МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**
**Высшая школа электроники и компьютерных наук**
**Кафедра системного программирования**

РАБОТА ПРОВЕРЕНА
Рецензент
Руководитель отдела «Системы управления» ООО «ДСТ-Урал»
_____ А.Р. Ишбулатов
«___»_____ 2025 г.

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой, д.ф.-м.н., профессор
_____ Л.Б. Соколинский
«___»_____ 2025 г.

# Разработка Android–приложения приборной панели для дисплейного модуля трактора

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 09.04.04.2025.308-557.ВКР

Научный руководитель,
доцент кафедры СП, к.ф.-м.н.
_____ А.Т. Латипова

Автор работы,
студент группы КЭ-228
_____ И.С. Захаров

Ученый секретарь
(нормоконтролер)
_____ И.Д. Володченко
«___»_____ 2025 г.

Челябинск, 2025 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
**«Южно-Уральский государственный университет**
**(национальный исследовательский университет)»**
**Высшая школа электроники и компьютерных наук**
**Кафедра системного программирования**

УТВЕРЖДАЮ
Зав. кафедрой СП
_____ Л.Б. Соколинский
27.01.2025 г.

**ЗАДАНИЕ**
**на выполнение выпускной квалификационной работы магистранта**
студенту группы КЭ-228
Захарову Ивану Сергеевичу,
обучающемуся по направлению
09.04.04 «Программная инженерия»

1. **Тема работы** (утверждена приказом ректора от 21.04.2025 г. № 648-13/12) Разработка Android-приложения приборной панели для дисплейного модуля трактора.
2. **Срок сдачи студентом законченной работы:** 02.06.2025 г.
3. **Исходные данные к работе**

3.1. Чернышев С.А. Основы Dart. // MADTeacher, 2021. 215 с.
3.2. Flutter documentation [Электронный ресурс] URL: https://docs.flutter.dev (дата обращения: 27.01.2025 г.).
3.3. Choi K. The Art of Dart: Master the Dart Programming Language. / Kenneth Choi, 2024. 182 с.

4. **Перечень подлежащих разработке вопросов**

4.1. Выполнить анализ предметной области.
4.2. Сформировать требования к приложению.
4.3. Выполнить проектирование приложения: описать архитектуру и принципиальную модель.
4.4. Реализовать приложение.
4.5. Провести тестирование приложения.

5. **Дата выдачи задания:** 27.01.2025 г.

**Научный руководитель,**
доцент кафедры СП, к.ф.-м.н. А.Т. Латипова

**Задание принял к исполнению** И.С. Захаров

**ОГЛАВЛЕНИЕ**

ВВЕДЕНИЕ ..... 4
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ ..... 6
1.1. Обзор аналогов ..... 6
1.2. Анализ основных технологических решений ..... 11
2. ФОРМИРОВАНИЕ ОСНОВНЫХ ТРЕБОВАНИЙ ..... 14
2.1. Функциональные требования ..... 14
2.2. Нефункциональные требования ..... 19
3. ПРОЕКТИРОВАНИЕ ..... 20
3.1. Архитектура предлагаемого решения ..... 20
3.1.1. Протокол UDS ..... 22
3.1.2. Многоуровневый обработчик данных шины CAN ..... 22
3.1.3. Универсальные виджеты ..... 24
4. РЕАЛИЗАЦИЯ ..... 25
4.1. Приборная панель, «плавающий» режим и камера ..... 25
4.2. Руководство ..... 28
4.3. Обновление приложения и ПО контроллера ..... 29
4.4. Все приложения ..... 31
4.5. Кэширование данных ..... 31
4.6. Уровни доступа ..... 32
4.7. Настройки ..... 33
4.7.1. Настройки машины ..... 34
4.7.2. Переменные ..... 35
4.7.3. Приложение ..... 35
4.7.4. Журнал событий ..... 37
4.8. Ошибки ..... 38
5. ТЕСТИРОВАНИЕ ..... 40
ЗАКЛЮЧЕНИЕ ..... 45
ЛИТЕРАТУРА ..... 46
ПРИЛОЖЕНИЕ. Типовые страницы приложения ..... 48

# ВВЕДЕНИЕ

## Актуальность

Одним из важнейших элементов практически всех современных автомобилей и самоходной техники, например, бульдозеров, кабелеукладчиков, погрузчиков, тракторов и т. д. является приоритетно цифровой дисплейный модуль. Данные о текущем состоянии машины поступают в модуль напрямую от датчиков (в случае аналогового дисплейного модуля) или от электронных блоков управления (далее – ЭБУ) и используются для управления исполнительными механизмами и устройствами [1].

На самоходной технике наиболее часто используются ЭБУ комбинированного типа, совмещающие в себе блоки управления двигателем, трансмиссией и навесным оборудованием, например, ножом кабелеукладчика, отвалом бульдозера и т. д.

Загружаемое в ЭБУ программное обеспечение (далее – ПО), называемое прошивкой, регламентирует алгоритм работы блока. ЭБУ может передавать ошибки в электронных системах и показатели датчиков в реальном времени, принимать различные настройки машины и запросы на выполнение тех или иных функций, необходимых для функционирования электронных систем машины [1].

Для взаимодействия с ЭБУ и обеспечения работоспособности перечисленных выше функций существуют приложения дисплейного модуля машины, представляющие из себя программное обеспечение с графическим пользовательским интерфейсом.

Предприятию требуется приложение приборной панели для дисплейного модуля нового сельскохозяйственного трактора модели FT-7.

Актуальность проекта обусловлена заказом предприятия.

## Постановка задачи

Целью выпускной квалификационной работы является разработка лаунчер-приложения приборной панели для дисплейного модуля сельско-

хозяйственного трактора FT-7 на базе Android 10. Для достижения поставленной цели необходимо решить следующие задачи:

1) провести анализ интерфейсов и функционала аналогичных программ дисплейных модулей других производителей самоходной техники, выявить их достоинства и недостатки, сделать вывод;

2) провести анализ существующих технологий разработки мобильных приложений для операционной системы Android и выбрать из них оптимальные для выполнения выпускной квалификационной работы, обосновать выбор;

3) определить функциональные и нефункциональные требования к разрабатываемому приложению;

4) разработать структурно-функциональную схему проекта;

5) выполнить программную реализацию проекта;

6) провести тестирование разработанного приложения на корректность выполнения задач, выполнить отладку.

**Структура и содержание работы**

Работа состоит из введения, пяти глав, заключения, списка литературы и приложения. Объем работы составляет 55 страниц, объем списка литературы – 15 источников.

В первой главе рассматриваются аналогичные дисплейные приложения схожих машин, их преимущества и недостатки.

Вторая глава посвящена определению требований к приложению с учетом технического задания.

В третьей главе проводится проектирование приложения.

В четвертой главе описывается процесс реализации приложения.

В последней главе проводится функциональное тестирование разработанного приложения и описываются его результаты.

# 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1. Обзор аналогов

Приложение приборной панели колесного трактора Fendt 716 Vario позволяет оператору полностью контролировать настройки и состояние трактора. Приложение отображает показания датчиков от контроллера, переменные процесса и ошибки, позволяет редактировать параметры контроллера, тем самым настраивая поведение машины и задавая условия работы, проводить калибровки оборудования, предоставляет функционал мультимедиа, а также может расширяться дополнительными программными модулями в зависимости от комплектации трактора.

Интерфейс приложения дисплейного модуля трактора Fendt 716 Vario представлен на рисунке 1.

Рисунок 1 – Интерфейс приложения дисплейного модуля трактора Fendt 716 Vario

Достоинства приложения:

1) возможность создания и кастомизации страниц, т.е. выбор определенных «виджетов» с настройками и соответствующими показометрами;

2) создание, редактирование и выполнение макросов – последовательностей выполнения тех или иных функций трактора;

3) вывод изображения с 6 камер (максимально);

4) создание и редактирование рабочих профилей – различных конфигураций настроечных параметров машины, между которыми оператор может переключаться в зависимости от работ;

5) все «виджеты», вкладки и показометры имеют соответствующие картинки стандарта ISO 7000;

6) программирование специальных дополнительных кнопок на выполнение последовательностеи действий (более «ручной» аналог макросов);

7) базовый функционал управления рабочим полем: определение границ поля и линии пути, отображение заданных параметров на карте при помощи спутниковой навигации;

8) наличие «помощника разворота»: система позволяет настроить параметры разворота в соответствии с шириной и смещением орудия, размерами поля и линией пути, отобразить полученный маршрут на карте поля и следовать намеченной траектории без участия оператора;

9) удаленный доступ к приложению.

Недостатки приложения:

1) «перегруженность» интерфейса и очень большая вложенность меню: неопытный пользователь может запутаться в функционале и долго искать нужное;

2) отсутствие всплывающих «подсказок» и предупреждений;

3) невозможность представления данных различных датчиков в другом виде.

Приложение приборной панели трактора John Deere 9R предоставляет оператору схожие с приложениями других современных тракторов, включая вышеупомянутый Fendt 716 Vario, возможности взаимодействия с машиной. Последние модели тракторов John Deere оснащаются несколь-

кими дисплейными модулями, поэтому некоторый функционал приложения главного модуля, такой как мультимедиа, отображение ламп и главных показометров, может быть делегирован приложениям второстепенных дисплейных модулей. За дополнительную плату становятся доступны опциональные программные модули, такие как AutoTrac (спутниковая система автовождения по заданному маршруту на поле) и Section Control (система автоматического контроля секций гидрораспределителя в зависимости от контура поля) [2].

Интерфейс приложения главного дисплейного модуля серии GEN 4 трактора John Deere 9R представлен на рисунке 2.

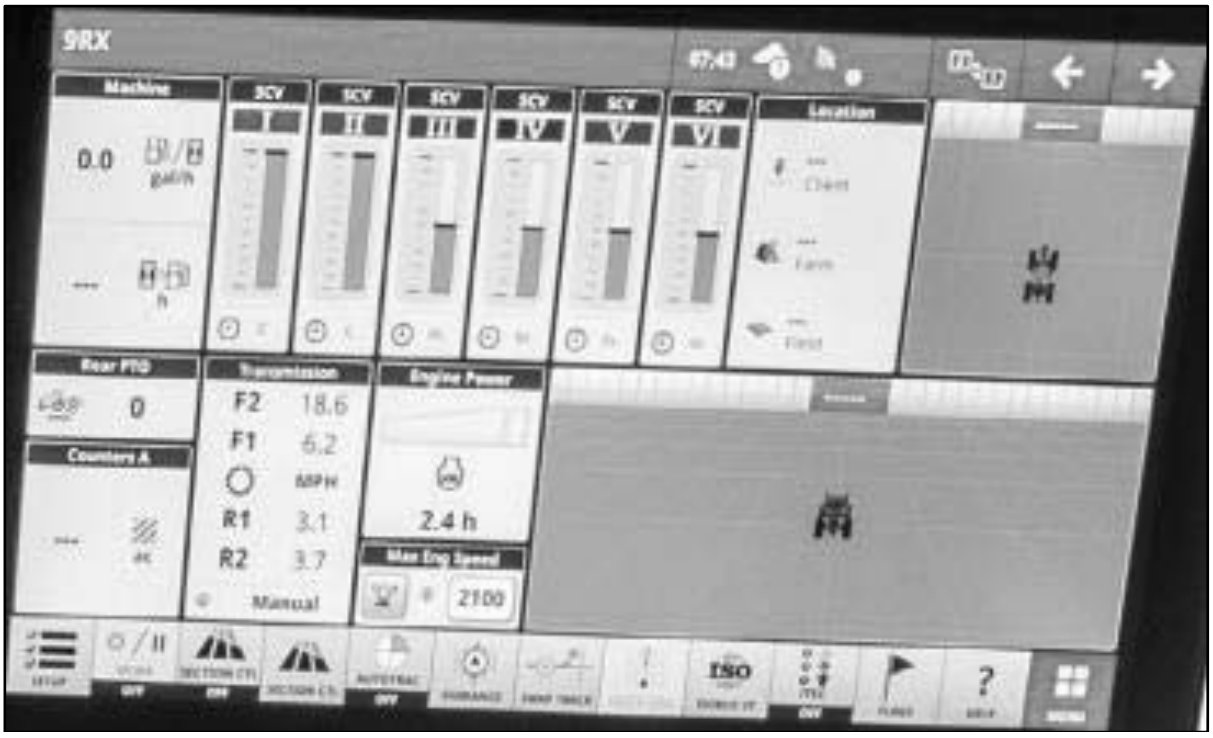


Рисунок 2 – Интерфейс приложения главного дисплейного модуля серии GEN 4 трактора John Deere 9R

Достоинства приложения:

1) возможность создания и кастомизации страниц, т.е. выбор определенных «виджетов» с настройками и соответствующими показометрами для отображения на странице;

2) создание, редактирование и выполнение макросов – последовательностей выполнения тех или иных функций трактора;

3) вывод изображения с 6 камер (максимально), возможность запрограммировать вывод изображения с той или иной камеры по триггеру или циклично с интералом;

4) интегрированная система мультимедиа: связь с телефоном по Bluetooth для вызозов и проигрывания музыки;

5) встроенное руководство по эксплуатации приложения с поиском по ключевым словам;

6) базовый функционал управления рабочим полем: определение границ поля и линии пути;

7) автоматический контроль секций гидрораспределителя в зависимости от контура поля;

8) удаленный доступ к приложению.

Недостатки приложения:

1) «перегруженность» интерфейса: неопытный пользователь может запутаться в функционале и долго искать нужное;

2) невозможность менять внешний вид показометров.

Дисплейное приложение бульдозеров Liebherr 8-го поколения рассматривается как референс к разрабатываемому проекту. Оно предоставляет весь необходимый функционал для работы на бульдозере, за исключением сервисных функций, настроек и калибровок, которые былли вынесены в отдельное сервисное приложение. Так как компания Liebherr не производит сельскохозяйственных тракторов, то за референс был взят интерфейс приложения для бульдозеров, так как по своей сути бульдозер более близок к гусеничному трактору, чем остальная техника, и он больше остальных аналогов похож концептуально на уже разработанные предприятием решения.

Интерфейс приложения дисплейного модуля бульдозера Liebherr PR756 Gen 8 представлен на рисунке 3.

Рисунок 3 – Интерфейс приложения дисплейного модуля бульдозера Liebherr PR756 Gen 8

Достоинства приложения:

1) простой и удобный интерфейс, напоминающий приборную панель легкового автомобиля;

2) стандартная легкочитаемая панель ламп и окно ошибок;

3) небольшая вложенность меню: много функционала в быстром доступе на каждой вкладке;

4) картинки стандарта ISO 7000.

Недостатки приложения:

1) отсутствие возможности персонализации;

2) устаревший непривлекательный интерфейс;

3) много полезного функционала отсутствует, так как было выделено в отдельное сервисное приложение.

Стоит отметить, что дисплейные модули вышеперечисленной техники работают на операционной системе Linux, а значит им практически недоступны привычные мобильные приложения, возможности Android и са-

мостоятельное расширение функционала модуля оператором или владельцем, поэтому, в целях расширения возможностей конечного пользователя, на предприятии используются дисплейные модули на базе Android, приложением-лаунчером которого должно стать разрабатываемое решение.

Вывод: ввиду запуска в производство нового сельскохозяйственного трактора, предприятию требуется приложение приборной панели для дисплейного модуля этого трактора, лишенное критических недостатков обозреваемых систем и перенявшее их важные достоинства. Также важно учесть, что некоторые функциональные возможности рассмотренных приложений будут отсутствовать в разрабатываемом проекте ввиду необходимости разработки программных и аппаратных решений на стороне контроллера для поддержки данного функционала, что будет постепенно осуществляться в будущем.

### 1.2. Анализ основных технологических решений

На сегодняшний день в сфере разработки мобильных приложений для ОС Android существует множество кроссплатформенных решений, каждое из которых обладает своими особенностями. Среди наиболее популярных платформ можно выделить React Native, Xamarin, Ionic, Flutter и другие. В данном обзоре мы рассмотрим ключевые из них, проанализировав их основные достоинства и недостатки.

React Native, созданный и поддерживаемый Facebook, – это кроссплатформенный фреймворк для разработки приложений, что позволяет использовать единую кодовую базу на всех платформах. React Native функционирует как фоновый процесс на устройстве, интерпретируя JavaScript-код и взаимодействуя с нативной платформой через асинхронный мост, передавая данные в сериализованном виде [3].

Достоинства React Native:

1) высокая производительность готового приложения, близкая к нативной;

2) подробная документация и большое количество учебных материалов;

3) моментальная «горячая перезагрузка», позволяющая сразу видеть изменения в коде без полной перекомпиляции, что ускоряет разработку и отладку.

Недостатки React Native:

1) в JavaScript отсутстует null safety, что может привести к непредвиденным ошибкам во время работы;

2) порог вхождения выше, чем у Flutter.

Flutter – кроссплатформенный фреймворк, разработанный компанией Google с открытым исходным кодом. Flutter использует собственный движок Flutter Engine, предоставляющий API для взаимодействия с нативными функциями, исключая необходимость в дополнительном мосте и сериализации данных [4].

Достоинства Flutter:

1) высокая производительность готового приложения. Вероятно, более высокая, чем у React Native, ввиду отсутствия моста для взаимодействия с нативными функциями и необходимости сериализации;

2) подробная документация и большое количество учебных материалов;

3) активное и быстрорастущее сообщество, готовое помочь в решении проблем;

4) моментальная «горячая перезагрузка», значительно ускоряющая разработку и тестирование;

5) низкий порог входа.

Недостатки Flutter:

1) некоторые функции недостаточно оттестированы и отлажены, что может приводить к нестабильности;

2) большое количество открытых сообществом багов в репозитории.

Кроссплатформенный фреймворк Xamarin (.NET MAUI), разработанный компанией Xamarin (позже куплена Microsoft), является расширением платформы .NET. Обращение к нативным функциям Android из приложения происходит посредством прослойки Managed Callable Wrappers (MCW), а обращение Android Runtime (ART) к приложению Xamarin – через Android Callable Wrappers (ACW) [5].

Достоинства Xamarin:

1) «live reload», ускоряющая разработку и тестирование;

2) низкий порог входа, особенно для разработчиков, уже знакомых с C# и .NET.

Недостатки Xamarin:

1) небольшая и постоянно уменьшающаяся популярность у сообщества разработчиков;

2) скудное количество учебного материала, небольшая и неполная документация;

3) сильно ограниченное число пакетов и расширений по сравнению с Flutter и React Native.

**Вывод по первой главе**

При выборе фреймворка для проекта ключевыми критериями являются скорость разработки, удобство использования и поддержка сообщества. Flutter выделяется среди аналогов благодаря высокой производительности, активному развитию и низкому порогу входа. Кроме того, личный опыт работы с этим фреймворком и его растущая популярность стали решающими факторами при выборе. Единственным языком разработки в Flutter является Dart.

## 2. ФОРМИРОВАНИЕ ОСНОВНЫХ ТРЕБОВАНИЙ

### 2.1. Функциональные требования

Техническое задание регламентирует следующие функциональные требования к приложению.

1. Связь с шиной CAN посредством интерфейса USB или Bluetooth с возможностью автоподключения.

2. Должны быть реализованы страницы: «Приборная панель», «Система позиционно-силового регулирования (EHR)», «Гидравлическая система рабочего оборудования», «Система охлаждения FanDrive», «Система круиз-контроля», «Калибровки и настройки машины», «Натяжение гусеничных лент», «Настройка BOM» (вал отбора мощности), «Конфигурация света», «Настройки приложения», «Все приложения», «Ошибки», «Журнал событий», «Камера», «Переменные», «Руководство по эксплуатации» и «Обновление ПО».

3. Страница «Калибровки и настройки машины» должна включать в себя следующие страницы: «ARES (Система автоматического регулирования оборотов ДВС)», «ECO-режим», «Гидромоторы», «Гидронасосы», «Органы управления», «Информация о машине» и «Обновление ПО», и интерфейс предоставления сервисного уровня доступа для настройки скрытых от оператора сервисных параметров.

4. Разрабатываясь как Android-лаунчер, приложение должно иметь возможность просматривать и открывать другие сторонние приложения дисплейного модуля и переходить в их Android-настройки.

5. В верхней части экрана располагается привычная водителям и операторам самоходной техники панель ламп.

6. Отображение уведомлений об ошибках, предупреждениях и информационных сообщениях.

7. Должен быть реализован журнал событий, записывающий все зафиксированные ошибки, действия оператора в приложении и внутренние события.

8. Возможность обновить приложение, ПО и драйвер контроллера системы управления (далее – СУ).

9. Компактный «плавающий» режим приложения для отображения основных индикаторов, ламп и ошибок поверх открытых сторонних приложений дисплейного модуля, необходимый для непрерывного контроля состояния машины.

10. Автоматическое подключение к шине CAN, восстановление связи при потере соединения.

11. Кэширование полученной информации от контроллера для быстрой загрузки приложения при следующих запусках.

12. Возможность связи с контроллером СУ с помощью Bluetooth через установленный на шине Bluetooth-модуль для возможности запуска на других мобильных устройствах в случае неисправности дисплейного модуля.

13. Использование протокола UDS для двунаправленной коммуникации между приложением и контроллером.

Функциональные требования к отдельным страницам и окнам приложения представлены в таблице 1.

Таблица 1 – Функциональные требования к страницам, окнам приложения

| **Название окна / страницы** | **Функциональные требования** |
|---|---|
| Приборная панель | 1. Отображение основных показометров и цифровых значений параметров состояния машины, таких как температура масла в трансмиссии, температура охлаждающей жидкости ДВС, скорость машины, уровень топлива и т.д.<br>2. Окно изображения с камер(ы) с возможностью регулировки яркости, насыщенности и т.д.<br>3. Отображение ламп рядом с соответствующими показометрами и цифровыми значениями параметров состояния машины |
| «Плавающий» режим приложения | 1. Автоматический запуск при приостановке и отключение при возобновлении работы главного приложения<br>2. Отображение основной необходимой информации с приборной панели<br>3. Отображение активных ошибок с возможностью их принятия |
| Калибровки и настройки машины | 1. Переход на выбранную вложенную страницу<br>2. Сохранение всех измененных параметров<br>3. Сброс всех измененных параметров<br>4. Сброс и получение повышенного уровня доступа |

| Название окна / страницы | Функциональные требования |
|---|---|
| Все приложения | 1. Список всех приложений дисплейного модуля<br>2. Открытие, удаление и переход в Android-настройки выбранного приложения |
| Уровень доступа | 1. Ввод пароля для уровней доступа Мастер и Сервис с валидацией<br>2. Сброс уровня доступа на Оператор |
| Ошибки | 1. Просмотр и принятие активных ошибок СУ, ДВС и БУС (блок управления светом)<br>2. Просмотр (все уровни доступа) и очистка (уровень доступа Сервис) сохраненных ошибок |
| Натяжение гусеничных лент | Вывод значений параметров натяжения правой и левой гусеничных лент |
| Система позиционно-силового регулирования (EHR) | 1. Выбор режима регулировки: смешанный, по уровню пробуксовки<br>2. Для смешанного режима регулировки: настройка чувствительности и допуска<br>3. Для режима регулировки по уровню пробуксовки: настройка чувствительности и порога пробуксовки<br>4. Общие настройки: задание транспортного и рабочего положения, скорости подъема и опускания рабочего оборудования |
| Гидравлическая система рабочего оборудования | 1. Включение и отключение управления секциями гидрораспределителя рабочего оборудования<br>2. Выбор режима секции: стандартный, постоянный (константный), по времени с настройкой времени<br>3. Синхронизация управления секциями<br>4. Настройка % подачи тока в прямом и обратном направлениях движения секций |
| Система охлаждения FanDrive | 1. Выбор режима охлаждения: выкл., автоматический, максимальное принудительное охлаждение, зимний режим<br>2. Выбор одного из предопределенных в контроллере режима автопродувки от засоров, возможность задать интервал и длительность продувки в ручном режиме<br>3. Включение принудительной продувки<br>4. Отображение основных показометров, логически связанных с системой охлаждения (процент управления FanDrive, температура охлаждающей жидкости, ток на FanDrive и т.д.) |
| Система круиз-контроля | 1. Задание скорости круиз-контроля<br>2. Задание ширины орудия<br>3. Задание оборотов регулятором оборотов двигателя<br>4. Включение системы автоматического регулирования оборотов ARES<br>5. Отображение текущих и заданных скорости и оборотов двигателя |
| Настройка ВОМ | 1. Выбор режима работы ВОМ<br>2. Отображение текущих оборотов ВОМ и соответствующих ламп неисправностей ВОМ |
| ECO-режим | 1. Включение и отключение ECO-режима двигателя<br>2. Настройка времени активации режима<br>3. Отображение соответствующих ламп и показометров |

| Название окна / страницы | Функциональные требования |
|---|---|
| Конфигурация света | 1. Задание конфигурации рабочего света<br>2. Состояние фонарей (вкл/выкл, короткое замыкание, обрыв) |
| Переменные | 1. Отображение всех переменных для текущего уровня доступа и обновление их значений в реальном времени<br>2. Возможность выбора отдельных меню или переменных для индивидуального отображения<br>3. Управление выборками: сохранение текущей, удаление, переключение |
| Настройки приложения | 1. Переключение темы приложения<br>2. Выбор интерфейса связи с контроллером (USB или Bluetooth)<br>3. Возможность очистки кэша приложения<br>4. Основная информация о приложении: компания-разработчик, версия с возможностью проверить наличие и установить обновление, условия использования продукта<br>5. Контактная информация |
| Система автоматического регулирования оборотов ДВС ARES | 1. Включение и отключение системы авторегулирования оборотов двигателя<br>2. Задание коэффициента потерь<br>3. Отображение текущих: нагрузка на двигатель, заданные регулятором обороты ДВС и заданные системой ARES обороты ДВС |
| Гидромоторы (уровень доступа мастер) | 1. Выбор режима настройки подаваемых на левый и правый гидромоторы токов<br>2. Настройка минимальных и максимальных подаваемых токов на гидромоторы в ручном межиме<br>3. Включение автокалибровки токов |
| Гидронасосы (уровень доступа мастер) | 1. Выбор режима настройки подаваемых на левый и правый гидронасосы токов<br>2. Настройка минимальных и максимальных подаваемых токов на гидронасосы в прямом и обратном направлении в ручном межиме;<br>3. Включение автокалибровки токов |
| Органы управления | 1. Отображение текущих отклонений и состояния плавающего режима стиков управления секциями гидрораспределителя<br>2. Отображение минимальных, максимальных и текущих напряжений педалей газа, тормоза и разворота, калибровка минимальных и максимальных напряжений |
| Информация о машине | 1. Отображение информации о наработке и пробеге машины, предстоящем техобслуживании, серийном номере и версии программного обеспечения контроллера<br>2. Сброс суточной наработки и пробега |
| Обновление ПО | 1. Информация о текущей версии приложения, ПО и драйвере контроллера<br>2. Список доступных для скачивания и установки обновлений с краткой сводкой нововведений и исправлений<br>3. Обновление дисплейного приложения, обновление и откат ПО и драйвера контроллера |
| Руководство по эксплуатации | 1. Просмотр и навигация в руководстве по эксплуатации приложения в формате PDF<br>2. Поиск по ключевым словам |

| Название окна / страницы | Функциональные требования |
|---|---|
| Камера | Вывод изображения с активной камеры и индикация отсутствия связи с камерой, возможность настроить изображение |
| Журнал событий | 1. Просмотр всех зафиксированных внутренних событий приложения (успешная/неуспешная выгрузка с контроллера, обрыв связи и т.д.), действий оператора в приложении (настройка, сохранение и сброс параметров, очистка данных и т.д.), ошибок и предупреждений<br>2. Возможность очистить отображаемый журнал (уровень доступа Оператор)<br>3. Сортировка событий по дате и типу, фильтр по типу; обновление содержимого журнала по мере появления новых событий |

Сформированные функциональные требования можно графически описать диаграммой вариантов использования, представленную на рисунке 4. Представление системы в виде диаграммы вариантов использования позволяет понять, как она будет функционировать [6], т.е. как будет происходить взаимодействие между системой и пользователем.

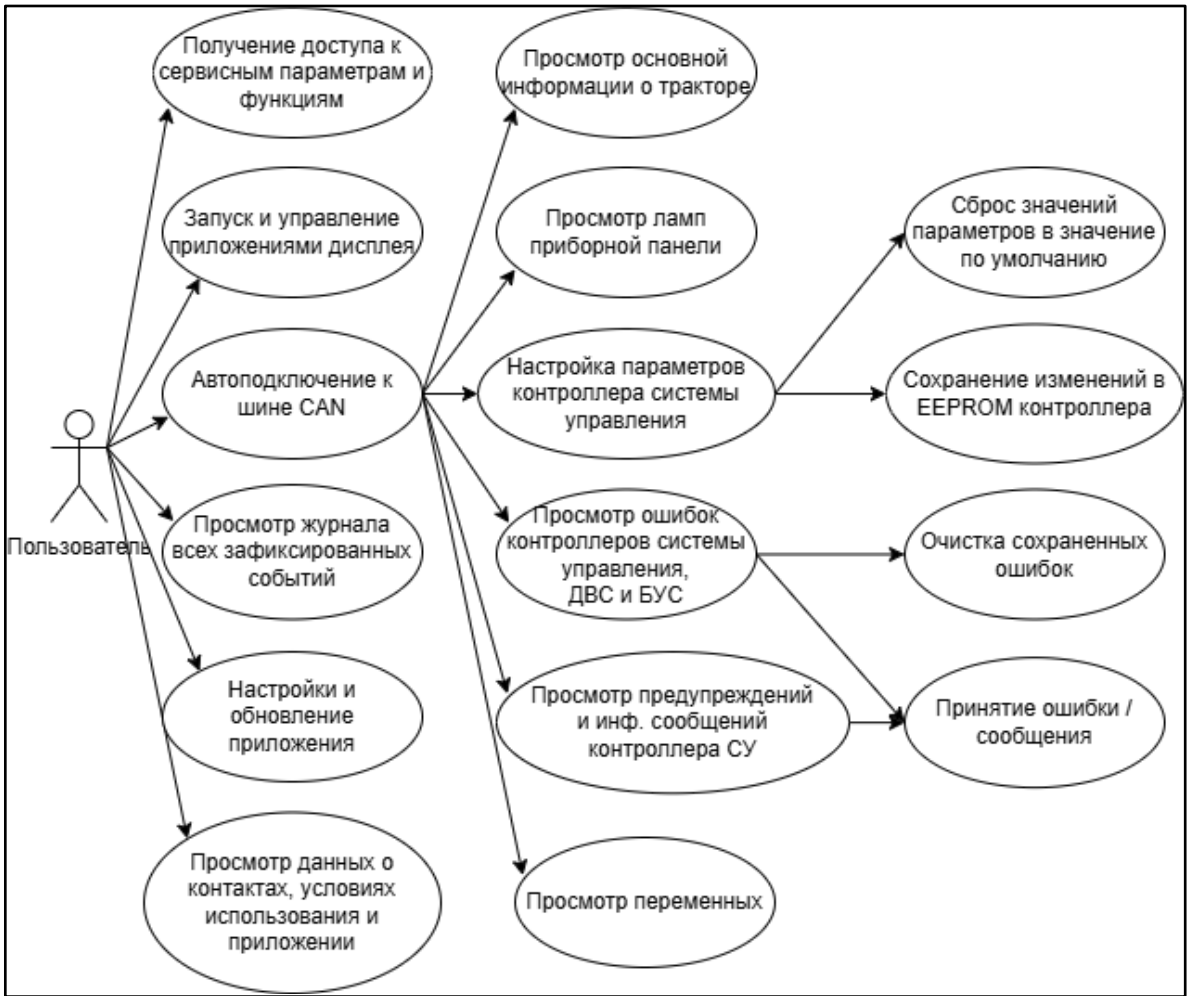


Рисунок 4 – Диаграмма вариантов использования

## 2.2. Нефункциональные требования

В результате аналитического обзора предметной области с учетом технического задания были выявлены следующие нефункциональные требования.

1. Совместимость с Android версии 8.1 и выше.
2. Схожесть с разработанным дизайнером макетом.
3. Адаптация интерфейса под различные разрешения и соотношения сторон дисплейного модуля.

**Вывод по второй главе**

Были сформулированы основные функциональные и нефункциональные требования к разрабатываемому приложению. Функциональные требования охватывают широкий спектр задач, включая связь с шиной CAN посредством интерфейсов USB или Bluetooth, реализацию множества страниц с настройками и мониторингом параметров машины, а также такие функции, как кэширование данных, уровни доступа и др. В результате анализа требований была построена диаграмма вариантов использования, которая наглядно демонстрирует взаимодействие пользователя с системой. Это позволило структурировать функционал приложения и выделить ключевые сценарии его использования. Все требования учтены с целью создания надежного, удобного и функционального решения, отвечающего потребностям операторов и потенциальных клиентов предприятия.

# 3. ПРОЕКТИРОВАНИЕ

## 3.1. Архитектура предлагаемого решения

Во Flutter виджеты служат основной единицей композиции пользовательского интерфейса, и одной из ключевых задач при разработке является эффективное управление их состояниями. В данном приложении большинство виджетов реализованы как независимые компоненты, не зависящие от состояния других элементов интерфейса (за исключением редких случаев, требующих синхронизации). Такой подход упрощает поддержку и модификацию кода, поскольку изменения в одном виджете не влекут за собой каскадных обновлений в других. Для управления состояниями был выбран наиболее простой и встроенный во фреймворк подход, основанный на использовании StatefulWidget (виджет, способный хранить изменяемое состояние), Listenable (объект, который может уведомлять слушателей о своих изменениях) и Stream (механизм асинхронной передачи событий с данными) и др.

Из основных шаблонов (Fundamental Patterns) [7] используются:

– неизменяемый интерфейс (Immutable Interface) – для предотвращения неконтролируемых изменений состояния и упрощения отладки;

– канал событий (Event Channel) – для организации коммуникации между различными частями приложения;

– строитель (Builder) – для динамического создания виджетов в зависимости от контекста;

– отложенная инициализация (Lazy Initialization) – для оптимизации производительности, особенно при работе с ресурсоемкими объектами;

– синглтон (Singleton) – для случаев, когда требуется глобальный доступ к определенному сервису (например, обработчики и парсеры данных с шины CAN).

Для соблюдения принципов чистого кода и модульности [8] приложение структурировано в соответствии с концепцией многоуровневой архитектуры. Оно разделено на два основных уровня:

– уровень данных (backend) – отвечает за взаимодействие с внешними источниками (API, локальное хранилище, данные с шины CAN), обеспечивая получение, сохранение и преобразование данных;

– уровень пользовательского интерфейса (frontend), который включает в себя не только отображение данных, но и простую бизнес-логику, такую как валидация ввода, фильтрация контента и обработка пользовательских действий.

Для удобства проект структурно разделен на несколько частей:

1) assets – каталог, который содержит все использующиеся в приложении, картинки и внутренние документы;

2) fonts – каталог шрифтов, в т.ч. уникального шрифта, содержащего все необходимые иконки стандарта ISO 7000 [9];

3) lib – каталог основного кода приложения, включающий в себя следующие каталоги:

3.1) api – каталог классов, с помощью которых осуществляется взаимодействие с сервером предприятия;

3.2) can – каталог обработчиков данных с шины CAN;

3.3) theme – каталог тем приложения и стилей элементов;

3.4) pages – каталог интерфейсов страниц приложения;

3.5) helpers – каталог всех переменных приложения, вспомогательных функций, классов, интерфейсов диалоговых окон, поведений элементов и генератора сервисного пароля;

3.6) widgets – каталог используемых по всему приложению универсальных виджетов, макетов элементов и виджетов «плавающего режима»;

4) pubspec.yaml – файл, предоставляющий возможность указать внешние ресурсы для проекта.

### 3.1.1. Протокол UDS

В программное обеспечение контроллера системы управления внедрен протокол UDS (Unified Diagnostic Services), обмен данными по которому в данный момент поддерживает только диагностическое приложение предприятия. UDS – это современный протокол диагностики электронных блоков управления в автомобилях и самоходной технике, который стандартизирован по ISO 14229 [10] и функционирует на прикладном уровне модели OSI, используя в качестве транспортного протокола CAN. Протокол подразумевает предоставление сервисов, которые выполняются через запрос-ответ между контролером и диагностическим приложением, где каждый запрос содержит идентификатор сервиса [11].

При работе приложений приборных панелей других выпускаемых предприятияем типов техники, нагрузка на шину CAN может доходить до 65-70% на некоторых машинах, что обусловлено большим количеством передаваемых параметров, высокой частотой обновления данных и ограниченной пропускной способностью шины CAN (256 Кб/с). Такая высокая нагрузка отрицательно влияет на качество и постоянство связи приложения и контроллера и может стать причиной потери данных в процессе коммуникации. В связи с этим, было принято решение использовать диагностический протокол UDS для осуществления коммуникации между разрабатываемым приложением и контроллером, частично сохраняя одностороннюю коммуникацию «контроллер-приложение» для обратной совместимости.

### 3.1.2. Многоуровневый обработчик данных шины CAN

Обработка входящих и отправление исходящих данных будет осуществляться в несколько этапов. Это позволит сделать обработку асинхронной и избежать потери данных, которая может возникнуть при высокой нагрузке шины CAN, когда полный цикл обработки данных может занимать слишком много времени. Сырые входные данные будут поступать

в обработчик потока данных USB в класс UsbLayer или обработчик потока данных Bluetooth в класс BluetoothLayer, в котором из всего потока будет выделяться только полезная информация. Далее данные поступают в обработчик класса CanLayer, который выделит тип сообщения, адрес и данные и сформирует кадр данных стандарта CAN. Далее, в зависимости от адреса и типа сообщения, оно поступает в обработчик параметров, передаваемых кадром данных, класса CanWrapper, в обработчик ошибок класса ErrorWrapper или обработчик сообщений диагностического протокола UDS класса UdsLayer. Отправление исходящих данных (протокол UDS) будет осуществляться таким же образом в обратном порядке. Диаграмма последовательностей взаимодействия приложения и контроллера на языке графического описания UML представлена на рисунке 5.

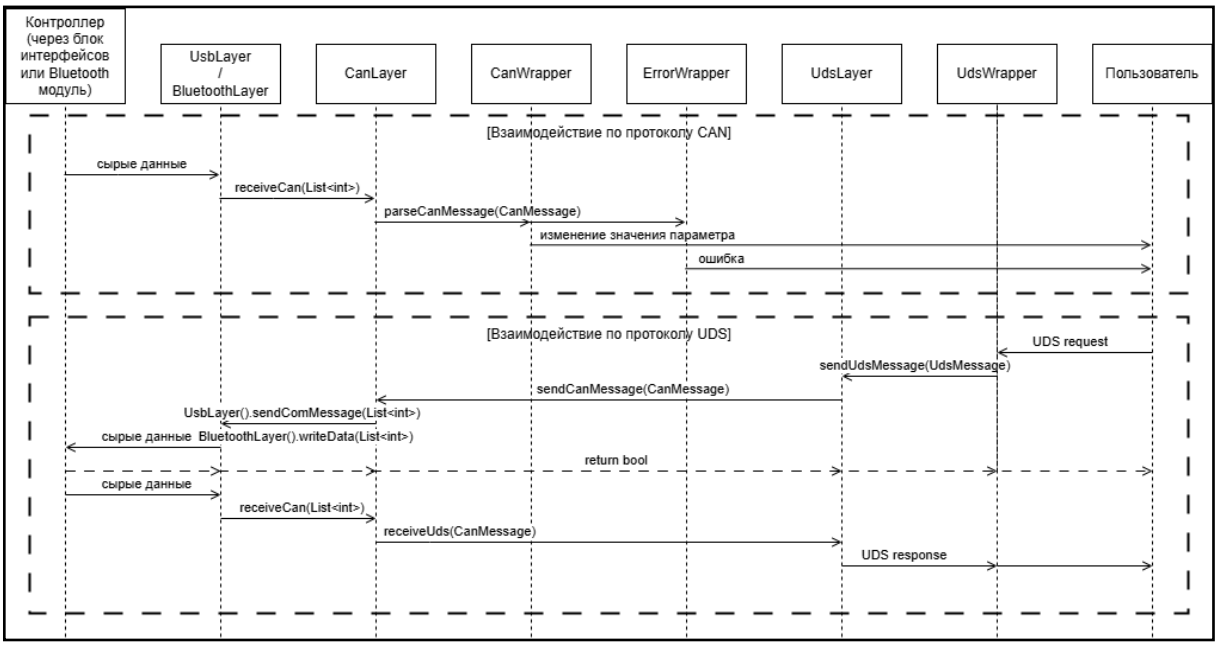


Рисунок 5 – Диаграмма последовательностей взаимодействия приложения и контроллера

Такой подход позволит избежать создания массивного обработчика, делегировав его обязанности другим классам, ускорить процессы разработки, тестирования и отладки приложения.

### 3.1.3. Универсальные виджеты

Поскольку ключевая функциональность приложения приборной панели сосредоточена вокруг отображения и настройки разнообразных параметров, было принято решение использовать универсальные виджеты вместо разработки индивидуальных интерфейсных элементов для каждого случая. Такой подход не только делает код более структурированным и поддерживаемым, но и существенно ускоряет процесс разработки, поскольку позволяет повторно использовать готовые компоненты в разных частях приложения без дублирования логики.

Были реализованы виджеты: ползунок, переключатель, кнопка включения/выключения, цифровой показометр, круговой показометр, четвертькруглый показометр, лампа приоритета, линейный показометр, настраиваемый параметр с полем ввода. Для универсальности все виджеты поддерживают любой реализованный в приложении тип параметра и переменной и предоставляют возможность настройки множества свойств для их различного поведения и внешнего вида на разных страницах приложения: отображение картинки и ее положение относительно виджета, отображение приоритетной лампы вместо картинки, режим «только для чтения», ориентация (для ползунка, линейного и четвертькруглого показометра), стиль каретки (для ползунка), стиль указателя (круговой и четвертькруглый показометр) а также все цвета, размеры и другие визуальные атрибуты виджета.

Такой подход не только упрощает разработку, но и обеспечивает единообразие интерфейса, снижает вероятность ошибок и облегчает дальнейшее масштабирование функционала.

**Вывод по третьей главе**

Были определены и обусловлены протоколы взаимодействия приложения и контроллера, алгоритм обработки данных шины CAN с приведением необходимых классов и их задач, и подход к реализации интерфейса, основанный на создании универсальных переиспользуемых виджетов.

# 4. РЕАЛИЗАЦИЯ

## 4.1. Приборная панель, «плавающий» режим и камера

Главной страницей приложения является «Приборная панель», отображающая основную необходимую для работы и контроля состояния машины информацию. Для быстрого доступа к странице из любого другого места в приложении, кнопка страницы имеет фиксированное место в меню навигации. Интерфейс страницы Приборная панель представлен на рисунке 6.

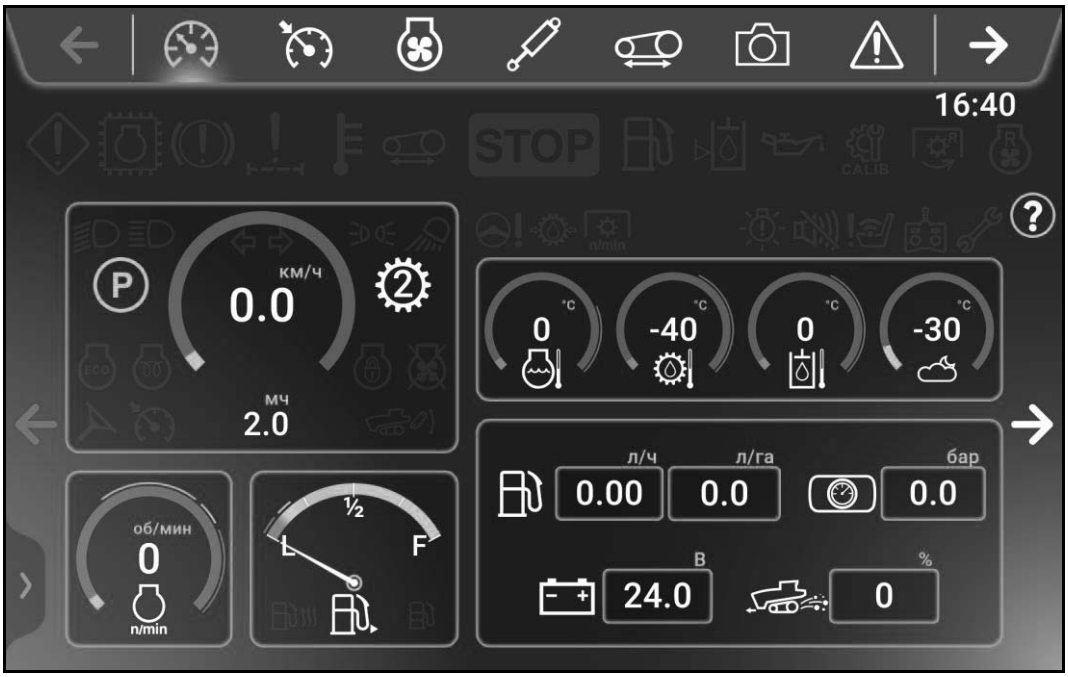


Рисунок 6 – Интерфейс страницы «Приборная панель»

Как было сказано выше, компактный «плавающий» режим приложения необходим для отображения основной информации приборной панели, т.е. показометров основных параметров работы, ошибок и ламп поверх открытых сторонних приложений дисплейного модуля для непрерывного контроля состояния машины, даже когда оператор закрывает основное приложение. Интерфейс «плавающей» приборной панели на фоне стороннего приложения представлен на рисунке 7.

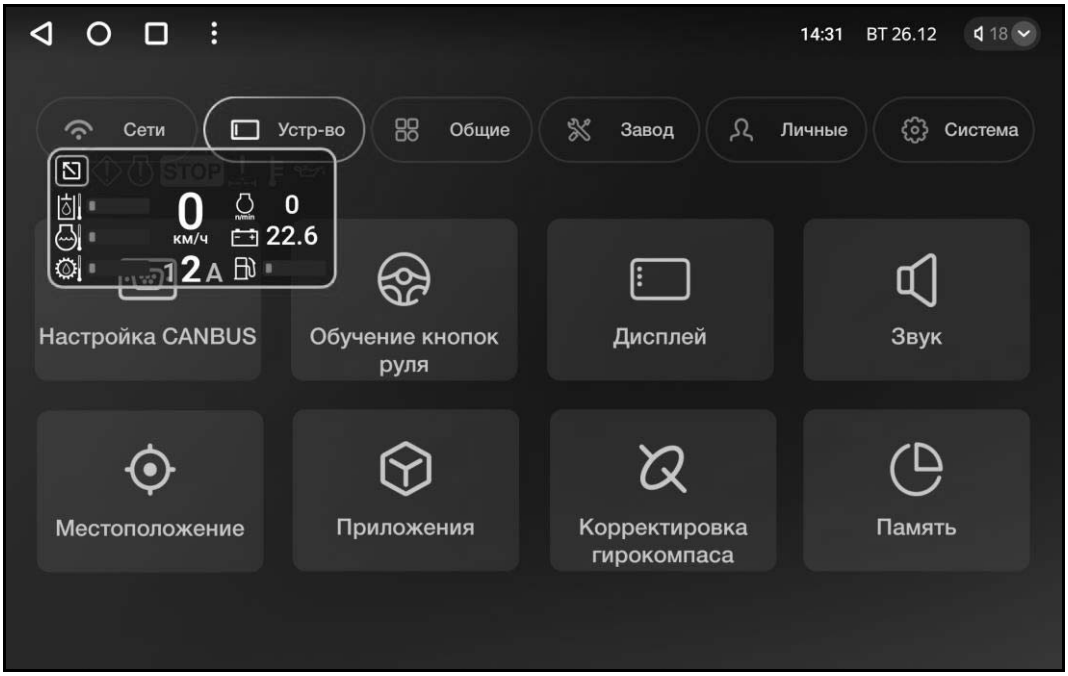


Рисунок 7 – Интерфейс «плавающей» приборной панели

«Плавающий» режим был реализован с помощью пакета flutter_floatwing и компонента Service. Пакет запускает фоновый сервис, предоставляет возможность объявить вторую точку входа в приложение и создает для «плавающего» окна новый, независимый от главного, экземпляр движка Flutter. По сути, это новое приложение с жизненным циклом, зависимым от жизненного цикла главного приложения. Создаваемый пакетом для каждого окна новый экземпляр движка Flutter позволяет избежать возможных конфликтов при использовании единственного экземпляра движка разными компонентами, когда одна из компонент может вызвать для него метод освобождения памяти dispose и перестать существовать.

Создание вторых экземпляров обработчиков CAN шины может привести к переполнению ОЗУ и экстренному завершению работы приложения менеджером ресурсов системы Android. «Плавающий» режим приборной панели не предоставляет каких-либо настроек или различных страниц, поэтому во избежание создания новых обработчиков был разработан небольшой протокол взаимодействия обработчиков главного приложения и сервиса «плавающего» режима с помощью поддерживаемого пакетом об-

щения между изолятами (классы SendPort и ReceivePort) [12]. Был объявлен достаточный для функционирования «плавающего» режима перечень событий, при возникновении которых в ReceivePort «плавающей» приборной панели будут поступать сообщения с их описанием. В этот перечень входят: изменение статуса подключения к шине CAN, изменение значения одного из представленных в «плавающем» режиме параметра, изменение статуса одной из представленных в «плавающем» режиме лампы, получение новой ошибки и принятие ошибки, которое работает и в обратную сторону. Для быстрого доступа к главному приложению в «плавающем» режиме предусмотрена кнопка запуска MainActivity главного приложения.

Комплектация трактора включает в себя переднюю и заднюю камеры, переключающиеся с блока кнопок управления, в связи с чем было необходимо реализовать автоматическую реинициализацию камеры. Также, согласно техническому заданию на разработку, был реализован экран с уменьшенной приборной панелью и окном изображения с камеры с функцией автовключения при начале движения задним ходом (рисунок 8).

Рисунок 8 – Интерфейс страницы «Приборная панель» с изображением с камеры

Для доступа к полноэкранному изображению с камеры была реализована отдельная страница, представленная на рисунке 9.

Рисунок 9 – Интерфейс страницы «Камера»

Правильное взаимодействие с камерой предполагает освобождение ее ресурсов в случае отсутствия необходимости ее отображения, поэтому при выходе из приложения или выключении зажигания, ресурсы используемой камеры будут освобождены для возможности ее использования другими приложениями.

Также, был реализован функционал настройки изображения с камеры, а именно: яркости, контраста, насыщенности и цветовой гаммы, так как некоторые камеры могут давать неестественное выходное изображение.

### 4.2. Руководство

Для обеспечения возможности ознакомления с функционалом приложения и конкретными настройками без необходимости использования сторонних средств, было реализовано окно просмотра электронной версии

руководства пользователя приложения (оператора), доступное из любой страницы приложения, представленное на рисунке 10.

Рисунок 10 – Интерфейс окна просмотр руководства

Исходный PDF-файл руководства содержит заголовки, автоматически интерпретирующиеся форматом как «Закладки», что позволяет программно обращаться к конкретным пунктам руководства, а значит отображать оператору информацию сразу о том функционале, с которым требуется помощь. Окно просмотра руководства поддерживает свободную навигацию и навигацию по заголовкам, поиск в тексте по ключевым словам, а также внешние и внутренние гиперссылки исходного документа.

### 4.3. Обновление приложения и ПО контроллера

Так как приложение не будет распространяться через какие-либо магазины приложений, а обновление через приложение предприятия «Центр загрузок» неинтуитивно и может быть недоступно ввиду отсутствия «Центра загрузок» на устройстве, была реализована возможность обновления приложения из самого приложения. Приложение уведомит оператора при наличии доступных обновлений.

Окно обновления включает в себя 2 вкладки: обновление приложения и обновление ПО контроллера. При наличии подключения к интернету, приложение выгружает с сервера предприятия все доступные версии приложения и ПО контроллера и сопровождающую информацию о них, исходя из типа приложения, типа машины, минимальной требуемой версии ПО контроллера (при обновлении приложения) и минимальной требуемой версии приложения (при обновлении ПО контроллера). При обнаружении несовместимой версии приложения или версии ПО контроллера, приложение уведомит оператора о необходимости обновления или перепрошивки контроллера, или отката версии при невозможности обновления. Интерфейс окна обновления представлен на рисунке 11.

Рисунок 11 – Интерфейс окна обновления

В окне отображаются текущие версии, информация о доступных обновлениях (версия, название, дата выхода и ключевые изменения), индикация процесса скачивания и обновления. Алгоритм прошивки контроллера и его реализацию предоставило предприятие.

### 4.4. Все приложения

Страница «Все приложения» аналогично любым другим лаунчер-приложениям предоставляет интерфейс для управления сторонними приложениями устройства: запуск, удаление, переход в Android-настройки приложения. Это позволяет оператору использовать все возможности дисплейного модуля, в то же время находясь под контролем разрабатываемого приложения. Интерфейс страницы «Все приложения» представлен на рисунке 12.

Рисунок 12 – Интерфейс страницы «Все приложения»

### 4.5. Кэширование данных

Заранее приложение ничего не знает о ПО контроллера, и, взаимодействуя с ним по протоколу UDS, должно получить от него всю необходимую информацию, такую как: все параметры и переменные и их свойства, список описаний ошибок, список списков значений для параметров-списков, список используемых единиц измерения и т.д. Эта процедура занимает некоторое время, а именно от 8 до 15 секунд. Во избежание такой задержки на каждый запуск приложения (включения зажигания, а, следовательно, и дисплейного модуля) было реализовано кэширование всей

единожды полученной от контроллера информации в папку кэша приложения и проверка на действительность закэшированных данных текущему ПО контроллера. При каждом запуске приложение запрашивает от контроллера лишь информацию, касающуюся его идентификации, по которой оно определяет необходимость в повторном запросе и кэшировании всех данных. Таким образом, со второго запуска приложения, все параметры становятся доступными к изменению без задержки при условии, что версия ПО контроллера осталось неизменной с предыдущего запуска.

Также, вместе с данными ПО контроллера кэшируются: настройки изображения с камеры, серийный номер машины, порядок страниц в меню навигации (изменяемый оператором), доступные к скачиванию и скачанные файлы обновления приложения и ПО контроллера, настройки приложения и пр.

Самым удобным форматом для работы с кэшем является Json [13]. Так как кэшируются в основном объекты, не относящиеся к простым типам данных, для каждого класса таких объектов были написаны соответствующие методы toJson и fromJson.

### 4.6. Уровни доступа

Некоторый функционал приложения не предполагает к себе прямого доступа оператором, поэтому были реализованы 3 уровня доступа (от низшего к высшему соответственно): Оператор, Мастер, Сервис. Уровень Оператор дает доступ к базовому функционалу, уровень Мастер открывает доступ к некоторым настройкам движения (гидростатической трансмиссии) и некоторым параметрам работы машины, уровень Сервис включает в себя функционал уровня Мастер и дополнительно открывает доступ к сервисным параметрам работы машины и полному журналу событий, включая все действия оператора в приложении.

Пароль для получения уровня Мастер совпадает с серийным номером машины, а пароль для получения уровня Сервис доступен только через сервисную службу по сгенерированному временному 4-значному коду.

Соответственно, был реализован интерфейс управления уровнем доступа, представленный на рисунке 13.

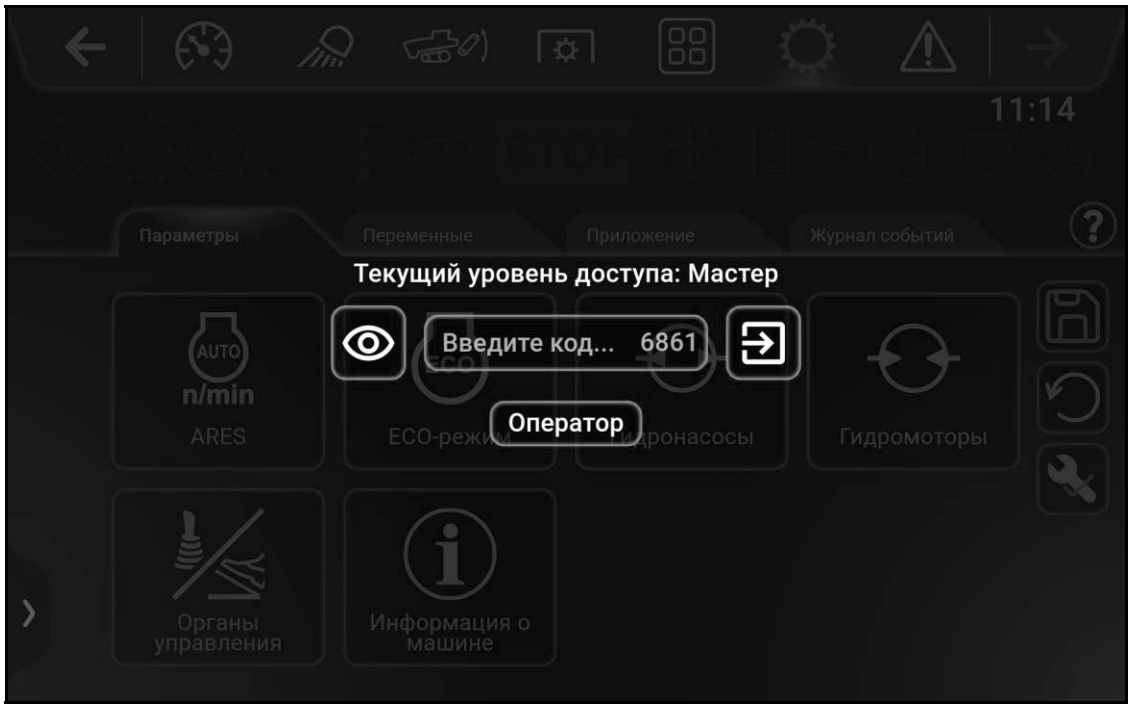


Рисунок 13 – Интерфейс управления уровнем доступа

Согласно техническому заданию был реализован автоматический сброс уровня доступа на Оператор по истечению 30 минут с момента получения более высокого уровня или по выключению зажигания, с уведомлением пользователя о произошедшем сбросе.

### 4.7. Настройки

Настройки приложения, параметры работы машины, журнал событий и настройки машины не первой важности для оператора были вынесены на одну страницу и разбиты на подменю, чтобы избежать переполнения меню навигации и сделать интерфейс приложения более удобным и интуитивным.

### 4.7.1. Настройки машины

Согласно функциональным требованиям, страница настроек машины («Параметры») включает в себя страницы: «ARES (Система автоматического регулирования оборотов ДВС)», «ECO-режим», «Гидромоторы», «Гидронасосы», «Органы управления», «Информация о машине» и «Обновление ПО». Интерфейс страницы настроек машины представлен на рисунке 14.

Рисунок 14 – Интерфейс страницы настроек машины («Параметры»)

Предполагается, что оператор будет редко менять настройки машины в этом разделе меню, это будет чаще делать квалифицированный специалист в процессе диагностики, настройки и калибровки, поэтому представленные в этом меню параметры, в отличии от других страниц, не сохраняются в EEPROM контроллера при изменении, а записываются в RAM контроллера и ожидают ручного сохранения мастером, подсвечиваясь цветом, отличным от неизмененных параметров. Также, для упрощения диагностики, параметры на этих страницах можно сбросить в значение при последнем включении зажигания.

### 4.7.2. Переменные

Страница «Переменные» отображает все неизменяемые параметры работы машины для текущего уровня доступа в виде древовидной структуры. Алгоритм обновления реализован так, что обновляются только значения параметров открытых меню или параметров, выбранных для мониторинга. Интерфейс страницы «Переменные» представлен на рисунке 15.

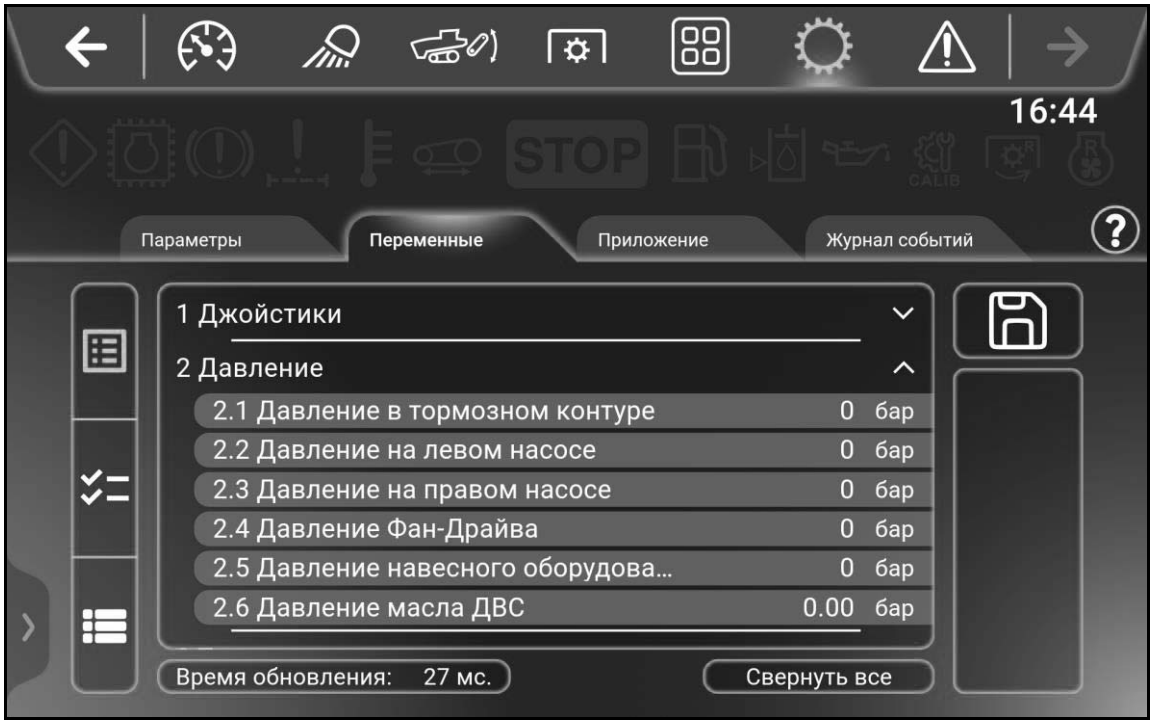


Рисунок 15 – Интерфейс страницы «Переменные»

Страница включает в себя 3 вкладки: «Список», «Выборка» и «Отображение». На вкладке «Список» отображаются все доступные переменные с обновляемыми в реальном времени значениями. Вкладка «Выборка» позволяет выбрать конкретные переменные для отображения на вкладке «Отображение». В правой части страницы есть интерфейс управления выборками: оператор может сохранить текущую выборку, удалить или выбрать любую другую. Выборки кэшируются при их создании и удалении.

### 4.7.3. Приложение

Страница «Приложение» включает в себя 3 вкладки: «Настройки приложения», «Информация о приложении» и «Контакты», и позволяет

изменить некоторые настройки и обновить приложение, ознакомиться с условиями использования и контактами предприятия. Интерфейс страницы «Приложение» представлен на рисунке 16.

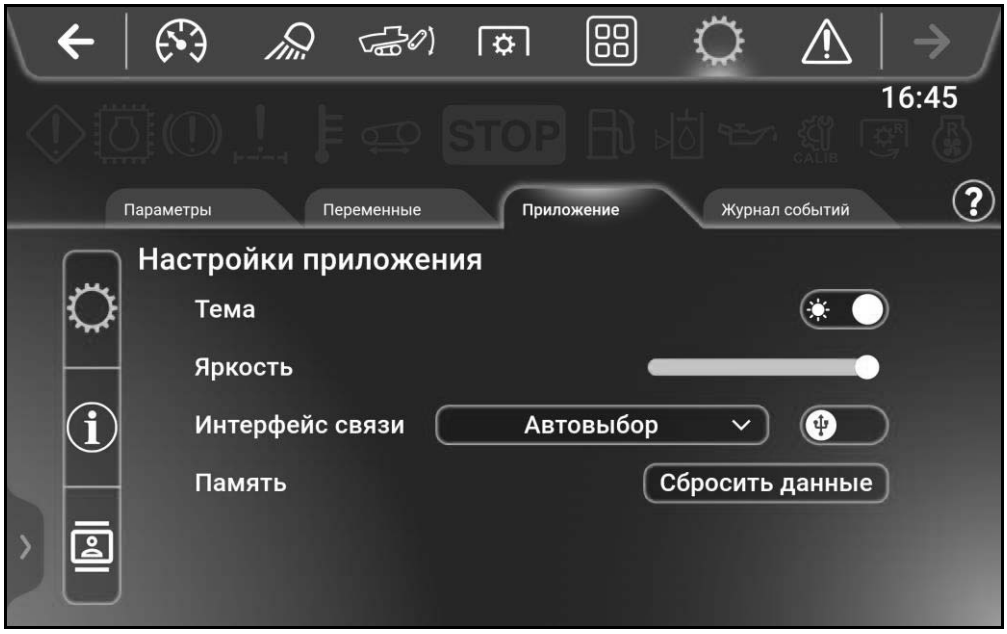


Рисунок 16 – Интерфейс страницы «Приложение»

Была добавлена возможность подключения и работы по BLE через Bluetooth-модуль, установленный в машине, нужная в случае неисправности дисплейного модуля и необходимости использования стороннего Android-устройства в процессе работы.

Для случая непредвиденных ошибок со стороны приложения при коммуникации с контроллером или желания оператора сбросить приложение к «заводским» настройкам была предусмотрена возможность очистки кэша приложения.

Если на устройстве установлены необходимые приложения с разрешением получать запросы на открытие определенных ссылок, на вкладке «Контакты» появятся кнопки, при нажатии на которые пользователь сможет, например, открыть сайт или позвонить в сервисную службу.

### 4.7.4. Журнал событий

Журнал событий – это критический компонент системы диагностики, предназначенный для регистрации отказов и аномалий в работе электронных систем управления и анализа причин инцидентов (например, обрыв связи с датчиками) [14].

Страница «Журнал событий» предоставляет интерфейс для ознакомления с зафиксированными в приложении событиями, такими как: полученная ошибка или предупреждение от контроллера, успешная или неуспешная загрузка данных из контроллера, обрыв связи, изменение или сброс параметров и др. Записи о событиях отображаются в виде бесконечного списка с указанием текущего просматриваемого временного интервала. Интерфейс страницы «Журнал событий» представлен на рисунке 17.

Рисунок 17 – Интерфейс страницы «Журнал событий»

События не хранятся в оперативной памяти и запись их в файл происходит в момент их возникновения, поэтому экстренное закрытие приложения по каким-либо причинам не приведет к потере части журнала.

Пользователь имеет возможность отсортировать записи по дате, времени и типу, отфильтровать по необходимым типам и обновить журнал. Пользователь с уровнем доступа Оператор может также очистить журнал

событий, который по-прежнему останется доступен более высоким уровням доступа.

### 4.8. Ошибки

Страница «Ошибки» включает в себя 4 вкладки: «Ошибки ЭБУ», «Ошибки ДВС», «Ошибки БУС» и «Состояние». Для быстрого доступа к странице из любого другого места в приложении, кнопка страницы имеет фиксированное место в меню навигации.

Интерфейс уведомления об активной ошибке и страницы «Ошибки» представлен на рисунке 18.

Рисунок 18 – Интерфейс страницы «Ошибки»

Приложение может получить ошибки от трех источников: контроллер системы управления, ЭБУ ДВС и БУС. Некоторые ошибки от контроллера системы управления с определенным кодом и статусом являются предупреждениями и попадают на вкладку «Состояние», например, уведомление о необходимости калибровки или уведомление о низком уровне топлива. В приложении предусмотрено всплывающее окно, поочередно с некоторым временным интервалом сообщающее о всех новых критических зарегистрированных ошибках, для привлечения к ним внимания оператора.

Так как уведомление частично перекрывает лампы и меню навигации, оператор сможет убрать его с экрана, но только после принятия всех ошибок, что необходимо для минимального подтверждения ознакомления оператором с каждой критической неисправностью.

**Вывод по четвертой главе**

В соответствии с функциональными требованиями, приведенными в работе и изложенными в техническом задании, было разработано Android-приложение приборной панели для дисплейного модуля трактора FT-7. Приложение создано на основе фреймворка Flutter с использованием языка Dart, что обеспечило высокую производительность и скорость разработки.

Были успешно реализованы все ключевые модули системы:

– приборная панель с отображением критических параметров работы машины и «плавающим» режимом;

– управление настройками машины (EHR, FanDrive, круиз-контроль, гидравлика и т.д.);

– система кэширования данных для ускорения получения доступа к функционалу приложения;

– модуль уровней доступа;

– модуль журналирования ошибок и внутренних событий;

– модуль общения с контроллером через интерфейс USB или Bluetooth;

– функционал Android-лаунчера.

В ходе реализации был применен многоуровневый подход к обработке CAN-сообщений, разработаны универсальные переиспользуемые виджеты и реализована система событийного взаимодействия между компонентами приложения.

Скриншоты остальных типовых страниц приложения представлены в приложении.

## 5. ТЕСТИРОВАНИЕ

Для проверки корректности работы приложения было проведено функциональное тестирование. Это вид тестирования, рассматривающий заранее указанное поведение и основывающийся на анализе спецификации системы [15], т.е. проверяется соответствие приложения заявленным функциональным требованиям. Набор тестов функционального тестирования представлен в таблице 2. Все тесты пройдены успешно.

Таблица 2 – Функциональные тесты

| № | Описание теста | Шаги | Ожидаемый результат |
|---|---|---|---|
| 1 | Автоматический запуск приложения при включении дисплейного модуля. | Включить зажигание на машине. | После загрузки системы дисплейного модуля приложение запустится автоматически. |
| 2 | Автоматический запуск диагностической сессии по протоколу UDS. | Включить зажигание на машине. | После запуска приложения, при наличии подключенного к USB порту дисплейного модуля блока интерфейсов, приложение отправит запрос контроллеру на открытие диагностической сессии UDS и, в случае успеха, получит от контроллера его основную идентификационную информацию. |
| 3 | Загрузка кэша при его наличии и совпадении реальной версии ПО контроллера с кэшированной. | Включить зажигание на машине. | После отрытия сессии UDS и идентификации ПО контроллера, приложение загрузит все данные из кэша и не начнет этап загрузки всех данных из контроллера. |
| 4 | Разрыв связи с контроллером. | 1. Включить зажигание на машине.<br>2. Физически разорвать связь дисплейного модуля с контроллером (отключить блок интерфейсов или контроллер). | Все параметры, получаемые по CAN протоколу, выставятся в значение по умолчанию, соответствующие виджеты приобретут тусклый оттенок и покажут подсказку об отсутствии связи по нажатию, появится всплывающее сообщение об отсутствии связи с контроллером, обновление ПО контроллера становится недоступным. |

| № | Описание теста | Шаги | Ожидаемый результат |
|---|---|---|---|
| 5 | Разрыв диагностической сессии UDS. | 1. Включить зажигание на машине.<br>2. Физически разорвать связь дисплейного модуля с контроллером (отключить блок интерфейсов или контроллер). | Все параметры, получаемые и настраиваемые по протоколу UDS, сохранят последнее значение, соответствующие виджеты приобретут тусклый оттенок и покажут подсказку об отсутствии данных по нажатию, обновление ПО контроллера становится недоступным. |
| 6 | Загрузка всех данных с контроллера по протоколу UDS при отсутствии кэша или несовпадении реальной версии ПО контроллера с кэшированной. | 1. Включить зажигание на машине. | После отрытия сессии UDS и идентификации ПО контроллера, приложение загрузит все данные из контроллера, соответствующие виджеты приобретут яркий оттенок и станут доступны для изменения, обновление ПО контроллера становится доступным. |
| 7 | Очистка данных при наличии кэша. | 1. Включить зажигание на машине.<br>2. Очистить данные в настройках приложения. | При следующем запуске приложения (включении зажигания), приложение вернется к исходному состоянию и потребуется загрузка всех данных из контроллера, даже если она уже была ранее выполнена. |
| 8 | Запуск, остановка и бесперебойная работа окна «плавающего» режима приложения. | 1. Включить зажигание на машине.<br>2. Открыть стороннее приложение на странице «Все приложения».<br>3. Вернуться в главное приложение приборной панели (на соответствующую кнопку в «плавающем» режиме или системную кнопку «Домой»). | При открытии стороннего приложения, запустится окно «плавающего» режима приложения, индикации будут соответствовать ожидаемым в течение неопределенного периода времени.<br>При возврате в главное приложение приборной панели, окно «плавающего» режима закроется. |
| 9 | Изменение параметров. | 1. Включить зажигание на машине.<br>2. Перейти на любую страницу с настраиваемыми параметрами.<br>3. Изменить значение параметра с помощью соответствующих кнопок или поля для ввода. | Значения параметров будут меняться со строгим соблюдением их диапазонов значений, шага изменения, множителя и делителя. Задание недопустимого значения параметра исключено. |

| № | Описание теста | Шаги | Ожидаемый результат |
|---|---|---|---|
| 10 | Просмотр руководства. | 1. Включить зажигание на машине.<br>2. Перейти на любую страницу, по которой необходима помощь руководства.<br>3. Нажать кнопку открытия руководства.<br>4. Воспользоваться поиском в содержимом руководства по словам, осуществить переходы по внутренним и внешним гиперссылкам. | Откроется окно просмотра руководства и руководство автоматически пролистается до соответствующего открытой странице раздела.<br>Поиск во время ввода находит и кратко отображает все вхождения введенной строки без учета регистра, навигация по результатам поиска работает корректно.<br>На переход по внутренней ссылке руководство пролистывается до соответствующего раздела, по внешней – открывается веб-браузер соответствующей веб-страницы. |
| 11 | Обновление приложения | 1. Включить зажигание на машине.<br>2. Открыть окно обновления (через «Настройки» – «Параметры» – «Информация о машине» или «Настройки» – «Приложение»).<br>3. Нажать кнопку загрузки обновления приложения, дождаться загрузки.<br>4. Нажать кнопку установки загруженного обновления. | Окно обновления отобразит список доступных к скачиванию обновлений приложения в соответствии с реальной или кэшированной (в случае отсутствия связи) версией ПО контроллера. По нажатию на загрузку обновления, приложение начнет загрузку соответствующего файла и сохранит его в директорию кэша по окончанию загрузки.<br>По нажатию на установку загруженного обновления, появится системное вспылывающее окно с предложением об установке обновления приложения, при принятии которого приложение будет обновлено и автоматически перезапущено с сохранением кэша. |
| 12 | Переключение камеры | 1. Включить зажигание на машине.<br>2. Открыть страницу «Камера» или второй экран страницы «Приборная панель».<br>5. Нажать кнопку переключения камеры на БКУ. | Ресурсы используемой камеры освободятся, в окне камеры сначала появится индикатор загрузки, потом, через небольшой временной интервал (около 5с), появится изображение со второй камеры (при ее наличии). |

| № | Описание теста | Шаги | Ожидаемый результат |
|---|---|---|---|
| 13 | Обновление ПО контроллера | 1. Включить зажигание на машине.<br>2. Открыть окно обновления (через «Настройки» – «Параметры» – «Информация о машине» или «Настройки» – «Приложение» – «Информация о приложении»).<br>3. Нажать кнопку загрузки обновления ПО контроллера, дождаться загрузки.<br>4. Нажать кнопку установки загруженного обновления.<br>5. Следовать инструкциям на дисплее. | Окно обновления отобразит список доступных к скачиванию обновлений ПО контроллера в соответствии с реальной или кэшированной (в случае отсутствия связи) версией ПО контроллера. По нажатию на загрузку обновления, приложение начнет загрузку соответствующего файла в оперативную память. По нажатию на установку загруженного обновления, появится вспылывающее окно с инструкциями по процессу обновления и индикация прогресса обновления. По успешному окончанию обновления, окно дождется подтверждения от оператора, и приложение автоматически переподключится к контроллеру и отправит запрос на открытие диагностической сессии UDS. При неуспешном обновлении, в окне отобразится причина неудачи с предложением повторить процедуру заного. |
| 14 | Получение уровней доступа Мастер и Сервис | 1. Включить зажигание на машине.<br>2. Перейти на страницу Настройки и открыть окно ввода пароля.<br>3. Ввести серийный номер машины или пароль по умолчанию для уровня Мастер или запросить пароль у сервисной службы, сообщив временный 4-значный код, для уровня Сервис (в тесте использовался генератор пароля, доступный только сотрудникам предприятия) и нажать «Войти». | Успех: отобразится индикация принятого пароля и нового уровня доступа, окно автоматически закроется. Страницы и настройки, требующие соответствующего полученному или более низкого уровня доступа, приобретут яркий оттенок и станут доступными для взаимодействия.<br>Неудача: отобразится причина неудачи (пустое поле, недопустимые символы или неправильный пароль) с предложением повторить попытку. |

| № | Описание теста | Шаги | Ожидаемый результат |
|---|---|---|---|
| 15 | Автоматический сброс уровня доступа до Оператор | 1. Включить зажигание на машине.<br>2. Перейти на страницу Настройки и открыть окно ввода пароля.<br>3. Ввести серийный номер машины или пароль по умолчанию для уровня Мастер или запросить пароль у сервисной службы, сообщив временный 4-значный код, для уровня Сервис (в тесте использовался генератор пароля, доступный только сотрудникам предприятия) и нажать «Войти»;<br>4. подождать 30 минут или выключить и повторно включить зажигание. | Если ранее был получен уровень доступа Мастер или Сервис, он автоматически будет сброшен до Оператор, и отобразится окно с причиной сброса (по таймеру или выключению зажигания). Страницы и настройки, требующие более высокого уровня доступа, приобретут тусклый оттенок и станут недоступными для взаимодействия. |
| 16 | Автосохранение измененного параметра в EEPROM контроллера по окончанию пользовательского ввода. | 1. Включить зажигание на машине.<br>2. Перейти на любую страницу с настраиваемыми параметрами, кроме «Настройки» – «Параметры».<br>3. Неоднократно изменить значение параметра за короткий промежуток времени. | Приложение отправит запрос на сохранение измененного значения параметра в EEPROM с задержкой в 1 секунду после последнего изменения. |

**Вывод по пятой главе**

В пятой главе было проведено функциональное тестирование приложения по сформулированному набору тестов. Все тесты из набора были выполнены успешно. Помимо приведенного набора, работниками различных отделов предприятия было проведено множество второстепенных тестов, также пройденных успешно. Можно сделать вывод, что разработанное приложение соответствует требованиям.

## ЗАКЛЮЧЕНИЕ

В рамках данного проекта в соответствии с техническим заданием было разработано Android-приложение приборной панели для дисплейного модуля сельскохозяйственного трактора модели FT-7, реализующее требуемый функционал и имеющее возможности для расширения. При этом были решены следующие задачи.

1. Произведен обзор существующих аналогичных приложений по предметной области.
2. Выбраны программные средства реализации и выполнено проектирование архитектуры приложения.
3. Разработано и протестировано приложение для дисплейного модуля трактора.

В настоящий момент приложение интегрировано в дисплейный модуль первого опытного образца трактора FT-7 и находится на стадии поддержки и расширения. В будущем планируется продолжать разработку приложения и его актуализацию вслед за нововведениями в программе контроллера системы управления. Так, например, со следующим обновлением планируется появление макросов – наборов действий для выполнения трактором, записанных оператором в процессе работы или созданных с дисплея, и профилей операторов, уже реализованных в другом, основанном на разработанном, дисплейном приложении, которые позволят перенастроить машину одним нажатием кнопки, что может быть необходимо при использовании трактора разными операторами или в разных условиях.

Грамотная архитектура приложения позволила без больших трудозатрат создать на своей основе приложения приборных панелей для дисплейных модулей двух других машин: автогрейдера от стороннего предприятия и разрабатываемого ратрака.

**ЛИТЕРАТУРА**

1. Ютт. В. Е., Рузавин Г. Е. Электронные системы управления ДВС и методы их диагностирования: // Учебное пособие для вузов. Изд-во Горячая линия-Телеком, 2007. 104 с.

2. John Deere 9 series tractors brochure | JohnDeere. [Электронный ресурс] URL: https://johndeere.widen.net/s/hmlx992bbl/202402_4867870-1en--9-series-tractor-brochure (дата обращения: 03.02.2025 г.).

3. React Native vs. Flutter vs. Xamarin | TheMobileReality. [Электронный ресурс] URL: https://themobilereality.com/blog/xamarin-vs-flutter-vs-react-native (дата обращения: 09.02.2025 г.).

4. Flutter под капотом | Habr. [Электронный ресурс] URL: https://habr.com/ru/articles/501862 (дата обращения: 09.02.2025 г.).

5. Подробно о Xamarin | Habr. [Электронный ресурс] URL: https://habr.com/ru/articles/188130 (дата обращения: 09.02.2025 г.).

6. IBM Documentation – Use-case diagrams. [Электронный ресурс] URL: https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case (дата обращения: 05.03.2025 г.).

7. Шпаргалка по шаблонам проектирования | Habr. [Электронный ресурс] URL: https://habr.com/ru/articles/210288 (дата обращения: 05.03.2025 г.).

8. Martin R. Clean Code: A Handbook of Agile Software Craftsmanship. / Robert C. Martin, 2008. – 464 с.

9. ISO 7000 Graphical symbols for use on equipment | ISO. [Электронный ресурс] URL: https://www.iso.org/obp/ui/#iso:pub:PUB400001:en (дата обращения: 06.03.2025 г.).

10. ISO 14229-1:2020 Road vehicles – Unified diagnostic services (UDS) | ISO. [Электронный ресурс] URL: https://www.iso.org/standard/72439.html (дата обращения: 15.03.2025 г.).

11. Automotive Diagnostic Protocols | University of Michigan, Transportation Research Institute. [Электронный ресурс] URL:

http://www.umtri.umich.edu/automotive-diagnostics (дата обращения: 15.03.2025 г.).

12. Isolates | Dart programming language. [Электронный ресурс] URL: https://dart.dev/language/isolates (дата обращения: 21.03.2025 г.).

13. JSON and serialization | Flutter Docs. [Электронный ресурс] URL: https://docs.flutter.dev/data-and-backend/serialization/json (дата обращения: 22.03.2025 г.).

14. ISO 25119-3:2018 Tractors and machinery for agriculture and forestry – Safety-related parts of control systems | ISO. [Электронный ресурс] URL: https://www.iso.org/standard/69027.html (дата обращения: 29.03.2025 г.).

15. Теория тестирования ПО | Habr. [Электронный ресурс] URL: https://habr.com/ru/articles/587620 (дата обращения: 11.04.2025 г.).

**ПРИЛОЖЕНИЕ. Типовые страницы приложения**

На рисунках 1–17 представлены скриншоты интерфейсов типовых страниц приложения.

Рисунок 1 – «Круиз-контроль»

Рисунок 2 – «FanDrive»

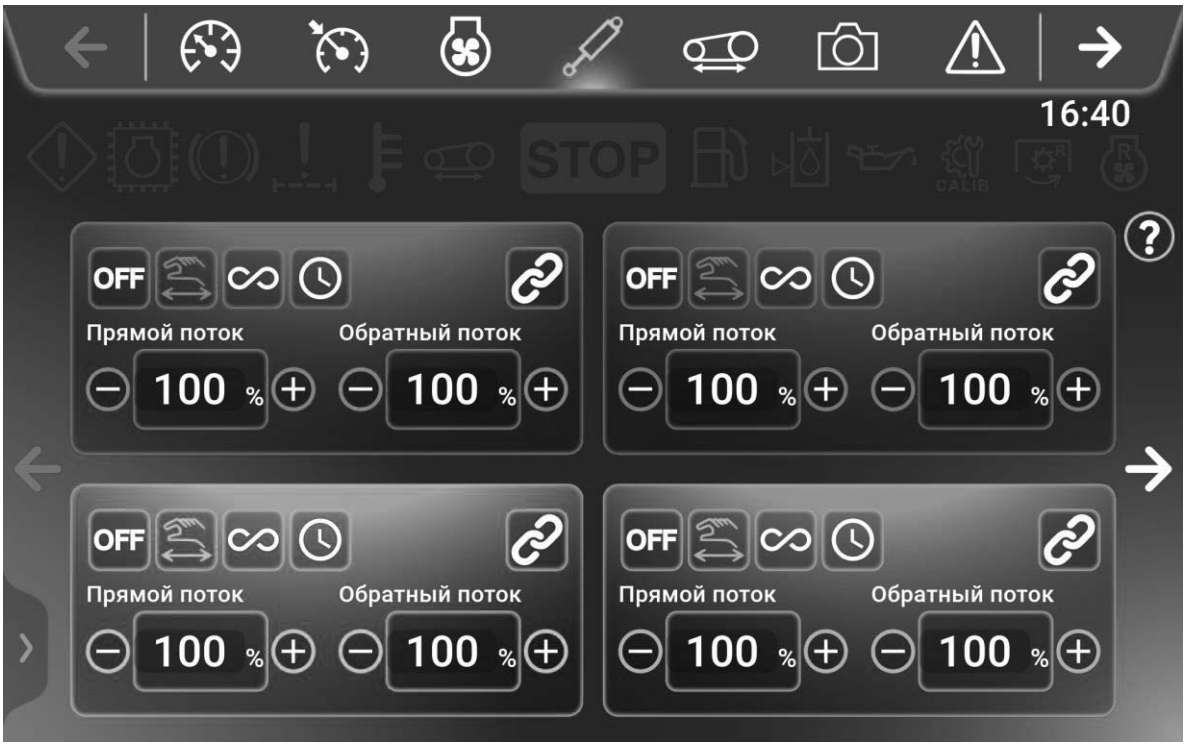


Рисунок 3 – «Регулировка секций гидрораспределителя рабочего оборудования»

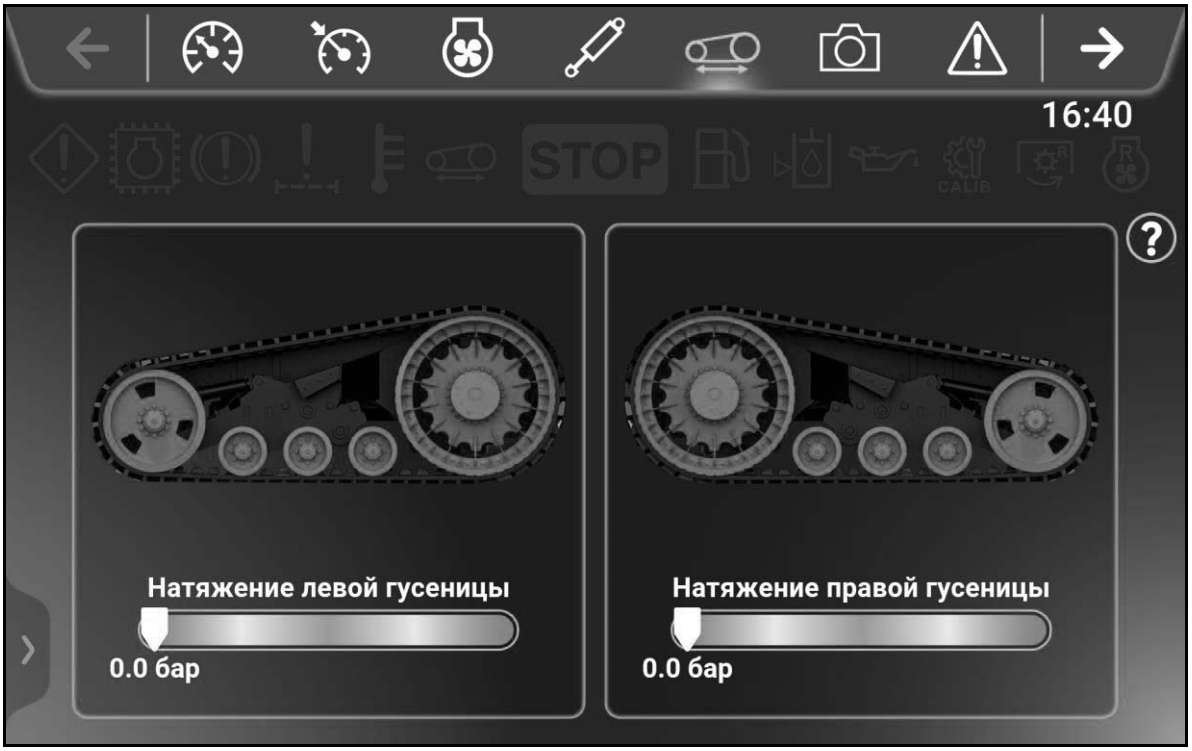


Рисунок 4 – «Натяжение гусеничных лент»

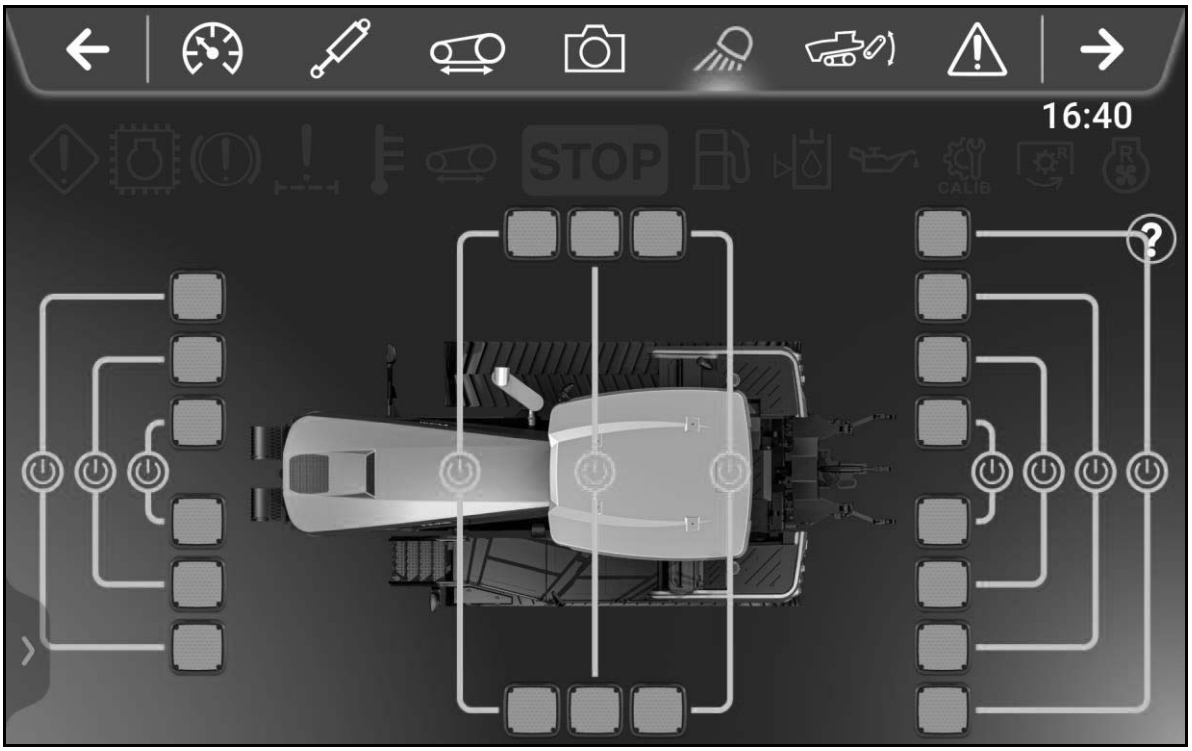


Рисунок 5 – «Конфигурация рабочего света»

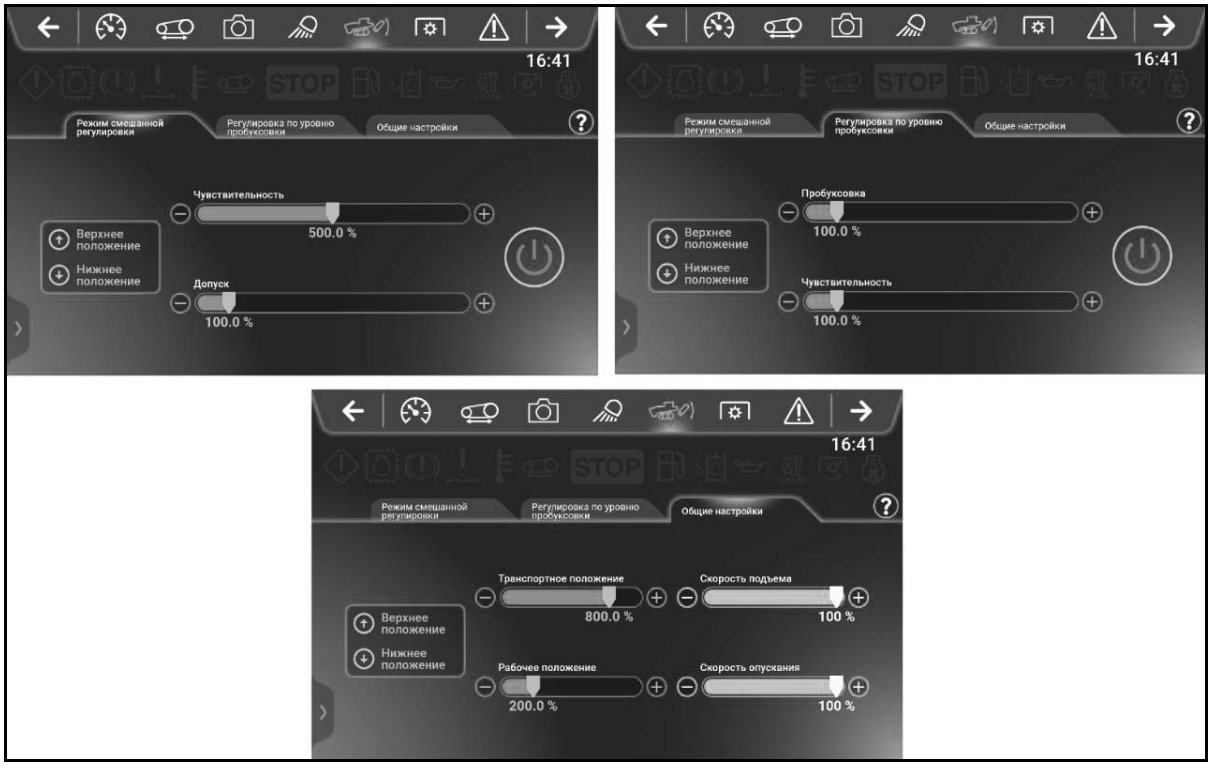


Рисунок 6 – «EHR» – «Режим смешанной регулировки», «Регулировка по уровню пробуксовки», «Общие настройки»

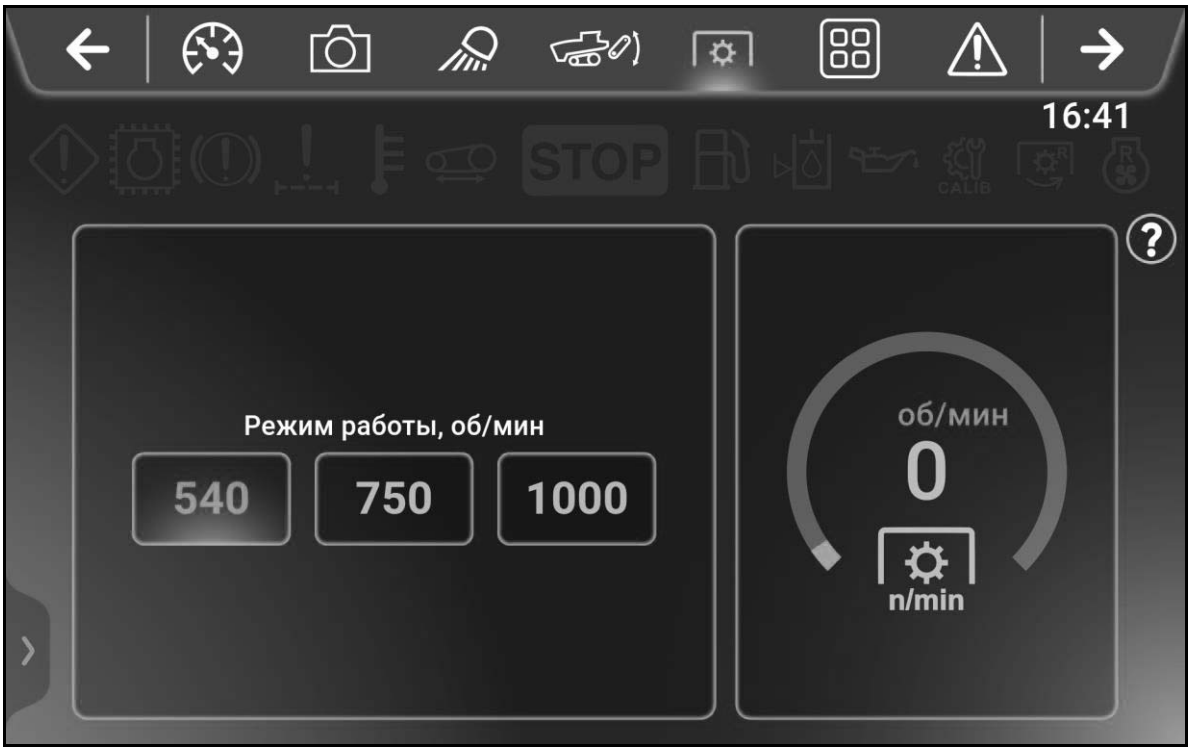


Рисунок 7 – «Вал отбора мощности»

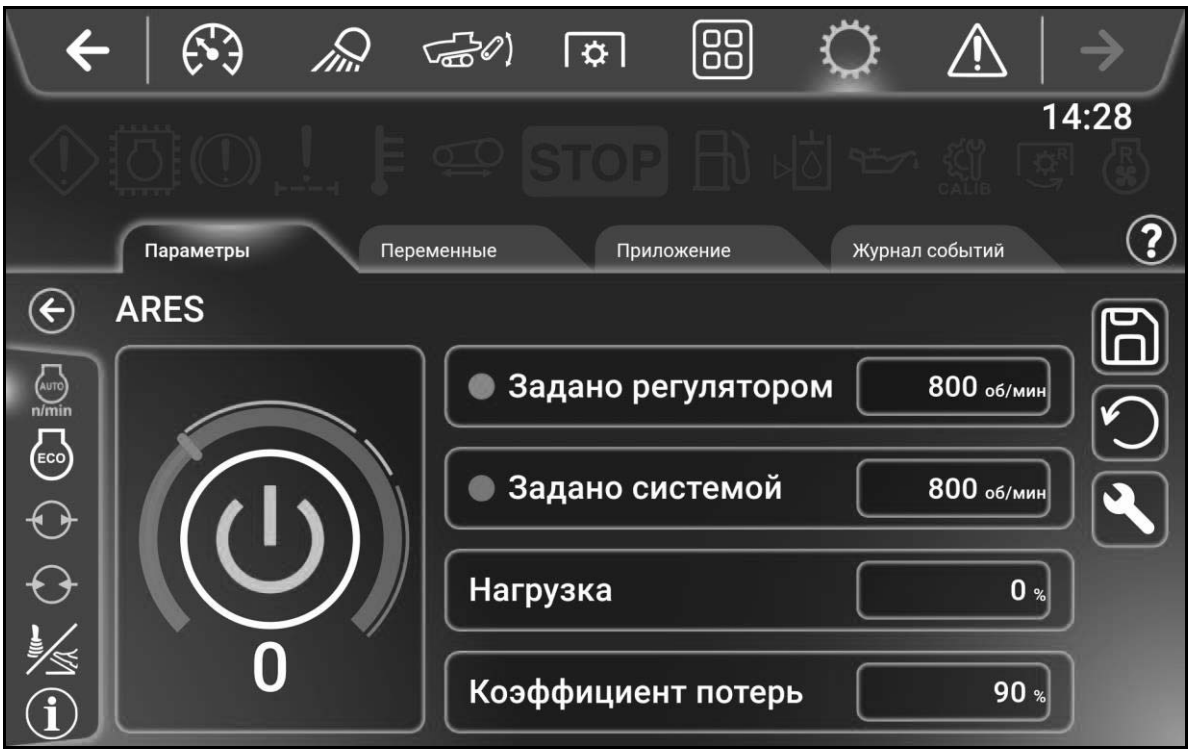


Рисунок 8 – «ARES»

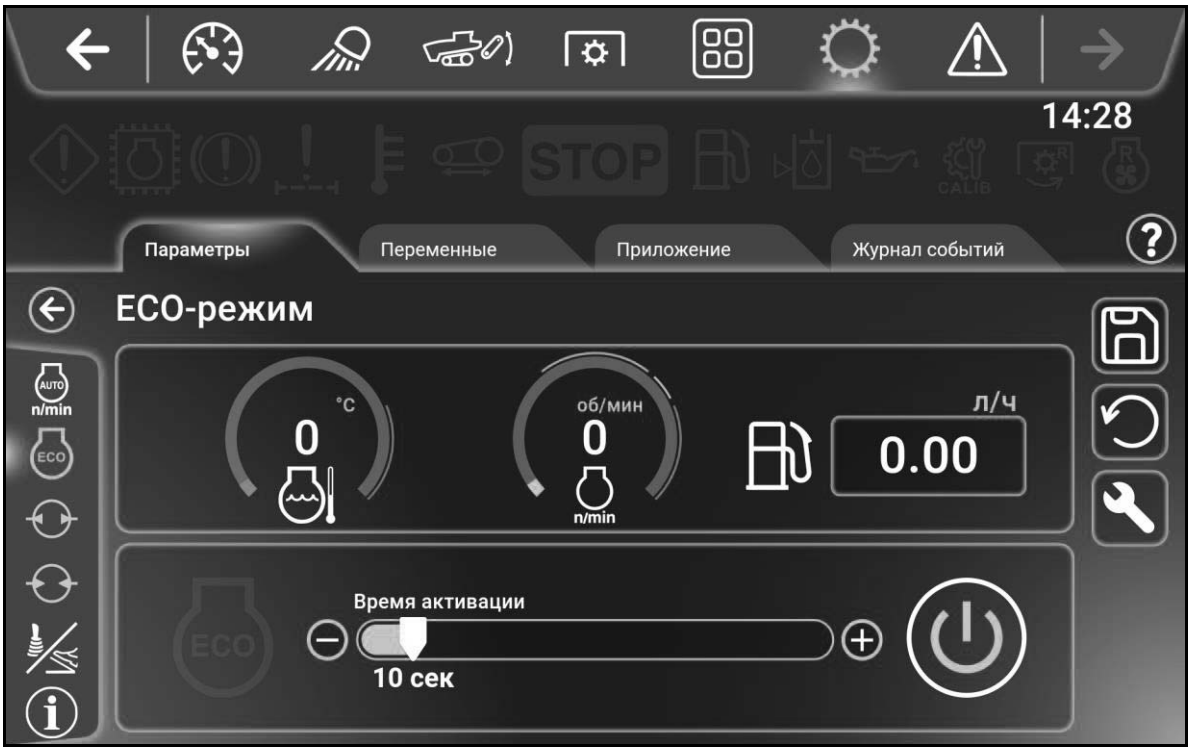


Рисунок 9 – «ECO-режим»

Рисунок 10 – «Гидронасосы»

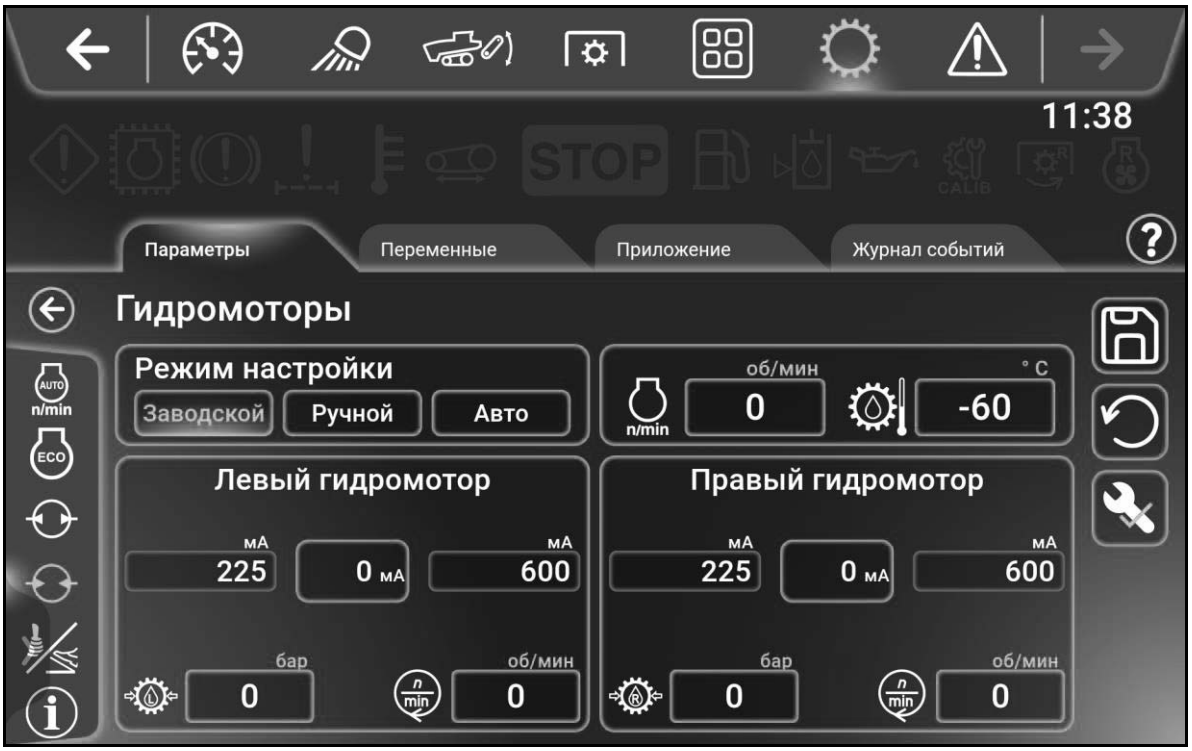


Рисунок 11 – «Гидромоторы»

Рисунок 12 – «Органы управления – Стики»

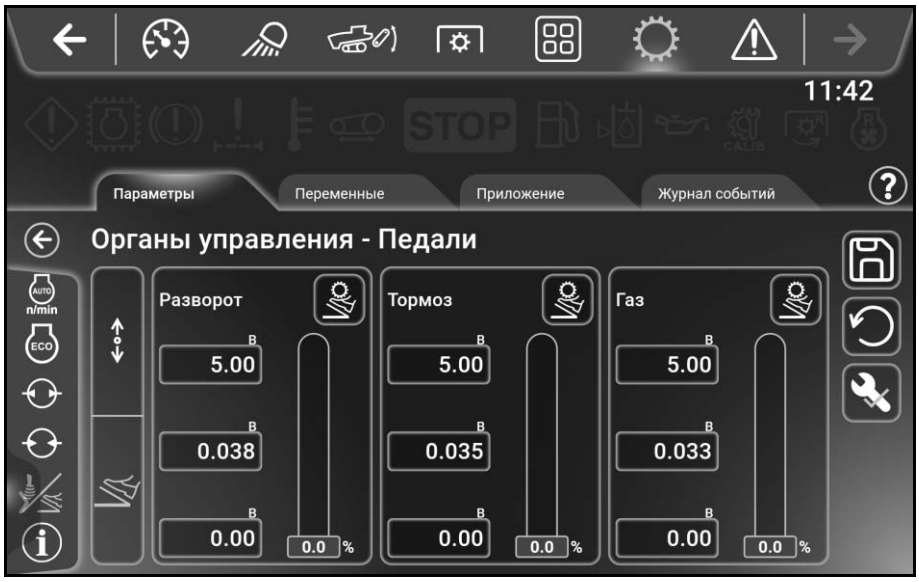


Рисунок 13 – «Органы управления – Педали»

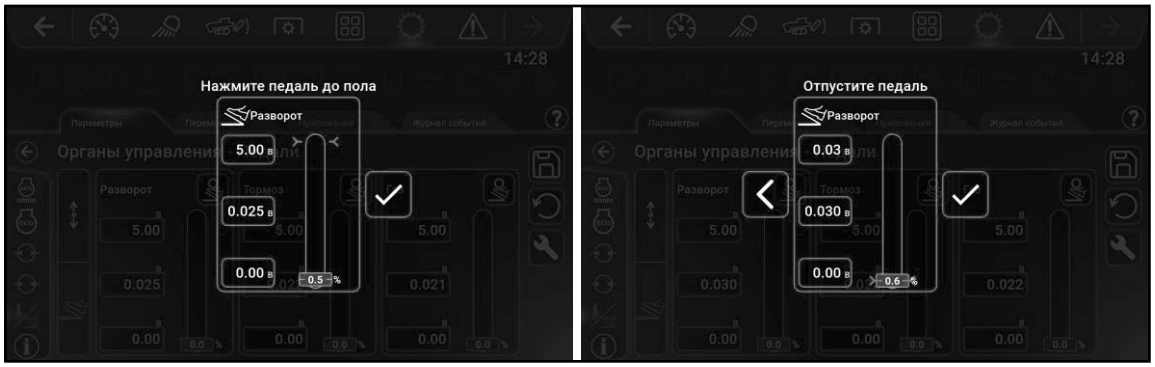


Рисунок 14 – «Калибровка педали»

Рисунок 15 – «Информация о машине»

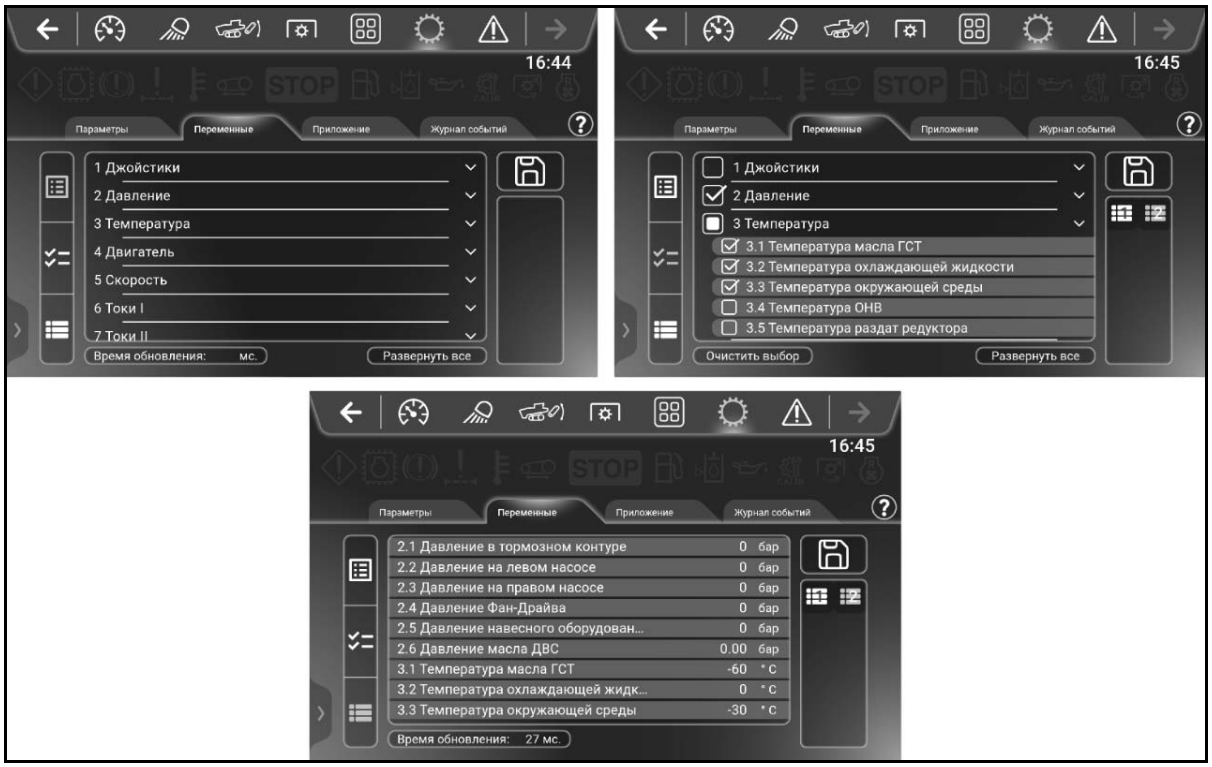


Рисунок 16 – «Переменные» – «Список», «Выборка», «Отображение»

Рисунок 17 – «Приложение» – «Настройки приложения», «Информация о приложении», «Контакты»