**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#__RefHeading___Toc2163_169777778)

[1 ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ И АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ 7](#__RefHeading___Toc2165_169777778)

[1.1 Обзор предметной области 7](#__RefHeading___Toc2167_169777778)

[1.1.1 Введение в предметную область 7](#__RefHeading___Toc2169_169777778)

[1.2 Анализ существующих решений 8](#__RefHeading___Toc2171_169777778)

[1.2.1 Mikron Mik32 HAL 8](#__RefHeading___Toc2173_169777778)

[1.2.2 Community Mik32 HAL 8](#__RefHeading___Toc2175_169777778)

[1.2.3 Rust STM32 HAL 8](#__RefHeading___Toc2177_169777778)

[1.2.4 Rust E310X HAL 8](#__RefHeading___Toc2179_169777778)

[1.2.5 Rust ESP32 HAL 8](#__RefHeading___Toc2181_169777778)

[1.3 Выводы по разделу 9](#__RefHeading___Toc2183_169777778)

[2 ПОСТАНОВКА ЗАДАЧИ 10](#__RefHeading___Toc2185_169777778)

[2.1 Назначение разрабатываемого ПО 10](#__RefHeading___Toc2187_169777778)

[2.2 Формирование требований к библиотеке 10](#__RefHeading___Toc2189_169777778)

[2.3 Выбор средств разработки 10](#__RefHeading___Toc2191_169777778)

[3 РАЗРАБОТКА 11](#__RefHeading___Toc2193_169777778)

[3.1 Проектирование 11](#__RefHeading___Toc2195_169777778)

[3.2 Разработка 11](#__RefHeading___Toc2197_169777778)

[3.2.1 Разработка модуля xxxx 11](#__RefHeading___Toc2199_169777778)

[4 ДЕМОНСТРАЦИЯ РАБОТЫ БИБЛИОТЕКИ 12](#__RefHeading___Toc2201_169777778)

[4.1 Пример работы модуля xxxx 12](#__RefHeading___Toc2203_169777778)

[ЗАКЛЮЧЕНИЕ 13](#__RefHeading___Toc2205_169777778)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 14](#__RefHeading___Toc2207_169777778)

[ПРИЛОЖЕНИЯ 15](#__RefHeading___Toc2209_169777778)

[ПРИЛОЖЕНИЕ А. xxxx 15](#__RefHeading___Toc2211_169777778)

**ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ**

Микроконтроллер — микросхема для программного управления электронными устройствами.

Встраиваемая система (встроенная система, англ. embedded system) — специализированная микропроцессорная система управления, контроля и мониторинга, концепция разработки которой заключается в том, что такая система будет работать, будучи встроенной непосредственно в устройство, которым она управляет.

Метапрограммирование — вид программирования, связанный с созданием программ, которые порождают другие программы как результат своей работы.

Висячий указатель или висячая ссылка (англ. Dangling pointer, wild pointer, dangling reference) — указатель, не указывающий на допустимый объект соответствующего типа.

Гонка данных (англ. data race) — это состояние в многопоточных программах, когда два или более потоков одновременно пытаются читать и записывать в одну и ту же область памяти без должной синхронизации.

**ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ**

Hardware Abstraction Layer (HAL, Слой аппаратных абстракций) — слой абстрагирования, реализованный в программном обеспечении, находящийся между физическим уровнем аппаратного обеспечения и программным обеспечением, запускаемом на устройстве.

ВВЕДЕНИЕ

Современные микроконтроллеры являются ключевыми компонентами встроенных систем, обеспечивая управление широким спектром устройств – от простых датчиков до сложных промышленных контроллеров и интернета вещей. В условиях стремительного развития embedded-технологий особую важность приобретает не только аппаратная часть микроконтроллеров, но и программные средства для их эффективного использования. Одним из перспективных представителей данного класса устройств является микроконтроллер Mik32 Amur, обладающий сбалансированными характеристиками производительности, энергопотребления и стоимости.

Mik32 Amur является первым и единственным в России микроконтроллером первого уровня. Данный статус позволяет ему иметь приоритет в государственных закупках и гарантированную поставку в условиях ограниченной доступности зарубежных аналогов. Также это означает поддержку отечественной экосистемы разработки, включая инструменты отладки и программирования, и соответствие микроконтроллера требованиям информационной безопасности для применения в системах критической инфраструктуры.

Эффективное использование возможностей микроконтроллера в значительной степени зависит от качества программного обеспечения, в частности – от уровня аппаратной абстракции. HAL представляет собой промежуточный слой между аппаратными ресурсами микроконтроллера и пользовательским приложением, предоставляя удобный и безопасный интерфейс для работы с периферией. Наличие качественного HAL существенно сокращает время разработки, минимизирует количество ошибок, связанных с прямым доступом к регистрам, и облегчает портирование кода между различными платформами.

В последние годы язык программирования Rust приобретает все большую популярность в области разработки встроенного программного обеспечения. По сравнению с традиционными языками, такими как C и C++, Rust предлагает уникальное сочетание высокой производительности, строгой безопасности работы с памятью и современных возможностей метапрограммирования. Система владения и заимствования (ownership и borrowing) исключает целый класс ошибок, связанных с висячими указателями, некорректным доступом к памяти и гонками данных, что особенно критично в embedded-системах с ограниченными ресурсами. Кроме того, Rust предоставляет развитые средства для работы с низкоуровневыми конструкциями, позволяя писать эффективный код, близкий к железу, но без ущерба для безопасности и надёжности.

Актуальность данной работы обусловлена несколькими ключевыми факторами. Во-первых, несмотря на растущую популярность языка Rust в embedded-разработке, для многих микроконтроллеров, включая Mik32 Amur, отсутствуют готовые и полнофункциональные решения HAL на этом языке. Во-вторых, существующие реализации HAL для Mik32 Amur, как правило, написаны на C и не в полной мере используют современные возможности языков программирования, такие как строгая типизация, безопасность памяти и выразительные абстракции. В-третьих, переход на Rust в embedded-сфере соответствует общемировому тренду на повышение надёжности и безопасности программного обеспечения, особенно в критически важных системах.

Разработка HAL для Mik32 Amur на языке Rust позволит не только заполнить существующий пробел в инструментарии для данного микроконтроллера, но и продемонстрировать преимущества Rust перед традиционными языками embedded-разработки. Создаваемая библиотека сможет стать основой для будущих проектов на Mik32 Amur, обеспечивая разработчиков удобным, безопасным и эффективным инструментом для работы с аппаратными ресурсами микроконтроллера. Кроме того, данный проект внесёт вклад в развитие экосистемы Rust для встроенных систем, расширяя область его применения и демонстрируя его возможности на практике.

Таким образом, данная дипломная работа направлена на создание современного, надёжного и удобного HAL для микроконтроллера Mik32 Amur с использованием языка программирования Rust, что соответствует актуальным тенденциям в области разработки встроенного программного обеспечения и открывает новые возможности для сообщества embedded-разработчиков.

1. ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ И АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ

## Обзор предметной области

### Введение в предметную область

xxxx

## Анализ существующих решений

### Mikron Mik32 HAL

### Community Mik32 HAL

### Rust STM32 HAL

### Rust E310X HAL

### Rust ESP32 HAL

xxxx

## Выводы по разделу

xxxx

1. ПОСТАНОВКА ЗАДАЧИ

## Назначение разрабатываемого ПО

## Формирование требований к библиотеке

## Выбор средств разработки

xxxx

1. РАЗРАБОТКА

## Проектирование

## Разработка

### Разработка модуля xxxx

xxxx

1. ДЕМОНСТРАЦИЯ РАБОТЫ БИБЛИОТЕКИ

## Пример работы модуля xxxx

ЗАКЛЮЧЕНИЕ

Xxxx

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Микроконтроллер // [Электронный ресурс]: Wikipedia. – URL: <https://ru.wikipedia.org/wiki/Микроконтроллер> (дата обращения: 01.06.2025)
2. Встраеваемая система // [Электронный ресурс]: Wikipedia. – URL: <https://ru.wikipedia.org/wiki/Встраиваемая_система> (дата обращения: 01.06.2025)
3. Слой аппаратных абстракций // [Электронный ресурс]: Wikipedia. – URL: <https://ru.wikipedia.org/wiki/Слой_аппаратных_абстракций> (дата обращения: 01.06.2025)
4. Метапрограммирование // [Электронный ресурс]: Wikipedia. – URL: <https://ru.wikipedia.org/wiki/Метапрограммирование> (дата обращения: 01.06.2025)
5. Висячий указатель // [Электронный ресурс]: Wikipedia. – URL: <https://ru.wikipedia.org/wiki/Висячий_указатель> (дата обращения: 01.06.2025)
6. Data Race // [Электронный ресурс]: SecurityLab. – URL: <https://www.securitylab.ru/glossary/data_race/> (дата обращения: 01.06.2025)

ПРИЛОЖЕНИЯ

## ПРИЛОЖЕНИЕ А. xxxx