**­РЕФЕРАТ**

РПЗ 30 страница, 37 рисунков, 1 таблица, 19 источников, 1 приложение

ПРОТОКОЛ, АВТОМОБИЛЬ, СКАННЕР, ДАТЧИК

Объектом исследования являются данные, получаемые с компьютера автомобиля.

Цель работы – исследование 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

1 Выбор платформы для разрабатываемого продукта….………………………7

2 Сравнительный анализ существующих приложений по подбору одежды.....9

3 Принцип работы разрабатываемого приложения……………………………21

4 Выбор программной составляющей приложения…………….......................23

4.1 Выбор языка………………………………………………………………...23

4.2 Выбор архитектуры………………………………………………………...24

ЗАКЛЮЧЕНИЕ………………………………………………………………….28

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ……………………………….29

Приложение А (Черновик технического задания ВКРБ)……………………..31

**ОБОЗНАЧЕНИЯ, ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ**

БД? – база данных.

ОС – операционная система.

ООП – объектно-ориентированное программирование.

ПО – программное обеспечение.

Android – операционная система на ядре Linux.

Windows – операционная система для компьютеров от корпорации Microsoft.

ПК – персональный компьютер.

ПП – программный продукт.

ЭБУ – электронный блок управления.

**Введение**

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

Исследовательская работа посвящена анализу процесса общения между компьютером автомобиля и разрабатываемым ПО.

**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. Он также выполняет 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.

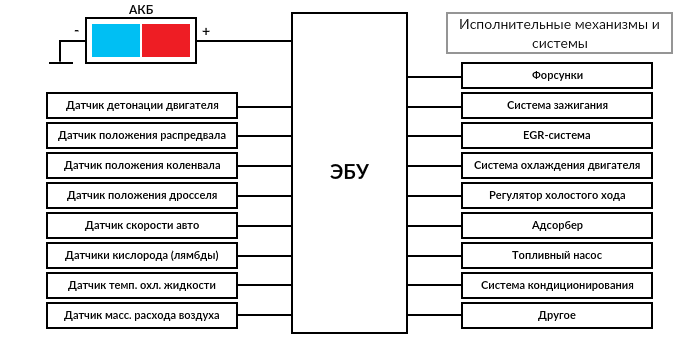


Рисунок 1 – Принцип действия ЭБУ двигателя.

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.

**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.

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

**2.1 Протокол LIN.**

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.

**2.2 Протокол CAN.**

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.

**2.3 Протокол FlexRay.**

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.

**2.4 Кадры данных рассматриваемых протоколов.**

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

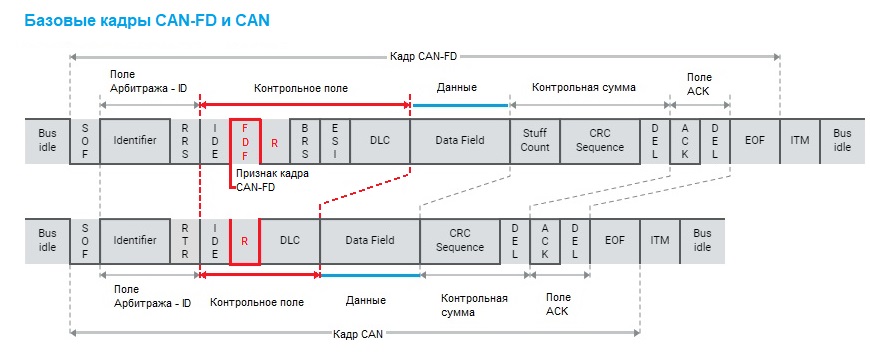


Рисунок 2 – Формат кадра данных CAN

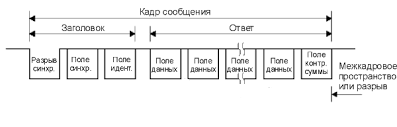


Рисунок 3 – Формат кадра данных LIN

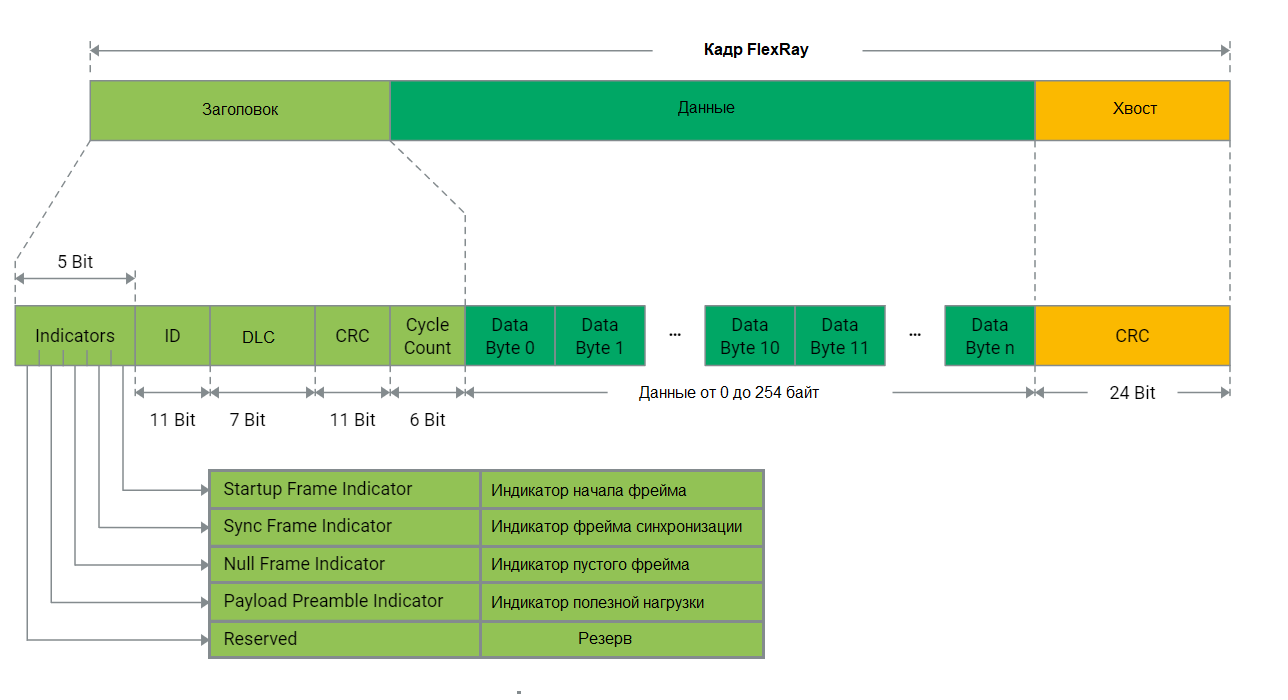


Рисунок 4 – Формат кадра данных Flex Ray

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.

**2.5 Сравнительный анализ.**

Проведем сравнительный анализ перечисленных протоколов, визуализированный в таблице 1:

Таблица 2 – 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.

**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);
* Регулятор 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 кол-ва поступающего воздуха в камеры сгорания смеси топлива, проблемы, связанные с этим датчиком, могут выражаться в том числе в повышенном расходе топлива, падении мощности двигателя.

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.

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.

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.

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

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

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.

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.

**4 OBD-II.**

**4.1 Назначение и структура.**

OBD-II – это стандарт который описывает тип разъема для диагностики и его распиновку, указанную на рисунке 11 а так же включающую в себя пин питания от автомобиля, для подключенного к нему сканнеру, 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.

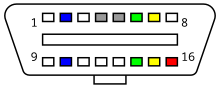
[](https://en.wikipedia.org/wiki/File:OBD-II_type_A_female_connector_pinout.svg)

Рисунок 11 – Распиновка OBD-II

Поясним распиновку, указанную на рисунке 11. 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.

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.

МБ НАПИСАТЬ КАКИЕ Я БУДУ ИСПОЛЬЗОВАТЬ И ЧТО ОНИ МОГУТ

**4.2 Стандарты OBD-II.**

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.

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.

Таким образом все распиновки протокола OBD-II используют тот же разъем, но разные контакты.

**4.3 Структура запросов и ответов OBD-II.**

**5 Выбор устройства для коммуникации. Elm327.**

**Заключение**

В работе был проведен сравнительный анализ мобильных приложений по подбору одежды. В результате сравнительного анализа, было выявлено, что большинство систем не учитывают погодные условия для выдачи рекомендаций по подбору одежды. Также на основе плюсов и минусов рассмотренных систем, были определены основные функции разрабатываемого приложения.

Были рассмотрены операционные системы и платформы, языки программирования и архитектуры. В результате чего были выбраны платформа и операционная система, на которой будет работать приложение, а также выбран язык программирования, на котором будет разрабатываться приложение и подобрана архитектура, наиболее подходящая по функциям системы.

На основе проведенной научно-исследовательской работы был составлен черновой вариант технического задания на разработку “Программного приложения для подбора одежды в соответствии с погодой”.

**Список используемых источников**