

На основе вышеперечисленной информации была спроектирована структурная схема программного обеспечения, показанная на рисунке 19.

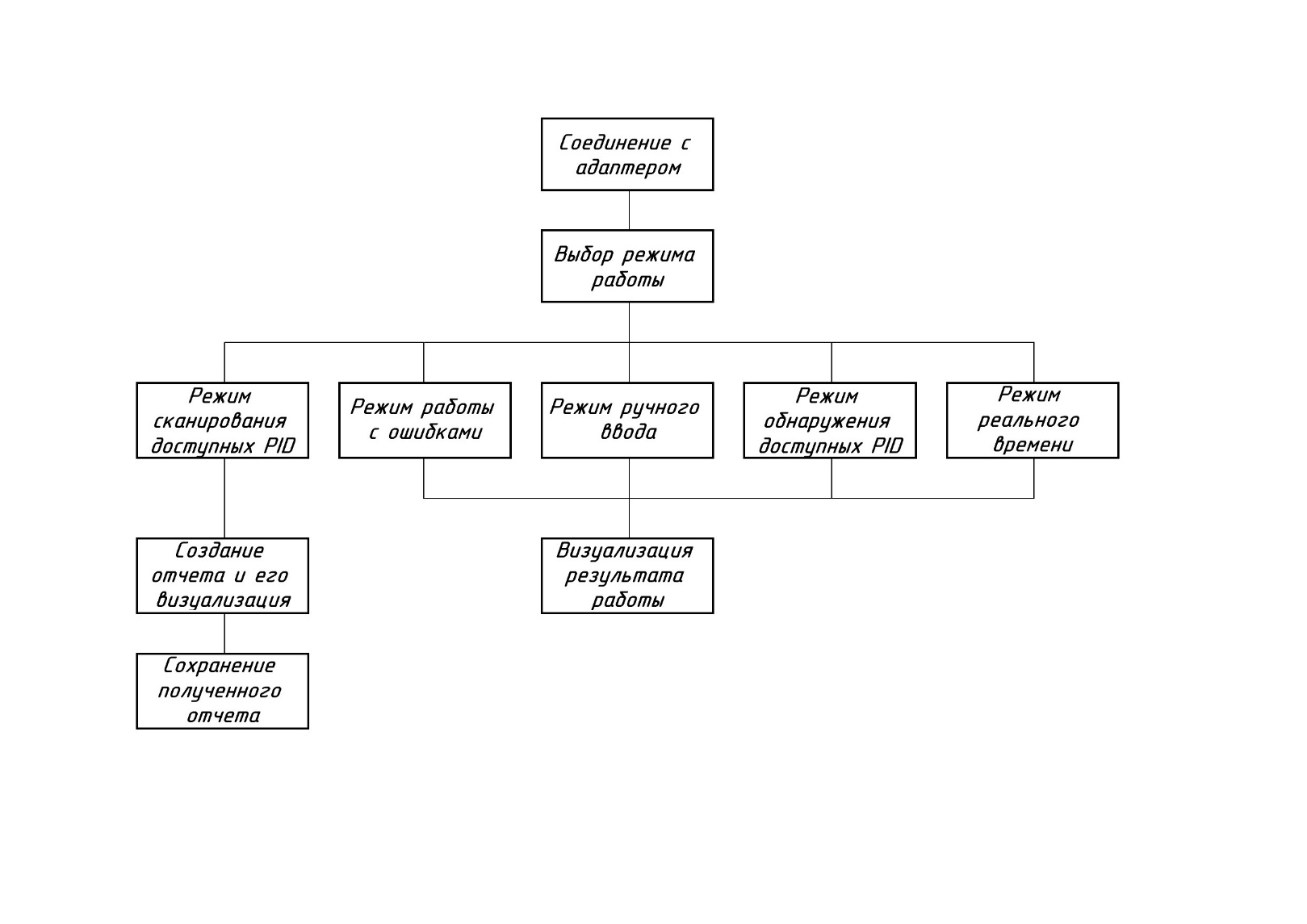


Рисунок 19 – Схема структурная

## 2.3 Разработка диаграммы деятельности

Диаграмма деятельности поможет детально рассмотреть процесс использования ПО, возможные проблемы и ошибки возникающие в процессе его работы.

Первым этапом пользователь вводит название адаптера, после чего система устанавливает соединение с адаптером по введенному ранее имени, на этом этапе могут возникать ошибки, возникающие из-за неправильных данных, введенных пользователем.

Если ошибок нет, происходит переход на выбор режима системы. На этом этапе пользователь выбирает режим работы программной системы, ошибки, возникающие после выбора, связаны с нехваткой данных о доступных автомобилю PID. Это свойственно для режимов реального времени, сканирования по доступным PID.

Если ошибок нет, происходит подготовка к работе с выбранным режимом, единственный тип ошибок, который может произойти на данном этапе – разрыв соединения с OBD2 адаптером.

При отсутствии ошибок, пользователю дается возможность ввода необходимых для выбранного режима данных, ошибками, возникающими на данном этапе являются ошибки введенных пользователем данных. Они могут возникать в случаях, если пользователь ввел неправильный формат PID, PID, введенный пользователем, не поддерживается сканнером и автомобилем, при пустом поле ввода и многих других факторах.

Если ошибок не обнаружено происходит выполнение необходимых запросов и получение необходимых ответов, на данном этапе ошибка может произойти при каком-либо из действий. Другими словами, каждый из отправленных запросов может повлечь за собой ошибки как при отправке, этого самого запроса, так и при получении ответа на него.

При отсутствии ошибок происходит визуализация результатов, если при ней так же не обнаружено ошибок, пользователь получает запрошенные им данные в понятном для человека, наглядном виде.

На основе вышеперечисленной информации была спроектирована диаграмма деятельности программного обеспечения, показанная на рисунке 20.

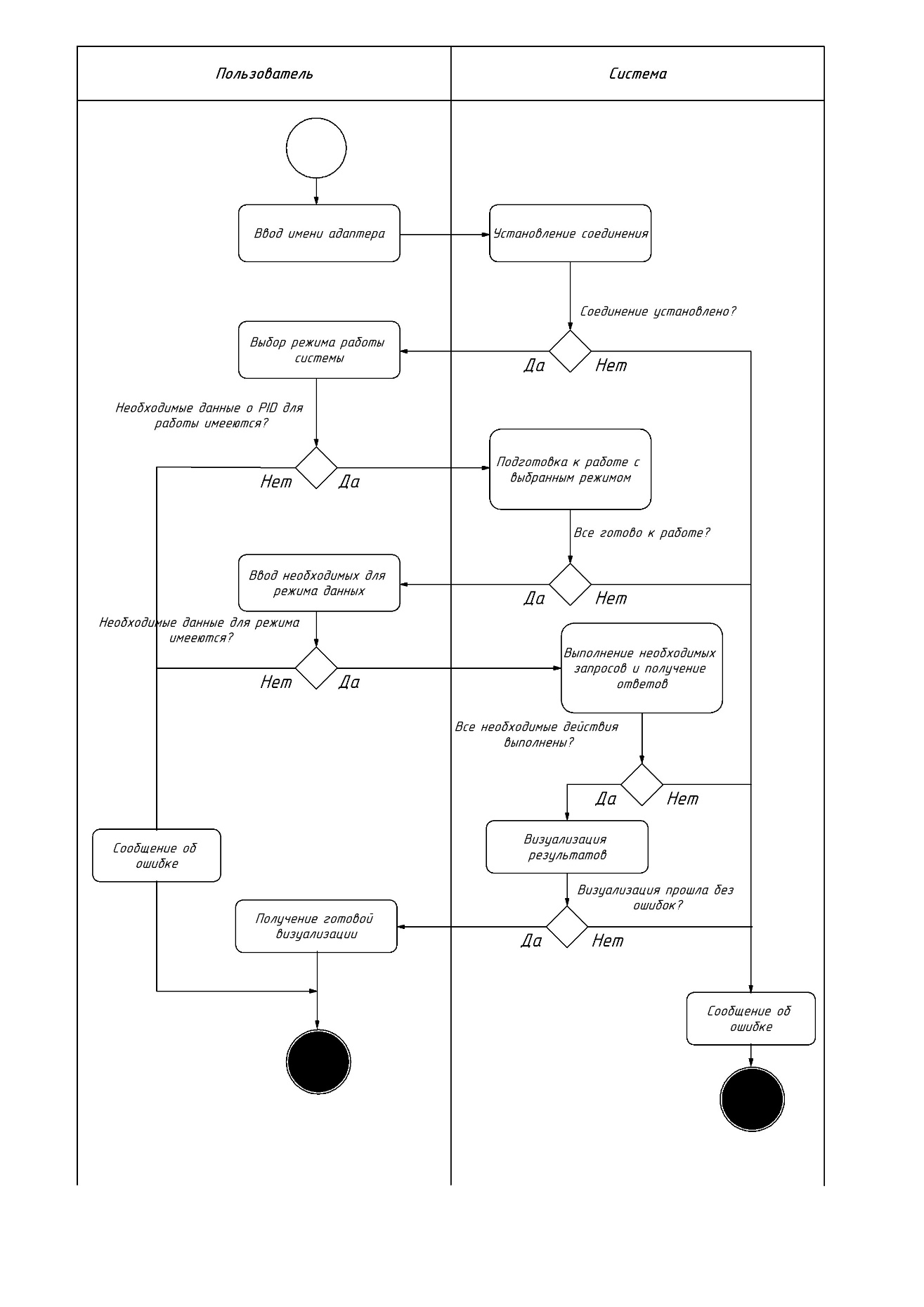


Рисунок 20 – Диаграмма деятельности

## 2.4 Разработка диаграммы функциональной

Для более точного понимания принципа работы программного обеспечения и взаимодействия его модулей, создадим для него функциональную диаграмму.

Функциональная диаграмма IDEF0 отлично отражает взаимосвязи между функциями в разрабатываемом программном обеспечении. Блоки в диаграмме располагаются в виде “ступенек”, позволяющем понять последовательность их работы. У каждой стороны каждого блока есть определенное назначение, левая сторона предназначена для входов, правая – для выходов, сторона сверху – управление, снизу – для механизмов.

Начнем с обозначения входных данных нашего ПО, ими будут выступать название сканнера OBD2, к которому будет выполняться подключение по Bluetooth, и запрос на компьютер автомобиля. Выходными данными будут визуализация ответов автомобиля и файл-отчет по полученным ответам. Управляющие данные, имеющие место быть в разрабатываемом программном обеспечении – режим отправки данных и интервал отправки запросов. Механизмы – модули программного обеспечения.

Так же рассмотрим наше ПО более детализировано. Название адаптера и запросы для его настройки попадают сразу же в блок установки соединения с адаптером, после установки стабильного соединения, начинает работу блок общения с автомобилем, получающий в качестве управляющих данных режим отправки данных и интервал отправки запросов, исходя из которых он выполняет все необходимые действия и на выходе отправляет декодированные данные с компьютера автомобиля на блок вывода результатов, который в свою очередь, с учетом управляющих данных в виде режима отправки данных дает на выходе визуализированный ответ автомобиля и файл-отчет по полученным ответам, если управляющие данные соответствуют необходимым.

Для более хорошего понимания системы, так же декомпозируем блок установления соединения с OBD2 адаптером. Он будет состоять из блоков обнаружения устройства по имени, установки связи с устройством и отправки команд конфигурации адаптера. Блок обнаружения устройства получает на вход название адаптера OBD2, после чего находит среди доступных Bluetooth устройств это название и передает его MAC-адрес блоку установки связи с устройством, который в свою очередь устанавливает связь с устройством по найденному адресу, далее блок отправки команд конфигурации производит настройку адаптера с помощью специальных команд и завершает работу блока установления соединения с OBD2 адаптером.

На основе вышеперечисленной информации была спроектирована функциональная диаграмма программного обеспечения, показанная на рисунке 21.

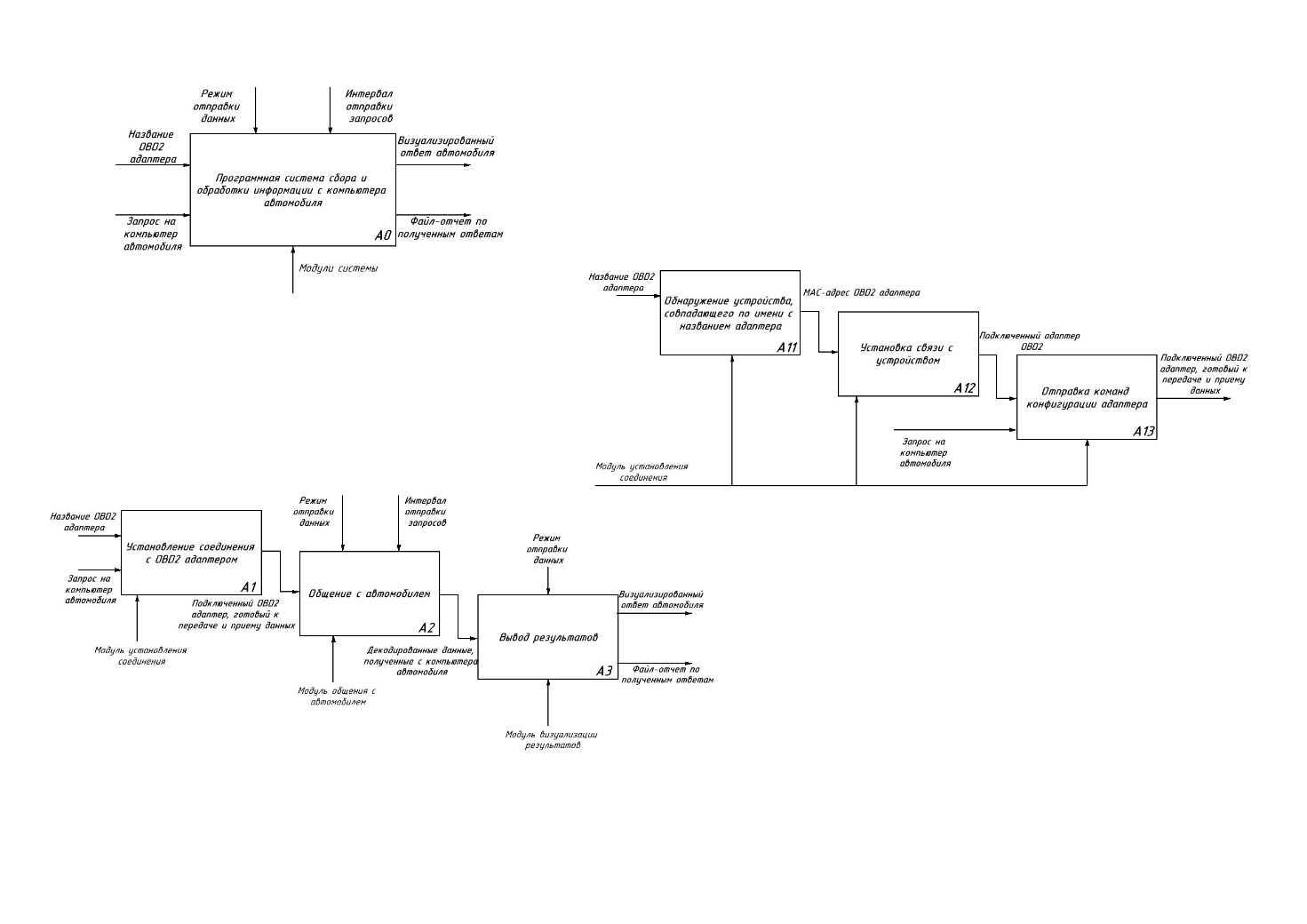


Рисунок 21 – Диаграмма функциональная

## 2.5 Разработка схемы взаимодействия модулей

## 2.6 Разработка схем алгоритмов системы

Разработка схем алгоритмов поможет подробнее рассмотреть модули, перечисленные в предыдущих этапах разработки, их принцип работы и примерную структуру.

Функция соединения, отвечающая за установку соединения в модуле соединения, сперва ищет имя адаптера среди доступных Bluetooth устройств и сопоставляет ему MAC-адрес, после чего производит подключение по этому адресу и наконец отправляет команды настройки OBD2 сканера.

Функция отправки команды и получения ответа ответственна за непосредственную передачу закодированной utf-8 последовательности символов на автомобиль и декодированием полученной utf-8 последовательности. Она используется совместно с функцией парсинга полученного ответа, которая в свою очередь делит полученный ответ на части по два HEX-символа, с помощью двух определенных из этих символов определяет тип запроса на который пришел этот ответ и находит подходящую функцию декодирования пришедших HEX-символов, вызывает ее и получает понятный для человека численный формат полученных данных.

Функция запроса доступных PID отправляет с помощью функции отправки команды и получения ответа заранее выбранный набор команд, при получении ответа на каждую из этих команд, ответ сначала очищается от лишних символов, иногда попадающих в него в качестве помех, затем вызывает парсер ответов на запрос о доступных PID, и после чего добавляет полученные данные в множество доступных PID.

Функция парсинга ответа о доступных PID декодирует ответ, получаемый в функции запроса доступных PID. Сперва эта функция удаляет первые 4 HEX символа служебной информации, затем переводит оставшуюся шестнадцатеричную строку в ее двоичный эквивалент. Каждый бит из полученного эквивалента проверяется на равенство “1”, и, в случае если условие выполняется, в множество для доступных бит добавляется эквивалент этому биту в формате PID.

На основе вышеперечисленной информации были спроектированы схемы алгоритмов модулей программного обеспечения, показанные на рисунке 4.

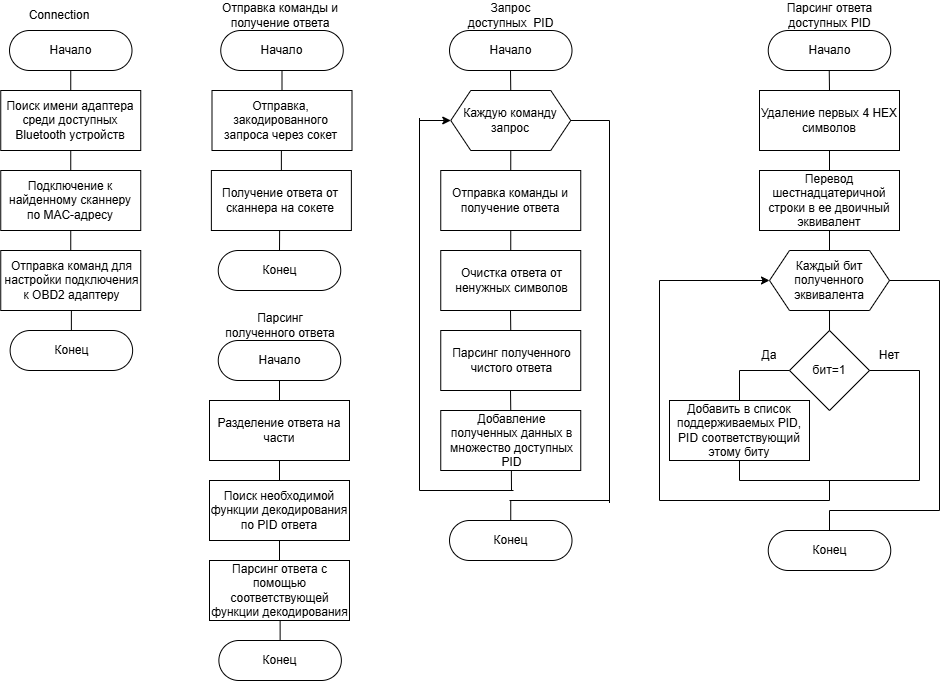


Рисунок 23 – Схемы определенных алгоритмов системы

## 2.7 Комплексное тестирование программной системы

# Заключение

В работе был проведен анализ процесса коммуникации с компьютером автомобиля, протоколов и технологий, задействованных в этом. В результате анализа, была выделена основная информация необходимая для понимания процесса коммуникации автомобиля с автомобильным сканнером, процесса коммуникации внутри самого автомобиля и протоколов, использованных в ней. Так же был проведен сравнительный анализ устройств, использующихся для коммуникации с автомобилем и выбрано оптимальное.

Был рассмотрен принцип шифрования и расшифровки команд и ответов, отправляемых автомобилю и получаемых от автомобиля соответственно, расшифровка кодов ошибок применяемых в OBD-II.

На основе проведенной научно-исследовательской работы был составлен черновой вариант технического задания на разработку “Программной системы сбора и обработки информации с компьютера автомобиля”.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Что такое ЭБУ в автомобиле [Электронный ресурс]. – URL: <https://auto.ru/mag/article/chto-takoe-block-upravleniya-v-avtomobile> (дата обращения: 15.09.2024)

2. Vehicle communication buses [Электронный ресурс]. – URL: <https://servicems.eu/en/news/post/854-Vehicle-communication-buses--FlexRay-CAN-LI.html> (дата обращения: 17.09.2024)

3. Introduction to LIN bus protocol [Электронный ресурс]. – URL: <https://www.csselectronics.com/pages/lin-bus-protocol-intro-basics> (дата обращения: 23.09.2024)

4. CAN bus explained [Электронный ресурс]. – URL: <https://www.csselectronics.com/pages/can-bus-simple-intro-tutorial> (дата обращения: 29.09.2024)

5. Exploring FlexRay Automotive communication bus [Электронный ресурс]. – URL: <https://www.logic-fruit.com/blog/flexray/flexray-automotive-communication/?srsltid=AfmBOoqsDFAsAsqkKNHHrenvdaaBLzPaC92B-ZNgA8Ck9dUd6CwFxXuL> (дата обращения: 01.10.2024)

6. LIN, CAN, FlexRay debug [Электронный ресурс]. – URL: <https://www.tek.com/en/documents/application-note/debugging-can-lin-and-flexray-automotive-buses-oscilloscope?anv=2> (дата обращения: 05.10.2024)

7. Датчики в автомобиле. Какие бывают и за что отвечают? [Электронный ресурс]. – URL: <https://koleso.ru/articles/datchiki-v-avtomobile-kakie-byivayut-i-za-chto-otvechayut/> (дата обращения: 08.10.2024)

8. Разъем OBD-II [Электронный ресурс]. – URL: <https://pinov.net/connector/OBD-2> (дата обращения: 13.10.2024)

9. OBD-II Protocols [Электронный ресурс]. – URL: <https://www.obdtester.com/obd2_protocols> (дата обращения: 15.10.2024)

10. OBD-II PID Overview [Электронный ресурс]. – URL: <https://www.csselectronics.com/pages/obd2-pid-table-on-board-diagnostics-j1979> (дата обращения: 18.10.2024)

11. Расшифровка диагностических кодов протокола OBD-II [Электронный ресурс]. – URL: <https://chiptuner.ru/content/obdcod/> (дата обращения: 21.10.2024)

12. ELM327 v1.5 datasheet [Электронный ресурс]. – URL: <https://www.alldatasheet.com/datasheet-pdf/pdf/542978/ELM/ELM327.html> (дата обращения: 07.11.2024)

13. ELM327 v2.1 datasheet [Электронный ресурс]. – URL: <https://www.electronicwings.com/components/elm327-obd2-v2.1-bluetooth-interface/1/datasheet> (дата обращения: 08.11.2024)