

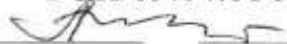
Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РТФ

Школа профессионального и академического образования

ДОПУСТИТЬ К ЗАЩИТЕ ПЕРЕД ГЭК

РОП 09.04.01 Аксенов К.А.

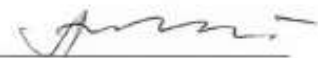


(подпись) (Ф.И.О.)
« » 2023 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

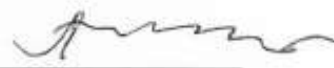
Разработка симулятора виртуальной реальности

Научный руководитель:
к.т.н.



Аксенов К.А.

Нормоконтролер:



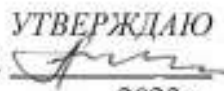
Аксенов К.А.

Студент группы РИМ-210961



Гришин Р.Ю.

Институт радиоэлектроники и информационных технологий - РТФ
Школа профессионального и академического образования
Направление подготовки: 09.04.01 Информатика и вычислительная техника
Образовательная программа: Информационно-управляющие системы

Директор ДИТиА  **УТВЕРЖДАЮ**
«___» _____ 2023 г.

ЗАДАНИЕ

на выполнение магистерской диссертации

студента Гришина Романа Юрьевича группы РИМ-210961

1. **Тема магистерской диссертации** Разработка симулятора виртуальной реальности

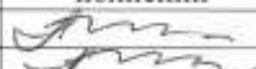
Утверждена распоряжением по ИРИТ-РТФ от 02.12.2022 г. № 33.02-05/286

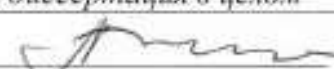
2. **Руководитель** Аксенов Константин Александрович, к.т.н., доцент департамента информационных технологий и автоматике ИРИТ-РТФ, УрФУ

3. **Исходные данные к работе**

4. **Перечень демонстрационных материалов** Презентации

5. **Календарный план**

№ п/п	Наименование этапов выполнения работы	Срок выполнения этапов работы	Отметка о выполнении
1.	<u>1 раздел (глава)</u>	до 31.01.2023 г.	
2.	<u>2 раздел (глава)</u>	до 09.04.2023 г.	
3.	<u>Магистерская диссертация в целом</u>	до 14.05. 2023 г.	

Руководитель 
(подпись)

Аксенов К.А.

Задание принял к исполнению _____
дата

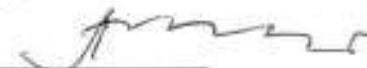

(подпись)

6. **Магистерская диссертация закончена** «15» мая 2023 г. считаю возможным допустить Гришина Романа Юрьевича к защите ее магистерской диссертации в Государственной экзаменационной комиссии.

Руководитель 
(подпись)

Аксенов К.А.

7. **Допустить** Гришина Романа Юрьевича к защите магистерской диссертации в Государственной экзаменационной комиссии (протокол заседания кафедры №___ от «___» ____ 20__ г.).

РОП 09.04.01 
(подпись)

Аксенов К.А.

ОТЗЫВ

руководителя магистерской диссертации

Магистерская диссертация выполнена на тему

Разработка симулятора виртуальной реальности

Студентом Гришиным Романом Юрьевичем группы РИМ-210961

Направление (специальность): Информатика и вычислительная техника

Образовательная программа: Информационно-управляющие системы

Руководитель: Аксенов Константин Александрович, к.т.н., доцент департамента информационных технологий и автоматизации ИРИТ-РТФ, УрФУ

Оценка соответствия требованиям ФГОС ВО подготовленной магистерской диссертации

Результаты освоения основной профессиональной образовательной программы, представленные в ВКР	Соответствует	В основном соответствует	Не соответствует
В части общекультурных компетенций:	+		
В части общепрофессиональных компетенций:	+		
В части профессиональных компетенций:	+		

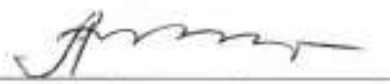
Отмеченные достоинства: Студент проявил самостоятельность при выполнении поставленных задач, показал умение применять теоретические знания на практике. Разработал и собрал прототип симулятора виртуальной реальности, на отечественных компонентах. Работа является актуальной и востребованной на сегодняшний день.

Отмеченные недостатки: Существенных недостатков в выпускной квалификационной работе не выявлено.

Дополнительная информация для ГЭК: Работа соответствует требованиям, предъявляемым к выпускным квалификационным работам.

Заключение. Выполненная работа является самостоятельным, законченным исследованием, достойным представления перед ГЭК. Студент Гришин Роман Юрьевич заслуживает присуждения степени **магистр** по направлению 09.04.01 Информатика и вычислительная техника.

Руководитель:


(подпись)

« 15 » май 2023 г.

Оглавление

РЕФЕРАТ	6
ОПРЕДЕЛЕНИЯ.....	7
ВВЕДЕНИЕ	8
ОСНОВНАЯ ЧАСТЬ	11
1 ПОСТАНОВКА ЗАДАЧИ.....	11
1.1 Обзор аналогов	12
1.1.1 Складной автосимулятор GameSTUL! HomeRevolution... 12	
1.1.2 Многофункциональный автосимулятор GameSTUL! DUAL 12	
1.1.3 XD Motion VR..... 13	
1.1.4 Автотренажер Forward КАМАЗ Категория «С»..... 14	
1.1.5 Статический кабинный автотренажер Forward КАМАЗ. 15	
1.1.6 Статический кабинный автотренажер КАМАЗ	16
1.1.7 Динамический автотренажер Forward КАМАЗ на 6-ти степенной платформе Категория "С, D, E"	17
1.1.8 Автотренажер Forward КАМАЗ открытого типа..... 18	
1.2 Техническое задание	20
1.2.1 Описание симулятора виртуальной реальности	20
1.1.2.1 Задачи, которые предстоит решить для создания симулятора. 20	
1.1.2.2 Основные функциональные возможности симулятора	20
2 ВЫБОР СРЕДСТВ РАЗРАБОТКИ.....	22
2.1 Исследование предметной области.....	22
2.2 Выбор средств разработки	26
3 СОЗДАНИЕ ПРОТОТИПА СИМУЛЯТОРА	37
ЗАКЛЮЧЕНИЕ	52
Приложение А	55
Приложение Б	60
Приложение В.....	72

РЕФЕРАТ

В Тема магистерской диссертации: «Разработка симулятора виртуальной реальности»

Диссертация содержит 52 с., 28 рисунков, 1 таблиц, 12 источника, 3 приложения.

РАЗРАБОТКА СИМУЛЯТОРА ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

Целью магистерской диссертации является разработка симулятор виртуальной реальности

В результате проведенной работы был разработан симулятор для программы Euro Track Simulator 2, позволяющий выводить данные из запущенного приложения на ПК, при помощи аддитивных технологий была модифицирована оригинальная приборная панель Камаз 4310, которая и послужила основой симулятора, в качестве платы управления использовалась отечественная Iskra Mega на базе микроконтроллера Atmega2560, устройство будет использоваться в качестве обучающего инструмента вождения. Симулятор выводит данные на оригинальную панель в соответствии с данными из приложения.

Устройство имеет возможность дальнейшей доработки и расширения функционала. Возможность создать полный кабинный автотренажёр.

Устройство было внедрено (приложение В).

ОПРЕДЕЛЕНИЯ

В настоящей пояснительной записке применяются следующие термины с соответствующими определениями, обозначения и сокращения:

Аддитивные технологии – технологии послойного создания моделей с использованием 3D печати

3D-модель - это объемное цифровое изображение необходимого объекта, как реального, так и вымышленного. Создание 3D моделей происходит в специальном программном обеспечении для 3D моделирования.

3D-печать - также известная как аддитивное производство, это производственный процесс, при котором устройство, 3D-принтер создает трехмерные объекты путем нанесения материала слоями, в соответствии с цифровой 3D-моделью объекта

Слайсер – программа, предназначенная для перевода файлов формата STL в формат понятный 3D принтерам в виде строчек команд gcode.

ПК – персональный компьютер, однопользовательская ЭВМ, имеющая эксплуатационные характеристики бытового прибора и универсальные функциональные возможности. ПК может быть или портативным.

ВВЕДЕНИЕ

В современном мире практически все люди используют личный автомобильный транспорт, чтобы получить права, разрешающие управление автотранспортными средствами, проходят обучение в автошколах. Вступившие в силу изменения в экзамене отменили этап автодрома, теперь экзаменуемых сразу же направляют на сдачу экзамена в городе. При ответственном подходе студент в течение 2-3 месяцев осваивает азы управления транспортным средством, сдает экзамен и получает водительское удостоверение. Симуляторы вождения имеются во многих автошколах, их использование помогает быстрее получить необходимые навыки, научиться управлять автомобилем в различных ситуациях и погодных условиях. Современные симуляторы вождения представляют собой программно-аппаратный комплекс, полностью имитирующий условия движения по автодорогам разного типа и качества покрытия. В состав стандартного автомобильного комплекта входят:

- монитор или несколько экранов, на которые транслируется видео изображение, соответствующее картинке за окном автомобиля в реальных условиях;
- кресло с различного уровня настройками, которое выглядит как водительское сиденье;
- органы управления, идентичные реальным. В современных автотренажерах предусмотрены руль, рукоятка переключения передач, педали, рычаги управления подворотниками и другие элементы;

обрабатывает информацию от органов управления и проецирует данные на монитор центральный блок, работающий под управлением специализированного программного обеспечения.

На любом этапе обучения существует возможность загружать разные компьютерные программы моделирования дорожных условий. Такой функционал помогает оттачивать навыки вождения в различных условиях.

Обучение вождению с использованием онлайн тренажера практикуется во многих автошколах. Современные программно-аппаратные комплексы необходимы, так как помогают получить следующие навыки:

- начинающий водитель учится чувствовать автомобиль, управлять транспортным средством любой категории на низких и высоких оборотах, переключать скорости без больших нагрузок на сцепление;
- автотренажер для водителя помогает научиться грамотно проходить повороты, выбирать нужную скорость и траекторию движения;
- с помощью подобных комплексов можно научиться держать дорогу, не прижиматься к обочине и не выезжать на встречную полосу;
- вождение в экстренных условиях, на скользкой и влажной дороге безопаснее отрабатывать именно на автосимуляторах.

Обучаясь управлению транспортным средством на симуляторе, будущий водитель получает необходимую тренировку, которая очень помогает ему в реальных ситуациях на дороге. Время, затраченное на учебные занятия в тренажере, не бывает бесполезным, особенно для новичков.

В начале учебной программы необходимо пользоваться автосимулятором, желательно ежедневно. Постепенно работу на тренажере можно совмещать с реальной ездой на автодроме и городские трассы. Польза

от использования симуляторов для обучения вождению легковых и грузовых автомобилей очевидна:

- начинающий водитель оттачивает свои навыки, приобретает необходимый опыт управления транспортным средством;
- внештатные и стандартные ситуации на дорогах, требующие немедленной реакции и принятия решения, не становятся причиной проблем. Водитель подготовлен к разным вариантам развития событий;
- движение в плотном городском потоке автомобилей становится стандартной ситуацией. Даже начинающий водитель уверенно держит технику на дороге в условиях плотного трафика;
- использование поворотников становится более грамотным и правильным, если учиться управлять ими на автосимуляторе. Современные тренажеры для автошколы предусматривают такую возможность;

безошибочная парковка по зеркалам – еще один плюс работы на тренажере до начала реального вождения

Для многих курсантов автошкол первые выезды на городские трассы – это большой стресс. Автомобиль постоянно глохнет на светофорах, инструктору приходится помогать справляться с различными ситуациями. Для начинающего водителя, тренировавшегося на автосимуляторе, переход к управлению транспортным средством дается проще и легче.

ОСНОВНАЯ ЧАСТЬ

1 ПОСТАНОВКА ЗАДАЧИ

Целью ВКР является: разработка симулятора виртуальной реальности, устройства вывода данных с программы симулятора, запущенного на пользовательском компьютере.

Актуальность работы заключается в отсутствии симуляторов или комплектующих иностранного производства, а также повышением цен существующих российских симуляторов из-за введения санкционной политики в отношении Российской Федерации.

Экономическая значимость работы - использовать доступные отечественные ресурсы для создания, не уступающего по функционалу устройства, так как аналоги являются дорогими и требуют загрузки специального коммерческого ПО.

1.1 Обзор аналогов

В ходе анализа рынка были найдены следующие аналоги:

1.1.1 Складной автосимулятор GameSTUL! HomeRevolution

Представлен на рисунке 1.



Рисунок 1 - GameSTUL! HomeRevolution

GameSTUL™ Home Revolution — кресло-трансформер, выполняющее несколько разных функций. В собранном положении руль и педали убираются под кресло. Не имеет экранов и приборов для вывода информации из симуляторов движения. Производится российской компанией и продается в розницу по цене 93200 рублей

1.1.2 Многофункциональный автосимулятор GameSTUL! DUAL

Представлен на рисунке 2.



Рисунок 2 – GameSTUL! DUAL

GameSTUL! DUAL представляет из себя статичную конструкцию из круглой трубы, позволяющая менять посадку. Позволяет ощутить себя в кресле пилота болида. Изменяема геометрия и регулировки, позволяют настроить посадку и педали под пользователя. Поставляется без устройств и экранов вывода. Производится в России, продаётся в розницу по цене 138800рублей

1.1.3 XD Motion VR

Представлен на рисунке 3.



Рисунок 3 – XD Motion VR

Кресло на подвижной платформе с планетом управления и экраном (содержит 3D телевизор и специальные очки для объемного изображения), представляет собой готовое бизнес-решение, так как по заказу оборудуется купюр приёмником и может использоваться, как аттракцион. Благодаря наличию динамической платформы

ощущает все неровности дороги, удары от столкновений с машинами соперников, заносы на поворотах и динамические перегрузки.

Предполагаемая стоимость по сравнению с аналогичными устройствами около 4 млн рублей, цена суточной аренды на сайте производителя составляет 100 тыс. рублей.

1.1.4 Автотренажер Forward КАМАЗ Категория «С»

Представлен на рисунке 4.



Рисунок 4 – Автотренажёр Forward КАМАЗ категория «С»

Симулятор грузового автомобиля, от производителя Forward для категории С, представляет собой передвижную платформу с экраном, креслом, рулём и педалями. Руль имеет электроусилитель, оригинальная приборная панель.

Производится в России цена от 685 тыс. рублей.

1.1.5 Статический кабинный автотренажер Forward КАМАЗ

Представлен на рисунке 5.



Рисунок 5 – Статический кабинный автотренажёр Forward Камаз

Симулятор грузового автомобиля, от производителя Forward для категории С, представляет собой передвижную платформу на базе оригинальной кабины автомобиля КАМАЗ, с лицевыми экранами диагональю 42 дюйма и боковыми экранами с диагональю 32 дюйма. Имеется электроусилитель руля место инструктора, оригинальная приборная панель, а также система видеонаблюдения и двухсторонней связи. Производится в России цена в розничной продаже от 1,8 млн рублей.

1.1.6 Статический кабинный автотренажер КАМАЗ

Представлен на рисунке 6.



Рисунок 6 – Статический кабинный автотренажер КАМАЗ

Статический симулятор КАМАЗа на передвижной платформе, включает в себя экран, кресло педали руль, виртуальную приборную панель и возможностью смены автомобиля на экране (КАМАЗ, ГАЗ, ЗИЛ, УРАЛ). Производится в России цена базовой комплектации от 550 тыс. рублей.

1.1.7 Динамический автотренажер Forward КАМАЗ на 6-ти степенной платформе Категория "С, D, E"

Представлен на рисунке 7.



Рисунок 7 - Динамический автотренажер Forward КАМАЗ

Динамический симулятор КАМАЗ для категории С на неподвижной платформе, на базе оригинальной кабины автомобиля, с лицевыми экранами диагональю 42 дюйма и боковыми экранами с диагональю 32 дюйма. Имеется электроусилитель руля место инструктора, оригинальная приборная панель, а также система видеонаблюдения и двухсторонней связи. Динамическая 6 осевая платформа позволяет погрузиться в управление виртуальным автомобилем, понять, как будет себя вести настоящая техника. Производится в России цена в розничной продаже от 5,3 млн рублей.

1.1.8 Автотренажер Forward КАМАЗ открытого типа

Представлен на рисунке 8.



Рисунок 8 – Автотренажер Forward открытого типа

Симулятор КАМАЗ для категории С на неподвижной платформе, с лицевыми экранами диагональю 42 дюйма и боковыми экранами с диагональю 32 дюйма. Имеется электроусилитель руля, оригинальная

приборная панель. Производится в России цена в розничной продаже от 1,2 млн рублей.

1.2 Техническое задание

1.2.1 Описание симулятора виртуальной реальности

Симулятор виртуальной реальности представляет собой устройство вывода информации с персонального компьютера, из программы симулятора дорожного движения. Драйвер для симулятора через СОМ порт передаёт нужные данные, при помощи программного обеспечения они выводятся на нужные нам пины вспомогательной платы. Таким образом происходит вывод информации и управление приборами в приборной панели.

1.1.2.1 Задачи, которые предстоит решить для создания симулятора

- Построить схему электрической цепи для подключения платы управления и приборной панели
- Создать программу, позволяющую устройству получать данные с пользовательского компьютера
- Модернизировать старые аналоговые приборы для уменьшения стоимости конечного продукта
- Спроектировать элементы корпуса и крепления платы управления

1.1.2.2 Основные функциональные возможности симулятора

Основываясь на описании симуляторов и исследовании аналогов, были сформулированные задачи для устройства:

1. Задача «Создание электрической цепи подключения». Создать электрическую цепь, включающую в себя управляющую плату, а также приборы и устройства вывода сигналов.

2. Задача «Создание удобного интуитивно понятного управления». Создать программу управления, не требующую вмешательства от пользователя.

3. Задача «Модернизировать аналоговые устройства вывода»

Разработанный прототип позволит, выводить параметры из симулятора, запущенного на ПК, практически без задержки, что позволит погрузиться в управление автосимулятором.

2 ВЫБОР СРЕДСТВ РАЗРАБОТКИ

2.1 Исследование предметной области

Для создания симулятора необходимо изучить возможности управления панелью приборов и их модификации при необходимости, написать или модифицировать существующий драйвер, собрать электрическую схему для подключения платы, получающей данные и управляющей панелью приборов.

Для создания симулятора были рассмотрены разные оригинальные приборные панели от грузовых автомобилей:

1. Электронная панель приборов для КАМАЗ 5490

Представлена на рисунке 9.



Рисунок 9 – Панель Камаз 5490

Стоимость данной приборной панели на рынке бывших в употреблении запчастей порядка 50 тыс. рублей.

2. Электронная панель приборов для КАМАЗ 65115

Представлена на рисунке 10.



Рисунок 10 – Панель Камаз 65115

Стоимость данной приборной панели от 20 тыс. рублей

3. Аналоговая панель приборов для КАМАЗ 4310, 5320.

Представлена на рисунке 11.



Рисунок 11 – Панель Камаз 4310,5320

Стоимость на рынке бывших в употреблении запчастей около 3 тыс. рублей.

Опираясь на экономическую составляющую, выбор был сделан на аналоговой приборной панели, потому что модернизация всех её приборов вместе экономически выгодней приобретения электронной панели.

Были исследованы электрические схемы развязки приборной панели, а также её узлов в отдельности. Принято решение о модернизации под более низкое потребление и адаптации части узлов.

Для модернизации было решено создать цифровые модели узлов, которые получают модернизацию.

Для создания цифровых 3D модели используют два метода:

1. 3D сканирование – процесс, когда периферийное устройство, анализирует форму предмета и на его основе строит цифровую трёхмерную модель.

2. 3D моделирование – процесс разработки и создания трёхмерной модели человеком, с использованием специализированных программ. При этом модель может как соответствовать объектам из реального мира, так и быть полностью абстрактной.

1. 3D сканирование

Плюсы:

- Универсальность применения, можно использовать в любой области: производстве, медицине, дизайне разработке продуктов, реверс-инжиниринг.
- Скорость создания виртуальной модели.
- В большинстве моделей не нужен непосредственный контакт с объектом, так как происходит лазерное сканирование.

Минусы:

- Для сканирования нужна существующая модель, твердотельный объект.
- При сканировании прозрачных или бликующих объектов, нужно специальное покрытие или нанесение маркеров.
- После сканирования нужна обработка цифровой модели вручную человеком.

- Большой вес конечного файла цифровой 3D модели

2. 3D моделирование

Плюсы:

- Реалистичное отображение продукта. Объект отрисовывается максимально детально, со всеми его особенностями. Наглядно можно увидеть возможности перемещения и изменения положения предмета в пространстве, принцип действия и т.д.
- Возможность детально рассмотреть со всех сторон. Без создания изображений и изготовления объекта. 3D-модель можно изучать с разных ракурсов, поворачивая и разворачивая, как вздумается, выявляя дефекты.
- Предварительная оценка нестыковок модели, предварительная проверка, как детали будут подходить друг к другу. Моделирование в 3D дает возможность заметить такие недочеты еще на стадии проектирования, не затрачивая лишние ресурсы.
- Экономия средств. Нарисовать проект в 3D гораздо дешевле, чем сразу же изготовить образец для демонстрации.

Минусы:

- Необходимость больших временных затрат для создания моделей.
- Необходимость отслеживать взаимное расположение объектов моделирования, чтобы избежать дефектов.
- Высокие требования к аппаратной составляющей части компьютера.

Электрическая цепь для платы управления создаётся на основе схемы расположения элементов приборной панели и пинов управляющей платы.

Согласно техническому заданию, можно принять решение, что для разработки цифровой модели лучше использовать 3D сканированием, так как у нас имеются реальные элементы, которые нужно изменить.

2.2 Выбор средств разработки

Для редактирования цифровой модели, полученной после сканирования были рассмотрены следующие программы:

- 3D Studio Max — это профессиональный программный пакет, созданный компанией Autodesk, для полноценной работы с 3D-графикой, включает в себя мощный инструментарий не только для непосредственного трехмерного моделирования, но и для создания качественной анимации. В базовый пакет также входит подсистема визуализации, позволяющая добиться довольно реалистичных эффектов.

- AutoCAD - это программное обеспечение для трехмерного компьютерного моделирования от Autodesk, которое разработано для проектирования изделий, зданий, планирования производства, гражданской инфраструктуры и строительства. Оно является частью коллекции 3D CAD программ Autodesk, используемой командами разработчиков продуктов, производственными предприятиями, средствами массовой информации и индустрией развлечений, инженерами, архитекторами, преподавателями и студентами, предпринимателями, медицинскими работниками и многими другими.

- Компас 3D - это российская система трехмерного проектирования, ставшая стандартом для тысяч предприятий и десятков тысяч профессиональных пользователей, включает в себя различные дополнительные пакеты расширения, широко используется для проектирования изделий основного и вспомогательного производств в таких отраслях промышленности, как машиностроение, приборостроение,

авиастроение, судостроение, станкостроение, вагоностроение, металлургия, промышленное и гражданское строительство, товары народного потребления и т. д.

- SolidWorks - программный комплекс САПР для автоматизации работ промышленного предприятия на этапах конструкторской и технологической подготовки производства. Обеспечивает разработку изделий любой степени сложности и назначения. Включает в себя мощный инструментарий для создания цифровых моделей.

- SketchUp - программа для 3D дизайна и архитектурного проектирования. В основном используется для моделирования жилых домов, мебели, интерьера. Есть инструменты для проектирования лестниц, электропроводки, санитарно-технических коммуникаций и оборудования.

- FreeCAD – параметрическая САПР общего назначения с открытым кодом (на базе лицензии LGPLv2+). Основой геометрического моделирования твёрдых тел в FreeCAD является принцип граничного представления, в то же время имеется поддержка полигональных сеток. Геометрическим ядром FreeCAD является OpenCASCADE. Кроме задач машиностроения, FreeCAD может использоваться для таких задач, как архитектурное проектирование, или инженерный анализ методом конечных элементов. •

- Blender - профессиональное свободное и открытое программное обеспечение для создания трёхмерной компьютерной графики, включающее в себя средства моделирования, скульптинга, анимации, симуляции, рендеринга, постобработки и монтажа видео со звуком, компоновки с помощью «узлов» (Node Compositing), а также создания 2D-анимаций. В настоящее время пользуется большой популярностью среди бесплатных 3D-редакторов в связи с его быстрым стабильным развитием и технической поддержкой.

- 123D Design - был набором любительских инструментов САПР и 3D-моделирования, созданных Autodesk. По своему охвату он аналогичен Trimble SketchUp и основан на Autodesk Inventor. Помимо основных возможностей рисования и моделирования, в нем также есть поддержка сборки и ограничений и экспорт в STL. Для программы доступна также библиотека готовых блоков и объектов.

Были рассмотрены следующие вспомогательные программы для преобразования цифровой модели в код понятный для печатающего устройства, 3D принтера:

- Creality Slicer - это запатентованный программный продукт от известного производителя 3d принтеров Creality. Большинство производителей 3D-принтеров стараются создать собственное программное обеспечение для слайсинга моделей. Хотя он может показаться более упрощенным, но разработан специально для работы с 3d принтерами Creality.
- Ultimaker Cura - это слайсер 3D-моделей с открытым исходным кодом для 3D принтеров, имеет множество расширений. Это позволило расширить интеграцию со сторонними приложениями САПР.
- Picaso polygon x – это программное обеспечение, которое подготавливает цифровые 3D-модели для печати на принтерах PICASO 3D Designer и Designer PRO 250 и управляет устройством.

При выборе платы управления были рассмотрены следующие решения:

- Raspberry Pi Zero - это, по сути, такая же самая оригинальная плата Raspberry Pi Model A+, только уменьшенная до размеров платежной карты. Этот микрокомпьютер может запустить полноценный дистрибутив Linux и в то же время он достаточно дешевый.

Плата имеет точно такие же возможности, как и Raspberry Pi A+, на ней можно запустить Raspbian или любую другую совместимую операционную систему. Процессора с частотой 1 ГГц и 512 Мб оперативной памяти вряд-ли хватит на что-либо серьезное, но вы можете использовать устройство в качестве микроконтроллера, альтернативы Arduino.

Это один из самых мощных микрокомпьютеров, поэтому он используется для сложных задач, которые не всегда по плечу для Arduino, например, обработка данных и визуализация.

- NodeMCU - это очень интересный микроконтроллер, который предлагает множество возможностей, особенно учитывая его цену. В отличие от Arduino, который использует 8 битный ATMEGA с частотой 16 МГц, здесь используется чипсет ESP8266 с 32 битным процессором Tensilica Xtensa LX106 80 МГц, также здесь есть Wifi, встроенная поддержка TCP/IP, 4 мегабайта встроенного хранилища и 20 кб ОЗУ. В дополнение ко всему тут есть 10 выходов GPIO. К устройству можно подключать различные компоненты, такие как мониторы, сенсоры или сервоприводы. Как и Arduino, его очень просто использовать, достаточно написать код и загрузить его на устройство по USB. Программы пишутся на Lua, это интерпретируемый язык программирования, похожий на Python.

- Teensy - это самая популярная альтернатива Arduino, которая может использоваться для создания любого DIY проекта. Здесь есть загрузчик, с помощью которого вы можете загружать в память программу даже с USB флешки. С помощью него вы можете эмулировать любое USB устройство, а в качестве процессора используется ARM Cortex M4 с частотой 180 МГц, и 256 Кб оперативной памяти. И здесь тоже можно использовать Arduino IDE.

- Beaglebone - Устройство использует ARM Cortex V8 с частотой 700 МГц и 256 мегабайт оперативной памяти DDR2, а также флешку

объемом 4 Гб. В качестве языков программирования могут использоваться Python, C, C++, PHP, JavaScript. Устройство поддерживает установку SD карты, а также есть USB порт, через который можно подключать различные расширения, например, Ethernet или другой компьютер.

- STM32 - это дешевый 32-битный микроконтроллер, от STMicroelectronics, одна из лучших альтернатив Arduino. Здесь используется своя среда разработки Keli, а также программатор ST-Link. Устройство использует чип ARM Cortex 32-bit M3 с тактовой частотой 24 МГц и 8 кб оперативной памяти. Среди других преимуществ можно отметить низкое потребление энергии, обработку цифровых сигналов и так далее.

- Arduino Micro не сильно отличается от своих собратьев. На плате расположены 12 аналоговых вводов и 20 цифровых вводов/выводов. 7 из них могут быть использованы как ШИМ выходы. Так же имеется кварцевый генератор с частотой 16 МГц, кнопка перезагрузки и разъем микро-USB. Основное отличие ардуино микро от ардуино нано заключается в самом микроконтроллере. Здесь использован ATmega32u4. Он имеет встроенную поддержку USB соединения. Благодаря этому Arduino Micro может определяться компьютером не только как последовательный COM порт, но и как периферийное устройство. Можно сказать, что это уменьшенная копия Arduino Leonardo. Версия Arduino Pro Micro отличается меньшими размерами и имеет 22 пина. Из них 5 могут быть использованы в качестве ШИМ выводов и 4 как аналоговые входы.

- Iskra Mega это российская версия Arduino Mega. Платформа выполнена на базе более продвинутого чипа ATmega2560, имеет 54 цифровых входа/выхода (из которых 15 могут использоваться как выходы ШИМ), 16 аналоговых входов/выходов, 4 UART (аппаратные последовательные порты) могут взаимодействовать с ПК, кварцевый

генератор с частотой 16 МГц, соединение micro USB, разъем питания, разъем ICSP, и кнопка сброса.

Для создания твердотельной модели были рассмотрены следующие материалы:

PLA (Полилактид) — биоразлагаемый пластик. Производится в основном из сахарного тростника или кукурузы. Может также производиться из других натуральных продуктов, например: картофельный крахмал, целлюлоза.

Плюсы:

- Не имеет усадки при печати, что позволяет точно соответствовать размерам напечатанного изделия смоделированному.
- Не требует подогрева основания стола и не боится сквозняков при печати, значит может использоваться для печати без специальных условий печати.
- Нетоксичен, во время печати не сильно пахнет , что позволяет печатать им в квартире без использования специальной вытяжки.
- Производится из натуральных компонентов, может использоваться для контакта с пищевыми продуктами.
- Биоразлагаемый, вещи из данного пластика не наносят вреда окружающей среде при утилизации.
- Имеет хорошую твердость конечных изделий

Минусы:

- Под воздействием воздуха и ультрафиолета, как и любой натуральный материал, со временем становится более хрупким, вследствие чего не рекомендуется для длительного применения при больших физических нагрузках или использования без защитного покрытия на открытом воздухе.

- Низкая температура размягчения (50°C) — в салоне машины, оставленной на солнце в жаркий день, легко размягчается и теряет форму.
- Узкий температурный диапазон использования (-20 — $+40^{\circ}\text{C}$).
- Высокая твердость пластика затрудняет его механическую обработку.
- Пластик некоторых производителей, из-за высокого содержания остаточных мономеров, склонен к образованию пробок в цельнометаллических хотэндах.
- Исходя из достоинств и недостатков данного пластика, можем обозначить следующие способы его применения.
- 3D-печать крупногабаритных изделий.

ABS (акрилонитрилбутадиенстирол) — ударопрочный пластик, популярен в промышленности и 3D-печати. Изделия из ABS достаточно прочны, поэтому его часто используют для печати функциональных объектов, имеющих практическое применение.

Плюсы:

- Хорошее сочетание прочности и упругости позволяет использовать его для изготовления механических изделий рассчитанных на долгий срок эксплуатации.
- Широкий диапазон используемых температур позволяет эксплуатировать изделия из него в технических целях.
- Простота механической обработки, в комплексе с химическим сглаживанием поверхности недорогими растворителями типа ацетона, позволяют делать декоративные изделия или корпуса с высоким качеством поверхности.

Минусы:

- Плохо переносит воздействие ультрафиолетового излучения, желтеет на солнечном свете, что ограничивает применение неокрашенных поверхностей на улице
- Не любит сквозняков при печати, что ограничивает применение дешевых принтеров с открытым корпусом.
- Из-за относительно высокой усадки склонен к деламации (расслоению), требует наличия подогреваемого стола, без него возникают проблемы с прилипанием к столу первого слоя.
- В процессе печати может образовываться неприятных запах, печатать лучше в проветриваемом помещении, или оснащать принтер специальной системой вытяжной вентиляции, с выводом за пределы квартиры.

HIPS (высокопрочный полистирол) — достаточно мягкий пластик, создавался для использования совместно с ABS, для поддержек при двухэкструдерной 3D-печати. Этому способствовали его следующие свойства: одинаковая с ABS температура экструзии, низкая спекаемость с ABS, наличие растворителя (D-Limonene), который растворяет HIPS и не растворяет ABS.

Плюсы:

- Меньшая усадка, чем у ABS, что делает его пригодным для печати точных изделий.
- Меньшая плотность, чем у PLA, что позволяет печатать изделия, где необходима легкость конструкции.
- Мягкость поверхности, которая гарантирует простоту механической обработки.
- Матовость, которая придает эффект сглаженности изделиям.
- Температура размягчения почти как у ABS, что позволяет использовать его в уличных условиях.

Минусы:

- Как и ABS, требует подогреваемой платформы и подвержен деламинации, хоть и в меньшей степени.
- Меньшая, чем у ABS, прочность на изгиб и, как следствие, большая хрупкость изделий.
- Низкая устойчивость к ультрафиолетовому излучению, что ограничивает использование изделий на солнечном свете.

PETG (полиэтилентерефталат-гликоль) — относительно новый, по сравнению с тем же ABS, материал, но уже завоевавший заслуженное признание у 3D-печатников. Пластик достаточно ударопрочный, а спекаемость слоев получается такой, что при нагрузке изделие часто ломается против слоев, а не вдоль.

Плюсы:

- Отсутствие запаха при печати — позволяет печатать в домашних условиях не используя дополнительную вытяжку.
- Отсутствие усадки обеспечивает высокую точность размеров принтов.
- Очень сильное спекание между слоями — можно печатать тонкостенные изделия с высокой прочностью.
- Стойкость к ультрафиолету — напечатанные модели можно использовать вне помещений.
- Широкий температурный диапазон эксплуатации.
- При печати не требуется закрытая камера.
- Хорошее скольжение и ударопрочность — можно печатать шестерни, втулки и другие детали механизмов.
- Не токсичен, можно печатать изделия предназначенные для контакта с пищей.

Минусы:

- Высокая текучесть требует тщательной настройки ретрактов.
- Высокая температура печати быстро выводит из строя фторопластовую вставку в хотэнде и заставляет задуматься о переходе на цельнометаллические термобарьеры.

- Прочность и температура размягчения ниже, чем у ABS.

SBS (стиролбутадиен–стирол) — еще один из относительно новых игроков на рынке пластиков для 3D-печати. Характеризуется низкой токсичностью и усадкой, а также высокой прочностью. Основное его преимущество в его прозрачности. Изделия, напечатанные этим пластиком и обработанные сольвентом, приобретают прозрачность окрашенного стекла.

Плюсы:

- Относительно низкая усадка, позволяющая печатать в принтерах с открытым корпусом.
- Высокая адгезия к столу.
- Возможность контакта с пищевыми изделиями.
- Ударопрочность.
- Красивые цвета, позволяющие создавать уникальные предметы декора.
- Прозрачность после обработки, возможность использования в светильниках.
- Широкий диапазон температур эксплуатации, морозостойкость.
- Простота постобработки как химическими, так и механическими методами.

Минусы:

- Слабая межслойная адгезия, требует сопел с большим диаметром отверстия, либо печати со 100% заполнением.
- Относительно высокая температура печати, как и у PETG.

Основываясь на личном опыте и соответствии требованиям технического задания для создания симулятора и модернизации элементов приборной панели было выбрано российское ПО Компас 3D, аналог зарубежной платы Iskra Mega из-за наличия большого количества разъемов, которые могут пригодиться в дальнейшей модификации симулятора и обладает значительным вычислительным потенциалом. Ultimaker Cura был выбран так как он имеет множество расширений и является свободным и бесплатным ПО.

3 СОЗДАНИЕ ПРОТОТИПА СИМУЛЯТОРА

Был проведен анализ элементов приборной панели (рисунок 12) и их электрической принципиальной схемы (рисунок 13).

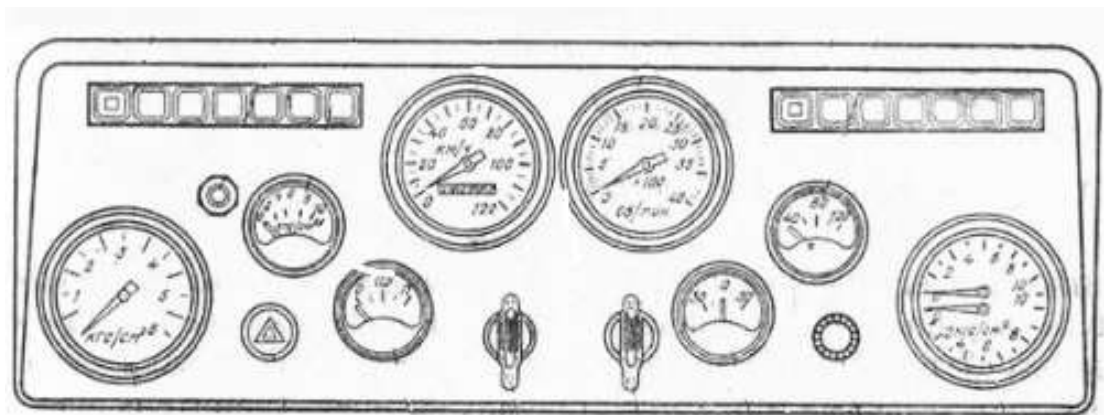


Рисунок 12 – Элементы приборной панели.

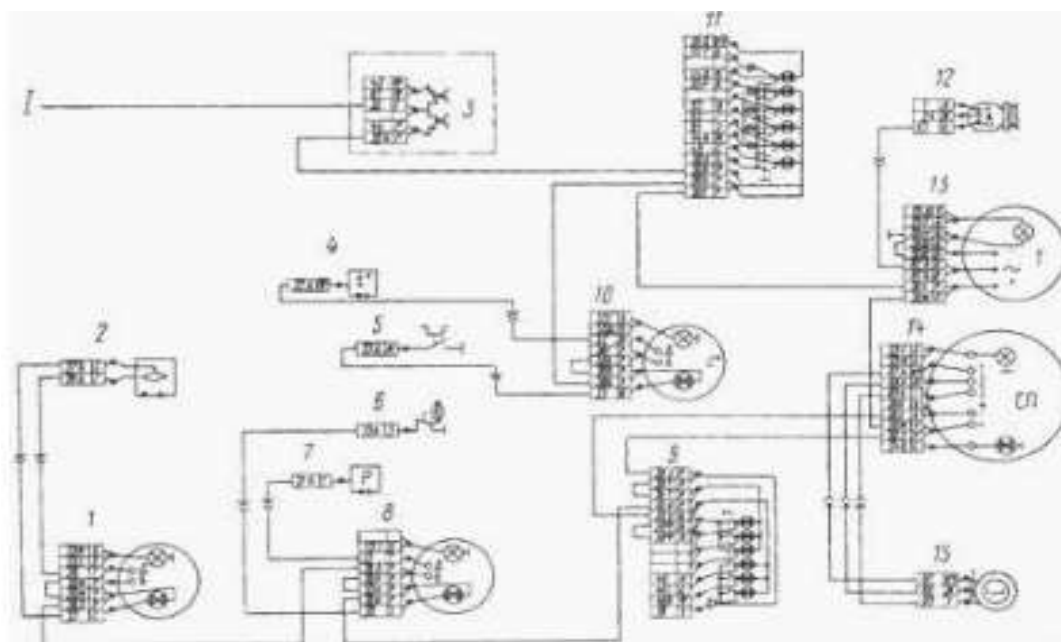


Рисунок 13 – Электрическая принципиальная схема

В соответствии, с тем что штатное питание бортовой цепи грузовых автотранспортных средств, а значит и штатное питание приборной панели составляет 24 вольта, а плата Iskra Mega питается от 12 вольт и способна управлять 5 вольтовыми линиями без дополнительных плат расширений, поэтому было решено произвести модификацию основных контрольных

приборов: спидометра, тахометра, контрольных ламп и уровня топлива, так как они имеют внутреннее устройство, которое повысит экономические затраты на создание прототипа. Спидометр и тахометр на реальном автотранспортном средстве управляются посредством генерации сигнала для управления 3 фазным двигателем, встроенным внутри, именно поэтому, если не модифицировать приборы потребуется дополнительный блок питания, реле на каждый элемент который будет использовать питание, драйверы двигателей, потому что спидометр, тахометр и указатель уровня топлива используют 3 фазные двигатели. Лампы накаливания в блоках контрольных ламп было решено заменить, на диоды так как ими проще управлять.

Модернизация элементов

Созданные на основе 3D сканирования модифицированные контрольные лампы (рисунки 14, 15, 16) были использованы оригинальные звонки контрольных ламп, печать производилась прозрачным пластиком для светопропускаемости и черным для предотвращения ошибочного восприятия, спроектированы гнезда диодов (рисунки 17,18).



Рисунок 14 – Контрольные лампы с поворотниками, блокировками мостов, указателям поднятого кузова 3D модель.

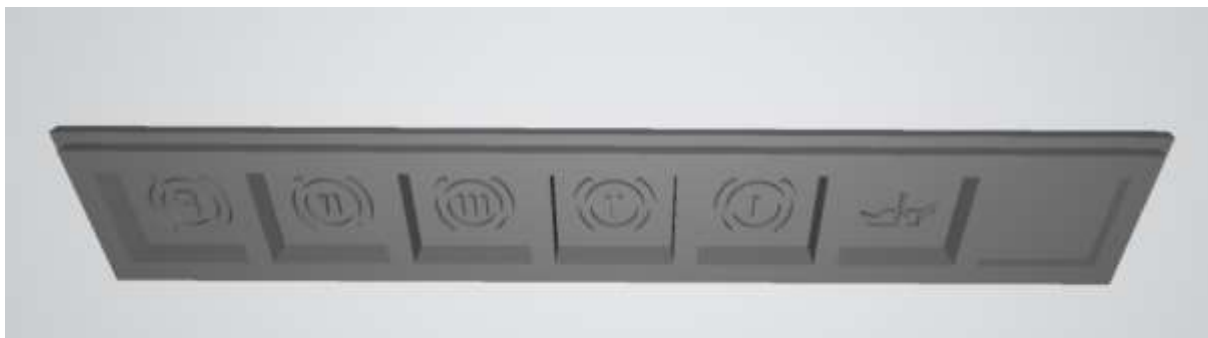


Рисунок 15 – Контрольные лампы с тормозными контурами, высокой температурой масла, стояночным тормозом 3D модель



Рисунок 16 – Готовый блок контрольных ламп.

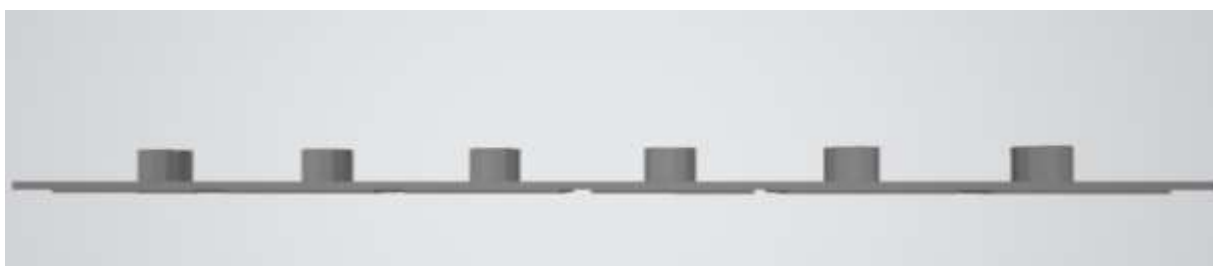


Рисунок 17 – Посадочные гнёзда диодов 3D модель

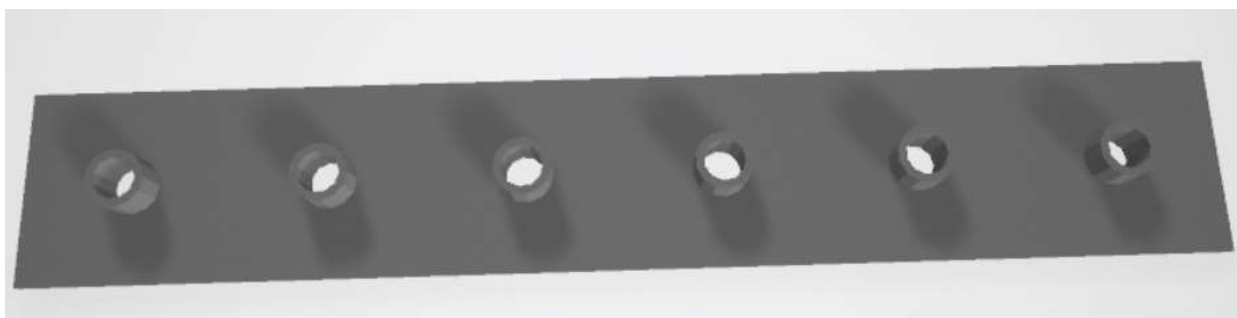


Рисунок 18 – Посадочные гнёзда диодов 3D модель

Так же были разработаны крепления сервоприводов внутри стандартных элементов панели приборов.

Крепление представлено на рисунке 19.

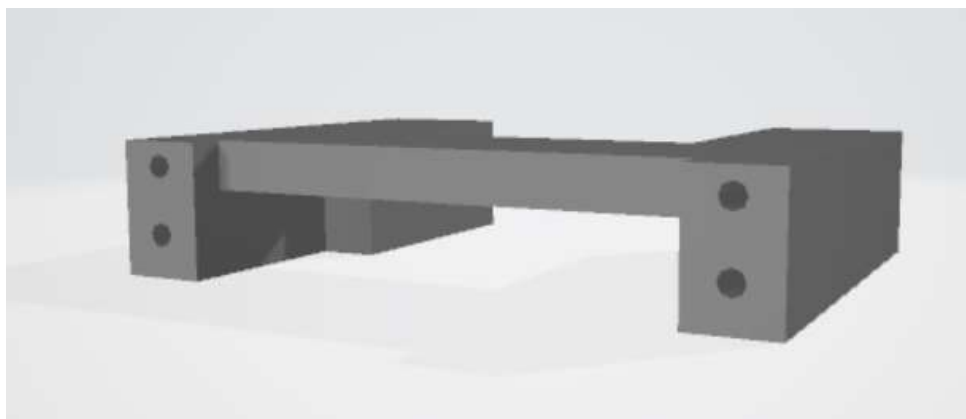


Рисунок 19 – Крепления сервоприводов

Параметры печати:

Для создания модели при помощи аддитивных технологий необходимо в специальной программе слайсере Ultimaker Cura задать параметры, используемого принтера (рисунок 20) в данном случае 3D печать будет производиться на принтере ZAV PRO. Также нужно установить параметры соответствующие выбранному материалу печати, в соответствии с рекомендацией производителя (рисунки 21, 22), температурный режим для печатающей части сопла по рекомендациям составляет от 220 до 240 градусов, скорость печати была выбранной оптимально для данной печати. Так как на джойстик будет постоянная нагрузка, был выбран параметр с полным заполнением без пустот.

ZAV PRO	
Принтер	Extruder 1
Параметры принтера	
X (Ширина)	300.0 mm
Y (Глубина)	200.0 mm
Z (Высота)	270.0 mm
Форма стола	Rectangular
Начало координат в центре	☐
Нагреваемый стол	☒
Подогреваемый объем печати	☐
Вариант G-кода	Marlin
Параметры головы	
X минимум	-26 mm
Y минимум	-21 mm
X максимум	36 mm
Y максимум	41 mm
Высота портала	999999.0 mm
Количество экструдеров	1
Применить смещения экструдера к G-Code	☒

Рисунок 20 – Параметры используемого 3D принтера

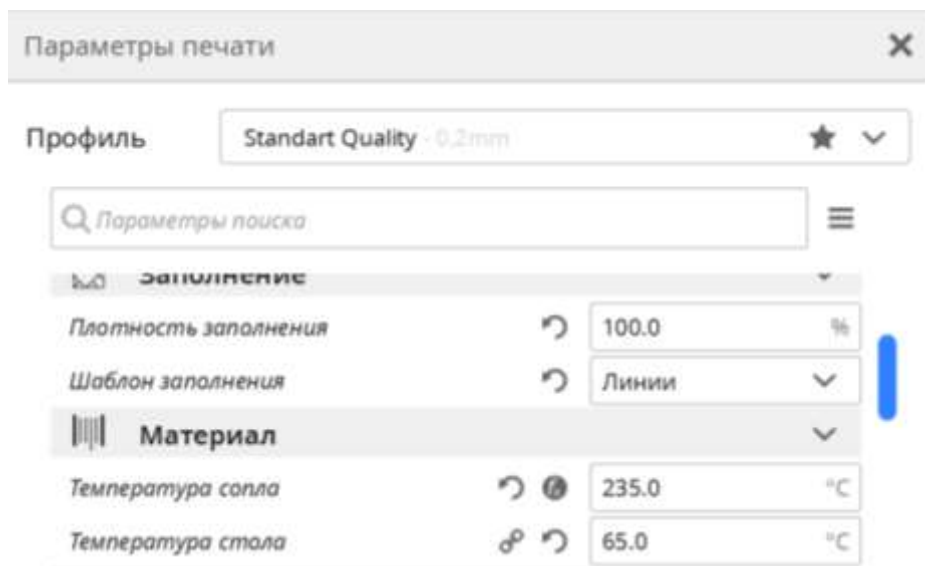


Рисунок 21 – Параметры печати, температурные режимы

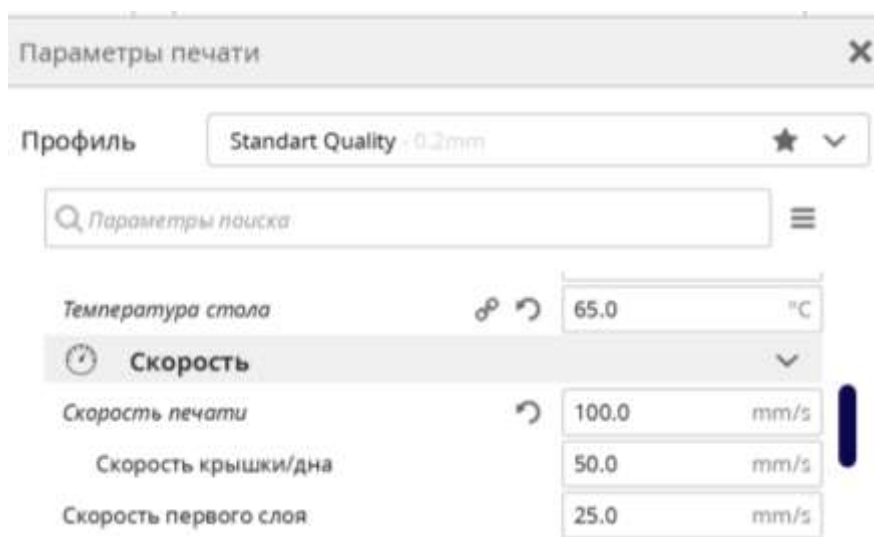


Рисунок 22 – Параметры печати, скорость печати.

Схема подключения платы

Для подключения модифицированных контрольных ламп и модернизированных контрольных приборов воспользуемся пинами Iskra Mega, аналоговые выходы A0, A1, A2 используются для вывода аналоговых сигналов управляющих контрольными приборами, встроенными в них сервоприводами. Сигнальные линии для контрольных ламп расположены с

пинов 2–13, как представлено на схеме (рисунок 23), питание моторов происходит от напряжения 3.3 вольта, диодов от 5 вольт, минусы от всех устройств выведены в пины GND.

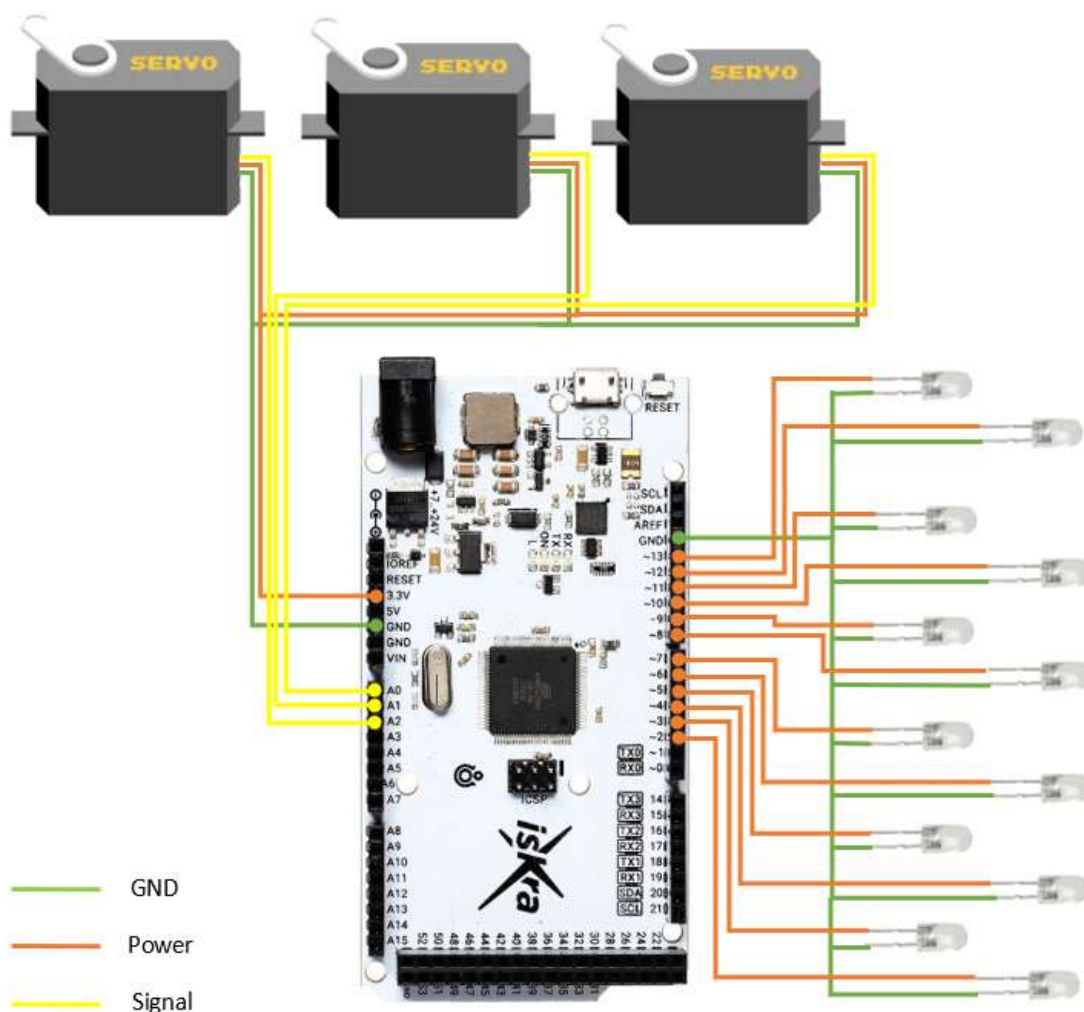


Рисунок 23 – Схема подключения элементов

Программа для взаимодействия с ПК была написана в Arduino IDE - интегрированной среде разработки, она подходит для Windows, MacOS и Linux, разработанной на C и C++ предназначенной для создания и загрузки программного кода изначально на Arduino и совместимые платы других производителей. Используемая плата Iskra Mega базируется на аналогичном чипсете, поэтому эта среда разработки нам тоже подходит. Arduino IDE – эта интегрированная среда разработки имеет множество библиотек

программного обеспечения, которые в свою очередь предоставляют множество общих процедур ввода и вывода и взаимодействия с ПК.

Приложение симулятор для ПК:

Для тестирования нашего автосимулятора было выбрано одно из самых популярных приложений Euro Truck Simulator 2.

В данном приложении разработчики постарались максимально приблизить действия программы к реальной ситуации на дорогах. Трафик во многих случаях ведёт себя неожиданно. Во время движения по дорогам компьютер выполняет не только правила дорожного движения, но и ряд других приёмов, которые взяты из реальной жизни. Например, когда игрок выезжает на главную дорогу со второстепенной, виртуальный водитель может пропустить его, предварительно дав знак, помигав фарами дальнего света и снизив скорость. Так же присутствуют временные и постоянные дорожные знаки, ограничивающие скоростной режим. Максимальная скорость всех грузовиков в игре по умолчанию ограничена технически до 90 км/ч (отключается в настройках игры). В исключительных случаях она может увеличиваться при движении на спусках и снижаться при движении на подъёмах, при работающем круиз-контроле, однако на ровной дороге скорость возвращается до заданной. В приложении представлены следующие производители грузовых автомобилей DAF, Iveco, MAN, Mercedes-Benz, Renault, Scania и Volvo. Так же разработчики этого симулятора добавили инструменты разработчика, который позволяет разработчикам и пользователям передавать данные из игры в любые сторонние приложения посредством API Windows. Иначе пришлось полностью использовать промежуточную программу, которая позволяла бы выполнять потоковую передачу данных посредством использования текстовых файлов, хранящихся в системе пользователя. Это создает ненужную нагрузку на запоминающее устройство пользователя, не говоря уже о количестве перезаписей, которые могут повредить твердотельные накопители.

Этот подключаемый модуль после настройки, передает поток данных через файл с отображением в памяти. Это специальный файловый поток Windows, который полностью находится в оперативной памяти и может быть прочитан несколькими приложениями.

Схема взаимодействия программы, драйвера и платы управления:

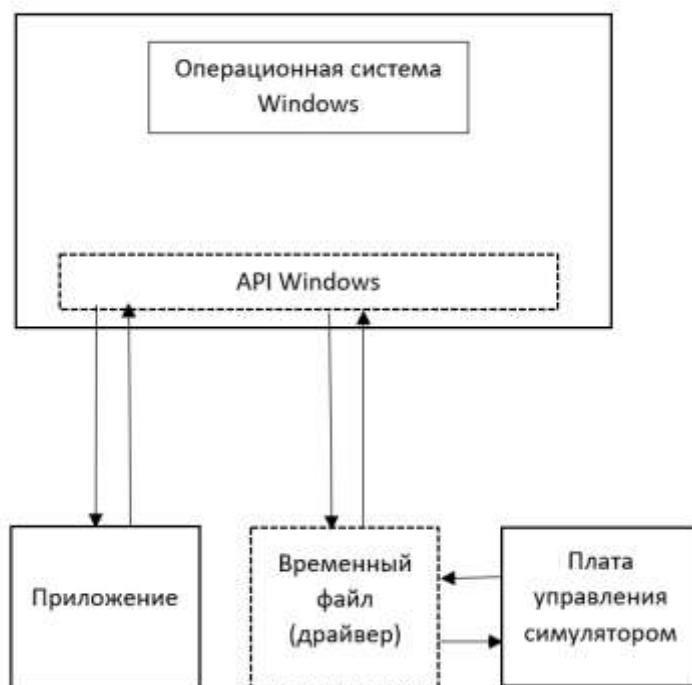


Рисунок 24 – Схема взаимодействия

При запуске приложение, запускается и драйвер (приложение Б) расположенный в корне приложения, он взаимодействует с API Windows, отправляя информационный поток на плату по соответствующему порту.

Код программы:

Были исследованы файлы симулятора и драйвера. В результате обнаружены необходимые нам переменные, с которыми и будем в дальнейшем взаимодействовать (рисунок 25).

```

REG_CHAN(speed, float);
REG_CHAN(engine_rpm, float);
REG_CHAN(engine_gear, s32);
REG_CHAN(parking_brake, bool);
REG_CHAN(motor_brake, bool);
REG_CHAN(brake_air_pressure, float);
REG_CHAN(brake_air_pressure_warning, bool);
REG_CHAN(brake_air_pressure_emergency, bool);
REG_CHAN(brake_temperature, float);
REG_CHAN(fuel, float);
REG_CHAN(fuel_warning, bool);
REG_CHAN(fuel_average_consumption, float);
REG_CHAN(oil_pressure, float);
REG_CHAN(oil_pressure_warning, bool);
REG_CHAN(oil_temperature, float);
REG_CHAN(water_temperature, float);
REG_CHAN(water_temperature_warning, bool);
REG_CHAN(battery_voltage, float);
REG_CHAN(battery_voltage_warning, bool);
REG_CHAN(electric_enabled, bool);
REG_CHAN(engine_enabled, bool);
REG_CHAN(light_lblinker, bool);
REG_CHAN(light_rblinker, bool);
REG_CHAN(light_parking, bool);
REG_CHAN(light_low_beam, bool);
REG_CHAN(light_high_beam, bool);
REG_CHAN(light_brake, bool);
REG_CHAN(light_reverse, bool);
REG_CHAN(odometer, float);

```

Рисунок 25 – Переменные из симулятора

В результате выявленных переменных, были назначены следующие пины платы управления для панели симулятора (рисунок 26)

```

const int SPEEDO_PIN    = A0; // Управление спидометр
const int RPM_PIN       = A1; // Управление тахометр
const int FUEL_PIN      = A2; // Уровень топлива
const int HIGH_BEAM     = 2;  // Дальний свет
const int LOW_BEAM      = 3;  // Ближний свет
const int LEFT_INDICATOR = 4;  // Левый поворотник
const int RIGHT_INDICATOR = 5; // Правый поворотник
const int PARKING_BRAKE = 6;  // Ручной тормоз
const int FUEL_WARNING   = 7; // Мало топлива
const int MOTOR_BRAKE    = 8; // Торможение двигателем
const int BRAKE_AIR_2    = 9; // Ошибка давления в тормозной системе 1 контур
const int BRAKE_AIR_1    = 10; // Ошибка давления в тормозной системе 2 контур
const int BATTERY_LOW    = 11; // Аккумуляторная батарея разряжена
const int OIL_TEMP       = 12; // Высокая температура масла
const int WATER_TEMP     = 13; // Высокая температура охлаждающей жидкости

```

Рисунок 26 – Назначение пинов

Модернизирован драйвер с параметрами, передаваемыми на плату управления, параметры скорректированы в соответствии с используемыми в симуляторе двигателями, на основе аналитических расчётов, так же указан соответствующий COM порт для создания соединения UART для взаимодействия платы управления симулятором с ПК (рисунок 27)

```
comport str COM6
factor_speed          flt    4.35
factor_engine_rpm     flt    0.04
factor_brake_air_pressure flt    1.0
factor_brake_temperature flt    1.0
factor_fuel_ratio      flt   80.0
factor_oil_pressure    flt    1.0
factor_oil_temperature flt    1.0
factor_water_temperature flt    1.0
factor_battery_voltage flt    5.0
```

Рисунок 27 – Параметры выходов и порт.

Была реализована программа для платы управления на языке Arduino C она представляет собой язык C++ с фреймворком Wiring, он имеет некоторые отличия по части написания кода, который компилируется и собирается с особенностями, облегчающими написание работающей программы - имеется набор библиотек, включающий в себя функции и объекты. Программа (приложение А) загружена на плату управление и подключена посредством кабеля к ПК.

Устройство обнаруживается операционной системой в штатном режиме

(Рисунок 28)

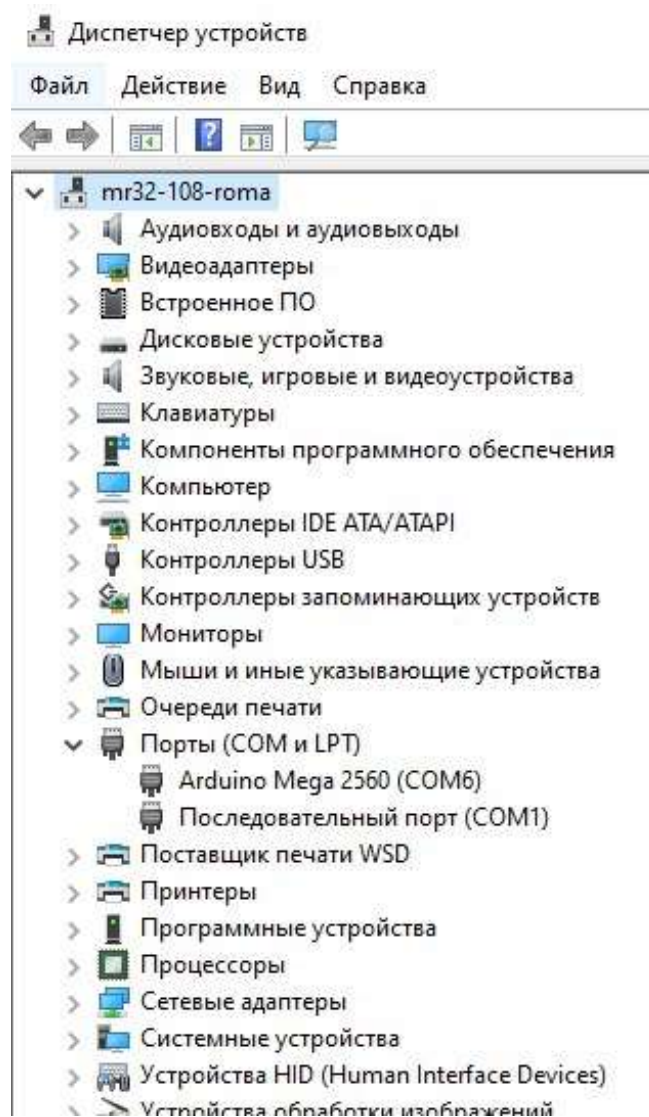


Рисунок 28 – Определение устройства системой

В результате при запуске симулятора мы видим уведомление о том, что используются функции разработчика и параметры приложения будут передаваться (рисунок 29).

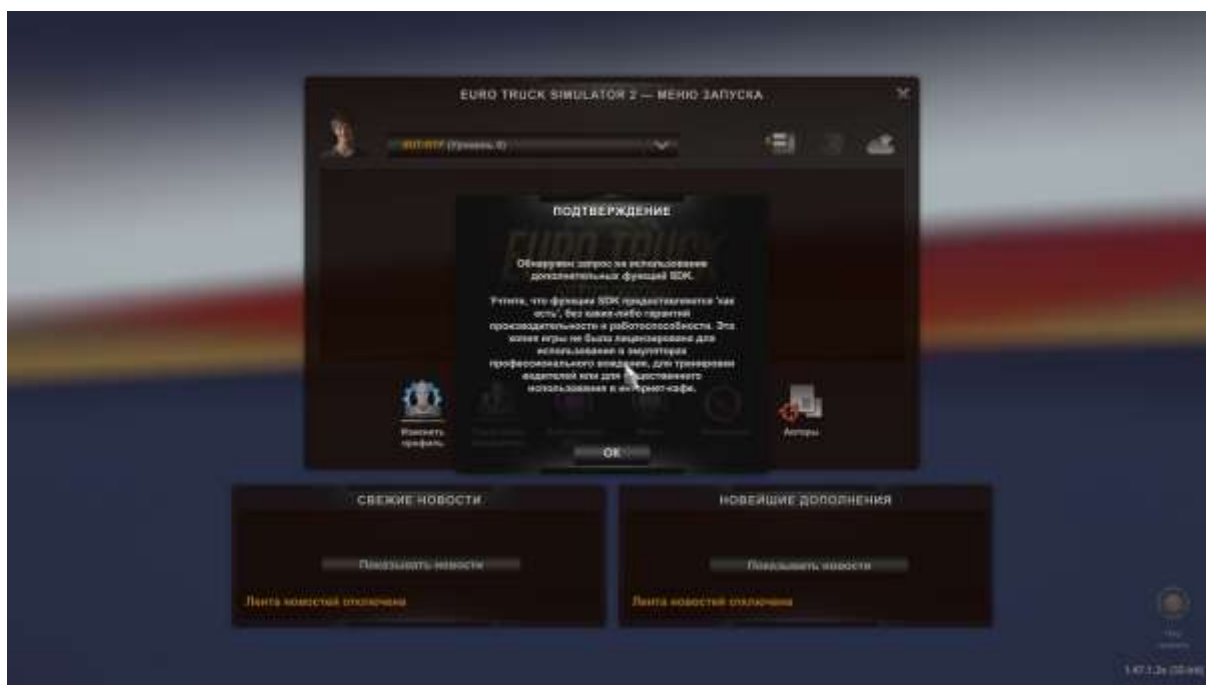


Рисунок 29 – Вход в приложение с настроенными функциями передачи данных

Запущенный симулятор с параметрами и реальная панель представлены на (рисунках 30, 31).



Рисунок 30 – оригинальная приборная панель, с включённой аварийной сигнализацией, фарами, запущенным двигателем и активированным стояночным тормозом.



Рисунок 31 – симулятор с включённой аварийной сигнализацией, фарами, запущенным двигателем и активированным стояночным тормозом.

В сравнении с аналогами (таблица 1) разработанный симулятор не уступает в функционале, устройством вывода является реальная приборная панель от КамАЗ 4310, 5320.

Наименование	Экран	Оригинальное устройство вывода	Устройство ввода	Загрузка ПО	Наличие динамической платформы	Цена
Складной автосимулятор GameSTUL! HomeRevolution v.3	нет	нет	есть	Можно использовать после подключения	нет	93200 рублей
Многофункциональный автосимулятор GameSTUL! DUAL	нет	нет	есть	Можно использовать после подключения	нет	138800 рублей
XD Motion VR	есть	есть	есть	Можно использовать после подключения	да	4 млн рублей

Автотренажер Forward КАМАЗ Категория «С»	есть	есть	есть	Можно использовать после подключения	нет	685000 рублей
Статический кабинный автотренажер Forward КАМАЗ	есть	есть	есть	Можно использовать после подключения	нет	1,8 млн рублей
Статический кабинный автотренажер КАМАЗ	есть	есть	есть	Можно использовать после подключения	нет	550000 рублей
Динамический автотренажер Forward КАМАЗ на 6-ти степенной платформе Категория "С, D, E"	есть	есть	есть	Можно использовать после подключения	есть	5,3 млн рублей
Автотренажер Forward КАМАЗ открытого типа	есть	есть	есть	Можно использовать после подключения	нет	1,2 млн рублей
Разработанный симулятор	есть	есть	есть	Можно использовать после подключения	нет	150000 рублей

Таблица 1

ЗАКЛЮЧЕНИЕ

В результате исследования и разработки удалось реализовать симулятор, взаимодействующий с программой на ПК и выводящей данные на реальную приборную панель (рисунки 30, 31)

Разработанный симулятор не имеет динамической платформы, но не уступает по функционалу, имея стоимость выше, чем у устройств без вывода информации в соотношении с имеющимися аналогами, что позволяет утверждать об экономической выгоде создания такого устройства. Данный автосимулятор увеличивает вовлеченность и погружение в симулятор вождения, это непосредственный плюс для пользователя. Данным автосимулятором заинтересовались и он был внедрен (приложение В).

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Герберт Шилдт. С++ для начинающих. Шаг за шагом// ЭКОМ Паблишерз, 2013. №2. С.640.
2. Указать API Windows [Электронный ресурс] – Режим доступа: <https://learn.microsoft.com/ru-ru/windows/win32/apiindex/windows-api-list>.
3. В.А.Безруков Win32 API Программирование/учебное пособие – СПбГУ ИТМО, 2009. С.90.
4. Arduino Mega 2560 [Электронный ресурс] //Arduino [Электронный ресурс]//Сайт пользователей «Arduino». - Режим доступа: <http://arduino.ru/Hardware/ArduinoBoardMega2560>
5. Arduino [Электронный ресурс]//Сайт программной среды и фирмы «Arduino». - Режим доступа: <https://www.arduino.cc>
6. Программирование Arduino [Электронный ресурс]//Сайт пользователей «Arduino». - Режим доступа: <http://arduino.ru/Reference>
7. Прерывания на микроконтроллере Arduino [Электронный ресурс]// Сайт по машинному обучению и программированию. - Режим доступа: <http://robotosha.ru/arduino/arduino-interrupts.html>
8. Аналоговые измерения с Arduino [Электронный ресурс]//Сайт пользователей«Arduino». - Режим доступа: <http://robotosha.ru/arduino/analogmeasurements-arduino.html>
9. Функции Arduino [Электронный ресурс]//Сайт <http://codius.ru>. Режим доступа: <http://codius.ru/articles.html>
10. Чагина А. В., Большаков В. П. 3D-моделирование в КОМПАС-3D версий v17 и выше/учебное пособие для вузов, 2021. С. 256.
11. Руководство по ремонту автомобилей КамАЗ. [Электронный ресурс]// <https://autoinfo24.ru/rukovodstva-po-remontu/otchestvennyye-avtomobili/kamaz>

12. Iskra Mega [Электронный ресурс]// Сайт <https://amperka.ru/>.
Режим доступа: <https://amperka.ru/product/iskra-mega#docs>

Код платы управления

```
#include <Servo.h>

const int SPEEDO_PIN    = A0; // Управление спидометр
const int RPM_PIN       = A1; // Управление тахометр
const int FUEL_PIN      = A2; // Уровень топлива
const int HIGH_BEAM     = 2;  // Дальний свет
const int LOW_BEAM      = 3;  // Ближний свет
const int LEFT_INDICATOR = 4;  // Левый поворотник
const int RIGHT_INDICATOR = 5; // Правый поворотник
const int PARKING_BRAKE = 6;  // Ручной тормоз
const int FUEL_WARNING  = 7;  // Мало топлива
const int MOTOR_BRAKE   = 8;  // Торможение двигателем
const int BRAKE_AIR_2   = 9;  // Ошибка давления в тормозной системе 1
контур
const int BRAKE_AIR_1   = 10; // Ошибка давления в тормозной системе 2
контур
const int BATTERY_LOW   = 11; // Аккумуляторная батарея разряжена
const int OIL_TEMP      = 12; // Высокая температура масла
const int WATER_TEMP    = 13; // Высокая температура охлаждающей
жидкости

Servo speedo;
Servo rpm;
Servo fuel;
#define PACKET_SYNC 0xFF
#define PACKET_VER 2
#define SERVO_DIR_NORMAL false
#define SERVO_DIR_INVERT true

int serial_byte;
void setup()
{
    Serial.begin(115200);

    speedo.attach(SPEEDO_PIN);
    speedo.write(180);
    rpm.attach(RPM_PIN);
    rpm.write(180);
    fuel.attach(FUEL_PIN);
    fuel.write(180);
```

```
pinMode(LEFT_INDICATOR, OUTPUT);
pinMode(RIGHT_INDICATOR, OUTPUT);
pinMode(PARKING_BRAKE, OUTPUT);
pinMode(FUEL_WARNING, OUTPUT);
pinMode(LOW_BEAM, OUTPUT);
pinMode(HIGH_BEAM, OUTPUT);
pinMode(MOTOR_BRAKE, OUTPUT);
pinMode(BRAKE_AIR_2, OUTPUT);
pinMode(BRAKE_AIR_1, OUTPUT);
pinMode(BATTERY_LOW, OUTPUT);
pinMode(OIL_TEMP, OUTPUT);
pinMode(WATER_TEMP, OUTPUT);
```

```
digitalWrite(LEFT_INDICATOR, 0);
digitalWrite(RIGHT_INDICATOR, 0);
digitalWrite(PARKING_BRAKE, 0);
digitalWrite(FUEL_WARNING, 0);
digitalWrite(LOW_BEAM, 0);
digitalWrite(HIGH_BEAM, 0);
digitalWrite(MOTOR_BRAKE, 0);
digitalWrite(BRAKE_AIR_2, 0);
digitalWrite(BRAKE_AIR_1, 0);
digitalWrite(BATTERY_LOW, 0);
digitalWrite(OIL_TEMP, 0);
digitalWrite(WATER_TEMP, 0);
```

```
delay(500);
```

```
speedo.write(0);
rpm.write(0);
fuel.write(0);
digitalWrite(LEFT_INDICATOR, 1);
digitalWrite(RIGHT_INDICATOR, 1);
digitalWrite(PARKING_BRAKE, 1);
digitalWrite(FUEL_WARNING, 1);
digitalWrite(LOW_BEAM, 1);
digitalWrite(HIGH_BEAM, 1);
digitalWrite(MOTOR_BRAKE, 1);
digitalWrite(BRAKE_AIR_2, 1);
digitalWrite(BRAKE_AIR_1, 1);
digitalWrite(BATTERY_LOW, 1);
digitalWrite(OIL_TEMP, 1);
digitalWrite(WATER_TEMP, 1);
```

```

delay(500);

speedo.write(180);
rpm.write(180);
fuel.write(180);
digitalWrite(LEFT_INDICATOR, 0);
digitalWrite(RIGHT_INDICATOR, 0);
digitalWrite(PARKING_BRAKE, 0);
digitalWrite(FUEL_WARNING, 0);
digitalWrite(LOW_BEAM, 0);
digitalWrite(HIGH_BEAM, 0);
digitalWrite(MOTOR_BRAKE, 0);
digitalWrite(BRAKE_AIR_2, 0);
digitalWrite(BRAKE_AIR_1, 0);
digitalWrite(BATTERY_LOW, 0);
digitalWrite(OIL_TEMP, 0);
digitalWrite(WATER_TEMP, 0);

}

void read_serial_byte_set_servo(Servo& servo, bool invert)
{
    serial_byte = Serial.read();
    serial_byte = (serial_byte < 0) ? 0 : ((serial_byte > 180) ? 180 : serial_byte);
    if (invert)
        servo.write(180 - serial_byte);
    else
        servo.write(serial_byte);
}

void skip_serial_byte()
{
    (void)Serial.read();
}

void digitalWriteFromBit(int port, int value, int shift)
{
    digitalWrite(port, (value >> shift) & 0x01);
}

void loop()
{
    if (Serial.available() < 16)

```

```

    return;

serial_byte = Serial.read();
if (serial_byte != PACKET_SYNC)
    return;

serial_byte = Serial.read();
if (serial_byte != PACKET_VER)
{
    return;
}

read_serial_byte_set_servo(speedo, SERVO_DIR_INVERT); // Speed
read_serial_byte_set_servo(rpm, SERVO_DIR_INVERT); // RPM
read_serial_byte_set_servo(fuel, SERVO_DIR_INVERT); //FUEL

skip_serial_byte();
skip_serial_byte();
skip_serial_byte();
skip_serial_byte();
skip_serial_byte();
skip_serial_byte();
skip_serial_byte();

serial_byte = Serial.read();
digitalWriteFromBit(LIGHT_P, serial_byte, 6);
digitalWriteFromBit(LEFT_INDICATOR, serial_byte, 5);
digitalWriteFromBit(RIGHT_INDICATOR, serial_byte, 4);
digitalWriteFromBit(LOW_BEAM, serial_byte, 3);
digitalWriteFromBit(HIGH_BEAM, serial_byte, 2);
digitalWriteFromBit(LIGHT_B, serial_byte, 1);
digitalWriteFromBit(LIGHT_R, serial_byte, 0);

serial_byte = Serial.read();
digitalWriteFromBit(PARKING_BRAKE, serial_byte, 7);
digitalWriteFromBit(MOTOR_BRAKE, serial_byte, 6);
digitalWriteFromBit(BRAKE_AIR_2, serial_byte, 5);
digitalWriteFromBit(BRAKE_AIR_1, serial_byte, 4);
digitalWriteFromBit(FUEL_WARNING, serial_byte, 3);
digitalWriteFromBit(BATTERY_LOW, serial_byte, 2);
digitalWriteFromBit(OIL_TEMP, serial_byte, 1);
digitalWriteFromBit(WATER_TEMP, serial_byte, 0);

```

```
serial_byte = Serial.read();  
}
```

Код драйвера

```
#define WINVER 0x0500
#define _WIN32_WINNT 0x0500
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#include <stdarg.h>
#include <sstream>

#include <scssdk_telemetry.h>
#include <eurotrucks2/scssdk_telemetry_eut2.h>

#include "serial.hpp"
#include "options.hpp"

static const float METERS_PER_SEC_TO_MILES_PER_HOUR = 2.2369f;
static const float METERS_PER_SEC_TO_KM_PER_HOUR = 3.6f;

// Game log file
scs_log_t game_log = NULL;

// COM port file
Serial serial_port;

unsigned long last_update = 0;

// Packet
#define PACKET_MAX_SIZE 720 // Can send a maximum of 720 bytes per
frame at 20 FPS over a 115200 bps line
#define PACKET_SYNC 0xFF
#define PACKET_VER 2

unsigned char packet[PACKET_MAX_SIZE];
```

```
#define PACKBOOL(A,B,C,D,E,F,G,H) \
(((unsigned char)(A) << 7) | ((unsigned char)(B) << 6) | \
((unsigned char)(C) << 5) | ((unsigned char)(D) << 4) | \
((unsigned char)(E) << 3) | ((unsigned char)(F) << 2) | \
((unsigned char)(G) << 1) | ((unsigned char)(H)))
```

```
// Telemetry data
```

```
struct telemetry_state_t
```

```
{
```

```
    float speed; // Meters per Second
```

```
        float engine_rpm; // RPM
```

```
    int engine_gear; // >0 Forward, 0 Neutral, <0 Reverse
```

```
    bool parking_brake;
```

```
    bool motor_brake;
```

```
    float brake_air_pressure; // PSI
```

```
    bool brake_air_pressure_warning;
```

```
    bool brake_air_pressure_emergency;
```

```
    float brake_temperature;
```

```
    float fuel; // Liters
```

```
    bool fuel_warning;
```

```
    float fuel_average_consumption; // Liters / KM
```

```
    float fuel_capacity; // Liters
```

```
    float oil_pressure; // PSI
```

```
    bool oil_pressure_warning;
```

```
    float oil_temperature; // Celsius
```

```
    float water_temperature; // Celsius
```

```
    bool water_temperature_warning;
```

```
    float battery_voltage; // Volts
```

```
    bool battery_voltage_warning;
```

```

bool electric_enabled;
bool engine_enabled;

bool light_lblinker;
bool light_rblinker;
bool light_parking;
bool light_low_beam; // Dipped headlights
bool light_high_beam; // Full headlights
bool light_brake;
bool light_reverse;

float odometer; // Kilometres

} telemetry;

```

Options option_file;

```

unsigned char float_to_byte(float value)
{
    return
        (value > 254.0f) ? 254 : // Sync byte is 255 so avoid that
        ((value < 0.0f) ? 0 :
        (unsigned char)(value));
}

```

```

void send_empty_packet()
{
    memset(packet, 0, PACKET_MAX_SIZE);
    packet[0] = PACKET_SYNC;
    packet[1] = PACKET_VER;

    serial_port.write(packet, 16);
}

```

```

SCSAPI_VOID telemetry_frame_end(const scs_event_t /*event*/, const void
*const /*event_info*/, const scs_context_t /*context*/)

```

```

{
    if (!serial_port.is_valid())
        return;

    const unsigned long now = GetTickCount();
    const unsigned long diff = now - last_update;
    if (diff < 50)
        return;

    last_update = now;

    const float speed_mph = telemetry.speed *
METERS_PER_SEC_TO_MILES_PER_HOUR;
    const float speed_kph = telemetry.speed *
METERS_PER_SEC_TO_KM_PER_HOUR;

    const float fuel_ratio = telemetry.fuel / telemetry.fuel_capacity;

    unsigned idx = 0;

    memset(packet, 0, PACKET_MAX_SIZE);

#define PUT_BYTE(X) packet[idx++] = (unsigned char)(X)
#define PUT_BYTE_FLOAT(X) packet[idx++] = (unsigned
char)(float_to_byte(X))
#define GETFLT OPT(X) (option_file.get_option_float(X, 1.0f))

    // Packet header
    PUT_BYTE(PACKET_SYNC);
    PUT_BYTE(PACKET_VER);

    // Convert data for output by servos
    // Adjust the constant values to map to servo range

    PUT_BYTE_FLOAT(fabs(telemetry.speed) *
GETFLT OPT("factor_speed")); // Absolute value for when reversing

```

```

    PUT_BYTE_FLOAT(telemetry.engine_rpm *
    GETFLTOPT("factor_engine_rpm"));
    PUT_BYTE_FLOAT(telemetry.brake_air_pressure *
    GETFLTOPT("factor_brake_air_pressure"));
    PUT_BYTE_FLOAT(telemetry.brake_temperature *
    GETFLTOPT("factor_brake_temperature"));
    PUT_BYTE_FLOAT(fuel_ratio * GETFLTOPT("factor_fuel_ratio"));
// Fuel level
    PUT_BYTE_FLOAT(telemetry.oil_pressure *
    GETFLTOPT("factor_oil_pressure"));
    PUT_BYTE_FLOAT(telemetry.oil_temperature *
    GETFLTOPT("factor_oil_temperature"));
    PUT_BYTE_FLOAT(telemetry.water_temperature *
    GETFLTOPT("factor_water_temperature"));
    PUT_BYTE_FLOAT(telemetry.battery_voltage *
    GETFLTOPT("factor_battery_voltage"));

// Pack data for LEDs into bytes

// Truck lights
PUT_BYTE(PACKBOOL(
    0, telemetry.light_parking,
    telemetry.light_lblinker, telemetry.light_rblinker,
    telemetry.light_low_beam, telemetry.light_high_beam,
    telemetry.light_brake, telemetry.light_reverse));

// Warning lights
PUT_BYTE(PACKBOOL(
    telemetry.parking_brake, telemetry.motor_brake,
    telemetry.brake_air_pressure_warning,
telemetry.brake_air_pressure_emergency,
    telemetry.fuel_warning, telemetry.battery_voltage_warning,
    telemetry.oil_pressure_warning, telemetry.water_temperature_warning));

// Enabled flags
PUT_BYTE(PACKBOOL(
    0,0,0,0,0, 0,

```

```

telemetry.electric_enabled, telemetry.engine_enabled));

// Build string for LCD display

std::stringstream ss;

ss.width(3);
ss << abs(int(speed_mph));
ss << " MPH";

ss << "  G ";

if (telemetry.engine_gear > 0)
{
    ss << "D";
    ss.width(2);
    ss << telemetry.engine_gear;
}
else if (telemetry.engine_gear == 0)
{
    ss << "N ";
}
else
{
    ss << "R";
    ss.width(2);
    ss << abs(telemetry.engine_gear);
}

ss << "\n";

ss.width(3);
ss << abs(int(speed_kph));
ss << " KPH";

ss << "  F ";

```

```

ss.width(3);
ss << int(fuel_ratio * 100.0f) << "%";

ss.width(3);
ss << abs(int(speed_kph));

std::string display_str(ss.str());

// Write length of string
PUT_BYTE(display_str.size());
// Followed by string
display_str.copy((char*)&packet[idx], PACKET_MAX_SIZE - idx);
idx += display_str.size();

serial_port.write(packet, idx);
}

```

```

SCSAPI_VOID telemetry_store_float(const scs_string_t /*name*/, const
scs_u32_t /*index*/, const scs_value_t *const value, const scs_context_t context)
{
    assert(value);
    assert(value->type == SCS_VALUE_TYPE_float);
    assert(context);
    *static_cast<float *>(context) = value->value_float.value;
}

```

```

SCSAPI_VOID telemetry_store_bool(const scs_string_t /*name*/, const
scs_u32_t /*index*/, const scs_value_t *const value, const scs_context_t context)
{
    assert(value);
    assert(value->type == SCS_VALUE_TYPE_bool);
}

```

```

    assert(context);
    *static_cast<bool *>(context) = (value->value_bool.value != 0);
}

SCSAPI_VOID telemetry_store_s32(const scs_string_t /*name*/, const
scs_u32_t /*index*/, const scs_value_t *const value, const scs_context_t context)
{
    assert(value);
    assert(value->type == SCS_VALUE_TYPE_s32);
    assert(context);
    *static_cast<int *>(context) = value->value_s32.value;
}

SCSAPI_VOID telemetry_configuration(const scs_event_t /*event*/, const void
*const event_info, const scs_context_t /*context*/)
{
    const struct scs_telemetry_configuration_t *const info = static_cast<const
scs_telemetry_configuration_t *>(event_info);

    for (const scs_named_value_t *current = info->attributes; current->name;
++current)
    {

#define GET_CONFIG(PROPERTY, TYPE) \
    if (strcmp(SCS_TELEMETRY_CONFIG_ATTRIBUTE_ ## PROPERTY, \
current->name) == 0) \
    { telemetry.PROPERTY = current->value.value_ ## TYPE.value; }

        GET_CONFIG(fuel_capacity, float);

    }
}

void get_cwd(std::string& str)
{
    char buffer[256];
    GetCurrentDirectory(256, buffer);
    str.assign(buffer);
}

```

```

}

SCSAPI_RESULT scs_telemetry_init(const scs_u32_t version, const
scs_telemetry_init_params_t *const params)
{
    if (version != SCS_TELEMETRY_VERSION_1_00) {
        return SCS_RESULT_unsupported;
    }

    const scs_telemetry_init_params_v100_t *const version_params =
static_cast<const scs_telemetry_init_params_v100_t *>(params);
    game_log = version_params->common.log;

    game_log(SCS_LOG_TYPE_message, "Plugin initialising");

    std::string cwd;
    get_cwd(cwd);

    game_log(SCS_LOG_TYPE_message, (std::string("Plugin CWD: ") +
cwd).c_str());

    std::string option_filepath(cwd + "\\plugins\\dash_plugin.txt");
    if (!option_file.read_file(option_filepath))
    {
        game_log(SCS_LOG_TYPE_error, (std::string("Error reading settings file: ")
+ option_filepath).c_str());
        return SCS_RESULT_generic_error;
    }

    const std::string com_port = option_file.get_option_string("comport",
"COM3");
    game_log(SCS_LOG_TYPE_message, (std::string("Using serial port: ") +
com_port).c_str());

    // Open COM port
    std::string errmsg;
    if (!serial_port.open(com_port, errmsg))
    {

```

```

    game_log(SCS_LOG_TYPE_error, errmsg.c_str());
    return SCS_RESULT_generic_error;
}

send_empty_packet();

// Register for in game events
bool registered =
    (version_params->register_for_event(
        SCS_TELEMETRY_EVENT_frame_end, telemetry_frame_end, NULL) ==
    SCS_RESULT_ok) &&
    (version_params->register_for_event(
        SCS_TELEMETRY_EVENT_configuration,      telemetry_configuration,
        NULL) == SCS_RESULT_ok);

// Register for truck channels

#define REG_CHAN(CHANNEL, TYPE) \
    registered &= (version_params->register_for_channel( \
        SCS_TELEMETRY_TRUCK_CHANNEL_ ## CHANNEL, SCS_U32_NIL, \
        SCS_VALUE_TYPE_ ## TYPE, \
        SCS_TELEMETRY_CHANNEL_FLAG_none, telemetry_store_ ## TYPE, \
        &telemetry.CHANNEL) == SCS_RESULT_ok)

REG_CHAN(speed, float);
REG_CHAN(engine_rpm, float);
REG_CHAN(engine_gear, s32);
REG_CHAN(parking_brake, bool);
REG_CHAN(motor_brake, bool);
REG_CHAN(brake_air_pressure, float);
REG_CHAN(brake_air_pressure_warning, bool);
REG_CHAN(brake_air_pressure_emergency, bool);
REG_CHAN(brake_temperature, float);
REG_CHAN(fuel, float);
REG_CHAN(fuel_warning, bool);
REG_CHAN(fuel_average_consumption, float);
REG_CHAN(oil_pressure, float);
REG_CHAN(oil_pressure_warning, bool);

```

```

REG_CHAN(oil_temperature,      float);
REG_CHAN(water_temperature,    float);
REG_CHAN(water_temperature_warning,  bool);
REG_CHAN(battery_voltage,      float);
REG_CHAN(battery_voltage_warning,  bool);
REG_CHAN(electric_enabled,      bool);
REG_CHAN(engine_enabled,        bool);
REG_CHAN(light_lblinker,        bool);
REG_CHAN(light_rblinker,        bool);
REG_CHAN(light_parking,         bool);
REG_CHAN(light_low_beam,        bool);
REG_CHAN(light_high_beam,       bool);
REG_CHAN(light_brake,           bool);
REG_CHAN(light_reverse,         bool);
REG_CHAN(odometer,             float);

```

```

if (!registered)
{
    game_log(SCS_LOG_TYPE_error, "Unable to register callbacks");
    return SCS_RESULT_generic_error;
}

```

```

memset(&telemetry, 0, sizeof(telemetry));

```

```

return SCS_RESULT_ok;
}

```

```

SCSAPI_VOID scs_telemetry_shutdown(void)

```

```

{
    send_empty_packet();
    serial_port.close();

```

```

    game_log(SCS_LOG_TYPE_message, "Plugin shutdown");
    game_log = NULL;
}

```

```
BOOL APIENTRY DllMain(HMODULE /*module*/, DWORD reason_for_call,
LPVOID /*reseved*/)
{
    if (reason_for_call == DLL_PROCESS_DETACH)
    {
        serial_port.close();
    }
    return TRUE;
}
```

В государственную аккредитационную
комиссию УРФУ

620002, Екатеринбург, ул. Мира, д. 32.

Акт о внедрении

Настоящим подтверждаем, что результаты диссертационного исследования Гришина Р.Ю. на тему: «Разработка симулятора виртуальной реальности» обладают актуальностью, представляют практический интерес и будут использованы при обучении на стендах нашей автошколы.

Директор ООО «Драйвер»

Веремехин И. Ю.

