

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(национальный исследовательский университет)
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой
_____ Г.И. Радченко
« _____ » _____ 2018 г.

Программно-аппаратный комплекс получения информации с датчиков,
сигнализирующих о состоянии автомобиля

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ – 230100.2018.154 ПЗ ВКР

Руководитель работы
доцент каф. «Электронные
вычислительные машины»
_____ В.А. Парасич
« _____ » _____ 2018 г.

Автор работы
студент группы КЭ-484
_____ В. П. Вертушков
« _____ » _____ 2018 г.

Нормоконтролер,
ст. преп. каф. «Электронные
вычислительные машины»
_____ В.В. Лурье
« _____ » _____ 2018 г.

Челябинск, 2018

АННОТАЦИЯ

Вертушков В.П. Разработка программно-аппаратного комплекса получения информации с датчиков, сигнализирующих о состоянии автомобиля. – Челябинск: ЮУрГУ, ВШЭЖН; 2018, 107 с., 23 ил., библиогр. список – 9 наим., 1 прил.

В данной выпускной квалификационной работе выполнена разработка программно-аппаратного комплекса получения информации с датчиков, сигнализирующих о состоянии автомобиля. В ходе выполнения было произведено исследование принципа работы диагностики автомобиля, был проведен анализ существующих на рынке решений, проведен анализ технических средств. Разработана система передачи информации по Internet на мобильное устройство. Создано мобильное приложение отображающее информацию о состоянии сигнализирующих датчиков автомобиля.

Пояснительная записка включает в себя введение, оглавление, основную часть, заключение и список литературы.

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		3

Оглавление

ВВЕДЕНИЕ	6
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	8
1.1 Обзор существующих решений/аналогов	8
1.2 Достоинства и недостатки существующих решений/аналогов	11
1.3 Собственное предложение решение проблемы	11
1.4 Достоинства и недостатки предложенного решения	12
2 ПРОЕКТИРОВАНИЕ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА	13
2.1 Проектирование программной части	14
2.2 Проектирование аппаратной части	16
2.2.1 Анализ задачи	16
2.2.2 Выявление необходимого функционала	17
2.2.3 Подбор необходимых технологий	18
2.2.3.1 Микроконтроллер	18
2.2.3.2 Организация ввода-вывода информации в микроконтроллере	23
2.2.3.3 Организация удаленного доступа	24
2.2.3.4 Организация подключения к автомобилю	29
2.2.4 Структурное проектирование	34
2.3 Проектирование логики комплекса	36
2.4 Проектирование базы данных	36
3 РЕАЛИЗАЦИЯ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА	41
3.1 Реализация клиентского приложения	41
3.1.1 Реализация интерфейса	41
3.1.2 Реализация логики приложения	45
3.2 Реализация обработчика сигналов	54
3.3 Реализация логики сервера	56
4 ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ	59
4.1 Тест 1 – диагностика ошибок	59

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		4

4.2 Тест 2 – Температура жидкостей	60
4.3 Тест 3 - Уровни жидкостей	61
4.4 Тест 4 – Напряжение узлов	62
ЗАКЛЮЧЕНИЕ	64
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	65

					<i>ЮУрГУ-09.20.06.2018.154 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		5

ВВЕДЕНИЕ

Актуальность

С каждым годом количество автомобильного транспорта на улицах города увеличивается. Усложняется техническая часть авто. Для определения неисправности узлов и агрегатов необходимо обращаться в сервисные центры. Очень часто на это тратится большое количество времени. Диагностика в наше время также стоит больших денег. Предприятия, обладающие большим автопарком, теряют прибыль за то время, пока автомобиль находится на диагностике. Для решения этой проблемы предложено разработать программно-аппаратный комплекс получения информации с датчиков, сигнализирующих о состоянии автомобиля

Описание проблемы

Современные электронные системы, предназначенные для управления узлами и агрегатами автомобиля, оснащены так называемыми системами самодиагностики, которые информируют водителя о появлении некоторых неисправностей. Так, например, на приборном щитке многих автомобилей имеется многофункциональный индикатор — лампочка Check Engine, которая обычно загорается при включении зажигания и гаснет через некоторое время после запуска двигателя. Если же при самодиагностике обнаружатся неисправные компоненты, то индикатор не погаснет. В случае возникновения некоторых неисправностей во время движения индикатор также загорается, а при однократной мелкой неисправности он может и погаснуть (сохранив ошибку в памяти для последующего считывания), но если он продолжает гореть, то не удастся избежать немедленной остановки, более глубокой диагностики и ремонта.

Системы диагностики на разных автомобилях могут различаться, но принцип действия всех систем схож: блоком управления считываются показания датчиков на разных режимах работы в процессе эксплуатации автомобиля (запуск, прогрев, холостой ход, разгон и торможение и т.д.).

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

Показания датчиков бывают статическими или динамическими. Статические показания датчиков обычно определяются неким пороговым значением — импульсом определенного уровня или «переключателем», а динамические, как правило, передают изменения параметра и проверяются на допустимые диапазоны. Все диагностические системы хранят и отображают статические данные — «коды ошибок» и динамические характеристики.

Очень часто, в наше время, можно встретить объявления: «Компьютерная диагностика вашего автомобиля. За данную услугу водители согласны отдать круглую сумму. Особенно завораживает само словосочетание «компьютерная диагностика», поскольку большинство людей под словом «компьютер» подразумевают что-то крайне сложное и умное, обладающее непререкаемым авторитетом. Впрочем, и многие вполне грамотные и неплохо знакомые с компьютером люди легко покупаются на подобные предложения.

Компании, обладающие большим автопарком, теряют внушительную часть прибыли на диагностику неисправностей своих автомобилей. В связи с этим, было принято решение разработать комплекс получения информации с датчиков, сигнализирующих о состоянии автомобиля. Информация должна обрабатываться в программном коде и представляться в обработанном виде на мобильном устройстве клиента. Это поможет избежать временных и финансовых издержек.

					<i>ЮУрГУ-09.20.06.2018.154 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		7

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Обзор существующих решений/аналогов

Существует несколько программ для вывода информации об автомобиле в мобильное приложение. Каждая обладает определенным набором функций.

Наиболее распространённой и функциональной программой является Torque [7]. Основными возможностями программы являются:

- функция чтения и стирания ошибок двигателя (Check Engine) из ЭБУ автомобиля;
- проведение полной диагностики двигателя автомобиля;
- функция отображения текущих параметров различных датчиков автомобиля в режиме реального времени (скорость, давление, обороты, температуры жидкостей и воздуха и многое другое);
- возможность сохранения динамических параметров автомобиля с возможностью последующего анализа;
- возможность добавления до 7 настраиваемых экранов с отображением различной информации с датчиков или устройств автомобиля;
- функция вывода текущей информации в виде цифровых значений, графиков, циферблатов;
- функция сохранения маршрута движения автомобиля.

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8



Рисунок 1. Демонстрация работы программы Torque

При использовании Torque информация может отображаться тремя способами – с имитацией аналогового прибора со стрелкой (Dial), в виде графика, а также в виде значения в цифровой форме (Display). При этом в режимах Dial/Display отображается текущее значение параметра, а график позволяет отследить динамику его изменения.

Вторым по популярности диагностическим приложением является DashCommand[8]. Программа обладает схожим функционалом:

- функция чтения и стирания ошибок двигателя (Check Engine) из ЭБУ автомобиля;
- проведение полной диагностики двигателя автомобиля;
- функция отображения текущих параметров различных датчиков автомобиля в режиме реального времени (скорость, давление, обороты, температуры жидкостей и воздуха и многое другое);
- возможность сохранения динамических параметров автомобиля с возможностью последующего анализа;
- функция вывода текущей информации в виде цифровых значений, графиков, циферблатов;

Данная программа представлена только для платформы IOS. В отличие от Torque, работающей по Bluetooth, DashCommand использует Wi-Fi протокол для передачи данных.



Рисунок 2. Демонстрация работы программы DashCommand (начальная форма)

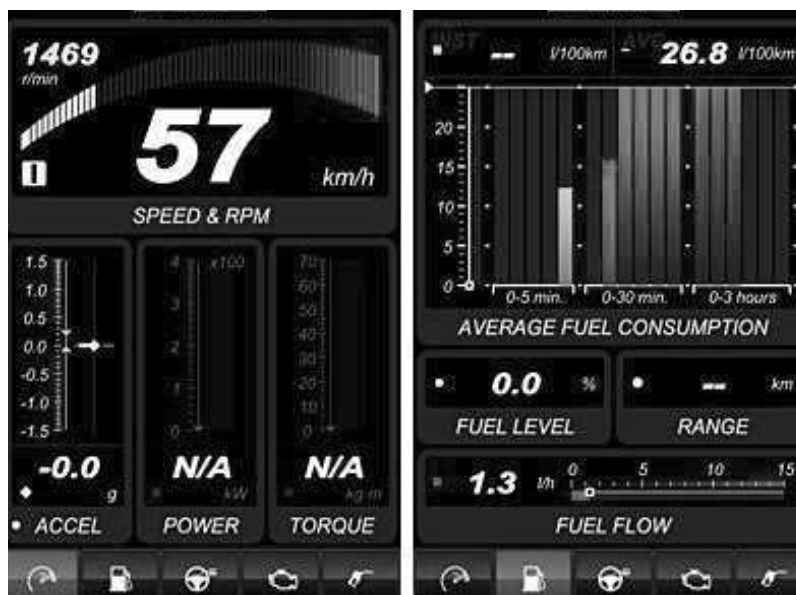


Рисунок 3. Демонстрация работы программы DashCommand (формы с выводом информации)

1.2 Достоинства и недостатки существующих решений/аналогов

Основными достоинствами Torque и DashCommand являются простота и доступность. Интерфейс данных программ создан для людей, которые не обладают специализированными знаниями. Приложения для платформы Android являются бесплатными. Функционала данных приложений хватает для диагностики большинства автомобилей. Отсюда высокая популярность данных продуктов.

У данных систем так же есть недостатки. Первым является то, что для работы нужно покупать непосредственно сканер для диагностики автомобиля, с которым будет связываться приложение. Вторым недостатком состоит в том, что системы работают при помощи протокола беспроводной связи Bluetooth. Следовательно, для получения информации нужно находиться на расстоянии не более 100 метров. Третьим минусом товаров представленных на рынке является то, что они не многопользовательские. При владении несколькими авто пользователю придется все время переключаться между ними, что является явным недостатком.

1.3 Собственное предложение решения проблемы

Для решения проблемы диагностики автомобилей предлагается разработка «Программно-аппаратный комплекс получения информации с датчиков, сигнализирующих о состоянии автомобиля. Информация, полученная с CAN-шины, будет анализироваться сервисом, и отправляться на мобильное устройство». Передача информации будет осуществлена с помощью Internet. Будет два вида интерфейса мобильного приложения: многопользовательское, однопользовательское.

В качестве входных параметров управления – температура ДВС, температура КПП, уровни рабочих жидкостей, информация о состоянии датчиков.

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

1.4 Достоинства и недостатки предложенного решения

Достоинствами является работа комплекса по протоколу Internet, что позволяет получать информацию об автомобиле, не находясь в непосредственной близости, приложение имеет возможность получать данные с нескольких автомобилей, что облегчает работу с автопарком.

Недостатки – для установки аппаратной части требуется работа специалиста, при отсутствии у пользователя интернета, невозможно получать информацию об автомобиле.

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

2 ПРОЕКТИРОВАНИЕ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА

Структура программно-аппаратного комплекса состоит из трех частей: анализатора ошибок, веб-приложения и мобильного приложения.

Анализатором ошибок является микроконтроллер, установленный в автомобиле и подключенный к диагностическому разъему. Его задачей является считывание данных, формирование пакетов данных и отправка этих пакетов на сервер по Internet. Отправка данных по Internet позволит пользователю не находиться рядом с автомобилем для получения информации о его состоянии. Это позволит отслеживать информацию о состоянии каждого автомобиля в автопарке, не тратя лишнего времени.

Веб-приложение служит для принятия данных с обработчика, хранения последних актуальных данных об автомобиле, а так же для отправки данных на устройство клиента. Веб-приложение принимает данные с обработчика и записывает их в БД. При запросе клиента оно отправляет данные на мобильное устройство.

Мобильное приложение является пользовательской частью комплекса. Оно нужно для отображения данных на устройстве клиента. Вся информация об автомобиле, содержащаяся на сервере, будет выводиться на экране мобильного телефона или планшета при запросе клиента.

На рисунке 4 представлена компонентная структура системы.

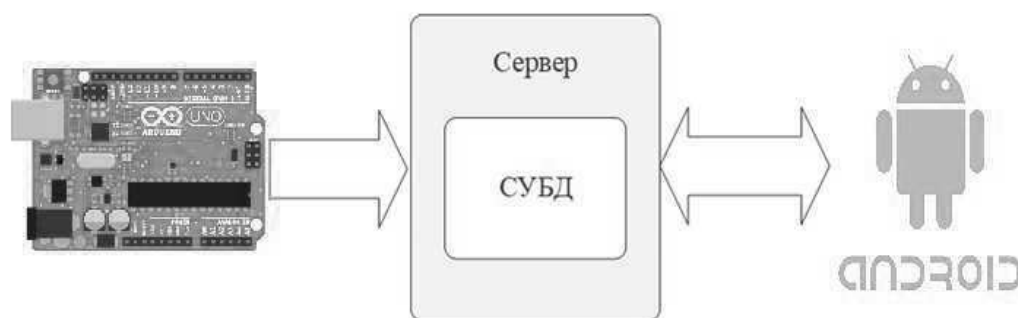


Рисунок 4. Компонентная архитектура системы

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		13

Ключевой этап разработки комплекса – проектирование. Для моего продукта я выделил следующие пункты:

- проектирование программной части
- проектирование аппаратной части
- проектирование логики программно-аппаратного комплекса
- проектирование базы данных

2.1 Проектирование программной части

Программная часть состоит из:

- интерфейса мобильного приложения;
- логики программы, осуществляющей передачу информации между автомобилем и пользовательским приложением, посредством клиент-серверной архитектуры. Сервера, получающего информацию и записывающего ее в БД, клиента, обрабатывающего информацию о состоянии автомобиля и отправляющего ее на сервер, клиента, получающего информацию и отображающего её на экране мобильного приложения;
- базы данных PostgreSQL Database, которая отвечает за хранение учетных записей об автомобилях и их владельцах.

Самой главной частью в программной подсистеме является веб-приложение.

На рисунке 5 представлена функциональная схема взаимодействия элементов.

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14



Рисунок 5. Функциональна схема

Данная схема отображает работу системы, где главным управляющим элементом является Веб-приложение.

Из функциональной схемы видно, что в центре системы стоит Веб-приложение, которое взаимодействует со всеми элементами системы. Данные приходящие с микроконтроллера, обрабатываются веб-приложением и посылаются на хранение в базу данных. Пользователи могут взаимодействовать со своими устройствами через клиентские приложения.

Для отображения информации необходимо разработать приложение для смартфонов, которое могло бы использовать Internet связь для принятия информации с сервера, и отображало бы эту информацию на экране.

В настоящее время на рынке наиболее распространены три операционные системы:

- Android;
- Windows Phone;
- iOS.

В данной работе было принято решение написания приложения для системы Android. Выбор был обусловлен тем, что смартфоны с данной системой являются самыми популярными на рынке.

Приложение должно отображать весь функционал комплекса. В результате разработки приложение должно выполнять следующие функции:

- вывод на экран ошибок автомобиля;
- вывод на экран температуры рабочих жидкостей;
- вывод на экран уровней рабочих жидкостей;
- вывод на экран напряжения узлов автомобиля.

2.2 Проектирование аппаратной части

2.2.1 Анализ задачи

Самая главная часть в системе обработчик сигналов, который собирает информацию о состоянии узлов и агрегатов.

Обработчик сигналов состоит из внутренних и внешних компонентов (модулей, датчиков). На рисунке 6 представлена структурная схема аппаратной части системы.

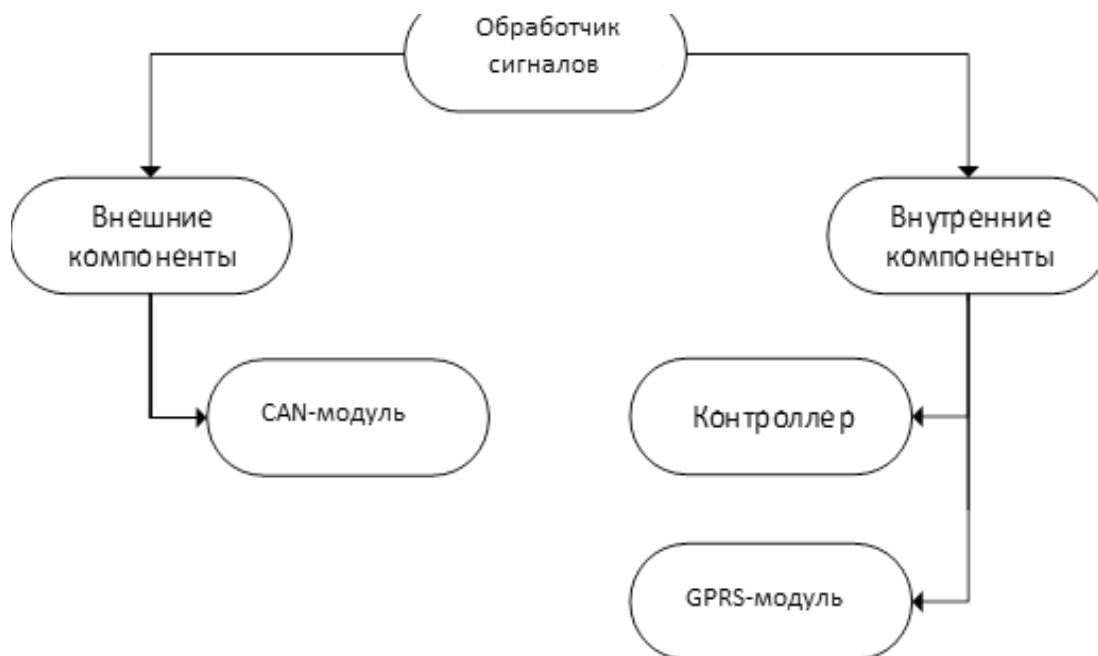


Рисунок 6. Структурная схема

2.2.2 Выявление необходимого функционала

Система должна поддерживать:

- Получение информации об ошибках работы узлов и агрегатов автомобиля:
 - система корректировки топливоподачи, смесь слишком бедная;
 - система корректировки топливоподачи, смесь слишком богатая;
 - обнаружены множественные пропуски воспламенения;
 - датчик детонации, неисправность цепи;
 - датчик положения коленчатого вала, неисправность цепи;
 - ошибка импульса датчика положения коленчатого вала;
 - датчик положения коленчатого вала, нет сигнала;
 - датчик положения распределительного вала, отсутствует сигнал;
 - неисправность цепи 1 и 4 управления зажиганием;
 - неисправность цепи 2 и 3 управления зажиганием;
 - отсутствует сигнал скорости автомобиля (только с МКПП);
 - неисправность цепи переключателя положения дроссельной заслонки;
 - ошибка записи контроллера электронной системы управления двигателем;
 - неисправность цепи выходного сигнала уровня топлива;
 - общая неисправность контроллера КПП, нарушение работы коробки передач (только с АКПП);
 - отсутствует сигнал скорости автомобиля (только с АКПП);
 - неисправность цепи привода заряда на холостом ходу;
 - механическая ошибка привода заряда на холостом ходу;
 - отсутствует сообщение контроллера КПП (только с АКПП);
 - бедная смесь в системе корректировки топливоподачи;

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		17

- богатая смесь в системе корректировки топливоподачи;
 - обнаружен пропуск зажигания;
 - работоспособность модуля датчика детонации;
 - рециркуляция отработавших газов, недостаточный расход;
 - рециркуляция отработавших газов, избыточный расход;
 - бедная смесь в режиме полной нагрузки;
 - отклонение сигнала датчика скорости колеса с ABS;
 - отсутствие сигнала от датчика скорости колеса с ABS.
- Получение информации о температуре рабочих жидкостей автомобиля:
 - температура масла ДВС;
 - температура масла КПП.
 - Получение информации об уровне рабочих жидкостей автомобиля:
 - топливо;
 - масло ДВС;
 - масло КПП;
 - тормозная жидкость.
 - Получение информации о напряжении аккумулятора и генератора.

2.2.3 Подбор необходимых технологий

2.2.3.1 Микроконтроллер

«В данной работе будет рассмотрен микроконтроллер фирмы Arduino модель Uno Rev3 на базе процессора ATmega328p с тактовой частотой 16 МГц, обладающий памятью 32 кБ и 20 контролируемых контактов ввода и вывода для взаимодействия с внешним миром. ATmega328p имеет следующие

					<i>ЮУрГУ-09.20.06.2018.154 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		18

характеристики:

- 8-разрядный высокопроизводительный МК с низким энергопотреблением;
- 32 Кбайт внутрисистемно-программируемой Flash-памяти;
- 1024 байт EEPROM;
- 2 Кбайт встроенной SRAM;
- периферия "на борту":
 - два 8-разрядных таймера/счетчика;
 - один 16-разрядный таймер/счетчик;
 - счетчик реального времени с отдельным генератором;
 - 5 каналов ШИМ (PWM);
 - 6-канальный АЦП;
 - программируемый последовательный USART;
 - последовательный интерфейс SPI;
 - поддержка байт-ориентированного обмена по двухпроводной последовательной шине (совместима с I2C);
 - программируемый сторожевой таймер (WATCHDOG) с отдельным встроенным генератором;
 - аналоговый компаратор;
 - прерывание и пробуждение при изменении состояния выводов;
 - схема сброса при включении питания и детектор кратковременных пропаданий питания;
 - встроенный откалиброванный генератор; о внешние и внутренние источники прерывания;
 - 5 режимов пониженного потребления: Idle, ADC Noise Reduction, Power-Save, Power-down и Standby;
- напряжение питания: 1,8 - 5,5 В;
- тактовая частота 0 - 20 МГц;
- температурный диапазон -40..85С;

					<i>ЮУрГУ-09.20.06.2018.154 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		19

- корпус PDIP-28;

Arduino — это открытая платформа, которая позволяет собирать всевозможные электронные устройства. Устройства могут работать как автономно, так и в связке с компьютером. Платформа состоит из аппаратной и программной частей; обе чрезвычайно гибки и просты в использовании. Все платы Arduino содержат основные компоненты, необходимые для программирования и совместной работы с другими схемами (рис. 7):

- микроконтроллер Atmel;
- USB-интерфейс для программирования и передачи данных;
- стабилизатор напряжения и выводы питания;
- контакты входов ввода-вывода; индикаторные светодиоды (Debug, Power, Rx, Tx);
- кнопку сброса;
- встроенный последовательный интерфейс программирования (ICSP).

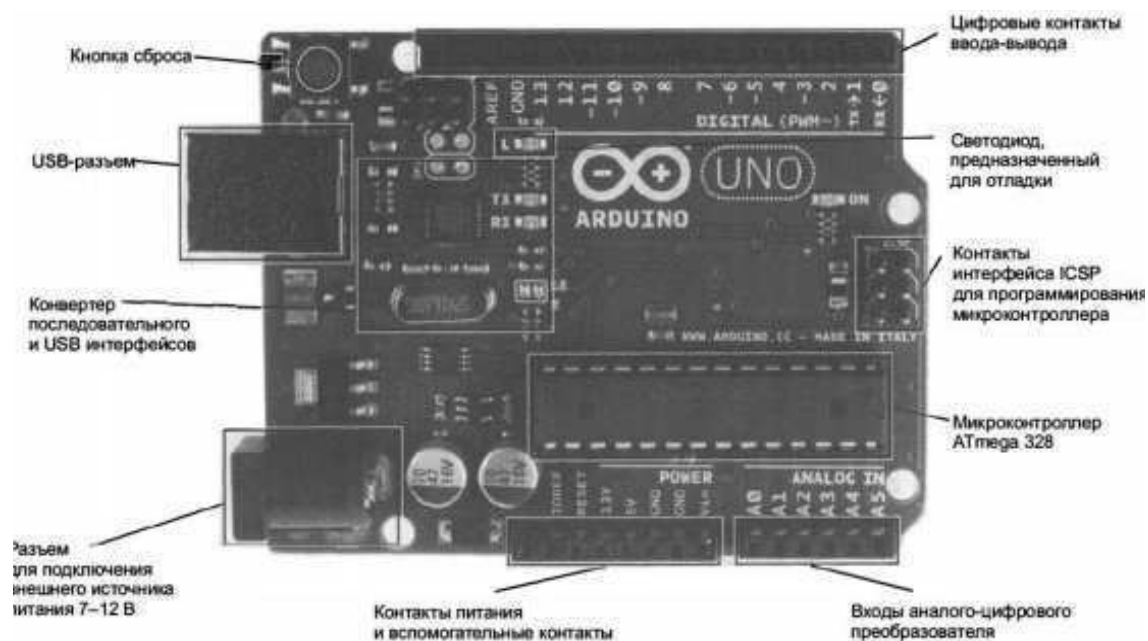


Рисунок 7. Компоненты платы Arduino Uno

Микроконтроллеры Atmel Основной элемент платы Arduino - микроконтроллер Atmel. На большинстве плат Arduino, включая Arduino Uno, установлен микроконтроллер ATmega. На плате Arduino Uno, изображенной на рис. 1.1, вы видите микроконтроллер ATmega 328. Исключением является

плата Due, укомплектованная микроконтроллером ARM Cortex. Микроконтроллер исполняет весь скомпилированный код программы. Язык Arduino предоставляет доступ к периферийным устройствам микроконтроллера:

- аналого-цифровым преобразователям (ADCs);
- цифровым портам ввода-вывода;
- коммуникационным шинам (включая I2c и SPI) и последовательным интерфейсам.

На плате все эти порты выведены на штырьковые контакты. К тактовым контактам микроконтроллера ATmega подключен кварцевый резонатор 16 МГц. На плате имеется входной контакт Reset. Его установка в логический ноль приводит к сбросу процессора. Это аналог кнопки Reset обычного компьютера.

Для программирования используется упрощённая версия C++, известная так же как Wiring. Разработку можно вести как с использованием бесплатной среды Arduino IDE, так и с помощью произвольного C/C++ инструментария. Поддерживаются операционные системы Windows, MacOS X и Linux.

Для программирования и общения с компьютером вам понадобится USB-кабель.

Arduino Uno может питаться как от USB подключения, так и от внешнего источника: батарейки или обычной электрической сети. Источник определяется автоматически. Платформа может работать при наличии напряжения от 6 до 20В. Однако при напряжении менее 7 В работа может быть неустойчивой, а напряжение более 12 В может привести к перегреву и повреждению. Поэтому рекомендуемый диапазон: 7–12 В. На Arduino доступны следующие контакты для доступа к питанию:

- Vin предоставляет тот же вольтаж, что используется для питания платформы. При подключении через USB будет равен 5 В;
- 5V предоставляет 5В вне зависимости от входного напряжения. На этом

напряжении работает процессор. Максимальный допустимый ток, получаемый с этого контакта — 800 мА;

- 3.3V предоставляет 3,3 В. Максимальный допустимый ток, получаемый с этого контакта — 50 мА;
- GND — земля.

Платформа оснащена 32 кБ flash-памяти, 2 кБ из которых отведено под так называемый bootloader. Он позволяет прошивать Arduino с обычного компьютера через USB. Эта память постоянна и не предназначена для изменения по ходу работы устройства. Её предназначение — хранение программы и сопутствующих статичных ресурсов. Также имеется 2 кБ SRAM- памяти, которые используются для хранения временных данных вроде переменных программы. По сути, это оперативная память платформы. SRAM- память очищается при обесточивании. Ещё имеется 1 кБ EEPROM-памяти для долговременного хранения данных. По своему назначению это аналог жёсткого диска для Arduino.

На платформе расположены 14 контактов (pins), которые могут быть использованы для цифрового ввода и вывода. Какую роль исполняет каждый контакт, зависит от вашей программы. Все они работают с напряжением 5 В, и рассчитаны на ток до 40 мА. Также каждый контакт имеет встроенный, но отключённый по умолчанию резистор на 20 - 50 кОм.

Помимо контактов цифрового ввода/вывода на Arduino имеется 6 контактов аналогового ввода, каждый из которых предоставляет разрешение в 1024 градации. По умолчанию значение меряется между землёй и 5 В, однако возможно изменить верхнюю границу, подав напряжение требуемой величины на специальный контакт AREF. Данный микроконтроллер подходит для выполнения поставленных задач, так как очень универсальный и выполняет ряд требований:

- низкая стоимость;
- высокая надежность;

					<i>ЮУрГУ-09.20.06.2018.154 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		22

- высокая степень миниатюризации;
- малое энергопотребление;
- работоспособность в жестких условиях эксплуатации;
- достаточная производительность для выполнения всех требуемых функций.

2.2.3.2 Организация ввода-вывода информации в микроконтроллере

В наше время, время высоких технологий и больших возможностей, существует большое количества способов взаимодействия с микроконтроллером, начиная от обычной кнопки и заканчивая сканером отпечатка пальца. Но не зависимо от вида ввода или вывода информации связь осуществляется либо цифровая, либо аналоговая. В микроконтроллере, который рассматривается в данной работе, имеется контакты, как для цифровой, так и для аналоговой связи. Все контакты могут служить цифровыми входами и выходами. Часть контактов Arduino могут также действовать в качестве аналоговых входов. Многие из контактов работают в режиме мультиплексирования и выполняют дополнительные функции: различные коммуникационные интерфейсы, последовательные интерфейсы, широтно-импульсные модуляторы и внешние прерывания. Цифровые контакты принимают сигналы в виде разных уровней напряжений, где логическая единица находится в промежутке напряжений от 2,7 – 5 Вольт, логический ноль от 0 – 0,5 Вольт. Промежуток напряжений от 0,5 – 2,7 Вольт соответствует неопределенному состоянию, то есть никто не может предсказать к какому уровню перейдет сигнал. Некоторые контакты обладают дополнительными ролями:

- Serial: 0-й и 1-й. Используются для приёма и передачи данных по USB;
- внешнее прерывание: 2-й и 3-й. Эти контакты могут быть настроены так, что они будут провоцировать вызов заданной функции при изменении

					<i>ЮУрГУ-09.20.06.2018.154 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		23

входного сигнала;

- PWM: 3-й, 5-й, 6-й, 9-й, 10-й и 11-й. Могут являться выходами с широтно-импульсной модуляцией (pulse-width modulation) с 256 градациями;
- LED: 13-й. К этому контакту подключен встроенный в плату светодиод. Если на контакт выводится 5 В, светодиод зажигается; при нуле — светодиод гаснет.

Микроконтроллеры ATmega, используемые в Arduino, содержат шестиканальный аналого-цифровой преобразователь (АЦП). Разрешение преобразователя составляет 10 бит, что позволяет на выходе получать значения от 0 до 1023. Основным применением аналоговых входов большинства платформ Arduino является чтение аналоговых датчиков, но в тоже время они имеют функциональность вводов/выводов широкого применения (GPIO) (то же, что и цифровые порты ввода/вывода 0 - 13). Таким образом, при необходимости применения дополнительных портов ввода/вывода имеется возможность сконфигурировать неиспользуемые аналоговые входы.

2.2.3.3 Организация удаленного доступа

Чтобы обеспечить удаленный доступ по сети Internet микроконтроллер должен быть подключен к Internet. Для решения данной цели есть два пути решения: установки дополнительного модуля называемый "Ethernet шилд", который обеспечит проводной выход в Internet и подключения GPRS модуля SIM900.

Ethernet Shield основан на чипе W51000, который имеет внутренний буфер на 16К. Скорость подключения достигает 10/100Мб. Шилд работает с использованием библиотеки Arduino Ethernet library, которая по умолчанию интегрирована в оболочку Arduino IDE.

На Ethernet шилде есть слот для установки micro SD карты, с помощью которой можно хранить большие массивы информации и загружать веб-сайты

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		24

непосредственно с Arduino. Необходимо помнить, что в этом случае надо использовать дополнительную библиотеку.

Кроме того, можно обеспечить питание Arduino с помощью Ethernet соединения. Для этого надо использовать Power over Ethernet (PoE) модуль, место для установки есть на Ethernet шилде.

Сводный список технических характеристик Ethernet шилда:

- для работы необходима плата Arduino;
- рабочее питание – 5 В (питается от платы Arduino);
- Ethernet контроллер: W5100 с буфером 16Кб;
- скорость подключения: 10/100Мб;
- подключается к Arduino через SPI порт.

Модуль SIM900 используется в различных автоматизированных системах. С помощью интерфейса UART осуществляется обмен данными с другими устройствами. Модуль обеспечивает возможность совершения звонков, обмен текстовыми сообщениями. Работа модуля реализуется на компоненте SIM900, созданным фирмой SIMCom Wireless Solution.

Технические характеристики:

- Диапазон напряжений 4,8-5,2В;
- В обычном режиме ток достигает 450 мА, максимальный ток в импульсном режиме 2 А;
- Поддержка 2G;
- Мощность передачи: 1 Вт 1800 и 1900 МГц, 2 Вт 850 и 900 МГц;
- Имеются встроенные протоколы TCP и UDP;
- GPRS multi-slot class 10/8;
- Питание 5В. Потребление тока: обычный режим 50–450 мА, в спящем режиме 1,5 мА, предельный импульсный 2 А;
- Рабочая температура от -30С до 75С.

Компоненты управления:

					<i>ЮУрГУ-09.20.06.2018.154 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		25

Переключателем Power select устанавливается источник питания: внешний, подключенный к коаксиальному соединителю или источник питания микроконтроллерного модуля Arduino.

Кнопкой Power key включается или выключается питание при нажатии и удерживании в течение 2 с.

Индикация:

О состоянии модуля сообщают 3 светодиода:

- PWR (зеленый) – индикатор питания модуля,
- Status (красный) – индикатор питания компонента SIM900,
- Net Light (зеленый) – соединение с сетью.
- Сообщения светодиода Net Light.
- Выключен – SIM900 не работает.

Контакты:

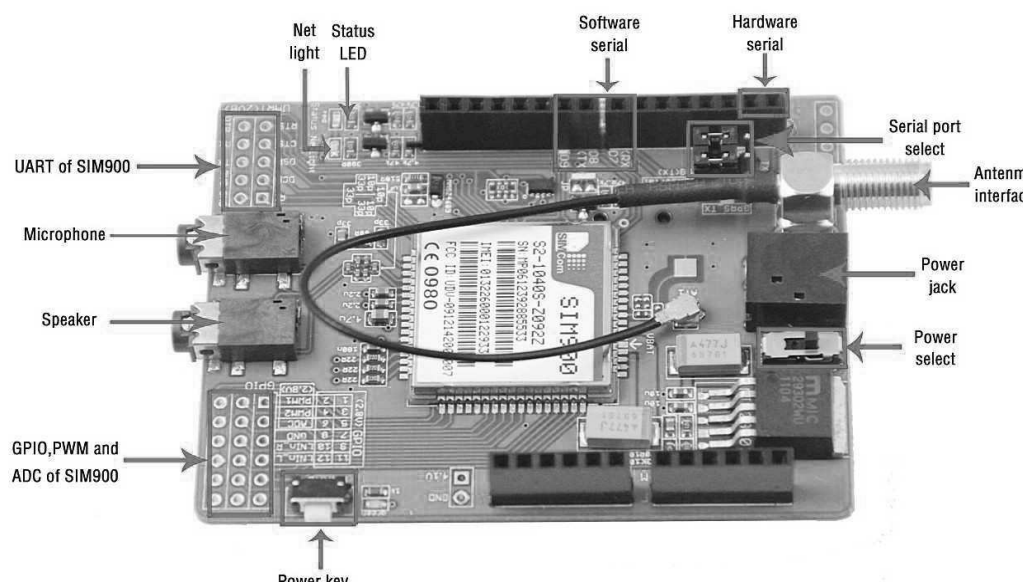


Рисунок 8. Компоненты платы SIM900

Компонент SIM900 содержит порт UART, его сигналы выведены на выводы компонента и соединены с перемычками, устанавливающими с

какими контактами модуля Shield GPRS/GSM SIM900 будет соединен порт UART компонента SIM900 с D0, D1 или D7, D8.

UART Shield GPRS/GSM можно подключить: к аппаратному интерфейсу МК через контакты TXD и RXD модуля Shield GPRS/GSM, для этого используются D0, D1. Или к программно эмулируемому средствами ардуино, для этого используются контакты D7 и D8 модуля Shield GPRS/GSM. Полный интерфейс UART имеет 10 сигналов выведенных на контакты в углу платы: R1, DCD, DSR, CTS, RTS, GND, 2V8, TX, RX, DTR.

12 подписанных контактов цифровых линий ввода-вывода GPIO расположены в углу платы. Имеется 2 контакта выходных сигналов с широтно-импульсной модуляцией PWM1, PWM2. Вход АЦП контакт ADC. Интерфейс встроенного счетчика времени имеет 4 контакта. Обозначение контактов: DISP_CLK, DISP_DATA, DISP_D/C, DISP_CS.

Контакт D9 используется для программного контроля включения или выключения SIM900.

На плате установлен соединитель для подключения антенны.

На нижней стороне модуля размещен разъем для Sim карты. Для установки Sim карты необходимо перевести фиксатор в положение OPEN и поднять крышку. В нее установить Sim карту, закрыть крышку и перевести ее в положение LOCK. Установить с помощью перемычек способ обмена данными UART. Подключите антенну. Соедините Shield GPRS/GSM SIM900 с основным прибором.

					<i>ЮУрГУ-09.20.06.2018.154 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		27



Рисунок 9 . Нижняя сторона платы SIM900

Перемычки:

Возле антенного гнезда расположены перемычки J11 и J12. С их помощью задаются контакты используемые интерфейсом UART компонента SIM900: TX, RX или D7, D8. Контакты TX, RX предназначены для подключения к интерфейсу UART являющемуся периферией МК. Благодаря программной эмуляции средствами ардуино интерфейса UART через контакты D7, D8 может происходить обмен сигналами в соответствии с требованиями UART, если аппаратный UART занят. Установку перемычек следует выполнять, ориентируясь на надписи возле контактов перемычек.

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		28

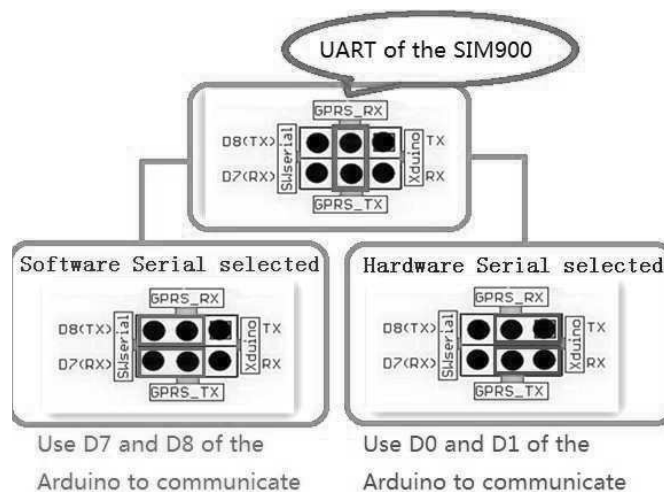


Рисунок 10. Перемычки платы SIM900

2.2.3.4 Организация подключения к автомобилю

Для подключения к автомобилю микроконтроллера Arduino Uno используется CAN-BUS Shield. Связь происходит при помощи интерфейсов DB9, расположенного на плате, и OBD-II, являющемся диагностическим разъемом автомобиля.

CAN-BUS Shield – это плата расширения для Arduino контроллеров, которая добавляет в проект CAN интерфейс - обще индустриальную шину передачи данных на длинные расстояния со средней скоростью и высокой надежностью. Плата основана на CAN контроллере с SPI интерфейсом MCP2515 и CAN трансивере MCP2551. Использование совместно с кабелем стандарта OBD-II позволит вам создать диагностический прибор или регистратор данных для автомобиля.

Характеристики CAN-BUS Shield:

- интерфейс SPI до 10 МГц;
- стандартный (11 бит) и расширенный (29 бит) данных и удаленных кадров;
- 2 приемных буфера с приоритетным хранением сообщений;

- 9-ти контактный разъем;
- выключатель питания;
- 2 светодиодных индикатора;
- 2 дополнительных разъема: UART и SPI.

Устройство CAN-BUS Shield:

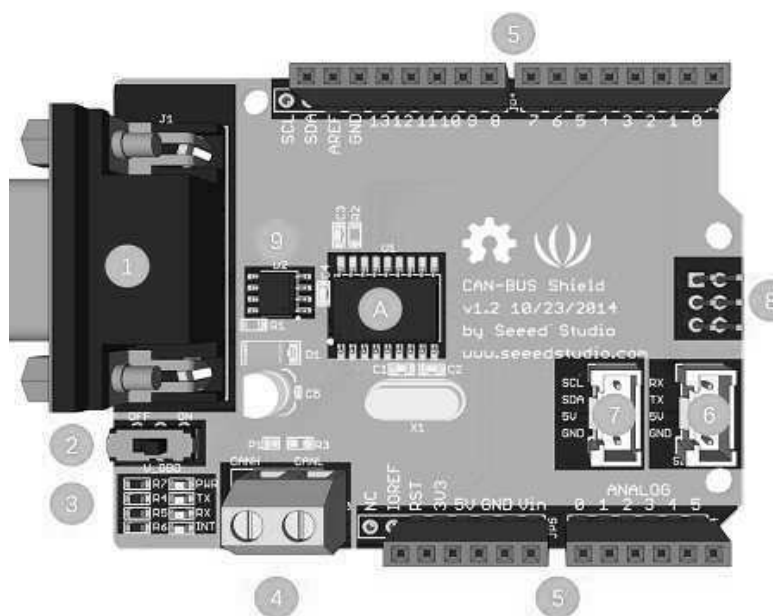


Рисунок 11. Компоненты платы CAN-BUS Shield

1. Интерфейс DB9 - для подключения к интерфейсу OBDII через кабель DBG-OBD.
2. V_OBD - он получает питание от интерфейса OBDII (от DB9)
3. Светодиодный индикатор:
 - PWR: мощность
 - TX: мигает, когда данные отправляются
 - RX: мигает при получении данных
 - INT: прерывание данных
4. Терминал - CAN_H и CAN_L
5. Вывод Arduino UNO
6. Разъем Serial Grove

7. Разъем I2C Grove
8. Выводы ICSP
9. IC - MCP2551, высокоскоростной CAN-трансивер
10. IC - MCP2515, автономный CAN-контроллер с интерфейсом SPI

Интерфейс DB9 и OBD-II:

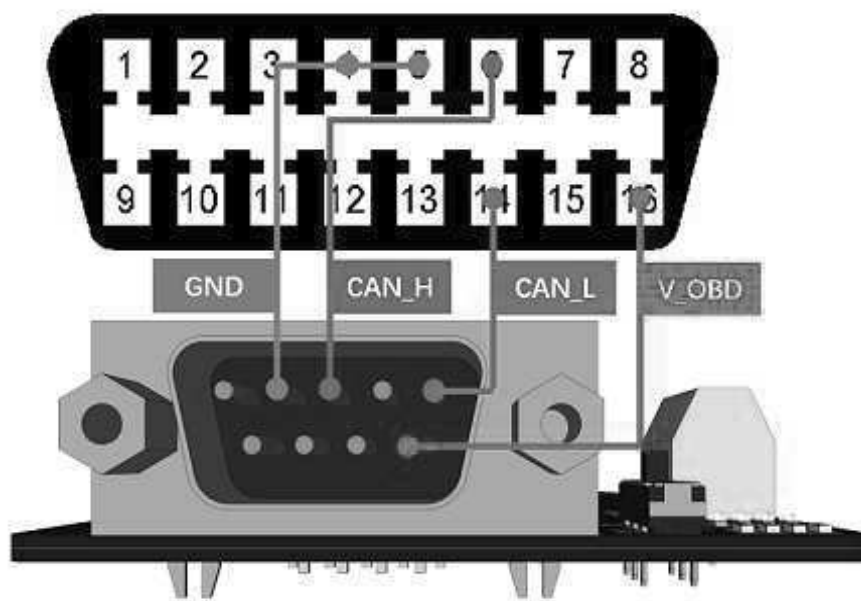


Рисунок 12. Соединение интерфейсов DB9 и OBD-II

- 2 пин интерфейса DB9 соединён с 4 и 5 пинами OBD-II. Соединены земли интерфейсов;
- 3 пин интерфейса DB9 соединён с 6 пином OBD-II. Линия шины CAN_H (высокая);
- 5 пин интерфейса DB9 соединён с 14 пином OBD-II. Линия шины CAN_L (низкая);
- 9 пин интерфейса DB9 соединён с 16 пином OBD-II. Индикация входящего вызова;

Устройство CAN-шины:

Промышленная сеть реального времени CAN представляет собой сеть с

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		31

общей средой передачи данных и состоит из узлов с собственными тактовыми генераторами, например, приборная панель или подсистема определения температуры в автомобиле (Рисунок 13). Любой узел сети CAN посылает сообщение по сети и каждый из узлов системы решает, относится ли к нему это сообщение. Для решения этой задачи в CAN имеется аппаратная реализация фильтрации сообщений. CAN контроллеры соединяются с помощью дифференциальной шины, которая имеет две линии - CAN_H (Can-High) и CAN_L (Can-Low), по которым передаются сигналы.

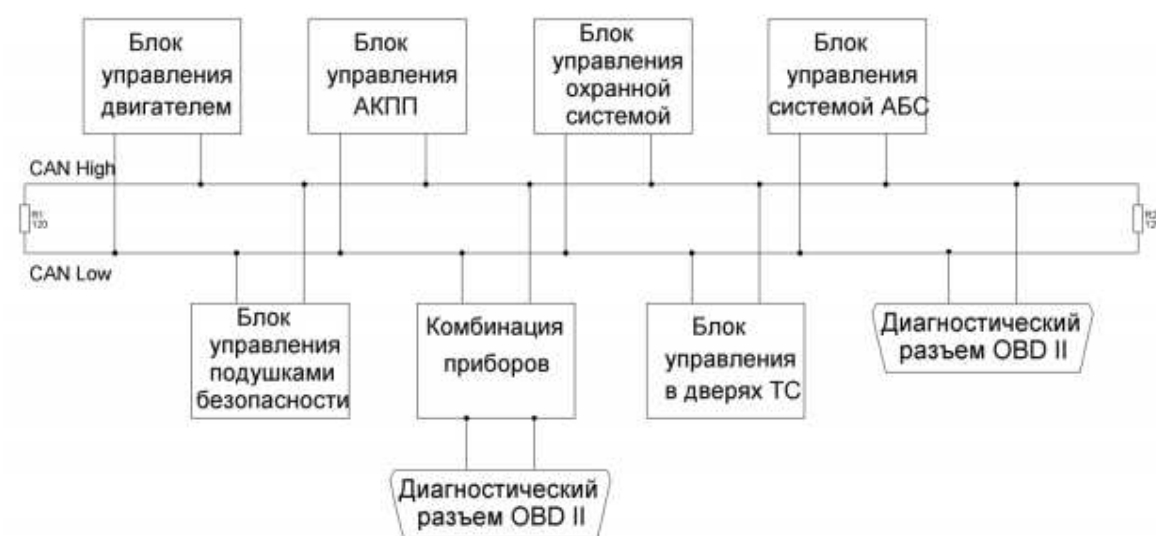


Рисунок 13. Соединение интерфейсов DB9 и OBD-II

Протокол CAN-шины реализован в двух версиях: версия А задает 11-битную идентификацию сообщений (т. е. в системе может быть 2048 сообщений), версия В — 29-битную (536 млн. сообщений).

Режимы работы:

1. Режим J1939. Данный режим предназначен для получения CAN-сообщений, передаваемых по протоколу J1939. Поддерживаются скорости от 10000 бит/с до 500000 бит/с (типовые значения: 62500, 12500, 250000, 500000). Поддерживаются 11-и и 29-и битные идентификаторы.

Для работы в этом режиме необходимо настройка параметров скорости шины и время задержки (время ожидания сообщения).

Сообщения получаются в следующем виде:

29 битные идентификаторы выводятся в следующем формате:

-«00000009 8 01 02 03 04 05 06 07 08», где «00000009» – идентификатор сообщения; «8» – количество принятых байт из шины; «01 02 03 04 05 06 07 08» – сообщение из восьми байт (слева младший байт, справа старший).

11 битные идентификаторы выводятся в виде:

-«009 8 01 02 03 04 05 06 07 08», где «009» – идентификатор сообщения; «8» – количество принятых байт из шины; 01 02 03 04 05 06 07 08 – сообщение из восьми байт (слева младший байт, справа старший).

2. Режим FMS – стандартный фильтр протокола J1939. В случае если производитель ТС (в основном это производители большегрузной техники, сельхоз оборудования) поддерживает стандарт FMS, выбор этого режима позволяет автоматически извлекать и расшифровывать сообщения, соответствующие стандарту FMS:

3. общий расход топлива - количество израсходованного топлива с момента создания ТС;

4. уровень топлива в баке, измеряется в процентах. 0% - пустой. 100% - полный;

5. температура охлаждающей жидкости;

6. обороты двигателя;

7. общий пробег.

Как правило, J1939 режимы используются, если данных, получаемых по стандарту FMS, недостаточно, либо стандарт FMS не поддерживается.

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		33

2.2.4 Структурное проектирование

На рисунке 14 представлена функциональная схема взаимодействия элементов. Проектируемое устройство расположено в рамке.

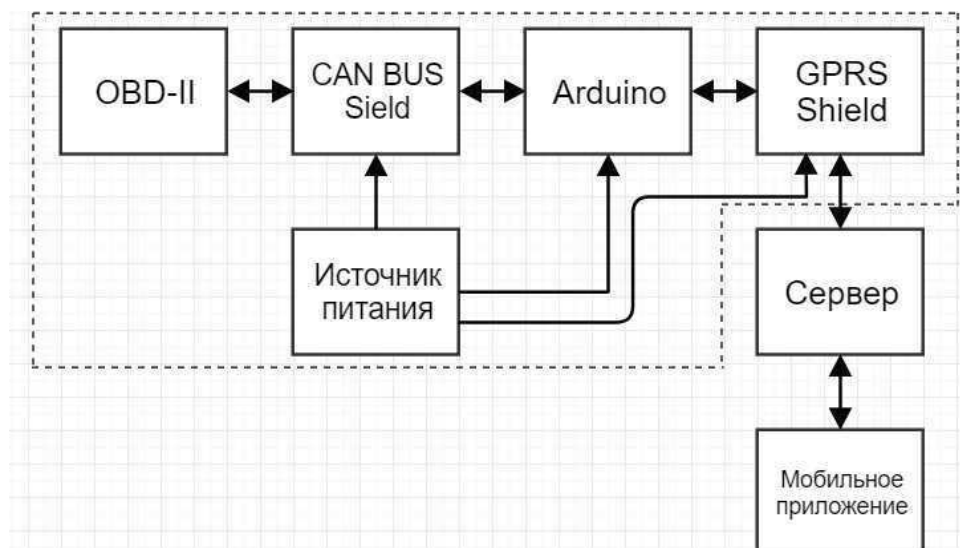


Рисунок 14. Функциональна схема

Данная схема отображает работу системы, где главным управляющим элементом является контроллер Arduino Uno.

На контроллер поступают данные с автомобиля при помощи CAN BUS shield. Посредством GPRS модуля, контроллер взаимодействует с сервисом, как на отправку данных, так и на прием.

Следующим этапом является разработка принципиальной электрической схемы.

Arduino Uno может быть запитан от USB либо от внешнего источника питания - тип источника выбирается автоматически. В качестве внешнего источника питания может использоваться сетевой AC/DC-адаптер или аккумулятор/батарея. Штекер адаптера (диаметр – 2.1мм) необходимо вставить в соответствующий разъем питания на плате.

Рассмотрев схему электрическую принципиальную Arduino Uno, видно, что устройство имеет стабилизатор напряжения LM1117-5, который дает на выходе стабилизированные 5В постоянного тока.

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		34

CAN BUS shield и GPRS модуль SIM900 питаются от источника питания постоянного тока 5В. На плате Arduino Uno имеется порт с питанием 5В. Следовательно, соединив данный выход с Arduino Uno с входами питания CAN BUS shield и GPRS модуля, мы решим проблему питания.

CAN BUS shield соединен с микроконтроллером Arduino Uno контактами D0/RX и D1/TX, для приема и передачи информации. По данным контактам передается информация, полученная с CAN-шины в режиме J1939.

GPRS модуль SIM900 соединяется с микроконтроллером Arduino Uno посредством программного USART, используя пины D7 и D8.

Результат разработки схемы электрической принципиальной устройства представлен на рисунке 15.

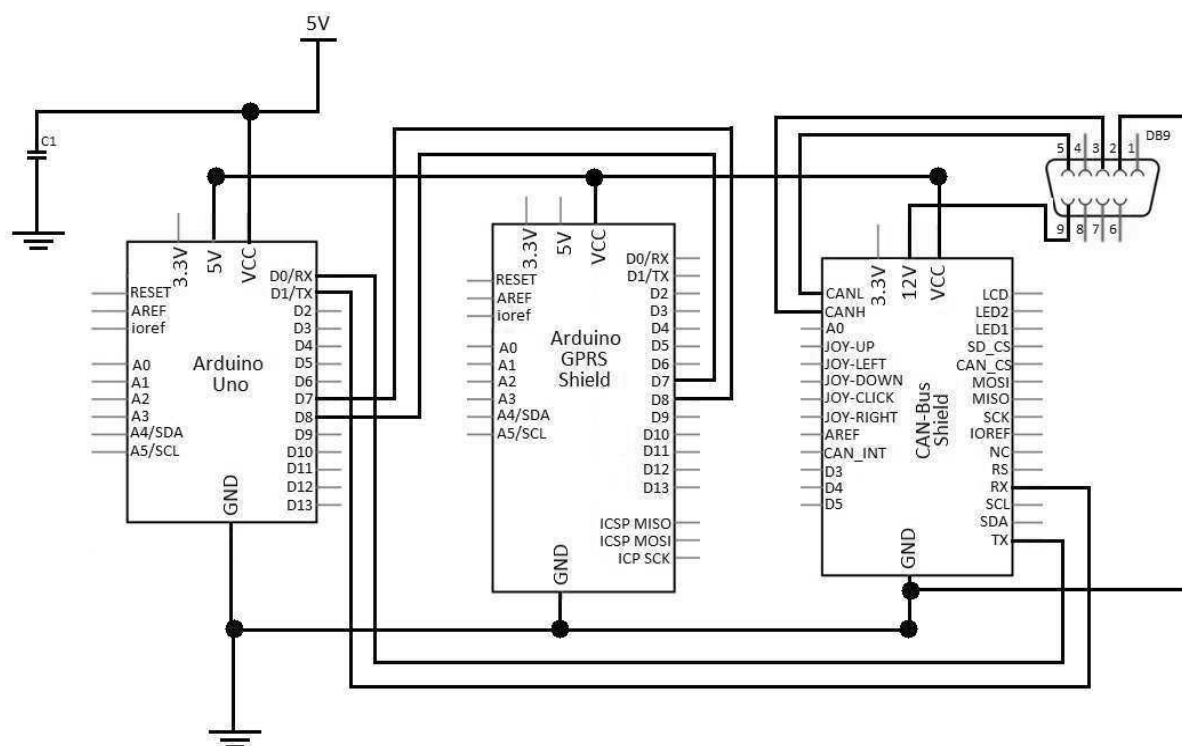


Рисунок 15 – Принципиальная электрическая схема

2.3 Проектирование логики комплекса

Методы и средства разработки:

- считывание данных обеспечивает AVR микроконтроллер, запрограммированный на языке C++;
- для разработки серверной части используется платформа .Net язык программирования C# и система WCF;
- для хранения данных будет использоваться PostgreSQL Database;
- для разработки клиентской части мобильного приложения будет использоваться язык программирования JAVA.

Логика программно-аппаратного комплекса включает в себя следующие функции:

- функция получения информации об ошибках работы узлов и агрегатов автомобиля;
- функция получения информации о температуре рабочих жидкостей автомобиля;
- функция получения информации об уровне рабочих жидкостей автомобиля;
- функция получения информации о напряжении аккумулятора и генератора.

2.4 Проектирование базы данных

Для функционирования приложения необходимо создание базы данных. Она отвечает за хранение учетных записей об автомобилях и их владельцах.

В таблице auto хранится следующая информация: состояние датчиков

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		36

сигнализирующих о состоянии автомобиля, температуры рабочих жидкостей, уровни рабочих жидкостей, напряжение узлов. Таблица user хранит информацию о пользователях. Таблица autouser является связующей. Она предназначена для реализации связи «многие ко многим».

Скриншот схемы базы данных представлен на рисунке 16.

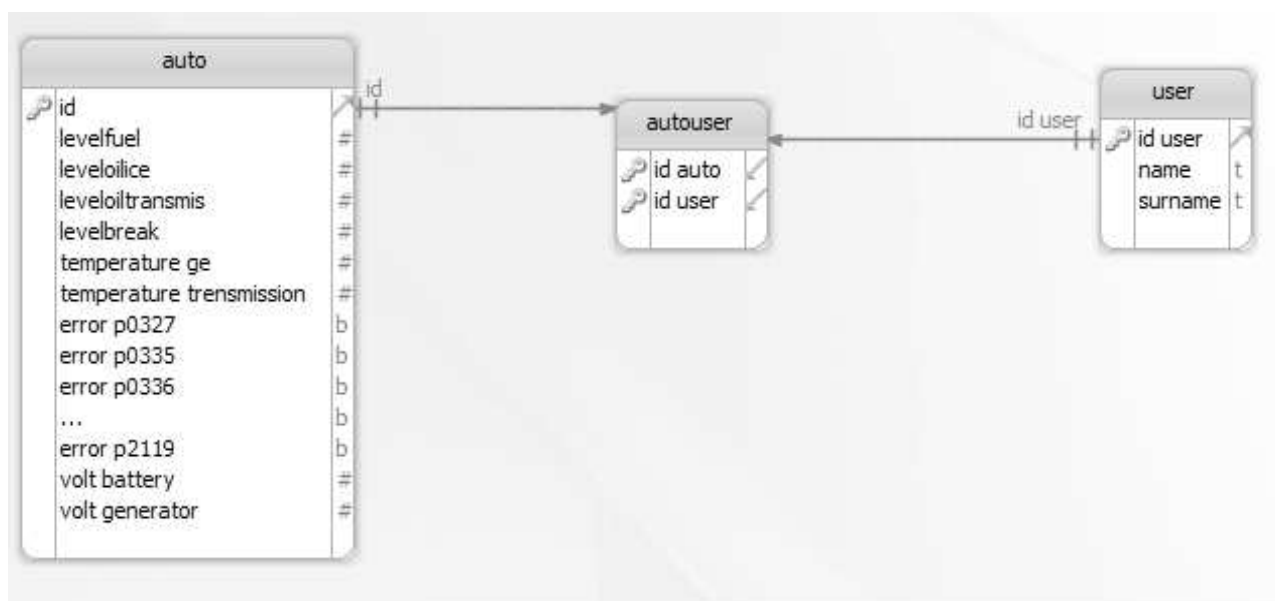


Рисунок 16 – Схема базы данных

Таблица auto содержит следующие поля:

- «id» тип integer – поле хранящее информацию об уникальном идентификаторе автомобиля. Является ключевым полем таблицы;
- «levelfuel» тип integer – поле хранящее информацию об уровне топлива в процентах;
- «levelOilICE» тип integer – поле хранящее информацию об уровне масла ДВС в процентах;
- «levelOilTransmis» тип integer – поле хранящее информацию об уровне масла КПП в процентах;
- «levelBreak» тип integer – поле хранящее информацию об уровне тормозной жидкости в процентах;

- «temperature ge» тип float - поле хранящее информацию о температуре ДВС в градусах Цельсия с точностью до десятых;
- «temperature transmission» тип float - поле хранящее информацию о температуре КПП в градусах Цельсия с точностью до десятых;
- «error p0327»- «error p2119» тип bool - поля хранящие информацию об ошибках автомобиля(1 – ошибка присутствует, 0 – ошибка отсутствует):
 - «error p0327» - датчик детонации, неисправность цепи;
 - «error p0335» - датчик положения коленчатого вала, неисправность цепи;
 - «error p0336» - ошибка импульса датчика положения коленчатого вала;
 - «error p0337» - датчик положения коленчатого вала, нет сигнала;
 - «error p0338» - система корректировки топливоподдачи, смесь слишком бедная;
 - «error p0339» - система корректировки топливоподдачи, смесь слишком богатая;
 - «error p0340» - обнаружены множественные пропуски воспламенения;
 - «error p0342» - датчик положения распределительного вала, отсутствует сигнал;
 - «error p0344» - работоспособность модуля датчика детонации;
 - «error p0351» - неисправность цепи 1 и 4 управления зажиганием;
 - «error p0352» - неисправность цепи 2 и 3 управления зажиганием;
 - «error p0400» - обнаружен пропуск зажигания;
 - «error p0401» - рециркуляция отработавших газов, недостаточный расход;
 - «error p0402» - рециркуляция отработавших газов, избыточный расход;
 - «error p0471» - бедная смесь в системе корректировки топливоподдачи;

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		38

- «error p0472» - богатая смесь в системе корректировки топливоподачи;
- «error p0501» - отсутствует сигнал скорости автомобиля (только с МКПП);
- «error p0510» - неисправность цепи переключателя положения дроссельной заслонки;
- «error p0605» - ошибка записи контроллера электронной системы управления двигателем;
- «error p0656» - неисправность цепи выходного сигнала уровня топлива;
- «error p0700» - общая неисправность контроллера КПП, нарушение работы коробки передач (только с АКПП);
- «error p0710» - отсутствует сообщение контроллера КПП (только с АКПП);
- «error p1504» - отсутствует сигнал скорости автомобиля (только с АКПП);
- «error p2066» - бедная смесь в режиме полной нагрузки;
- «error p2096» - отклонение сигнала датчика скорости колеса с ABS;
- «error p2097» - отсутствие сигнала от датчика скорости колеса с ABS;
- «error p2101» - неисправность цепи привода заряда на холостом ходу;
- «error p2118» - механическая ошибка привода заряда на холостом ходу.
- «volt battery» тип float – поле хранящее информацию о напряжении на аккумуляторе в вольтах с точностью до десятых;
- «volt generator» тип float – поле хранящее информацию о напряжении на генераторе в вольтах с точностью до десятых.

Таблица user содержит следующие поля:

					<i>ЮУрГУ-09.20.06.2018.154 ПЗ ВКР</i>	<i>Лист</i>
						39
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

- «id user» тип integer – поле хранящее информацию об уникальном идентификаторе пользователя. Является ключевым полем таблицы;
- «name» тип char – поле хранящее информацию об имени пользователя;
- «surname» тип char – поле хранящее информацию фамилии пользователя;

Таблица user содержит следующие поля:

- «id auto» тип integer – поле хранящее информацию об уникальном идентификаторе автомобиля;
- «id user» тип integer – поле хранящее информацию об уникальном идентификаторе пользователя;

Поля являются составным ключом.

					<i>ЮУрГУ-09.20.06.2018.154 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		40

3 РЕАЛИЗАЦИЯ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА

3.1 Реализация клиентского приложения

3.1.1 Реализация интерфейса

Интерфейс приложения написан на языке XML.

При запуске приложения, открывается главный экран. Предлагается выбрать автомобиль из списка автомобилей клиента. При нажатии на выбранный автомобиль, программа открывает пользовательскую форму приложения с функционалом.

Для отображения информации, создаются объекты. Для текстовых полей создается объект типа TextView.



Рисунок 17 – Стартовая форма

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		41


```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"

  <TextView
    android:id="@+id/textViewStart"                //уникальное имя
    android:text="Выберите автомобиль"
    android:textColor="#000"                      //цвет текста
    android:textSize="25sp"                       //размер текста
    android:layout_width="400dp"                  //высота объекта
    android:layout_height="35dp"                 //ширина объекта
    android:layout_marginTop="10dp"               //отступ от верхнего края
    android:layout_marginLeft="25dp"             //отступ от левого края
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"/>

  <Button
    android:id="@+id/buttonCar"                   //уникальное имя
    android:text="Рено логан"
    android:layout_width="120dp"                 //высота объекта
    android:layout_height="35dp"                //ширина объекта
    android:layout_marginTop="76dp"              //отступ от верхнего края
    android:layout_marginLeft="76dp"            //отступ от левого края
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"/>

</android.support.constraint.ConstraintLayout>

```

Листинг 1 – стартовая страница.

Далее открывается форма с функционалом, на которой расположены кнопки получения информации о текущем состоянии датчиков, уровне жидкостей, температуре жидкостей, напряжении на узлах. При нажатии на каждую из кнопок, открывается форма с информацией. Также, на каждой форме, кроме начальной, есть кнопка «назад», которая открывает предыдущую форму.

Кнопки для открытия функционала созданы так же, при помощи объектов типа Button. Кнопка назад является объектом типа ImageButton. Данный тип нужен для создания кнопок с изображением.

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		42



Рисунок 18 – Форма с выбором функционала

```

<ImageButton
    android:id="@+id/imageBack1" //уникальное имя
    android:layout_width="wrap_content" //ширина объекта
    android:layout_height="wrap_content" //высота объекта
    app:srcCompat="@android:drawable/ic_menu_revert" //имя ресурса
    android:layout_marginTop="5dp" //отступ от верхнего края
    android:layout_marginLeft="5dp" //отступ от левого края
    tools:layout_editor_absoluteX="2dp"
    tools:layout_editor_absoluteY="0dp" />

<Button
    android:id="@+id/buttonError" //уникальное имя
    android:layout_width="131dp" //высота объекта
    android:layout_height="105dp" //ширина объекта
    android:text="диагностика ошибок"
    android:layout_marginTop="76dp" //отступ от верхнего края
    android:layout_marginLeft="50dp" //отступ от левого края
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"/>

```

Листинг 2 - Описание кнопки «назад» и кнопки выбора функциональной задачи.

После нажатия на кнопку открывается окно с информацией, запрошенной пользователем. При необходимости выводится графическое отображение.

Графика создается при помощи объекта типа ImageView. Для того что бы

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		43

нужное изображение поместить в форму, нужно добавить это изображение в папку resources. Для этого правой клавишей нужно нажать на папку, выбрать пункт «добавить ресурс» и прописать путь до нужного изображения. После создания объекта типа ImageView в параметрах указать имя нужного изображения.



Рисунок 19 – Форма с выводом информации, на примере информации об уровнях жидкостей

```

<TextView
    android:id="@+id/textFuel" //уникальное имя
    android:layout_width="0dp" //высота объекта
    android:layout_height="35dp" //ширина объекта
    android:layout_marginTop="76dp" //отступ от верхнего края
    android:textSize="20sp" //размер текста
    android:textColor="#000" //цвет текста
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<ImageView
    android:id="@+id/imageViewFuel1" //уникальное имя
    android:layout_width="177dp" //высота объекта
    android:layout_height="144dp" //ширина объекта
    android:layout_marginTop="16dp" //отступ от верхнего края
    android:visibility="invisible" //видимость
  
```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		44

```

app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/sp1" //имя ресурса
/>

```

Листинг 3 – Вывод информации и графического отображения на дисплей мобильного устройства.

Таким образом реализован интерфейс приложения.

3.1.2 Реализация логики приложения

Логика приложения написана на языке Java.

Приложение состоит из шести рабочих окон: выбор автомобиля, меню, диагностика ошибок, уровни жидкостей, температуры жидкостей, напряжение узлов. Для каждого окна написан код на языке Java. Каждое окно представляет собой новый класс, содержащий программный код.

1. Выбор автомобиля – «public class ChoiceCar»

Для установки соответствия данного класса с XML версткой используется функция «setContentView(R.layout.activity_choicocar)». Данная функция устанавливает соединение с файлом «activity_choicocar», в котором описан интерфейс окна.

В теле класса описываются функциональные кнопки выбора автомобиля, из списка подключенных автомобилей к данному мобильному устройству. Данные кнопки имеют одинаковый функционал. Описание работы кнопок на примере кнопки «Рено Логан»:

- Кнопка «Рено Логан» - buttonError.

Функция «findViewById(R.id.button_car1)», находит в файле XML нужный элемент по уникальному имени «button_car1». Данный элемент ставится в соответствие элементу buttonCar1. После чего с помощью функции

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
						45
Изм.	Лист	№ докум.	Подпись	Дата		

«setOnClickListener» проверяется условие нажатия кнопки. Если кнопка нажата, то выполняется код тела функции. Переменной «IDcar» присваивается уникальный идентификатор, по которому будет производиться поиск автомобиля в БД при запросе. После этого создается новый элемент класса Intent. Класс Intent является встроенным классом, и нужен для описания одной операции. В этом случае данный класс используется для прихода на следующее окно при нажатии кнопки «Рено Logan». Для запуска нового окна необходимо создать экземпляр класса Intent и указать в первом параметре текущий класс, а во втором класс для перехода, в данном случае это «ActivityMenu». После этого вызывается метод startActivity(), который и запускает новый экран.

```
buttonCar1 = (Button) findViewById(R.id.button_car1);
buttonCar1.setOnClickListener(new View.OnClickListener() {
    public void onClick(View view) {
        IDcar = 1;
        Intent intent = new Intent(getApplicationContext(), ActivityMenu.class);
        startActivity(intent);
    }
});
```

Листинг 4 – Проверка условия нажатия кнопки «Рено Logan».

2. Меню – «public class ActivityMenu»

Для установки соответствия данного класса с XML версткой используется функция «setContentView(R.layout.activity_menu)». Данная функция устанавливает соединение с файлом «activity_menu», в котором описан интерфейс окна.

В теле класса описываются четыре функциональные кнопки:

- Кнопка «диагностика ошибок» - buttonError.

Функция «findViewById(R.id.button_error)», находит в файле XML нужный элемент по уникальному имени «button_error». Данный элемент ставится в соответствие элементу buttonError. После чего с помощью функции «setOnClickListener» проверяется условие нажатия кнопки. Если кнопка нажата, то выполняется код тела функции. В теле функции создается новый элемент класса Intent. Класс Intent является встроенным классом, и нужен для описания

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		46

одной операции. В этом случае данный класс используется для прихода на следующее окно при нажатии кнопки «Диагностика ошибок». Для запуска нового окна необходимо создать экземпляр класса Intent и указать в первом параметре текущий класс, а во втором класс для перехода, в данном случае это «ActivityError». После этого вызывается метод startActivity(), который и запускает новый экран.

```
buttonError = (Button) findViewById(R.id.button_error);
buttonError.setOnClickListener(new View.OnClickListener() {
    public void onClick(View view) {
        Intent intent = new Intent(getApplicationContext(),
        ActivityError.class);
        startActivity(intent);
    }
});
```

Листинг 5 – Проверка условия нажатия кнопки «Диагностика ошибок».

- Кнопка «Уровни жидкостей» - buttonLevel.

По аналогии с предыдущей кнопкой функция findViewById(R.id.button_level), находит в файле XML нужный элемент по уникальному имени «button_level». Данный элемент ставится в соответствие элементу buttonLevel. После чего с помощью функции setOnClickListener проверяется условие нажатия кнопки. Если кнопка нажата, то выполняется код тела функции. В теле функции создается новый элемент класса Intent. Указываются параметры: текущее окно и класс для перехода, в данном случае это «ActivityLevel». После этого вызывается метод startActivity(), который и запускает новый экран.

```
buttonLevel = (Button) findViewById(R.id.button_level);
buttonLevel.setOnClickListener(new View.OnClickListener() {
    public void onClick(View view) {
        Intent intent = new Intent(getApplicationContext(),
        ActivityLevel.class);
        startActivity(intent);
    }
});
```

Листинг 6 – Проверка условия нажатия кнопки «Уровни жидкостей».

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		47

- Кнопка «Температура жидкостей» - `buttonTemperature`.

По аналогии с предыдущей кнопкой функция `findViewById(R.id.button_temperature)`, находит в файле XML нужный элемент по уникальному имени «`button_temperature`». Данный элемент ставится в соответствие элементу `buttonTemperature`. После чего с помощью функции `setOnClickListener` проверяется условие нажатия кнопки. Если кнопка нажата, то выполняется код тела функции. В теле функции создается новый элемент класса `Intent`. Указываются параметры: текущее окно и класс для перехода, в данном случае это «`ActivityTemperature`». После этого вызывается метод `startActivity()`, который и запускает новый экран.

```
buttonTemperature = (Button) findViewById(R.id.button_temperature);
buttonTemperature.setOnClickListener(new View.OnClickListener() {
    public void onClick(View view) {
        Intent intent = new Intent(getApplicationContext(),
ActivityTemperature.class);
        startActivity(intent);
    }
});
```

Листинг 7 – Проверка условия нажатия кнопки «Температура жидкостей».

- Кнопка «Напряжение узлов» - `buttonVolt`.

По аналогии с предыдущей кнопкой функция `findViewById(R.id.button_volt)`, находит в файле XML нужный элемент по уникальному имени «`button_volt`». Данный элемент ставится в соответствие элементу `buttonVolt`. После чего с помощью функции `setOnClickListener` проверяется условие нажатия кнопки. Если кнопка нажата, то выполняется код тела функции. В теле функции создается новый элемент класса `Intent`. Указываются параметры: текущее окно и класс для перехода, в данном случае это «`ActivityVolt`». После этого вызывается метод `startActivity()`, который и запускает новый экран.

```
buttonVolt = (Button) findViewById(R.id.button_volt);
buttonVolt.setOnClickListener(new View.OnClickListener() {
    public void onClick(View view) {
        Intent intent = new Intent(getApplicationContext(), ActivityVolt.class);
        startActivity(intent);
    }
});
```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
						48
Изм.	Лист	№ докум.	Подпись	Дата		

```
});
```

Листинг 8 – Проверка условия нажатия кнопки «Напряжение узлов».

3. Диагностика ошибок – public class ActivityError.

Для установки соответствия данного класса с XML версткой используется функция `setContentView(R.layout.activity_error)`. Данная функция устанавливает соединение с файлом «*activity_error*», в котором описан интерфейс окна.

Функция «`findViewById(R.id.text_error1)`», находит в файле XML нужный элемент текстового поля по уникальному имени «*text_error1*». По аналогии находятся следующие элементы «*text_error2*» - «*text_error26*» Данные элементы ставятся в соответствие элементам `textError1` – `textError26`.

В теле класса описываются принятие данных с сервера и вывод их на экран устройства.

Функция `Void doInBackground()` принимает данные с сервера. Для этого в теле функции формируется GET запрос «`HttpGet get = new HttpGet(linc)`», где `linc` – это локальная переменная, содержащая GET-запрос на диагностику ошибок.

```
String linc = "http://wcfrestservice222.azurewebsites.net/Service1/ErrorOpen/";  
HttpGet get = new HttpGet(linc);
```

Листинг 9 – GET-запрос на диагностику ошибок.

Далее данные полученные с сервера в формате XML десериализуются с помощью встроенной функции «`DocumentBuilderFactory`». Каждая ошибка записывается в соответствующую переменную «`textResultError1`» - «`textResultError26`».

В конце текстовые поля содержащие информацию выводятся в формах на экране. Данное действие выполняется с помощью функции «`setText()`», параметром которой является текстовая переменная «`textResultError1`» - «`textResultError26`».

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		49

Функция «void Visiability(ImageView a)» необходима для появления изображения значка предупреждения при возникновении ошибки. Изначально изображение невидимо, но при возникновении ошибки, в теле функции выполняется действие «a.setVisibility(View.VISIBLE)», которое включает видимость объекта в нужной строке.

4. Уровни жидкостей – public class ActivityLevel.

Для установки соответствия данного класса с XML версткой используется функция «setContentView(R.layout.activity_level)». Данная функция устанавливает соединение с файлом «activity_level», в котором описан интерфейс окна.

Функция «findViewById(R.id.text_levelFuel)», находит в файле XML нужный элемент текстового поля по уникальному имени «text_levelFuel». По аналогии находятся следующие элементы «text_levelOilICE», «text_levelOilTracm», «text_levelBreak». Данные элементы ставятся в соответствие элементам «textLevelFuel», «textlevelOilICE», «textlevelOilTracm», «textlevelBreak», соответственно.

В теле класса описываются принятие данных с сервера и вывод их на экран устройства.

Функция Void doInBackground() принимает данные с сервера. Для этого в теле функции формируется GET запрос «HttpGet get = new HttpGet(linc)», где linc – это локальная переменная, содержащая GET-запрос на уровни жидкостей.

```
String linc = "http://wcfrestservice222.azurewebsites.net/Service1/LevelOpen/";  
HttpGet get = new HttpGet(linc);
```

Листинг 10 – GET-запрос на уровни жидкостей.

Далее данные полученные с сервера в формате XML десериализуются с помощью встроенной функции «DocumentBuilderFactory». Каждый уровень записывается в соответствующую переменную «textResultLevelFuel»- уровень

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		50

топлива, «textResultOilICE» - уровень масла ДВС, «textResultOilTracm» - уровень масла КПП, «textResultBreak» - уровень тормозной жидкости.

В конце текстовые поля содержащие информацию выводятся в формах на экране. Данное действие выполняется с помощью функции «setText()», параметром которой является текстовая переменная «textResultLevelFuel», «textResultOilICE», «textResultOilTracm», «textResultBreak».

Уровни жидкости отображаются на экране в виде шкалы. По аналогии с остальными объектами, функция «findViewById(R.id. imageView_levelFuel)», находит в файле XML нужный элемент текстового поля по уникальному имени «imageView_levelFuel». Так же находятся следующие элементы «imageView_levelOilICE1», «imageView_levelOilTracm1», «imageView_levelBreak1». Данные элементы ставятся в соответствие элементам «imageViewLevelFuel1» - шкала уровня топлива, «imageViewlevelOilICE1» - шкала уровня масла ДВС, «imageViewlevelOilTracm1» - Шкала уровня масла КПП, «imageViewlevelBreak1» - шкала уровня тормозной жидкости, соответственно. Данные шкалы отображают 0% по шкале. Соответственно создаются шкалы от 0% до 100%, для каждой рабочей жидкости отображаемой на экране. С помощью функция «findViewById()», каждое изображение подключаем к программному коду. Изначально каждая шкала невидима. Но при получении данных, в зависимости от принятого значения, видимой становится нужная шкала.

Функция «void Visiability(ImageView a)» необходима для появления нужной шкалы. В теле функции выполняется действие «a.setVisibility(View.VISIBLE)», которое включает видимость объекта переданного на вход функции. При получении значения менее 10% вызывается функция «void Visiability2(ImageView a, Imageview b)». Данная функция помимо отрисовки нужной шкалы, отрисовывает знак предупреждения.

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		51

5. Температура жидкостей – public class ActivityTemperature.

Для установки соответствия данного класса с XML версткой используется функция «setContentView(R.layout.activity_temperature)». Данная функция устанавливает соединение с файлом «activity_temperature», в котором описан интерфейс окна.

Функция «findViewById(R.id.text_temperatureICE)», находит в файле XML нужный элемент текстового поля по уникальному имени «text_temperatureICE». По аналогии находятся следующие элементы «text_temperatureTransm». Данные элементы ставятся в соответствие элементам «textTemperatureICE», «textTemperatureTransm», соответственно.

В теле класса описываются принятие данных с сервера и вывод их на экран устройства.

Функция Void doInBackground() принимает данные с сервера. Для этого в теле функции формируется GET запрос «HttpGet get = new HttpGet(linc)», где linc – это локальная переменная, содержащая GET-запрос на температуры жидкостей.

```
String linc =  
"http://wcfrestservice222.azurewebsites.net/Service1/TemperatureOpen/";  
  
HttpGet get = new HttpGet(linc);
```

Листинг 11 – GET-запрос на температуры жидкостей.

Далее данные полученные с сервера в формате XML десериализуются с помощью встроенной функции «DocumentBuilderFactory». Каждый уровень записывается в соответствующую переменную «textResultTemperatureICE»- температура масла ДВС, «textResultTemperatureTransm» - температура масла КПП.

В конце текстовые поля содержащие информацию выводятся в формах на экране. Данное действие выполняется с помощью функции «setText()», параметром которой является текстовая переменная «textResultTemperatureICE», «textResultTemperatureTransm».

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		52

Температуры жидкости отображаются на экране в виде шкалы. По аналогии с остальными объектами, функция «`findViewById(R.id.imageView_TemperatureICE1)`», находит в файле XML нужный элемент текстового поля по уникальному имени «`imageView_TemperatureICE1`». Так же находится следующий элемент «`imageView_temperatureTransm1`». Данные элементы ставятся в соответствие элементам «`imageViewTemperatureICE1`» - шкала температуры масла ДВС, «`imageTemperatureTransm1`» - шкала температуры масла КПП, соответственно. Данные шкалы отображают 0% по шкале. Соответственно создаются шкалы от 0% до 100%, для масла КПП и масла ДВС. С помощью функция «`findViewById()`», каждое изображение подключаем к программному коду. Изначально каждая шкала невидима. Но при получении данных, в зависимости от принятого значения, видимой становится нужная шкала.

Функция «`void Visiability(ImageView a)`» необходима для появления нужной шкалы. В теле функции выполняется действие «`a.setVisibility(View.VISIBLE)`», которое включает видимость объекта переданного на вход функции.

6. Напряжение узлов– `public class ActivityVolt`.

Для установки соответствия данного класса с XML версткой используется функция «`setContentView(R.layout.activity_volt)`». Данная функция устанавливает соединеине с файлом «`activity_volt`», в котором описан интерфейс окна.

Функция «`findViewById(R.id.text_VoltBattery)`», находит в файле XML нужный элемент текстового поля по уникальному имени «`text_VoltBattery`». По аналогии находятся следующие элементы «`text_VoltGenerator`». Данные элементы ставятся в соответствие элементам «`textVoltBattery`», «`textVoltGenerator`», соответственно.

В теле класса описываются принятие данных с сервера и вывод их на экран устройства.

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
						53
Изм.	Лист	№ докум.	Подпись	Дата		

Функция `Void doInBackground()` принимает данные с сервера. Для этого в теле функции формируется GET запрос «`HttpGet get = new HttpGet(linc)`», где `linc` – это локальная переменная, содержащая GET-запрос на напряжение узлов.

```
String linc = "http://wcfrestservice222.azurewebsites.net/Service1/VoltOpen/";  
HttpGet get = new HttpGet(linc);
```

Листинг 12 – GET-запрос на напряжение узлов.

Далее данные полученные с сервера в формате XML десериализуются с помощью встроенной функции «`DocumentBuilderFactory`». Напряжение аккумулятора записывается в переменную «`textResultVoltBattery`», напряжение генератора в «`textResultVoltGenerator`».

В конце текстовые поля содержащие информацию выводятся в формах на экране. Данное действие выполняется с помощью функции «`setText()`», параметром которой является текстовая переменная «`textResultVoltBattery`», «`textResultVoltGenerator`».

3.2 Реализация обработчика сигналов

Логика обработчика написана на языке C++.

Микроконтроллер через CAN BUS Shield соединяется с управляющим входом автомобиля. Через диагностический разъем OBD-II получает информацию о состоянии автомобиля. Режим работы CAN BUS Shield - J1939. Следовательно, информация получается, как битовая строка, где первые 12 бит – биты идентификатора сообщения, а за ними 8 байт информации (слева младший байт, справа старший).

```
ID: 12A Data: 10 F0 70 60 00 00 30 80  
ID: 135 Data: 00 08 0D 02 05 0F 00 00
```

Листинг 13 – Вид получаемой информации.

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		54

```

void setup()
{
  Serial.begin(115200);
  CAN0.begin(CAN_500KBPS);
  pinMode(3, INPUT);
}
void make()
{
  if(!digitalRead(3)      //Если контакт 3 низкий, считать буфером приема
  {
    CAN0.readMsgBuf(&len, rxBuf); // Чтение данных: len = длина данных, buf = байт данных
    rxId = CAN0.getCanId();      // Получение идентификатор сообщения
    Serial.print("ID: ");
    Serial.print(rxId, HEX);
    Serial.print("  Data: ");
    for(int i = 0; i<len; i++)   //Получение каждого байта данных
    {
      if(rxBuf[i] < 0x10)
      {
        Serial.print("0");
      }
      Serial.print(rxBuf[i], HEX);
    }
  }
}

```

Листинг 14 – Получение информации.

После, используя документацию [9], находятся идентификаторы нужной информации. Расшифровывается информация сообщения. Полученная информация формируется для отправки. Программа отправляет POST-запрос, содержащий информацию об автомобиле, на сервер. Отправка осуществляется с помощью GPRS-модуля.

Для отправки данных используется функция «Dispatch». Указывается тип POST, указывается адрес сервиса, в качестве сообщения передается класс «Auto», содержащий поля с информацией.

```

void Dispatch (){
  String a ="POST /Service1/getCarParameters"; //Тип запроса, тело запроса
  a.concat(Auto); //Класс Auto с информацией об автомобиле
  a.concat(" HTTP/1.1"); //Протокол передачи данных
  Serial.println(a);
  if (client.connect(server, 80)) {
    client.println(a);
    client.println("Host: wcfrestservice222.azurewebsites.net"); //Адрес
сервиса
    client.println("Connection: close");
  }
}

```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		55

```

while (true){
    if (client.available()) { //Проверка на получение доступа клиентом
char c = client.read();
    if (!client.connected()) { //Проверка на корректность соединения
        Serial.println("disconnecting.");
        client.stop();
        break;
    }
}

```

Листинг 15 – POST-запрос.

Таким образом реализован обработчик сигналов.

3.3 Реализация логики сервера

Сервер реализован на языке C#.

На сервере расположен класс содержащий поля характеризующие автомобиль. Данный класс содержит поля соответствующие значениям получаемой информации.

```

public class Car
{
    public int _id { get; set; }
    public int _TemperatureICE { get; set; }
    public int _TemperatureTransmission { get; set; }
    public int _TemperatureBattery { get; set; }
    public int _LevelFuel { get; set; }
    public int _LevelOilICE { get; set; }
    public int _LevelOilTransmission { get; set; }
    public int _LevelBreak { get; set; }
    public int _LevelFreeze { get; set; }
    public bool _ErrorPreasure { get; set; }
    public bool _ErrorVolt { get; set; }
    public bool _ErrorPumpGas { get; set; }
    public bool _ErrorCrankshaft { get; set; }
    public bool _ErrorNozzle { get; set; }
    public bool _ErrorKnockSensor { get; set; }
    public int _VoltBatarey { get; set; }
    public int _VoltGenerator { get; set; }
}

```

Листинг 16 – Класс Car, содержащий поля об автомобиле.

WCF-сервис получает данные с обработчика сигналов, считывает их и записывает в БД.

Получение данных происходит при помощи метода «getCarParameters». Метод получает данные отправленные обработчиком сигналов. Десериализует

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
						56
Изм.	Лист	№ докум.	Подпись	Дата		

информацию с помощью метода «XmlSerializer» и записывает в базу данных

```
[WebInvoke(UriTemplate = "/getCarParameters", Method = "POST")]
public Car getCarParameters() //Метод получения данных
{string body =
System.Text.Encoding.UTF8.GetString(ExecutionContext.Current.RequestContext.RequestMessage.GetBody<byte[]>());
XmlSerializer serializer = new XmlSerializer(typeof(Car)); //Десериализация сообщения
Stream read = new MemoryStream();
var writer = new StreamWriter(read);
writer.Write(body);
writer.Flush();
read.Position = 0;
Car temp = serializer.Deserialize(read) as Car; //Создание элемента класса Car
var search = carDataBase.Find(x => x._id == temp._id);
if (search == null)
    carDataBase.Add(temp); //Добавление экземпляра в БД
else
{
    carDataBase [carDataBase.IndexOf(search)] = temp;
}

return carDataBase.Find(x => x._id == temp._id);
}
```

Листинг 17 – Метод POST, для принятия информации с обработчика.

Сервис отправляет данные на устройство клиента при получении GET запроса. Программа обращается к БД, по ID находит нужный экземпляр автомобиля. В зависимости от запроса выбираются нужные параметры для отправки. После чего осуществляется отправка данных.

Отправка информации осуществляется при помощи индивидуальных методов:

- public Car.Volts VoltOpen – информация о напряжении;
- public Car.levels LevelOpen – информация об уровнях жидкостей;
- public Car.Errors ErrorOpen – информация об ошибках;
- public Car.temperatures TemperatureOpen – информация о температуре жидкостей.

Параметром данных функций является ID автомобиля.

```
[WebInvoke(UriTemplate = "/TemperatureOpen/{id}", Method = "GET")]
public Car.temperatures TemperatureOpen(string id) //Метод для отправки информации
о температуре рабочих жидкостей
{
    Car A = new Car();
}
```

					<i>ЮУрГУ-09.20.06.2018.154 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		57


```

A = carList.Find(x => x._id == Int32.Parse( id)); //Поиск нужного экземпляра в БД
Car.temperatures temp = new Car.temperatures(); //Формирование сообщения
temp._TemperatureBattery = A._TemperatureBattery;
temp._TemperatureICE = A._TemperatureICE;
temp._TemperatureTransmission = A._TemperatureTransmission;
return temp;
}

```

Листинг 18 – Метод GET, для отправки информации клиенту.

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		58

4 ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ

4.1 Тест 1 – диагностика ошибок

Тестирование разработанного программного продукта проводилось с помощью эмулятора мобильного устройства Galaxy Nexus на платформе Android.

В таблице 5 описано тестирование диагностики ошибок.

Таблица 5 – Тестирование диагностики ошибок

Свойство	Значение
Выполняемые действия	Приложение запущено, выбран автомобиль, нажимаем кнопку «Диагностика ошибок».
Ожидаемые результаты	При нажатии на кнопку «Диагностика ошибок», должно открыться окно, в котором отобразится информация о состоянии датчиков сигнализирующих, о состоянии автомобиля.
Полученные результаты	Нажатие на кнопку «Диагностика ошибок» открывает окно с информацией. При обнаружении ошибки появляется знак предупреждения в строке с данной ошибкой.



Рисунок 20 – скриншот окна диагностики ошибок

4.2 Тест 2 – Температура жидкостей

В таблице 6 описано тестирование получения температуры жидкостей.

Таблица 6 – получения температуры жидкостей

Свойство	Значение
Выполняемые действия	Приложение запущено, выбран автомобиль, нажимаем кнопку «Температуры жидкостей».
Ожидаемые результаты	При нажатии на кнопку «Температуры жидкостей», должно открыться окно, в котором отобразится информация о температурах масла ДВС и масла КПП.
Полученные результаты	Нажатие на кнопку «Диагностика ошибок» открывает окно с информацией. Информация отображена при помощи шкал. Стрелка показывает температуру, соответствующую цифровому значению, принятому с сервера.



Рисунок 21 – Скриншот формы температуры жидкостей

4.3 Тест 3 - Уровни жидкостей

В таблице 7 описано тестирование получения уровней жидкости

Таблица 7 – уровни жидкостей

Свойство	Значение
Выполняемые действия	Приложение запущено, выбран автомобиль, нажимаем кнопку «Уровни жидкостей».
Ожидаемые результаты	При нажатии на кнопку «Уровни жидкостей», должно открыться окно, в котором отобразится информация об уровнях топлива, масла ДВС, масла КПП, тормозной жидкости.
Полученные результаты	Нажатие на кнопку «Уровни жидкостей» открывает окно с информацией. Информация отображена при помощи шкал. Стрелка показывает уровень, соответствующий цифровому значению, принятому с сервера. Если значение ниже минимального уровня, загорается индикатор.



Рисунок 22 – Скриншот формы уровней жидкости

4.4 Тест 4 – Напряжение узлов

В таблице 8 описано тестирование получения напряжения узлов.

Таблица 8 – напряжение узлов

Свойство	Значение
Выполняемые действия	Приложение запущено, выбран автомобиль, нажимаем кнопку «Напряжение узлов».
Ожидаемые результаты	При нажатии на кнопку «Напряжение узлов», должно открыться окно, в котором отобразится информация о напряжениях на генераторе и аккумуляторе автомобиля.
Полученные результаты	Нажатие на кнопку «Напряжение узлов» открывает окно с информацией. Информация о напряжении, выводится в Вольтах и отображается с помощью текстового поля.



Рисунок 23 – Скриншот формы напряжение узлов

Результаты тестирования на рисунках показывают корректную работу программного продукта.

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		63

ЗАКЛЮЧЕНИЕ

Целью данной работы являлось создание программно-аппаратного комплекса получения информации с датчиков, сигнализирующих о состоянии автомобиля, с последующей передачей данных на мобильное устройство клиента.

В ходе выполненной работы решены следующие задачи:

- проведено исследование принципа работы диагностики автомобиля;
- проведен анализ существующих на рынке решений;
- в процессе проведения анализа действующих решений были сформулированы требования к разрабатываемой системе;
- спроектирована база данных, обслуживающая систему;
- разработана система передачи данных на мобильное устройство;
- разработана система передачи информации по Internet;
- проведен анализ технических средств;
- разработан алгоритм работы программно-аппаратного комплекса.

Программно-аппаратный комплекс осуществляет работу по Internet, данные принимаемые с нескольких автомобилей могут выводиться на одно мобильное устройство. Данные свойства помогают сократить временные издержки при диагностике состояния всех автомобилей автопарка.

Разработанный программно-аппаратный комплекс имеет следующие перспективы на будущее развитие:

- разработка приложений на платформы iOS и Windows Phone;
- монетизация за счет продажи устройств.

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		64

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Волгин В. В. Справочник по диагностике неисправностей автомобиля/ В. В. Волгин. – М.: Атласы автомобилей, 2006. – 96 с.
- 2 Бен-Ган И. PostgreSQL Database® 2012. Создание запросов. / И. Бен-Ган, Д. Сарка, Р. Талмейдж. – СПб.: BHV, 2012. – 702 с.
- 3 Новиков Ф.А. Основы моделирования на UML: учебное пособие. / Ф.А. Новиков, Д.Ю. Иванов. – СПб.: СПбГУ ИТМО, 2010. – 200 с.
- 4 Фримен А. ASP.NET MVC 5 с примерами на С# 5.0 для профессионалов / А. Фримен., Л.: Вильямс – СПб.: BHV 2015. – 736 с.
- 5 Щербаков А. К. Сеть CAN: популярные прикладные протоколы/ А. К. Щербаков. – М.: Современные Технологии Автоматизации, 2005. – 354 с.
- 6 Харди Б., Android. Программирование для профессионалов. / Б. Филлипс, К. Стюарт, К. Марсикано – СПб.: BHV, 2015. – 456 с.
- 7 Характеристики приложения «Torque». [Электронный ресурс]/ Электрон. текстовые, граф., дан. – Режим доступа: <https://torque-bhp.com>, свободный. — Загл. с экрана. — Яз. англ. (дата обращения: 03.02.2018).
- 8 Характеристики приложения «Palmer». [Электронный ресурс]/ Электрон. текстовые, граф., дан. – Режим доступа: <https://www.palmerperformance.com>, свободный. — Загл. с экрана. — Яз. англ. (дата обращения: 15.02.2018).
- 9 Коды ошибок автомобильной диагностики [Электронный ресурс]/ Электрон. текстовые, граф., дан. – Режим доступа: <http://autoscanner.by/kody-oshibok.html>, свободный. — Загл. с экрана. — Яз. рус. (дата обращения: 09.01.2018).

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		65

ПРИЛОЖЕНИЕ А

Листинг исходного кода проекта

Реализация сервисной части приложения:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.ServiceModel;
using System.ServiceModel.Activation;
using System.ServiceModel.Web;
using System.Text;
using System.IO;
using System.Xml.Serialization;
using Newtonsoft.Json;
namespace WcfRestService

{
    [ServiceContract]
    [AspNetCompatibilityRequirements(RequirementsMode =
AspNetCompatibilityRequirementsMode.Allowed)]
    [ServiceBehavior(InstanceContextMode = InstanceContextMode.PerCall)]
    public class Service1
    {

        static List<Car> carList = new List<Car>();
        [WebInvoke(UriTemplate = "/TemperatureOpen/{id}", Method = "GET")]
        public Car.temperatures TemperatureOpen(string id)
        {
            Car A = new Car();
            A = carList.Find(x => x._id == Int32.Parse( id));
            Car.temperatures temp = new Car.temperatures();
            temp._TemperatureBattery = A._TemperatureBattery;
            temp._TemperatureICE = A._TemperatureICE;
            temp._TemperatureTransmission = A._TemperatureTransmission;
            return temp;
        }
        [WebInvoke(UriTemplate = "/ErrorOpen/{id}", Method = "GET")]
        public Car.Errors ErrorOpen(string id)
        {
            Car A = new Car();
            A = carList.Find(x => x._id == Int32.Parse(id));
            Car.Errors temp = new Car.Errors();
            temp._ErrorCrankshaft = A._ErrorCrankshaft;
```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		66

```

temp._ErrorKnockSensor = A._ErrorKnockSensor;
temp._ErrorNozzle = A._ErrorNozzle;
temp._ErrorPreasure = A._ErrorPreasure;
temp._ErrorPumpGas = A._ErrorPumpGas;
temp._ErrorVolt = A._ErrorVolt;
return temp;
}
[WebInvoke(UriTemplate = "/LevelOpen/{id}", Method = "GET")]

```

```

public Car.levels LevelOpen(string id)
{
    Car A = new Car();
    A = carList.Find(x => x._id == Int32.Parse(id));
    Car.levels temp = new Car.levels();
    temp._LevelBreak = A._LevelBreak;
    temp._LevelFreeze = A._LevelFreeze;
    temp._LevelFuel = A._LevelFuel;
    temp._LevelOilICE = A._LevelOilICE;
    temp._LevelOilTransmission = A._LevelOilTransmission;
    return temp;
}

```

```

[WebInvoke(UriTemplate = "/VoltOpen/{id}", Method = "GET")]
public Car.Volts VoltOpen(string id)
{
    Car A = new Car();
    A = carList.Find(x => x._id == Int32.Parse(id));
    Car.Volts temp = new Car.Volts();
    temp._VoltBatarey = A._VoltBatarey;
    temp._VoltGenerator = A._VoltGenerator;
    return temp;
}

```

```

[OperationContract]
[WebInvoke(UriTemplate = "/getCarParameters", Method = "POST")]
public Car getCarParameters()
{
    string body =
System.Text.Encoding.UTF8.GetString(OperationContext.Current.RequestContext.R
equestMessage.GetBody<byte[]>());

    XmlSerializer serializer = new XmlSerializer(typeof(Car));
    Stream read = new MemoryStream();

```

					<i>ЮУрГУ-09.20.06.2018.154 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		67

```

var writer = new StreamWriter(read);
writer.Write(body);
writer.Flush();
read.Position = 0;
Car temp = serializer.Deserialize(read) as Car;
var search = carList.Find(x => x._id == temp._id);
if (search == null)
    carList.Add(temp);
else
{
    carList[carList.IndexOf(search)] = temp;
}

return carList.Find(x => x._id == temp._id);
}

```

```

[HttpGet(UriTemplate = "{id}")]
public Car Get(string id)
{
    throw new NotImplementedException();
}

```

```

[WebInvoke(UriTemplate = "{id}", Method = "PUT")]
public Car Update(string id, Car instance)
{
    throw new NotImplementedException();
}

```

```

[WebInvoke(UriTemplate = "{id}", Method = "DELETE")]
public void Delete(string id)
{
    throw new NotImplementedException();
}

```

```

}
}
namespace WcfRestService
{
    public class Car
    {
        public struct temperatures {

```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		68

```

public int _TemperatureICE { get; set; }
public int _TemperatureTransmission { get; set; }
public int _TemperatureBattery { get; set; }

}

public struct levels
{

public int _LevelFuel { get; set; }
public int _LevelOilICE { get; set; }
public int _LevelOilTransmission { get; set; }
public int _LevelBreak { get; set; }
public int _LevelFreeze { get; set; }

}

public struct Errors
{
public bool _ErrorPreasure { get; set; }
public bool _ErrorVolt { get; set; }
public bool _ErrorPumpGas { get; set; }
public bool _ErrorCrankshaft { get; set; }
public bool _ErrorNozzle { get; set; }
public bool _ErrorKnockSensor { get; set; }
}

public struct Volts
{
public int _VoltBatarey { get; set; }
public int _VoltGenerator { get; set; }
}

public int _id { get; set; }
public int _TemperatureICE { get; set; }
public int _TemperatureTransmission { get; set; }
public int _TemperatureBattery { get; set; }
public int _LevelFuel { get; set; }
public int _LevelOilICE { get; set; }
public int _LevelOilTransmission { get; set; }
public int _LevelBreak { get; set; }
public int _LevelFreeze { get; set; }
public bool _ErrorPreasure { get; set; }
public bool _ErrorVolt { get; set; }

```

```
public bool _ErrorPumpGas { get; set; }
public bool _ErrorCrankshaft { get; set; }
public bool _ErrorNozzle { get; set; }
public bool _ErrorKnockSensor { get; set; }
public int _VoltBatarey { get; set; }
public int _VoltGenerator { get; set; }

}
}
```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		70

Реализация обработчика сигналов:

```
#include <SPI.h>
#include<mcp_can.h>
#include <Ethernet.h>

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

char server[] = "wcfrestservice222.azurewebsites.net";

EthernetClient client;

MCP_CAN CAN(10);
INT32U canId = 0x000;
unsigned char len = 0;
unsigned char buf[8];
char str[20];
String BuildMessage="";
int MSGIdentifier=0;
void setup() {
  Serial.begin(15200);
  START_INIT:
  if(CAN_OK == CAN.begin(CAN_500KBPS))
  {
    Serial.println("CAN BUS Shield init ok!");
  }
  else
  {
    Serial.println("CAN BUS Shield init fail");
    Serial.println("Init CAN BUS Shield again");
    delay(100);
    goto START_INIT;
  }
}

void dispatch(b)
{
  String a ="GET /Service1/temper/";
  a.concat(b);
  a.concat(" HTTP/1.1");
  if (client.connect(server, 80)) {
    Serial.println("connected");
    client.println(a);
    client.println("Host: wcfrestservice222.azurewebsites.net");
```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		71

```

client.println("Connection: close");
client.println();
while (true){
if (client.available()) {
char c = client.read();
Serial.print(c);
if (!client.connected()) {
Serial.println();
Serial.println("disconnecting.");
client.stop();
break;
}
}
}
}else {
Serial.println("connection failed");
}

}

void loop()
{
char rndCoolantTemp=random(1,200);
char rndRPM=random(1,55);
char rndSpeed=random(0,255);
char rndIAT=random(0,255);
char rndMAF=random(0,255);
char rndAmbientAirTemp=random(0,200);
char rndCAT1Temp=random(1,55);

//GENERAL ROUTINE
unsigned char SupportedPID[8] = {1,2,3,4,5,6,7,8};
unsigned char MilCleared[7] = {4, 65, 63, 34, 224, 185,
147};

//SENSORS
unsigned char CoolantTemp[7] = {4, 65, 5, rndCoolantTemp, 0,
185, 147};
unsigned char rpm[7] = {4, 65, 12, rndRPM, 224, 185, 147};
unsigned char vspeed[7] = {4, 65, 13, rndSpeed, 224, 185,
147};
unsigned char IATSensor[7] = {4, 65, 15, rndIAT, 0, 185,
147};
unsigned char MAFSensor[7] = {4, 65, 16, rndMAF, 0, 185,
147};

```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		72

```

    unsigned char AmbientAirTemp[7] = {4, 65, 70,
    rndAmbientAirTemp, 0, 185, 147};
    unsigned char CAT1Temp[7] = {4, 65, 60, rndCAT1Temp,
    224, 185, 147};
    unsigned char CAT2Temp[7] = {4, 65, 61, rndCAT1Temp,
    224, 185, 147};
    unsigned char CAT3Temp[7] = {4, 65, 62, rndCAT1Temp,
    224, 185, 147};
    unsigned char CAT4Temp[7] = {4, 65, 63, rndCAT1Temp,
    224, 185, 147};

    if(CAN_MSGAVAIL == CAN.checkReceive())
    {

        CAN.readMsgBuf(&len, buf);
        canId = CAN.getCanId();
        Serial.print("<");Serial.print(canId);Serial.print(",");
        for(int i = 0; i<len; i++)
        {
            BuildMessage = BuildMessage + buf[i] + ",";
        }
        dispatch (BuildMessage);
    }

```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		73

Реализация мобильного приложения:

```
import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.TextView;

import com.google.gson.Gson;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

import org.xml.sax.InputSource;
import java.io.StringReader;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;

import org.apache.http.HttpEntity;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.utils.URLEncodedUtils;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.util.EntityUtils;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.Objects;
```

					<i>ЮУрГУ-09.20.06.2018.154 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		74

```

public class ActivityHome extends AppCompatActivity{

    Button buttonCar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
        buttonCar = (Button) findViewById(R.id.buttonCar);
        buttonCar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(getApplicationContext(), Hello.class);
                startActivity(intent);
            }
        });
        buttonCar2 = (Button) findViewById(R.id.buttonCar2);
        buttonCar2.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(getApplicationContext(), Hello.class);
                startActivity(intent);
            }
        });
    }

}

public class Hello extends AppCompatActivity{

    Button buttonError;
    Button buttonLevel;
    Button buttonTemperature;
    Button buttonVolt;
    ImageButton imageButtonBack;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hello);
        imageButtonBack = (ImageButton) findViewById(R.id.imageButtonBack);
        imageButtonBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		75

```

        Intent intent = new Intent(getApplicationContext(), ActivityHome.class);
        startActivity(intent);
    }
});
buttonError = (Button) findViewById(R.id.buttonError);
buttonError.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(getApplicationContext(), ActivityTwo.class);
        startActivity(intent);
    }
});
buttonLevel = (Button) findViewById(R.id.buttonLevel);
buttonLevel.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(getApplicationContext(), ActivityThree.class);
        startActivity(intent);
    }
});
buttonTemperature = (Button) findViewById(R.id.buttonTemperature);
buttonTemperature.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(getApplicationContext(), ActivityFour.class);
        startActivity(intent);
    }
});
buttonVolt = (Button) findViewById(R.id.buttonVolt);
buttonVolt.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(getApplicationContext(), ActivityFive.class);
        startActivity(intent);
    }
});

```

```
public class ActivityTwo extends AppCompatActivity {
```

```
    ImageButton imageButtonBack;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		76

```

setContentView(R.layout.activity_two);
imageButtonBack = (ImageButton) findViewById(R.id.imageButtonBack);
imageButtonBack.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(getApplicationContext(), Hello.class);
        startActivity(intent);
    }
});
new ActivityTwo.Mytask().execute();
}

```

```

private class Mytask extends AsyncTask<Void, Void, Void> {

    TextView Error1 = (TextView) findViewById(R.id.text);
    TextView Error2 = (TextView) findViewById(R.id.text3);
    TextView Error3 = (TextView) findViewById(R.id.text4);
    TextView Error4 = (TextView) findViewById(R.id.text5);
    TextView Error5 = (TextView) findViewById(R.id.text6);
    TextView Error6 = (TextView) findViewById(R.id.text7);
    ImageView imageViewError1 = (ImageView)
findViewById(R.id.imageViewError1);
    ImageView imageViewError2 = (ImageView)
findViewById(R.id.imageViewError2);
    ImageView imageViewError3 = (ImageView)
findViewById(R.id.imageViewError3);
    ImageView imageViewError4 = (ImageView)
findViewById(R.id.imageViewError4);
    ImageView imageViewError5 = (ImageView)
findViewById(R.id.imageViewError5);
    ImageView imageViewError6 = (ImageView)
findViewById(R.id.imageViewError6);

    String textResult;
    String textResult2;
    String textResult3;
    String textResult4;
    String textResult5;
    String textResult6;
    String textError1;
    String textError2;
    String textError3;
    String textError4;
    String textError5;

```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		77

```

String textError6;

private void Visiability(ImageView a)
{
    a.setVisibility(View.VISIBLE);
}

@Override
protected Void doInBackground(Void... voids) {
    String linc =
"http://wcfrestservice222.azurewebsites.net/Service1/ErrorOpen/1";

    HttpClient client = new DefaultHttpClient();
    HttpGet get = new HttpGet(linc);

    try {

        HttpResponse response = client.execute(get);
        HttpEntity resEntity = response.getEntity();

        String resultet = EntityUtils.toString(resEntity);

        DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
        DocumentBuilder builder;
        InputSource is;
        try {
            builder = factory.newDocumentBuilder();
            is = new InputSource(new StringReader(resultet));
            Document doc = builder.parse(is);
            NodeList list =doc.getElementsByTagName("_ErrorCrankshaft");
            textResult = "Ошибки положения коленвала: " +
(Objects.equals(list.item(0).getTextContent(), "true")?"Да":"Нет") ;
            texte = list.item(0).getTextContent();
            list = doc.getElementsByTagName("_ErrorKnockSensor");
            textResult2 = "Ошибки датчика детонации: " +
(Objects.equals(list.item(0).getTextContent(), "true")?"Да":"Нет") ;
            texte2 = list.item(0).getTextContent();
            list = doc.getElementsByTagName("_ErrorNozzle");
            textResult3 = "Ошибки положения форсунок: " +
(Objects.equals(list.item(0).getTextContent(), "true")?"Да":"Нет") ;
            texte3 = list.item(0).getTextContent();
            list = doc.getElementsByTagName("_ErrorPreasure");

```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		78

```

        textResult4 = "Ошибки давления в системе: " +
(Objects.equals(list.item(0).getTextContent(), "true")?"Да":"Нет") ;
        texte4 = list.item(0).getTextContent();
        list = doc.getElementsByTagName("_ErrorPumpGas");
        textResult5 = "Ошибки топливного насоса: " +
(Objects.equals(list.item(0).getTextContent(), "true")?"Да":"Нет") ;
        texte5 = list.item(0).getTextContent();
        list = doc.getElementsByTagName("_ErrorVolt");
        textResult6 = "Ошибки напряжения сети: " +
(Objects.equals(list.item(0).getTextContent(), "true")?"Да":"Нет");
        texte6 = list.item(0).getTextContent();

    } catch (ParserConfigurationException e) {}

    } catch (Exception e) {
        e.printStackTrace();
        text.setText(e.toString());
    }
    return null;
}

```

@Override

protected void onPostExecute(Void result) {

```

    if(Objects.equals(textError1, "true"))
    {
        Visiability(imageViewError1);
    }
    if(Objects.equals(textError2, "true"))
    {
        Visiability(imageViewError2);
    }
    if(Objects.equals(textError3, "true"))
    {
        Visiability(imageViewerror3);
    }
    if(Objects.equals(textError4, "true"))
    {
        Visiability(imageViewError4);
    }
    if(Objects.equals(textError5, "true"))
    {

```

						ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата			79

```

        Visiability(imageView5);
    }
    if(Objects.equals(textError6, "true"))
    {
        Visiability(imageViewError6);
    }

    text.setText(textResult);
    text2.setText(textResult2);
    text3.setText(textResult3);
    text4.setText(textResult4);
    text5.setText(textResult5);
    text6.setText(textResult6);

    super.onPostExecute(result);
    }
}
}
}
public class ActivityThree extends AppCompatActivity {

    ImageButton imageButtonBack;

    @RequiresApi(api = Build.VERSION_CODES.CUPCAKE)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_three);
        imageButtonBack = (ImageButton) findViewById(R.id.imageButtonBack);
        imageButtonBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(getApplicationContext(), Hello.class);
                startActivity(intent);
            }
        });
        new ActivityThree.Mytask().execute();
    }

    @RequiresApi(api = Build.VERSION_CODES.CUPCAKE)
    private class Mytask extends AsyncTask<Void, Void, Void> {
        TextView Fuel = (TextView) findViewById(R.id.textFuel);
        TextView OilICE = (TextView) findViewById(R.id.textOil1);
        TextView OilTransm = (TextView) findViewById(R.id.textOil2);
        TextView Break = (TextView) findViewById(R.id.textBreak);
    }
}

```

```

        ImageView imageViewFuel1 = (ImageView)
findViewById(R.id.imageViewFuel1);
        ImageView imageViewOilICE1 = (ImageView)
findViewById(R.id.imageViewOilICE1);
        ImageView imageViewOilTrasm1 = (ImageView)
findViewById(R.id.imageViewOiltransm1);
        ImageView imageBreak1 = (ImageView) findViewById(R.id.imageBreak1);
        ImageView imageViewFuel2 = (ImageView)
findViewById(R.id.imageViewFuel2);
        ImageView imageViewOilICE2 = (ImageView)
findViewById(R.id.imageViewOilICE2);
        ImageView imageViewOilTrasm3 = (ImageView)
findViewById(R.id.imageViewOilTrasm2);
        ImageView imageBreak2 = (ImageView) findViewById(R.id.imageBreak2);
        ImageView imageViewFuel3 = (ImageView)
findViewById(R.id.imageViewFuel3);
        ImageView imageViewOilICE2 = (ImageView)
findViewById(R.id.imageViewOilICE2);
        ImageView imageViewOilTrasm3 = (ImageView)
findViewById(R.id.imageViewOilTrasm3);
        ImageView imageBreak3 = (ImageView) findViewById(R.id.imageBreak3);
        ImageView imageViewFuel3 = (ImageView)
findViewById(R.id.imageViewFuel3);
        ImageView imageViewOilICE4 = (ImageView)
findViewById(R.id.imageViewOilICE4);
        ImageView imageViewOilTrasm4 = (ImageView)
findViewById(R.id.imageViewOilTrasm4);
        ImageView imageBreak4 = (ImageView) findViewById(R.id.imageBreak4);
        ImageView imageViewFuel5 = (ImageView)
findViewById(R.id.imageViewFuel5);
        ImageView imageViewOilICE5 = (ImageView)
findViewById(R.id.imageViewOilICE5);
        ImageView imageViewOilTrasm5 = (ImageView)
findViewById(R.id.imageViewOilTrasm5);
        ImageView imageViewBreak5 = (ImageView)
findViewById(R.id.imageBreak5);
        ImageView imageWarning1 = (ImageView)
findViewById(R.id.imageWarning1);
        ImageView imageWarning2 = (ImageView)
findViewById(R.id.imageWarning2);
        ImageView imageWarning3 = (ImageView)
findViewById(R.id.imageWarning3);
        ImageView imageWarning4 = (ImageView)
findViewById(R.id.imageWarning4);

```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		81


```
String textResult;  
String textResult2;  
String textResult3;  
String textResult4;  
String textResult5;  
Integer textFuel;  
Integer textOilICE;  
Integer textOilTransm;  
Integer textBreak;
```

```
private void Visiability(ImageView a)  
{  
    a.setVisibility(View.VISIBLE);  
}  
private void Visiability2(ImageView a, ImageView b)  
{  
    a.setVisibility(View.VISIBLE);  
    b.setVisibility(View.VISIBLE);  
}
```

```
@RequiresApi(api = Build.VERSION_CODES.FROYO)  
@Override  
protected void doInBackground(Void... voids) {  
    String linc =  
"http://wcfrestservice222.azurewebsites.net/Service1/LevelOpen/1";  
  
    HttpClient client = new DefaultHttpClient();  
    HttpGet get = new HttpGet(linc);  
  
    try {  
  
        HttpResponse response = client.execute(get);  
        HttpEntity resEntity = response.getEntity();  
  
        String resultat = EntityUtils.toString(resEntity);  
  
        DocumentBuilderFactory factory =  
DocumentBuilderFactory.newInstance();  
        DocumentBuilder builder;  
        InputSource is;  
        try {
```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		82

```

builder = factory.newDocumentBuilder();
is = new InputSource(new StringReader(resultet));
Document doc = builder.parse(is);
NodeList list = doc.getElementsByTagName("_LevelFuel");
textFuel = Integer.parseInt(list.item(0).getTextContent());
textResult = "Уровень топлива " + "\n" + "\n";
list = doc.getElementsByTagName("_LevelOilICE");
textOilICE = Integer.parseInt(list.item(0).getTextContent());
textResult2 = "Уровень масла ДВС " + "\n" + "\n";
list = doc.getElementsByTagName("_LevelOilTransmission");
textOilTransm = Integer.parseInt(list.item(0).getTextContent());
textResult3 = "Уровень масла КПП " + "\n" + "\n";
list = doc.getElementsByTagName("_LevelBreak");
texteBreak = Integer.parseInt(list.item(0).getTextContent());
textResult4 = "Уровень тормозной \n жидкости " + "\n" + "\n";

} catch (ParserConfigurationException e) {}

} catch (Exception e) {
    e.printStackTrace();
    text.setText(e.toString());
}
return null;
}

@RequiresApi(api = Build.VERSION_CODES.KITKAT)
@Override
protected void onPostExecute(Void result) {

    if(Objects.equals(textFuel, 3))
    {
        Visiability(imageFuel1);
    }
    if(Objects.equals(textOikICE, 3))
    {
        Visiability(imageOikICE1);
    }
}

```

```

if(Objects.equals(textOilTransm, 3))
{
    Visiability(imageViewOilTransm1);
}
if(Objects.equals(textBreak, 3))
{
    Visiability(imageViewBreak1);
}
    if(Objects.equals(textFuel, 5))
    {
        Visiability(imageViewFuel2);
    }
    if(Objects.equals(textOikICE, 5))
    {
        Visiability(imageOikICE2);
    }
    if(Objects.equals(textOilTransm, 5))
    {
        Visiability(imageViewOilTransm2);
    }
    if(Objects.equals(textBreak, 5))
    {
        Visiability(imageViewBreak2);
    }
        if(Objects.equals(textFuel, 2))
        {
            Visiability(imageViewFuel3);
        }
        if(Objects.equals(textOikICE, 2))
        {
            Visiability(imageViewOikICE3);
        }
        if(Objects.equals(textOilTransm, 2))
        {
            Visiability(imageViewOilTransm3);
        }
        if(Objects.equals(textBreak, 2))
        {
            Visiability(imageViewBreak3);
        }
            if(Objects.equals(textFuel, 4))
            {
                Visiability(imageViewFuel3);
            }
}

```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		84

```

        if(Objects.equals(textOikICE, 4))
        {
            Visiability(imageViewOikICE3);
        }
        if(Objects.equals(textOilTransm, 4))
        {
            Visiability(imageViewOilTransm3);
        }
        if(Objects.equals(textBreak, 4))
        {
            Visiability(imageViewBreak3);
        }
        if(Objects.equals(textFuel, 1))
        {
            Visiability2(imageFuel4, imageViewWarning1);
        }
        if(Objects.equals(textOikICE, 1))
        {
            Visiability2(imageOikICE4, imageViewWarning2);
        }
        if(Objects.equals(textOilTransm, 1))
        {
            Visiability2(imageViewOilTransm, imageViewWarning3);
        }
        if(Objects.equals(textBreak, 1))
        {
            Visiability2(imageViewBreak, imageViewWarning4);
        }

        textFuel.setText(textResult);
        textOilICE.setText(textResult2);
        textOilTransm.setText(textResult3);
        textBreak.setText(textResult4);

        super.onPostExecute(result);
    }
}
}
public class ActivityFour extends AppCompatActivity {

    ImageButton imageButtonBack;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		85

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_four);
ImageButtonBack = (ImageButton) findViewById(R.id.imageButtonBack);
ImageButtonBack.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(getApplicationContext(), Hello.class);
        startActivity(intent);
    }
});
new ActivityFour.Mytask().execute();
}

```

```

private class Mytask extends AsyncTask<Void, Void, Void> {
    TextView ICE = (TextView) findViewById(R.id.textViewICE);
    TextView Transm = (TextView) findViewById(R.id.textViewTransm);

```

```

        ImageView imageViewICE1 = (ImageView)
findViewById(R.id.imageViewICE1);
        ImageView imageViewTransm1 = (ImageView)
findViewById(R.id.imageViewTransm1);
        ImageView imageViewICE2 = (ImageView)
findViewById(R.id.imageViewICE2);
        ImageView imageViewTransm2 = (ImageView)
findViewById(R.id.imageViewTransm2);
        ImageView imageViewICE3 = (ImageView)
findViewById(R.id.imageViewICE3);
        ImageView imageViewTransm3 = (ImageView)
findViewById(R.id.imageViewTransm3);
        ImageView imageViewICE4 = (ImageView)
findViewById(R.id.imageViewICE4);
        ImageView imageViewTransm4 = (ImageView)
findViewById(R.id.imageViewTransm4);

```

```

String textResult;
String textResult2;
Integer ICE;
Integer Transm;

```

```

private void Visiability(ImageView a)
{
    a.setVisibility(View.VISIBLE);
}

```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		86

```

@Override
protected Void doInBackground(Void... voids) {
    String linc =
"http://wcfrestservice222.azurewebsites.net/Service1/TemperatureOpen/1";

    HttpClient client = new DefaultHttpClient();
    HttpGet get = new HttpGet(linc);

    try {

        HttpResponse response = client.execute(get);
        HttpEntity resEntity = response.getEntity();

        String resultet = EntityUtils.toString(resEntity);

        DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
        DocumentBuilder builder;
        InputSource is;
        try {
            builder = factory.newDocumentBuilder();
            is = new InputSource(new StringReader(resultet));
            Document doc = builder.parse(is);

            NodeList list = doc.getElementsByTagName("_TemperatureICE");
            textICE = Integer.parseInt(list.item(0).getTextContent());
            textResult = "Температура ДВС ";
            list = doc.getElementsByTagName("_TemperatureTransmission");
            textTransm = Integer.parseInt(list.item(0).getTextContent());
            textResult2 = "Температура КПП ";
        } catch (ParserConfigurationException e) {}

        } catch (Exception e) {
            e.printStackTrace();
            text.setText(e.toString());
        }
        return null;
    }

@Override
protected void onPostExecute(Void result) {

```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		87

```

        if(Objects.equals(textICE, 10))
        {
            Visiability(imageViewICE1);
        }
        if(Objects.equals(textTransm, 10))
        {
            Visiability(imageViewTransm1);
        }
        if(Objects.equals(textICE, 7))
        {
            Visiability(imageViewICE2);
        }
        if(Objects.equals(textTransm, 7))
        {
            Visiability(imageViewTransm2);
        }
        if(Objects.equals(textICE, 5))
        {
            Visiability(imageViewICE3);
        }
        if(Objects.equals(textTransm, 5))
        {
            Visiability(imageViewTransm3);
        }
        if(Objects.equals(textICE, 1))
        {
            Visiability(imageViewICE4);
        }
        if(Objects.equals(textTransm, 1))
        {
            Visiability(imageViewTransm4);
        }

        ICE.setText(textResult);
        Transm.setText(textResult2);
        super.onPostExecute(result);
    }
}
}

```

```

public class ActivityFive extends AppCompatActivity {

```

```

    ImageButton imageButtonBack;

```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		88

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_five);
    ImageButtonBack = (ImageButton) findViewById(R.id.imageButtonBack);
    ImageButtonBack.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(getApplicationContext(), Hello.class);
            startActivity(intent);
        }
    });
    new ActivityFive.Mytask().execute();
}

private class Mytask extends AsyncTask<Void, Void, Void> {
    TextView textBatarey = (TextView) findViewById(R.id.textView2);
    TextView textGenerator = (TextView) findViewById(R.id.textView4);

    String textResult1;
    String textResult2;

    @Override
    protected Void doInBackground(Void... voids) {
        String linc =
"http://wcfrestservice222.azurewebsites.net/Service1/VoltOpen/1";

        HttpClient client = new DefaultHttpClient();
        HttpGet get = new HttpGet(linc);

        try {

            HttpResponse response = client.execute(get);
            HttpEntity resEntity = response.getEntity();

            String resultet = EntityUtils.toString(resEntity);

            DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder builder;
            InputSource is;

```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		89


```

try {
    builder = factory.newDocumentBuilder();
    is = new InputSource(new StringReader(resultet));
    Document doc = builder.parse(is);
    NodeList list = doc.getElementsByTagName("_VoltBatarey");
    textResult1 = "Напряжение АКБ : " + list.item(0).getTextContent() + "
В";
    list = doc.getElementsByTagName("_VoltGenerator");
    textResult2 = "Напряжение генератора : " +
list.item(0).getTextContent() + " В";
    } catch (ParserConfigurationException e) {}

    } catch (Exception e) {
        e.printStackTrace();
        text.setText(e.toString());
    }
    return null;
}

@Override
protected void onPostExecute(Void result) {

    textBatarey.setText(textResult1);
    textGenerator.setText(textResult2);
    super.onPostExecute(result);
}
}
}

```

XML верстка мобильного приложения:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
```

```
  tools:context="ru.vladimir74.vova.Hello">
```

```
<Button
  android:id="@+id/buttonCAR1"
  android:layout_width="131dp"
  android:layout_height="52dp"
  android:layout_marginBottom="390dp"
  android:layout_marginEnd="32dp"
  android:layout_marginStart="16dp"
  android:layout_marginTop="170dp"
  android:text="Лада приора"
  app:layout_constraintBottom_toBottomOf="parent"
  app:layout_constraintEnd_toEndOf="parent"
  app:layout_constraintHorizontal_bias="0.541"
  app:layout_constraintStart_toStartOf="parent"
  app:layout_constraintTop_toTopOf="parent"
  app:layout_constraintVertical_bias="0.0" />
```

```
<Button
  android:id="@+id/buttonCAR2"
  android:layout_width="131dp"
  android:layout_height="52dp"
  android:layout_marginBottom="390dp"
  android:layout_marginEnd="32dp"
  android:layout_marginStart="16dp"
  android:layout_marginTop="230dp"
  android:text="Рено логан"
  app:layout_constraintBottom_toBottomOf="parent"
  app:layout_constraintEnd_toEndOf="parent"
  app:layout_constraintHorizontal_bias="0.541"
  app:layout_constraintStart_toStartOf="parent"
  app:layout_constraintTop_toTopOf="parent"
```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		91

```
app:layout_constraintVertical_bias="0.0" />
```

```
<TextView  
    android:id="@+id/textViewName"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Выберите автомобиль"  
    android:textColor="#000"  
    android:textSize="25sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.502"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.032" />
```

```
</android.support.constraint.ConstraintLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.ConstraintLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"
```

```
    tools:context="ru.vladimir74.vova.Hello">
```

```
<ImageButton  
    android:id="@+id/imageButtonBack"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    app:srcCompat="@android:drawable/ic_menu_revert"  
    tools:layout_editor_absoluteX="2dp"  
    tools:layout_editor_absoluteY="0dp" />
```

```
<Button  
    android:id="@+id/buttonError"  
    android:layout_width="131dp"  
    android:layout_height="105dp"  
    android:layout_marginBottom="390dp"  
    android:layout_marginEnd="32dp"
```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		92

```
android:layout_marginStart="16dp"
android:layout_marginTop="50dp"
android:text="диагностика ошибок"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.121"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.0" />
```

<Button

```
android:id="@+id/buttonLevel"
android:layout_width="131dp"
android:layout_height="105dp"
android:layout_marginBottom="390dp"
android:layout_marginEnd="16dp"
android:layout_marginStart="32dp"
android:layout_marginTop="50dp"
android:text="Уровни жидкостей"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.946"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.0" />
```

<Button

```
android:id="@+id/buttonTemperature"
android:layout_width="132dp"
android:layout_height="105dp"
android:text="Температуры агрегатов"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.896"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.472" />
```

<Button

```
android:id="@+id/buttonVolt"
android:layout_width="132dp"
android:layout_height="105dp"
android:text="Напряжение узлов"
app:layout_constraintBottom_toBottomOf="parent"
```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		93

```
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.162"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.472" />
```

```
</android.support.constraint.ConstraintLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="ru.vladimir74.vova.ActivityTwo">
```

```
<TextView
android:id="@+id/textError1"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginTop="100dp"
android:textColor="#000"
android:textSize="20sp"
android:textStyle="normal"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
android:id="@+id/textError2"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginTop="140dp"
android:textColor="#000"
android:textSize="20sp"
android:textStyle="normal"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		94

```
android:id="@+id/textError3"  
android:layout_width="0dp"  
android:layout_height="wrap_content"  
android:layout_marginTop="180dp"  
android:textColor="#000"  
android:textSize="20sp"  
android:textStyle="normal"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.0"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

<TextView

```
android:id="@+id/textError4"  
android:layout_width="0dp"  
android:layout_height="wrap_content"  
android:layout_marginTop="220dp"  
android:textColor="#000"  
android:textSize="20sp"  
android:textStyle="normal"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.0"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

<TextView

```
android:id="@+id/textError5"  
android:layout_width="0dp"  
android:layout_height="wrap_content"  
android:layout_marginTop="260dp"  
android:textColor="#000"  
android:textSize="20sp"  
android:textStyle="normal"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.0"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

<TextView

```
android:id="@+id/textError6"  
android:layout_width="0dp"  
android:layout_height="wrap_content"  
android:layout_marginTop="300dp"  
android:textColor="#000"
```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		95

```
android:textSize="20sp"
android:textStyle="normal"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

<ImageButton

```
android:id="@+id/imageButtonBack"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@android:drawable/ic_menu_revert" />
```

<ImageView

```
android:id="@+id/imageViewError1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="100dp"
android:visibility="invisible"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.98"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/ic_action_name" />
```

<ImageView

```
android:id="@+id/imageViewError2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="140dp"
android:visibility="invisible"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.98"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/ic_action_name" />
```

<ImageView

```
android:id="@+id/imageViewError3"
android:layout_width="wrap_content"
```

					<i>ЮУрГУ-09.20.06.2018.154 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		96

```
android:layout_height="wrap_content"
android:layout_marginTop="180dp"
android:visibility="invisible"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.98"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/ic_action_name" />
```

<ImageView

```
android:id="@+id/imageViewError4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="220dp"
android:visibility="invisible"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.98"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/ic_action_name" />
```

<ImageView

```
android:id="@+id/imageViewError5"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="260dp"
android:visibility="invisible"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.98"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/ic_action_name" />
```

</android.support.constraint.ConstraintLayout>

<?xml version="1.0" encoding="utf-8"?>

```
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="ru.vladimir74.vova.ActivityThree">
```

<TextView

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		97


```
android:id="@+id/textLevelFuel"
android:layout_width="0dp"
android:layout_height="35dp"
android:layout_marginTop="76dp"
android:textSize="20sp"
android:textColor="#000"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

<TextView

```
android:id="@+id/textLevelOilICE"
android:layout_width="0dp"
android:layout_height="37dp"
android:layout_marginTop="180dp"
android:textSize="20sp"
android:textColor="#000"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

<TextView

```
android:id="@+id/textLEVELOilTrasm"
android:layout_width="0dp"
android:layout_height="39dp"
android:layout_marginTop="292dp"
android:textSize="20sp"
android:textColor="#000"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

<TextView

```
android:id="@+id/textLevelBreak"
android:layout_width="0dp"
android:layout_height="63dp"
android:layout_marginTop="396dp"
android:textColor="#000"
android:textSize="20sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		98

```
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

<ImageButton

```
android:id="@+id/imageButtonBack"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.0"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:srcCompat="@android:drawable/ic_menu_revert" />
```

<ImageView

```
android:id="@+id/imageViewFuel1"  
android:layout_width="177dp"  
android:layout_height="144dp"  
android:layout_marginTop="16dp"  
android:visibility="invisible"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="1.0"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:srcCompat="@drawable/sp1" />
```

<ImageView

```
android:id="@+id/imageViewOilICE1"  
android:layout_width="174dp"  
android:layout_height="144dp"  
android:layout_marginTop="128dp"  
android:visibility="invisible"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="1.0"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:srcCompat="@drawable/sp1" />
```

<ImageView

```
android:id="@+id/imageViewOilTracm1"  
android:layout_width="174dp"  
android:layout_height="144dp"  
android:layout_marginTop="236dp"  
android:visibility="invisible"  
app:layout_constraintEnd_toEndOf="parent"
```

					<i>ЮУрГУ-09.20.06.2018.154 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		99

```
app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/sp1" />
```

<ImageView

```
android:id="@+id/imageViewBreak"
android:layout_width="174dp"
android:layout_height="144dp"
android:layout_marginTop="348dp"
android:visibility="invisible"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/sp1" />
```

<ImageView

```
android:id="@+id/imageViewFuel2"
android:layout_width="174dp"
android:layout_height="144dp"
android:layout_marginTop="16dp"
android:visibility="invisible"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/sp2" />
```

<ImageView

```
android:id="@+id/imageViewOilICE2"
android:layout_width="174dp"
android:layout_height="144dp"
android:layout_marginTop="128dp"
android:visibility="invisible"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/sp2" />
```

<ImageView

```
android:id="@+id/imageViewOilTransm2"
android:layout_width="174dp"
```

					<i>ЮУрГУ-09.20.06.2018.154 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		100

```
android:layout_height="144dp"
android:layout_marginTop="236dp"
android:visibility="invisible"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/sp2" />
```

<ImageView

```
android:id="@+id/imageViewBreak2"
android:layout_width="174dp"
android:layout_height="144dp"
android:layout_marginTop="348dp"
android:visibility="invisible"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/sp2" />
```

<ImageView

```
android:id="@+id/imageViewFuel3"
android:layout_width="174dp"
android:layout_height="144dp"
android:layout_marginTop="16dp"
android:visibility="invisible"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/sp3" />
```

<ImageView

```
android:id="@+id/imageViewOilICE3"
android:layout_width="174dp"
android:layout_height="144dp"
android:layout_marginTop="128dp"
android:visibility="invisible"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/sp3" />
```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		101

```
<ImageView
    android:id="@+id/imageViewOilTransm3"
    android:layout_width="174dp"
    android:layout_height="144dp"
    android:layout_marginTop="236dp"
    android:visibility="invisible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/sp3" />
```

```
<ImageView
    android:id="@+id/imageViewBreak3"
    android:layout_width="174dp"
    android:layout_height="144dp"
    android:layout_marginTop="348dp"
    android:visibility="invisible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/sp3" />
```

```
<ImageView
    android:id="@+id/imageViewFuel4"
    android:layout_width="174dp"
    android:layout_height="144dp"
    android:layout_marginTop="16dp"
    android:visibility="invisible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/sp5" />
```

```
<ImageView
    android:id="@+id/imageViewOilICE4"
    android:layout_width="174dp"
    android:layout_height="144dp"
    android:layout_marginTop="128dp"
    android:visibility="invisible"
    app:layout_constraintEnd_toEndOf="parent"
```

					<i>ЮУрГУ-09.20.06.2018.154 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		102

```
app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/sp5" />
```

```
<ImageView
    android:id="@+id/imageViewOilTransm4"
    android:layout_width="174dp"
    android:layout_height="144dp"
    android:layout_marginTop="236dp"
    android:visibility="invisible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/sp5" />
```

```
<ImageView
    android:id="@+id/imageViewBreak4"
    android:layout_width="174dp"
    android:layout_height="144dp"
    android:layout_marginTop="348dp"
    android:visibility="invisible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/sp5" />
```

```
<ImageView
    android:id="@+id/imageViewFuel5"
    android:layout_width="174dp"
    android:layout_height="144dp"
    android:layout_marginTop="16dp"
    android:visibility="invisible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/rer" />
```

```
<ImageView
    android:id="@+id/imageViewOilICE5"
    android:layout_width="174dp"
```

					<i>ЮУрГУ-09.20.06.2018.154 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		103

```
android:layout_height="144dp"
android:layout_marginTop="128dp"
android:visibility="invisible"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/rer" />
```

<ImageView

```
android:id="@+id/imageViewOilTransm5"
android:layout_width="174dp"
android:layout_height="144dp"
android:layout_marginTop="236dp"
android:visibility="invisible"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.985"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/rer" />
```

<ImageView

```
android:id="@+id/imageViewBreak5"
android:layout_width="174dp"
android:layout_height="144dp"
android:layout_marginTop="348dp"
android:visibility="invisible"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/rer" />
```

<ImageView

```
android:id="@+id/imageViewWarning1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="56dp"
android:visibility="invisible"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.798"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/ic_action_name" />
```

					<i>ЮУрГУ-09.20.06.2018.154 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		104

```
<ImageView
    android:id="@+id/imageViewWarning2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="168dp"
    android:visibility="invisible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.798"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/ic_action_name" />
```

```
<ImageView
    android:id="@+id/imageViewWarning3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="272dp"
    android:visibility="invisible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.798"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/ic_action_name" />
```

```
<ImageView
    android:id="@+id/imageViewWarning4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="388dp"
    android:visibility="invisible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.798"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/ic_action_name" />
```

```
</android.support.constraint.ConstraintLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
```

					<i>ЮУрГУ-09.20.06.2018.154 ПЗ ВКР</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		105


```
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="ru.vladimir74.vova.ActivityFour">
```

```
<ImageButton
    android:id="@+id/imageButtonBack"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.006"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@android:drawable/ic_menu_revert" />
```

```
<TextView
    android:id="@+id/textViewICE"
    android:layout_width="302dp"
    android:layout_height="58dp"
    android:layout_marginTop="196dp"
    android:textColor="#000"
    android:textSize="25sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
    android:id="@+id/textViewTransm"
    android:layout_width="301dp"
    android:layout_height="58dp"
    android:layout_marginTop="436dp"
    android:textColor="#000"
    android:textSize="25sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.987"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<ImageView
    android:id="@+id/imageViewICE1"
    android:layout_width="300dp"
    android:layout_height="204dp"
    android:layout_marginTop="24dp"
    android:visibility="invisible"
```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		106

```
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/volt" />
```

```
<ImageView
    android:id="@+id/imageViewTransm1"
    android:layout_width="300dp"
    android:layout_height="204dp"
    android:layout_marginTop="256dp"
    android:visibility="invisible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/volt" />
```

```
<ImageView
    android:id="@+id/imageViewICE2"
    android:layout_width="300dp"
    android:layout_height="204dp"
    android:layout_marginTop="24dp"
    android:visibility="invisible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/volt2" />
```

```
<ImageView
    android:id="@+id/imageViewTransm2"
    android:layout_width="300dp"
    android:layout_height="204dp"
    android:layout_marginTop="256dp"
    android:visibility="invisible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/volt2" />
```

```
<ImageView
    android:id="@+id/imageViewICE3"
    android:layout_width="300dp"
    android:layout_height="204dp"
    android:layout_marginTop="24dp"
    android:visibility="invisible"
```

					ЮУрГУ-09.20.06.2018.154 ПЗ ВКР	Лист
Изм.	Лист	№ докум.	Подпись	Дата		107

```
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:srcCompat="@drawable/volt3" />
```

<ImageView

```
android:id="@+id/imageViewTransm3"  
android:layout_width="300dp"  
android:layout_height="204dp"  
android:layout_marginTop="256dp"  
android:visibility="invisible"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:srcCompat="@drawable/volt3" />
```

<ImageView

```
android:id="@+id/imageViewICE4"  
android:layout_width="300dp"  
android:layout_height="204dp"  
android:layout_marginTop="24dp"  
android:visibility="invisible"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:srcCompat="@drawable/volt4" />
```

<ImageView

```
android:id="@+id/imageViewTransm4"  
android:layout_width="300dp"  
android:layout_height="204dp"  
android:layout_marginTop="256dp"  
android:visibility="invisible"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:srcCompat="@drawable/volt4" />
```

</android.support.constraint.ConstraintLayout>