



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 «Электронные вычислительные машины»

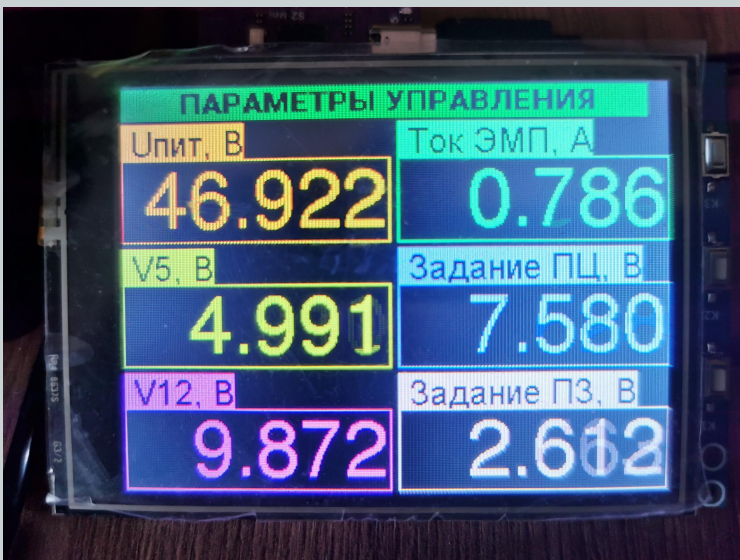
Разработка веб-интерфейса электронного блока управления электрогидравлического усилителя МОЩНОСТИ

Руководитель работы,
к.т.н., доцент каф. ЭВМ
Д.В. Топольский
Автор работы,
студент группы КЭ-406
Ж.С. Величко

Челябинск 2025

Актуальность

В условиях растущей автоматизации промышленных процессов возрастает потребность в удобных и функциональных интерфейсах взаимодействия человека с техникой. Существующие решения, такие как промышленные панели оператора, часто оказываются дорогостоящими, громоздкими и недостаточно гибкими для специфических задач.



Цель и задачи

Целью данной работы является разработка программного обеспечения, обеспечивающего эффективное и удобное взаимодействие пользователя с электронным блоком управления электрогидравлического усилителя мощности.

Для достижения поставленной цели бы сформулированы следующие задачи:

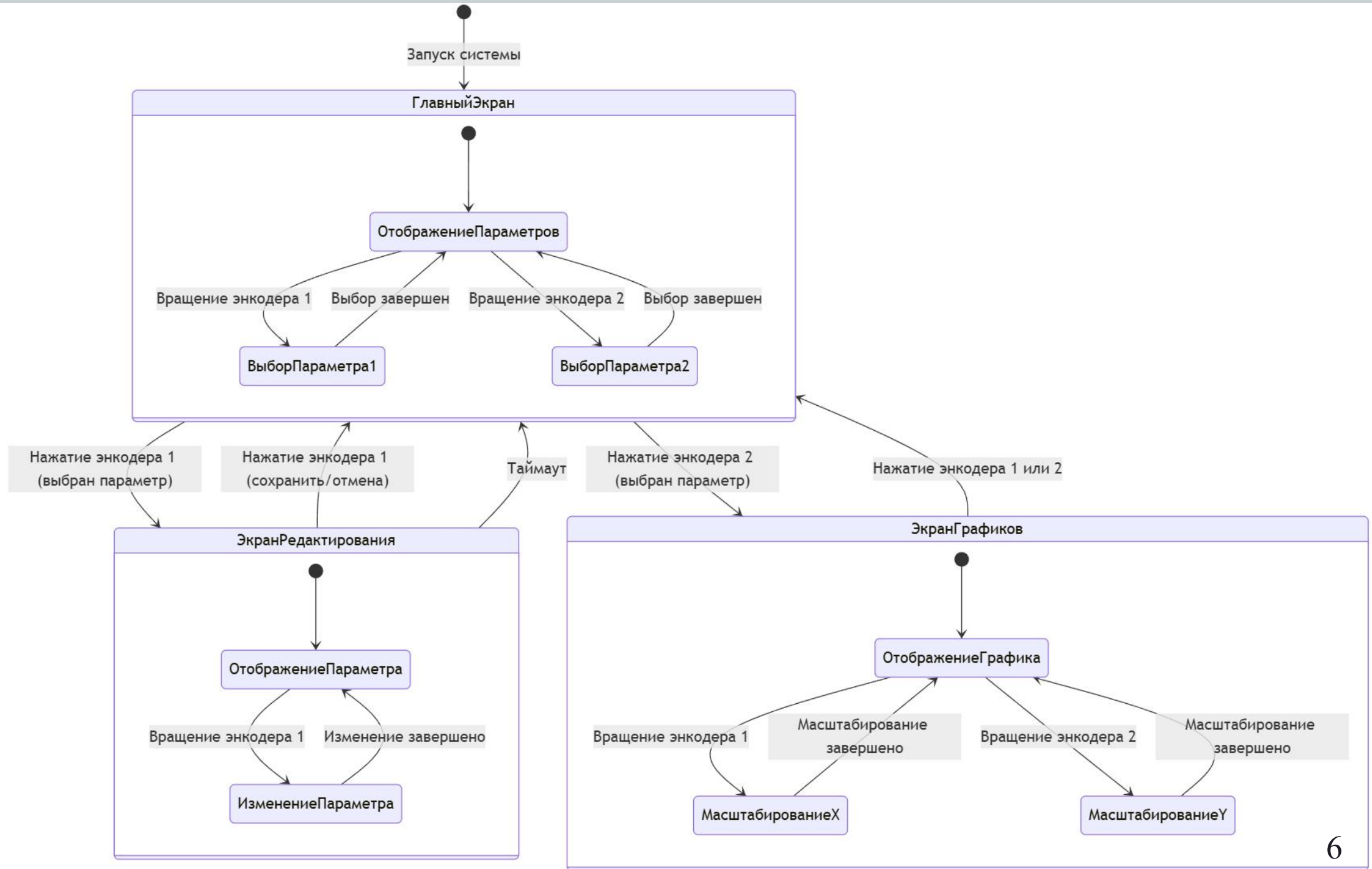
1. Проведение аналитического обзора.
2. Разработка алгоритмов, реализующие функционал интерфейса.
3. Разработка и отладка программного кода на C++.
4. Тестирование.

Обзор GUI-библиотек

Критерий	LVGL	uGFX	TouchGFX
Поддержка MCU/MPU	Широкая, не зависящая от конкретного производителя	Широкая, платформенно-независимая	Оптимизирована для STM32
Язык программирования	C (совместим с C++)	C	C++
Лицензия	MIT (бесплатная, открытый исходный код)	BSD (бесплатная, открытый исходный код)	Проприетарная (бесплатна для STM32)
Простота использования	Относительно простая, с обширной документацией и сообществом	Средняя, требует определенной квалификации	Упрощенный дизайн с помощью TouchGFX Designer
Занимаемая память	Оптимизированная, низкое потребление RAM/ROM	Оптимизированная, самая низкая среди представленных	Требует больше памяти, чем LVGL и uGFX

Обзор GUI-библиотек (продолжение)

Критерий	LVGL	uGFX	TouchGFX
Производительность	Высокая, поддержка аппаратного ускорения	Высокая, но может потребовать оптимизации	Высокая, благодаря оптимизации под STM32
Удобство разработки	Развиты API, широкий набор виджетов	Базовый набор виджетов, более низкоуровневый API	Визуальный редактор (TouchGFX Designer), генерация кода
Гибкость и кастомизация	Высокая, широкие возможности стилизации	Средняя, более ограниченные возможности стилизации	Средняя, ограничения связанные с генерацией кода
Сообщество и поддержка	Активное сообщество, обширная документация	Меньшее сообщество, документация менее полная	Поддержка от STMicroelectronics
Стоимость	Бесплатная	Бесплатная	Бесплатна для использования с STM32



Главный экран

Основная задача: отображение текущих параметров электрогидравлического усилителя мощности (ЭГУР).

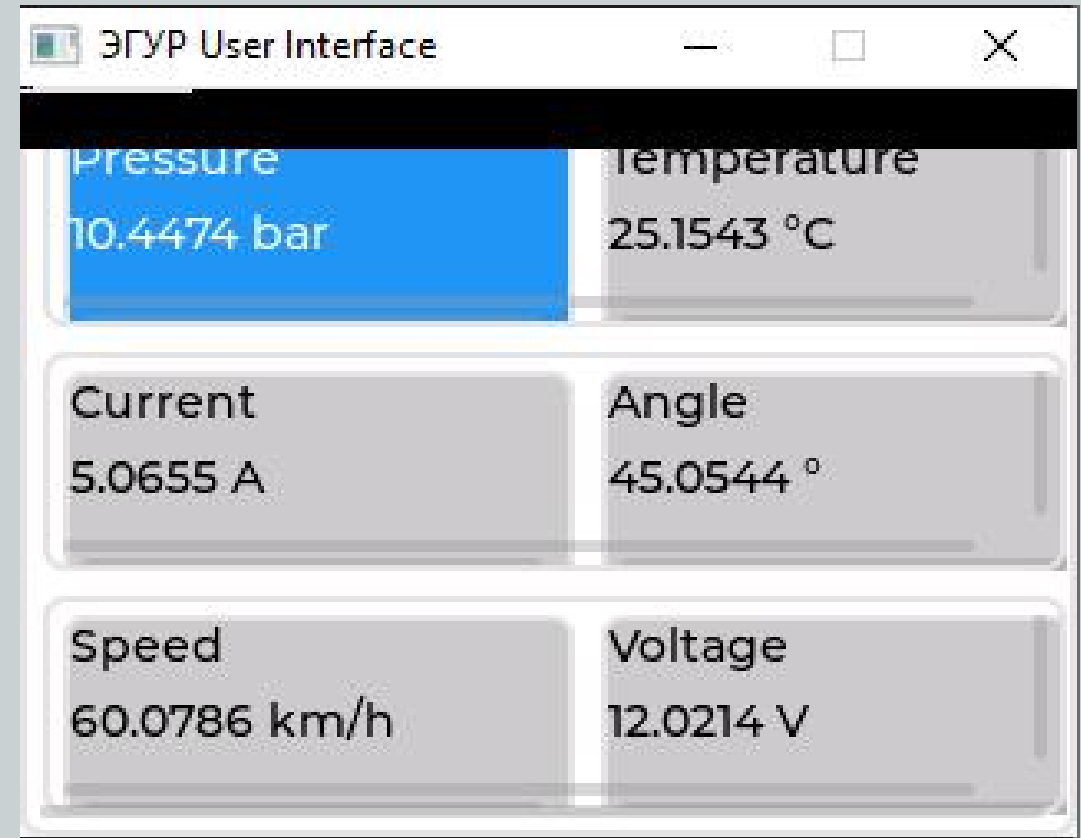
Структура: два столбца по три параметра (всего 6 параметров).

Отображаемые данные: числовые значения параметров с обновлением не менее 10 раз в секунду.

Выделение: выбранный параметр подсвечивается рамкой для дальнейшего редактирования или выбора графика.

Управление:

- первый энкодер — перемещение по списку параметров (выбор);
- второй энкодер — выбор параметра для отображения графика.



Экран редактирования

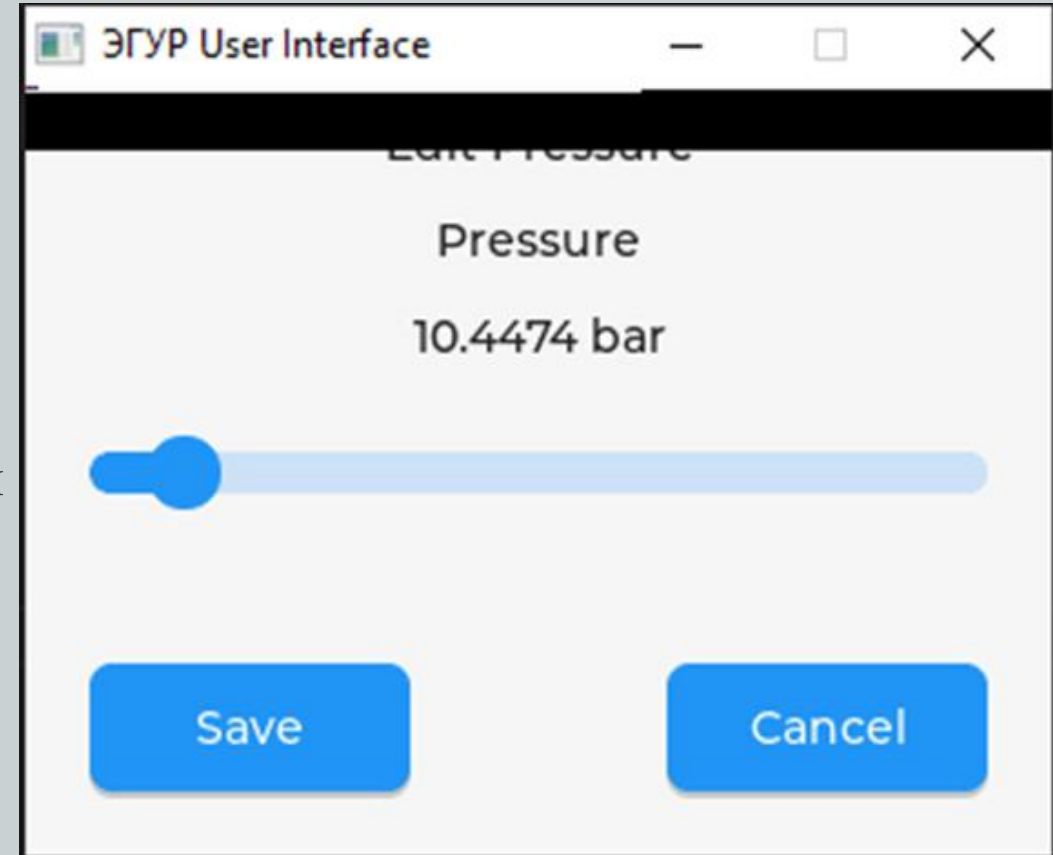
Функция: позволяет изменять выбранный параметр с проверкой на допустимый диапазон.

Интерфейс: отображается название параметра, текущее значение и кнопки сохранения/отмены.

Управление:

- первый энкодер — изменение значения параметра с шагом, равным 1/100 диапазона;
- кнопка первого энкодера — сохранение изменений и возврат на главный экран;
- кнопка второго энкодера — отмена изменений и возврат.

Обработка ошибок: при выходе за пределы параметра значение ограничивается.



Экран графиков

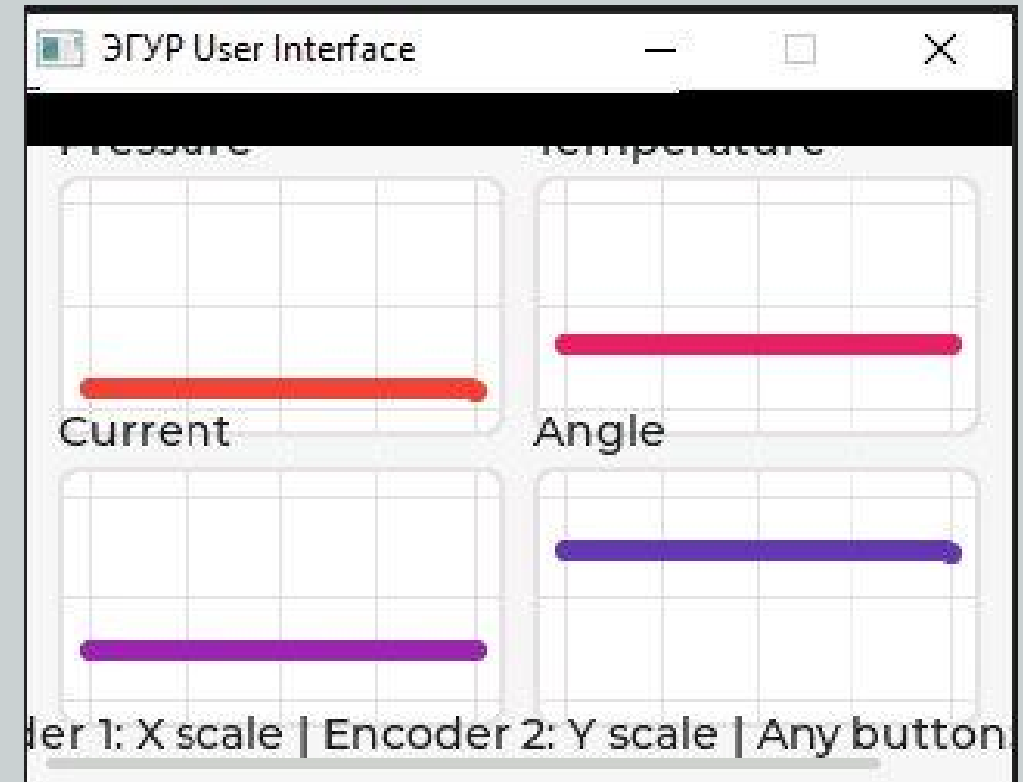
Назначение: визуализация динамики изменения параметров во времени.

Отображение: 4 линейных графика, каждый из которых соответствует одному параметру.

Управление:

- первый энкодер — выбор параметров для отображения на графике;
- второй энкодер — масштабирование по осям X (время) и Y (значение).

Особенности: плавное масштабирование и прокрутка, удобный визуальный анализ.



Архитектура ПО

Программное обеспечение веб-интерфейса построено по модульному принципу с использованием модели конечного автомата для управления состояниями интерфейса.

Основные модули:

- `main.cpp` — инициализация и цикл;
- `data_manager.cpp` — управление параметрами;
- `encoders.cpp` — обработка энкодеров;
- `ui/` — экраны интерфейса;
- `hal/` — аппаратный слой;
- `communication.cpp` — обмен данными.

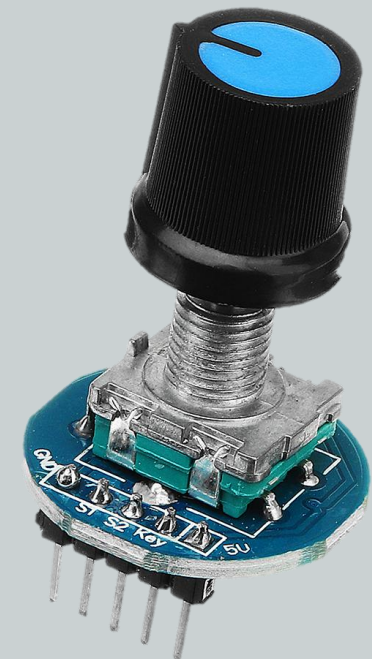
Управление

Управление осуществляется с помощью клавиатуры, эмулирующей работу энкодеров:

- энкодер 1: A (против часовой), D (по часовой), S (кнопка);
- энкодер 2: J (против часовой), L (по часовой), K (кнопка);
- ESC: Выход из приложения.

Алгоритмы обработки энкодеров:

- энкодер 1: выбор параметра, переход в режим редактирования;
- энкодер 2: выбор параметра для графика, переход в режим графика;
- в режиме редактирования: изменение значения параметра и сохранение;
- в режиме графиков: масштабирование и возврат на главный экран.



Отладка на ПК

Для отладки на ПК реализована эмуляция работы микроконтроллера.

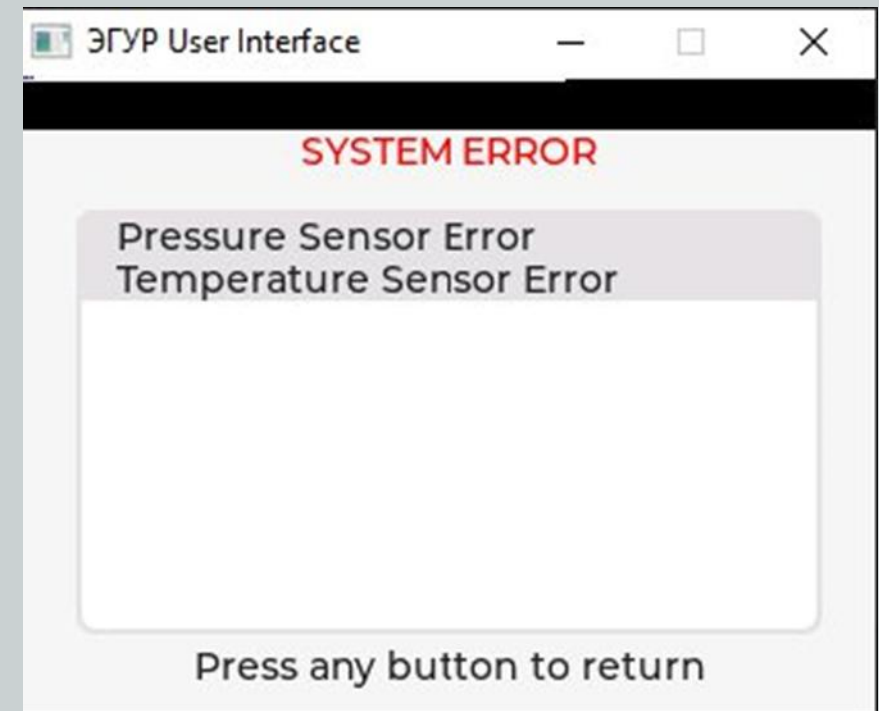
Эмуляция включает в себя:

- использование кросс-компилятора для сборки кода под архитектуру ПК;
- реализацию заглушек для аппаратных функций;
- генерацию симулированных данных для параметров ЭГУР.

Тестирование

Тестирование проводилось в режиме эмуляции на ПК с использованием SDL, и включало в себя:

- проверку обновления значений с частотой ≥ 10 Гц;
- проверку навигации и редактирования параметров;
- тестирование отображения и масштабирования графиков;
- имитацию ошибок и отображение сообщений об ошибках.



Вывод

В ходе выполнения работы успешно достигнута поставленная цель:

1. Выполнен обзор современных GUI-библиотек и выбран оптимальный инструмент — LVGL.
2. Разработаны алгоритмы, обеспечивающие требуемую функциональность интерфейса управления ЭГУР.
3. Реализован программный код с модульной структурой, обеспечивающий удобство поддержки и расширения.
4. Проведено функциональное тестирование в режиме эмуляции, подтверждающее корректность работы.



Спасибо за внимание!