**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

Факультет информационных технологий и программной инженерии

Кафедра программной инженерии и вычислительной техники

Курсовая работа

по дисциплине

«Проектирование и архитектура программных систем»

на тему

«Разработка HAL для микроконтроллера Mik32 Amur»

Выполнил:

студент группы ИКПИ-11

Крылов А.В.

Принял:

Смирнов. К.А.

Санкт-Петербург

2024 г.

**Оглавление**

[Словарь с терминологией 3](#__RefHeading___Toc300_4009604101)

[1. Актуальность 4](#__RefHeading___Toc302_4009604101)

[2. Общие сведения 5](#__RefHeading___Toc304_4009604101)

[2.1. Полное наименование системы и ее условное обозначение 5](#__RefHeading___Toc306_4009604101)

[2.2. Плановые сроки начала и окончания работы по созданию системы 5](#__RefHeading___Toc308_4009604101)

[3. Назначение и цели создания системы 6](#__RefHeading___Toc310_4009604101)

[3.1. Назначение системы 6](#__RefHeading___Toc312_4009604101)

[3.2. Цели создания системы 7](#__RefHeading___Toc314_4009604101)

[4. Характеристика объекта информатизации 8](#__RefHeading___Toc316_4009604101)

[4.1. Краткие сведения об объекте информатизации или ссылки на документы, содержащие такую информацию 8](#__RefHeading___Toc318_4009604101)

[4.2. Сведения об условиях эксплуатации объекта информатизации 9](#__RefHeading___Toc320_4009604101)

[5. Характеристика объекта информатизации 10](#__RefHeading___Toc322_4009604101)

[5.1. Перечень подсистем, их назначение и основные характеристики 10](#__RefHeading___Toc324_4009604101)

[5.2. Требования к функциям (задачам), выполняемым системой 13](#__RefHeading___Toc326_4009604101)

[5.3. Требования к способам и средствам связи для информационного обмена между компонентами системы 15](#__RefHeading___Toc328_4009604101)

[5.4. Требования к эргономике и технической эстетике 15](#__RefHeading___Toc330_4009604101)

[5.5. Требования к защите информации от несанкционированного доступа 15](#__RefHeading___Toc332_4009604101)

[5.6. Перспективы развития, модернизации системы 15](#__RefHeading___Toc334_4009604101)

[6. Требования к видам обеспечения 16](#__RefHeading___Toc336_4009604101)

[6.1. Требования к лингвистическому обеспечению 16](#__RefHeading___Toc338_4009604101)

[6.2. Требования к программному обеспечению 16](#__RefHeading___Toc340_4009604101)

[Заключение 17](#__RefHeading___Toc342_4009604101)

## Словарь с терминологией

**Библиотека** — сборник подпрограмм или объектов, используемых для разработки программного обеспечения.

**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.

**Интернет вещей** (англ. Internet of Things, IoT) — концепция сети передачи данных между физическими объектами («вещами»), оснащёнными встроенными средствами и технологиями для взаимодействия друг с другом или с внешней средой.

**Интерфейс** (англ. interface) — это устройство или система для взаимодействия между не связанными друг с другом объектами.

**Исполнительный** **механизм** — устройство в системе автоматического регулирования, дистанционного управления и других систем управления, осуществляющее механическое перемещение регулирующего органа.

**Компилируемый язык программирования** — язык программирования, исходный код которого преобразуется компилятором в машинный код.

**Фреймворк** — программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

**Язык ассемблера** (англ. assembly language) — это язык программирования низкого уровня, который представляет команды процессора в виде, доступном для чтения человеком.

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

**LCD-дисплей** — экран на основе жидких кристаллов.

## 1. Актуальность

Современные тенденции в области разработки встроенных систем и интернета вещей требуют высокой производительности, надёжности и безопасности программного обеспечения. Микроконтроллеры, такие как Mik32 Amur, становятся все более популярными благодаря своей способности эффективно обрабатывать данные и поддерживать различные протоколы коммуникации. Однако для реализации полного потенциала этих устройств необходимы надёжные программные решения, которые обеспечивают взаимодействие между аппаратным обеспечением и приложениями.

Разработка HAL для микроконтроллера Mik32 Amur актуальна по нескольким причинам:

* **Безопасность**: HAL минимизирует риски, связанные с управлением памятью, что критично для встроенных систем, где сбои могут привести к серьезным последствиям. Это особенно важно для приложений, работающих в условиях реального времени, где стабильность и нёдежность являются приоритетами;
* **Производительность**: Высокая производительность компилируемых языков позволяет эффективно использовать ресурсы микроконтроллера Mik32 Amur, что особенно важно для задач, требующих быстрой обработки данных и минимальной задержки;
* **Совместимость** **и** **гибкость**: Mik32 Amur поддерживает множество интерфейсов и протоколов, что делает HAL легко интегрируемым с существующими решениями и позволяющим разработчикам быстро адаптировать свои приложения под новые требования;
* **Поддержка** **сообщества** **и** **экосистемы**: Активное сообщество и растущая библиотека готовых решений создают благоприятные условия для разработки новых инструментов и библиотек, что способствует ускорению процесса разработки и повышению качества программных продуктов.

## 2. Общие сведения

### **2.1. Полное наименование системы и ее условное обозначение**

**Полное наименование:** “Слой аппаратных абстракций для микроконтроллера Mik32 Amur” – библиотека для написания пользовательских программ. Она позволяет разработчикам программного обеспечения получать доступ к внутренним ресурсам и интерфейсам коммуникации микроконтроллера. Также, при надобности, с помощью функционала библиотеки возможна огранизация работы с подключенными к микроконтроллеру устройствами периферии.

**Условное обозначение:** Mik32 Amur HAL.

### **2.2. Плановые сроки начала и окончания работы по созданию системы**

**Дата начала:** 11.01.2024 г.

**Дата окончания:** 01.05.2025 г.

## 3. Назначение и цели создания системы

### **3.1. Назначение системы**

HAL для микроконтроллера Mik32 Amur предназначен для упрощения разработки и взаимодействия с аппаратными компонентами. Он ориентирован как на индивидуальных разработчиков, так и на команды, работающие над проектами в области встроенных систем.

HAL обеспечивает разработчиков высокоуровневым интерфейсом, который позволяет легко и эффективно взаимодействовать с различными аппаратными модулями, такими как датчики, исполнительные механизмы и интерфейсы коммуникаций, что позволяет разработчикам сосредоточиться на логике приложения, а не на низкоуровневых деталях работы оборудования.

Кроме того, HAL предоставляет инструменты для анализа и оптимизации работы микроконтроллера, позволяя разработчикам адаптировать свои решения с учетом целевых требований производительности и энергопотребления.

HAL включает в себя широкий спектр инструментов для мониторинга и отладки, таких как вывод информации в консоль разработчика, средства тестирования и профилирования, что позволяет разработчикам проводить глубокий анализ производительности и выявлять потенциальные проблемы на ранних стадиях разработки.

Суммируя вышеизложенное, система HAL для микроконтроллера Mik32 Amur стремится предоставить высокотехнологичное решение для разработки встроенных приложений, учитывая сложные аспекты работы с аппаратными компонентами и предоставляя разработчикам инструменты для достижения оптимальной производительности и надёжности своих решений. Аналогами Mik32 Amur HAL являются различные библиотеки и фреймворки, такие как ESP IDF и Arduino Core.

### **3.2. Цели создания системы**

Целями создания Mik32 Amur HAL являются:

* Создание высокоуровневого интерфейса:

Разработка удобного и интуитивно понятного программного интерфейса для взаимодействия с аппаратными компонентами микроконтроллера.

* Повышение безопасности программного обеспечения:

Использование языка Rust в разработке HAL для снижения рисков, связанных с ошибками, такими как переполнение буфера и некорректное управление памятью.

* Оптимизация производительности:

Обеспечение эффективного использования ресурсов микроконтроллера, включая поддержку многопоточности и асинхронного программирования.

* Обеспечение совместимости и расширяемости:

Поддержка множества интерфейсов и протоколов, что обеспечит совместимость с различными устройствами и системами.

* Упрощение процесса тестирования и отладки:

Разработка инструментов, облегчающих тестирование и отладку программного обеспечения.

Таким образом, цели создания HAL для микроконтроллера Mik32 Amur направлены на разработку надёжной, безопасной и эффективной платформы, позволяющей упростить процесс разработки встроенных приложений.

## 4. Характеристика объекта информатизации

### **4.1. Краткие сведения об объекте информатизации или ссылки на документы, содержащие такую информацию**

**Основные функции:**

* Управление внутренними ресурсами микроконтроллера;
* Инструменты для оценки эффективности работы микроконтроллера;
* Интеграция с различными внешними модулями и протоколами для расширения функциональности.

**Технические характеристики:**

* Обеспечивает высокую производительность и надежность при работе с аппаратными ресурсами;
* Поддерживает различные интерфейсы и протоколы для интеграции с внешними устройствами, что позволяет легко расширять функциональность системы;
* Использует язык программирования Rust.

**Условия эксплуатации:**

* HAL может быть интегрирован в различные программные среды, что позволяет разработчикам легко адаптировать его к специфическим требованиям проектов;
* Система предназначена для работы в условиях, обеспечивающих стабильную и надёжную эксплуатацию микроконтроллера.

**Преимущества:**

* Обеспечение эффективного управления ресурсами и оптимизация работы микроконтроллера;
* Интеграция с различными внешними модулями для расширения возможностей системы;
* HAL обеспечивает надежное управление памятью, предотвращая переполнение буферов и другие ошибки, что способствует стабильной и безопасной работе приложений на микроконтроллере.

### **4.2. Сведения об условиях эксплуатации объекта информатизации**

**Температурные условия:**

Микроконтроллер Mik32 Amur должен функционировать в диапазоне температур, соответствующем спецификациям производителя.

**Электропитание:**

HAL должен быть спроектирован с учётом стабильного и надёжного питания. Микроконтроллер должен работать в условиях, когда напряжение питания находится в пределах допустимых значений, что обеспечивает защиту от перепадов напряжения и минимизирует риск повреждения оборудования.

**Электромагнитные помехи:**

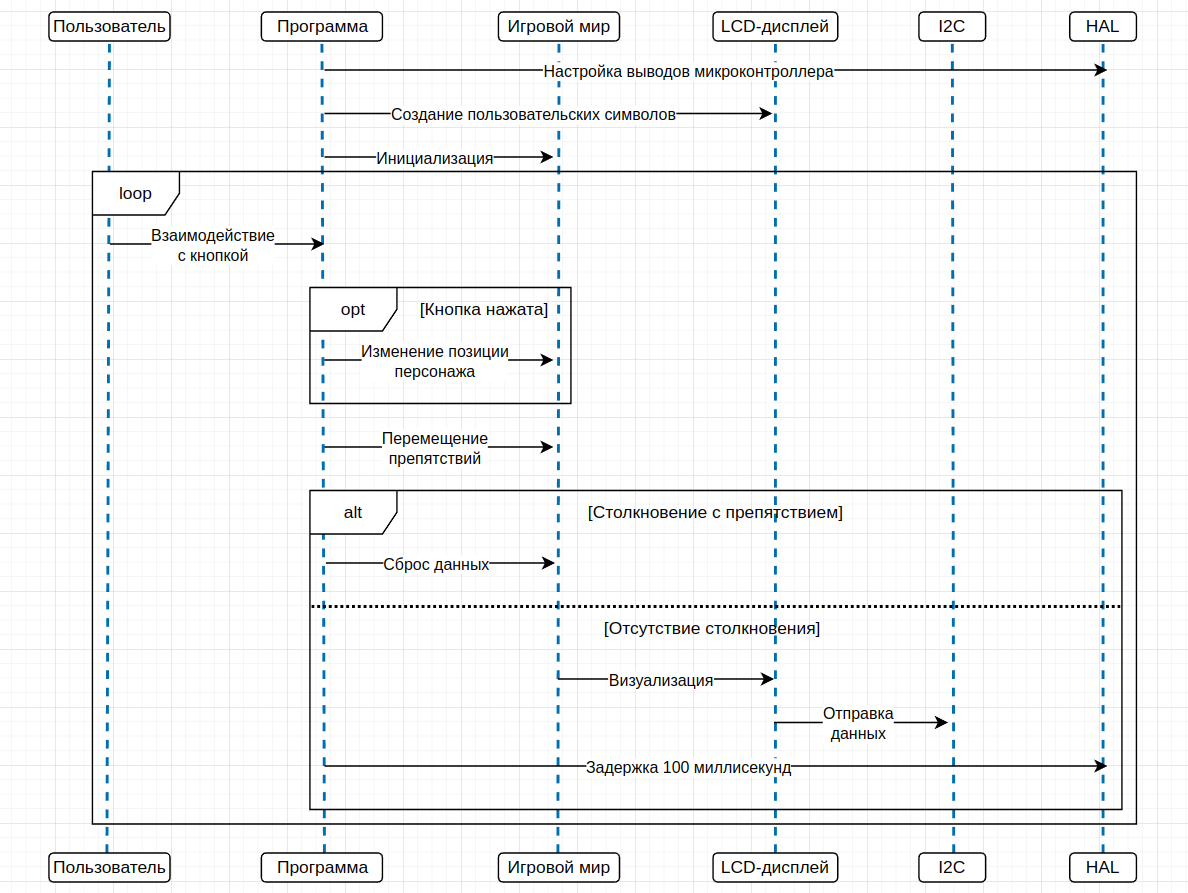
Должна быть обеспечена защита от электромагнитных помех и других источников интерференции, которые могут повлиять на работу микроконтроллера и его взаимодействие с внешними устройствами.

## 5. Характеристика объекта информатизации

### **5.1. Перечень подсистем, их назначение и основные характеристики**

* Подсистема управления вводом/выводом
* Подсистема управления часами и таймерами
* Подсистема управления памятью
* Подсистема управления прерываниями
* Подсистема управления внешними устройствами

На рисунках 1, 2, 3 и 4 представлены схемы для теоретической программы, написанной при помощи HAL. Программа реализует небольшую видеоигру, в которой пользователь через нажатие на подключённую к микроконтроллеру кнопку управляет персонажем, уклоняющимся от препятствий. Игровое состояние отображается на LCD-дисплее в реальном времени.

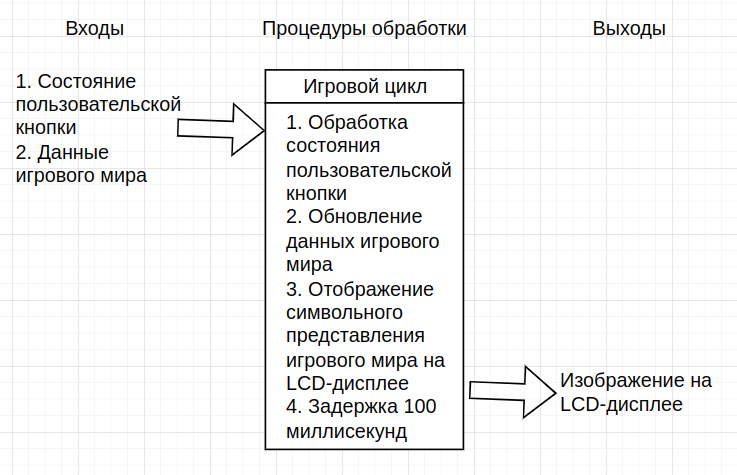


*Рисунок 1. Диаграмма взаимодействия между подсистемами*

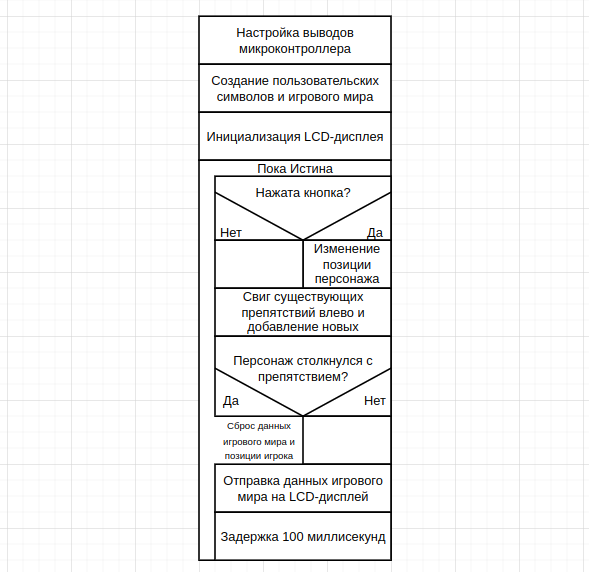


*Рисунок 2. Диаграмма организации данных в программе*

### **5.2. Требования к функциям (задачам), выполняемым системой**



*Рисунок 3. Схема HIPO для программы*



*Рисунок 4. Диаграмма Несси-Шнейдермана*

### **5.3. Требования к способам и средствам связи для информационного обмена между компонентами системы**

Поддержка различных интерфейсов коммуникации, таких как UART, SPI и I2C для обеспечения возможности взаимодействия с различными периферийными устройствами.

Разделение HAL на небольшие, автономные модули может облегчить коммуникацию между компонентами и обеспечить гибкость в разработке и масштабировании.

Внедрение механизмов мониторинга и журналирования обмена данными для отслеживания производительности, выявления проблем и обеспечения безопасности.

Способы связи должны быть гибкими и легко адаптируемыми для совместимости с различными технологиями и изменениями в инфраструктуре.

Реализация механизмов обработки ошибок для обеспечения надёжности и целостности обмена информацией.

### **5.4. Требования к эргономике и технической эстетике**

HAL должен удовлетворять следующим требованиям:

* Информирующие сообщения и сообщения об ошибках в процессе выполнения программы выводятся на русском языке.
* Использование стандарта API.

### **5.5. Требования к защите информации от несанкционированного доступа**

HAL предназначен для прямого взаимодействия с разработчиками программного обеспечения, вследствие чего не имеет требований к защите информации он несанкционированного доступа.

### **5.6. Перспективы развития, модернизации системы**

* Увеличение числа поддерживаемых интерфейсов коммуникации;
* Оптимизация работы HAL при помощи использования вставок кода на языке ассемблера.

## 6. Требования к видам обеспечения

### **6.1. Требования к лингвистическому обеспечению**

Разработка программного обеспечения должна вестись на высокоуровневом языке программирования.

HAL должен быть написан на языке Rust. Использование другого языка снизит производительность и безопасность работы с памятью микроконтроллера.

Программный интерфейс HAL может быть реализован средствами иных языков программирования, но Rust является рекомендацией, поскольку благодаря ему есть возможность писать асинхронный код, который может значительно снизить потребление энергии микроконтроллером.

### **6.2. Требования к программному обеспечению**

**Язык программирования и фреймворки:**

* + - * Rust 1.82.0+: Использование последней стабильной версии языка Rust.

**Коммуникация и API:**

* + - * rust-embedded-hal: Стандартизация API для взаимодействия с ресурсами микроконтроллера и переферией.

**Мониторинг и Журналирование:**

* + - * log crate: Вывод информирующих сообщений и сообщений об ошибках.

**Тестирование:**

* + - * cargo test: Официальный фреймворк языка Rust для тестирования кода.

**Управление Зависимостями и Виртуальные Окружения:**

* + - * cargo: Официальная система сборки и контроля зависимостей языка Rust.

## Заключение

Проектирование HAL для микроконтроллера Mik32 Amur представляет собой важный этап в создании эффективного и современного инструмента для взаимодействия с аппаратным обеспечением. Учитывая растущий интерес к встраиваемым системам и потребность в надежных решениях, HAL для Mik32 Amur является многообещающим проектом.

Формулирование требований к программному обеспечению учитывает современные стандарты безопасности, эффективность взаимодействия компонентов, а также гибкость и расширяемость системы. Выбор языка Rust, известного своей безопасностью и производительностью, обеспечивает надежность разработки и упрощает поддержку кода.

Особое внимание уделяется аспектам безопасности, включая управление памятью и предотвращение ошибок, что критически важно для работы с встраиваемыми системами. Также предусмотрены механизмы тестирования и журналирования, что позволяет быстро обнаруживать и устранять потенциальные проблемы.

HAL для Mik32 Amur нацелен на обеспечение удобства и безопасности при разработке приложений, использующих данный микроконтроллер. Его реализация открывает новые возможности для создания современных и востребованных решений в области встраиваемых систем, соответствующих требованиям стремительно развивающегося рынка.