МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество «Казахский национальный исследовательский технический университет имени К.И.Сатпаева»

Институт Автоматики и информационных технологий

Кафедра Робототехники и технических средств автоматики

Ганиев Шухрат Рустамович

Разработка системы контроля доступа с анализом рисков и методами обеспечения безопасности на базе ESP32

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту

6В07113 – Робототехника и мехатроника

Алматы 2025

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество «Казахский национальный исследовательский технический университет имени К.И.Сатпаева»

Институт Автоматики и информационных технологий

Кафедра Робототехники и технических средств автоматики

ДОПУЩЕН К ЗАЩИТЕ
Заведующий кафедрой РТиТСА
кандидат технических наук, профессор
_____Ожикенов К. А.
«___» _____ 2025 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту

На тему: «Разработка системы контроля доступа с анализом рисков и методами обеспечения безопасности на базе ESP32»

6B07113 – Робототехника и мехатроника

Выполнил	Ганиев Шухрат Рустамович
Рецензент	Научный руководитель
Зав. кафедрой академии «Общие научные дисциплины» к.т.н., доцент	к.ф.-м. н., ассоциированный профессор
_____ Сейдилдаева А.К.	_____ Бактыбаев М.К.
«30» май 2025 г.	«30» май 2025 г.

Алматы 2025

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество «Казахский национальный исследовательский технический университет имени К.И.Сатпаева»

Институт Автоматики и информационных технологий

Кафедра Робототехники и технических средств автоматики

6В07111 – Робототехника и мехатроника

УТВЕРЖДАЮ
Заведующий кафедрой РТиТСА
кандидат технических наук, профессор
_____Ожикенов К. А.
«___» _____ 2025 г.

ЗАДАНИЕ
на выполнение дипломного проекта

Студенту Ганиеву Шухрату Рустамовичу

Тема: Разработка системы контроля доступа с анализом рисков и методами обеспечения безопасности на базе ESP32

Утверждена приказом ректора_____ №521-о/о от «13» 11 2025 г.

Срок сдачи законченной работы «20» 05 2025 г.

Исходные данные к дипломному проекту: ESP32 DevKit v1, RFID -модуль RC522, электромагнитный замок, блок питания 5В 2А, реле(1-канальное).

Перечень подлежащих разработке в дипломном проекте вопросов:

а) проектирования архитектуры системы;

б) выбора компонентов;

в) программной реализации, тестирования, анализа рисков и визуализации данных.

Перечень графического материала (с точным указанием обязательных чертежей): представлены слайдов презентации работы

Рекомендуемая основная литература: из 23 наименований

ГРАФИК
подготовки дипломной работы (проекта)

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечания
Исследовательская часть	27.03.2025	Выполнено
Теоретическая часть	18.04.2025	Выполнено
Практическая часть	12.05.2025	Выполнено
Итог	20.05.2025	Выполнено

Подписи
консультантов и норм контролера на законченную дипломную работу (проект)с указанием относящихся к ним разделов работы (проекта)

Наименование разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Нормконтролер	Кальмаков Т.Т.	05.06.2025	

Научный руководитель: _____ Бактыбаев М.К.

Задание принял к исполнению обучающийся: _____ Ганиев Ш.Р.

Дата «05» 06 2025

4

АҢДАТПА

Бұл дипломдық жұмыста ESP3 микроконтроллері мен RFID модуліне негізделген зияткерлік қол жеткізуді басқару жүйесін әзірлеу қарастырылады. Жұмыстың негізгі мақсаты — Telegram-бот арқылы сандық растаумен бірге физикалық сәйкестендіруді біріктіретін көп деңгейлі қауіпсіздік жүйесін құру. Жоба аясында ESP32, FastAPI сервері, Telegram және дерекқор арасындағы өзара әрекеттесу архитектурасы жүзеге асырылды. Жүйе әртүрлі сценарийлерде, соның ішінде сәтсіздікке төзімділік пен аномальды кіру әрекеттерінде сәтті сыналды. Ұсынылған шешімді білім беру мекемелерінде, кеңсе ғимараттарында және тұрғын үй кешендерінде енгізуге болады.

АННОТАЦИЯ

В данной работе представлена разработка интеллектуальной системы контроля доступа, основанной на микроконтроллере ESP32 и RFID-технологии. Система обеспечивает управление физическим доступом с использованием бесконтактных карт и повышает безопасность за счёт многоуровневой проверки и анализа рисков. Дополнительный уровень безопасности реализован через Telegram-бота, позволяющего владельцу карты подтверждать или отклонять попытки доступа в режиме реального времени. Серверная часть разработана с использованием FastAPI и обеспечивает централизованную обработку запросов, хранение данных и анализ событий.

ANNOTATION

This bachelor’s thesis focuses on the development of an intelligent access control system based on the ESP32 microcontroller and an RFID module. The main goal is to design a multi-level security system combining physical identification with digital confirmation via a Telegram bot. The project implements an integrated architecture involving the ESP32, a FastAPI-based server, Telegram communication, and a database. The system was successfully tested in various scenarios, including failure resistance and abnormal entry attempts. The proposed solution is suitable for deployment in educational institutions, office buildings, and residential complexes.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В условиях стремительного развития цифровых технологий и увеличения количества угроз информационной безопасности возрастает потребность в эффективных и надёжных системах контроля доступа. Традиционные системы часто не обладают достаточной гибкостью, масштабируемостью и уровнем защиты от современных угроз. Использование микроконтроллеров нового поколения, таких как ESP32 [1], открывает широкие возможности для создания интеллектуальных и безопасных решений в области доступа к физическим объектам и информационным ресурсам.
ESP32 [1] сочетает в себе Wi-Fi и Bluetooth-модули, достаточную вычислительную мощность, разнообразные интерфейсы для подключения датчиков и низкую стоимость, что делает его идеальной платформой для встраиваемых систем безопасности. Актуальность данной темы обусловлена необходимостью разработки доступных, защищённых и масштабируемых решений для доступа в жилых, коммерческих и промышленных помещениях.

Современные системы контроля доступа (СКУД) играют важную роль в обеспечении безопасности на объектах различного назначения — от офисных зданий до промышленных предприятий и жилых комплексов. С развитием технологий и повсеместным внедрением концепции Интернета вещей (IoT) требования к таким системам стали более жёсткими.

Всё чаще используются микроконтроллеры с поддержкой сетевых протоколов и интеграцией с мобильными и облачными сервисами. Одним из таких решений является платформа ESP32 [1], которая сочетает высокую вычислительную мощность, малые размеры и поддержку Wi-Fi/Bluetooth. В сочетании с RFID [2]-технологией и ботами в мессенджерах (например, Telegram) становится возможной реализация дешёвых, надёжных и масштабируемых СКУД.

Цель данной работы — разработка доступной, функциональной и безопасной СКУД, основанной на ESP32 [1] и RFID [2], с Telegram-ботом в качестве средства дополнительной проверки доступа.

В ходе работы были решены *задачи* проектирования архитектуры системы, выбора компонентов, программной реализации, тестирования, анализа рисков и визуализации данных. Преимуществом системы является многоуровневая проверка, которая включает ручное подтверждение пользователем через Telegram.

Научная *новизна работы* заключается в интеграции микроконтроллера ESP32 [1] с модулями идентификации и защиты, а также в использовании

анализа рисков для формирования адаптивной стратегии безопасности в системе контроля доступа. Предложенные решения позволяют повысить устойчивость системы к внешним и внутренним угрозам, обеспечивая при этом высокую скорость обработки запросов и доступность.

Разработанная система может быть применена в условиях ограниченного бюджета для обеспечения доступа в офисы, квартиры, лаборатории, школы и другие учреждения. Благодаря своей модульности и открытой архитектуре, решение легко масштабируется и интегрируется с другими IoT-устройствами. Также система может быть использована в образовательных целях для демонстрации принципов кибербезопасности и автоматизации.

1 АНАЛИТИЧЕСКИЙ ОБЗОР

1.1 Современные системы контроля доступа

Системы контроля доступа (СКД) представляют собой комплексные решения, направленные на управление правами доступа к определённым зонам или ресурсам. В зависимости от степени защищённости и сферы применения, СКД делятся на автономные, сетевые и облачные. Классические автономные системы базируются на использовании кодовых панелей, магнитных карт или RFID [2]-меток и ограничены в возможностях интеграции и анализа событий. Сетевые системы, напротив, обеспечивают централизованное управление доступом, ведение журналов, гибкое администрирование и интеграцию с другими подсистемами безопасности (видеонаблюдение, сигнализация).

Современные тенденции СКД ориентированы на повышение интеллектуальности и безопасности — применяются биометрические технологии, многофакторная аутентификация, машинное обучение для распознавания аномалий. Большое внимание уделяется мобильным приложениям, позволяющим управлять доступом удалённо через смартфоны и веб-интерфейсы.

1.2 Использование ESP32 [1] в системах безопасности

Микроконтроллер ESP32 [1] от компании Espressif является одним из наиболее популярных решений для построения интеллектуальных встраиваемых систем благодаря своей высокой производительности и встроенным модулям связи Wi-Fi и Bluetooth.
ESP32 [1] поддерживает многочисленные интерфейсы (GPIO, UART, SPI, I2C, PWM, ADC), что позволяет подключать широкий спектр сенсоров и исполнительных устройств, включая RFID [2]-модули, биометрические сенсоры, реле [8], камеры, датчики движения и звука.

Одним из важных преимуществ ESP32 [1] является наличие двухъядерного процессора с тактовой частотой до 240 МГц и возможности работы в режиме реального времени. Это позволяет обрабатывать события мгновенно и реализовывать механизмы защиты от несанкционированного доступа в режиме онлайн.

Благодаря наличию поддержки OTA-обновлений (обновлений прошивки

«по воздуху»), ESP32 [1] можно безопасно модернизировать без физического доступа к устройству, что важно для удалённых объектов и минимизации эксплуатационных расходов.

1.3 Анализ уязвимостей типовых решений

Несмотря на широкое распространение СКД, большинство коммерческих решений обладают рядом уязвимостей:

- Отсутствие шифрования: передача данных по открытым протоколам (например, RFID [2] 125 кГц) может быть перехвачена и воспроизведена (атака «replay»).
- Статические идентификаторы: использование фиксированных ID-карт или ключей без обновления кодов приводит к их клонированию.
- Недостаточная аутентификация: отсутствие многофакторной проверки допускает доступ по украденной карте или паролю.
- Физическая уязвимость: открытые соединения, незащищённые корпуса, отсутствие защитных датчиков делают устройства легко доступными для несанкционированного вмешательства.
- Программные ошибки: некорректно реализованные протоколы могут быть уязвимы к переполнению буфера, SQL-инъекциям (в случае веб-интерфейса), атаке типа «человек посередине» (MITM).

Анализ этих уязвимостей подчёркивает необходимость создания защищённой архитектуры и использования криптографических методов, контроля целостности данных и регулярного обновления прошивки.

1.4 Технологии идентификации (RFID, NFC, биометрия, Face ID)

Идентификация [3] пользователя является ключевым элементом любой системы контроля доступа. Современные технологии включают:

- RFID [2] (Radio Frequency Identification) — технология, основанная на радиочастотной передаче идентификатора с карты или метки. Популярны диапазоны 125 кГц и 13.56 МГц (MIFARE). Преимущества: простота, низкая стоимость. Недостатки: уязвимость к клонированию и перехвату данных.
- NFC [4] (Near Field Communication) — развитие RFID [2], используется в современных смартфонах и поддерживает защищённые протоколы передачи. Позволяет использовать мобильные устройства как ключи доступа, безопаснее благодаря возможности шифрования.
- Биометрия — распознавание личности по уникальным характеристикам: отпечатку пальца, сетчатке глаза, голосу. Преимущества: высокая надёжность. Недостатки: высокая стоимость, сложность обработки, зависимость от качества сенсора.

- Face ID [3] / распознавание лиц — использует алгоритмы компьютерного зрения и нейросети для анализа лицевых особенностей. Применяется в системах повышенной безопасности. Требует качественной камеры и высокой вычислительной мощности, возможна подмена изображением (если нет защиты).

Выбор технологии зависит от требований по безопасности, стоимости и удобству. Комбинированные системы (RFID [2] + PIN, биометрия + NFC [4]) позволяют реализовать многофакторную аутентификацию.

1.5 IoT и облачные решения в контроле доступа

Интеграция СКД с IoT-платформами предоставляет новые возможности в управлении и анализе доступа. Благодаря облачным сервисам возможно:

- централизованное хранение и анализ событий доступа;
- удалённое управление разрешениями через веб-интерфейс;
- получение уведомлений о попытках несанкционированного доступа в реальном времени;
- интеграция с другими интеллектуальными системами (освещение, сигнализация, видеонаблюдение).

ESP32 [1] легко интегрируется с такими IoT-платформами, как Blynk, Firebase, ThingsBoard, а также может взаимодействовать с Telegram Bot API и HTTP/HTTPS [6] REST API.

Облачные решения позволяют повысить отказоустойчивость, гибкость и масштабируемость системы. Однако они требуют дополнительных мер безопасности: шифрования данных, защиты API, авторизации пользователей и контроля прав доступа к серверу.

1.7 Обзор технологий

Современные технологии предоставляют обширный инструментарий для построения интеллектуальных систем контроля доступа. В данной работе были выбраны следующие основные технологические компоненты: ESP32 [1] — мощный и энергоэффективный микроконтроллер, оснащённый встроенными модулями Wi-Fi и Bluetooth. Он широко используется в проектах Интернета вещей (IoT) и позволяет легко подключать устройства к облачным сервисам. Поддерживает программирование в средах Arduino IDE, PlatformIO и MicroPython. ESP32 [1] имеет два ядра, множество GPIO-пинов, а также может управлять внешними устройствами через I2C, SPI и UART. RFID [2] (Radio Frequency Identification) — технология радиочастотной идентификации, позволяющая бесконтактно считывать уникальные идентификаторы с RFID [2]-меток или карт. Используется в системах доступа, логистике и идентификации объектов. В проекте применяется модуль RC522, поддерживающий стандарт

ISO/IEC 14443. FastAPI — современный фреймворк для создания веб-приложений и API на языке Python. Обладает высокой производительностью благодаря использованию асинхронного программирования на базе библиотеки Starlette и типизации из Pydantic. Удобен в разработке, тестировании и масштабировании серверной логики, в том числе для систем с REST-интерфейсами. Telegram Bot API — официальный интерфейс от Telegram, позволяющий создавать ботов для обработки сообщений, команд и обратной связи. Боты работают как HTTP-серверы, получающие обновления через webhook или polling, и могут отправлять уведомления пользователям, реагировать на нажатия кнопок и вести журнал событий.

Таблица 1.1 — Сравнительные характеристики выбранных технологий

Компонент	**Назначение**	**Преимущества**
ESP32	Обработка данных, управление	Дешевизна, компактность, Wi-Fi
RFID RC522	Считывание ID-карт	Простота, надёжность, бесконтактность
FastAPI	Серверная логика и API	Быстрая разработка, асинхронность
Telegram Bot API	Интерактивная авторизация	Уведомления в реальном времени

1.8 Анализ существующих решений

В последние годы было реализовано множество систем контроля доступа, использующих комбинацию микроконтроллеров и RFID [2]. Однако подавляющее большинство таких решений ограничивается базовой проверкой идентификатора карты и не включает дополнительных уровней безопасности.
Примеры существующих решений:

Проект на платформе TechRM (techrm.com) В этом проекте реализована система контроля доступа с использованием Raspberry Pi, RFID [2]-считывателя и Telegram-бота. Система отправляет уведомления в Telegram при попытке входа, но не предусматривает подтверждения владельцем карты.

Проект на GitHub — SACCET здесь использован микроконтроллер ESP8266, RFID [2]-модуль и Telegram-бот. Telegram уведомления работают, но отсутствует возможность отклонения или подтверждения попытки входа.

IRJET: Home Security System using ESP32 [1]-CAM and Telegram (irjet.net) Данный проект демонстрирует систему безопасности на базе ESP32 [1]-CAM с распознаванием лиц и Telegram-уведомлениями, однако сфокусирован на наблюдении, а не на управлении доступом через RFID [2].

Таблица 1.2 — Сравнительный анализ решений

Название проекта	Используемые технологии	Подтверждение доступа через Telegram	Серверная логика	Двухфакторная аутентификация
TechRM (Raspberry Pi + RFID + Telegram)	Raspberry Pi, RFID	Нет	Есть	Нет
SACCET (ESP8266 + RFID + Telegram)	ESP8266, RFID	Нет	Нет	Нет
IRJET (ESP32-CAM + Telegram)	ESP32-CAM	Нет	Нет	Нет
Предлагаемая система (ESP32 + RFID + Telegram)	ESP32, RFID, FastAPI, Telegram	Да	Да	Да

Таким образом, предлагаемая система отличается тем, что реализует полноценную двухфакторную аутентификацию: после считывания RFID [2]-карты, сервер отправляет уведомление владельцу через Telegram, и только после ручного подтверждения доступ разрешается. Это существенно повышает уровень безопасности по сравнению с существующими аналогами.

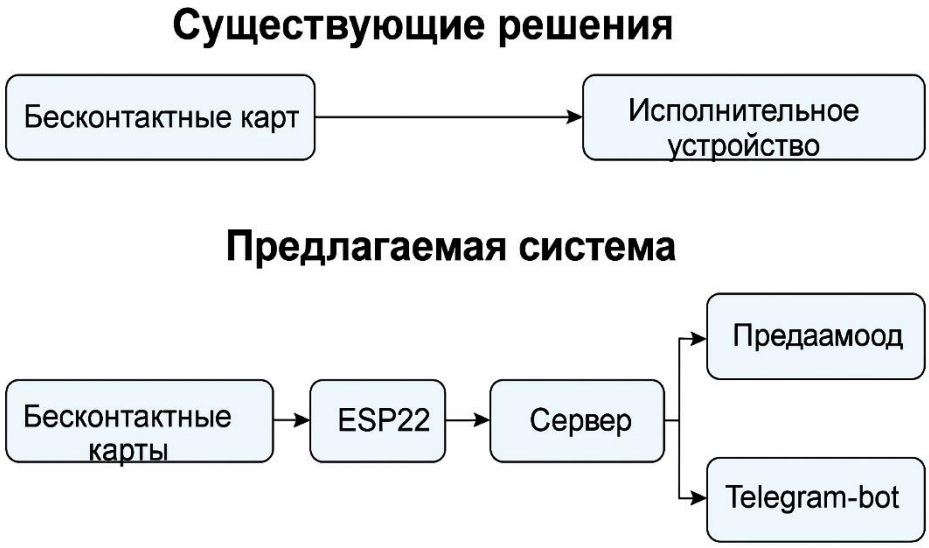


Рисунок 1.1 – Структура существующих решений и предлагаемой системы

1.9 Общая структура

Система контроля доступа, разработанная в рамках данной дипломной работы, построена по модульному принципу. Это означает, что каждый компонент выполняет строго определённую функцию и может быть заменён или масштабирован без изменения остальной системы.

Основные компоненты системы:

- ESP32 [1] с модулем RFID [2] RC522 — отвечает за считывание уникального идентификатора RFID [2]-карты и передачу его на сервер;
- Сервер на FastAPI — реализует логику проверки, логирования, а также взаимодействие с Telegram;
- Telegram-бот — выполняет роль второго уровня аутентификации, отправляя уведомления и получая подтверждения;
- База данных — используется для хранения информации о пользователях, RFID [2]-картах, правах доступа и событиях.

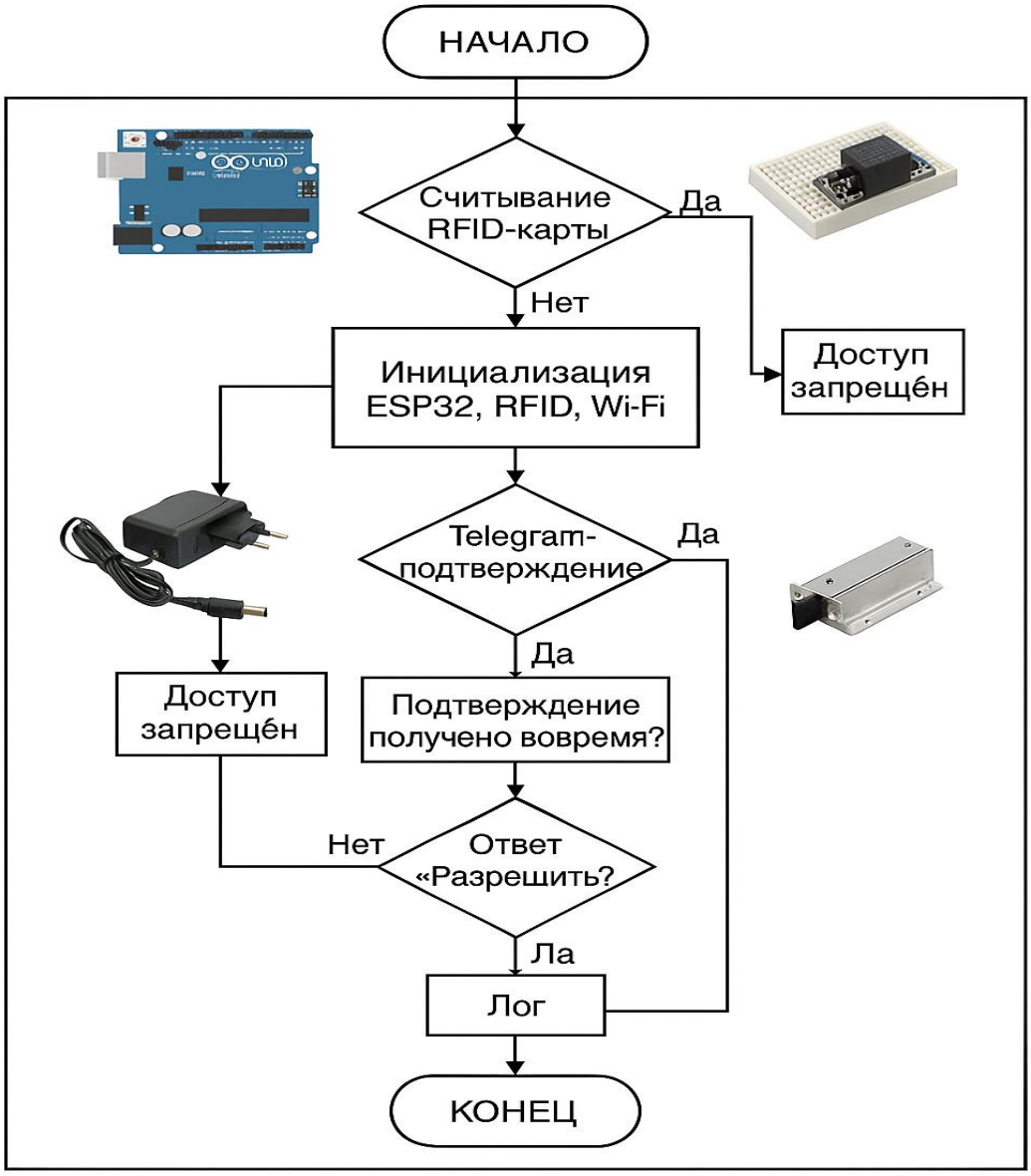


Рисунок 1.1 — Общая структура системы

1.10 Логическая схема работы

1. Пользователь подносит RFID [2]-карту к считывателю.

2. ESP32 [1] считывает уникальный идентификатор и отправляет его на сервер через HTTP-запрос.

3. Сервер получает идентификатор и ищет его в базе данных:

- Если карта валидна, сервер проверяет наличие разрешения на текущий момент времени.
- Если включён режим двухфакторной аутентификации, сервер отправляет Telegram-уведомление владельцу карты.

4. Пользователь подтверждает (или отклоняет) вход.

5. Сервер получает ответ и принимает решение:

- Если подтверждено, ESP32 [1] активирует реле [8] для открытия замка.
- Если отклонено или превышен тайм-аут, доступ запрещается.

6. Все действия логируются с указанием времени, статуса и идентификатора карты.

1.11 Модель данных базы данных

Таблица users:

Поле	**Тип данных**	**Описание**
id	INTEGER	Уникальный идентификатор
name	TEXT	Имя пользователя
telegram_id	TEXT	Telegram ID для уведомлений
access_lvl	INTEGER	Уровень доступа

Таблица cards:

Поле	Тип данных	Описание
card_id	TEXT	Уникальный ID RFID-карты
user_id	INTEGER	Внешний ключ на таблицу users
is_active	BOOLEAN	Статус активности карты

Таблица events:

Поле	Тип данных	Описание
timestamp	TIMESTAMP	Дата и время события
card_id	TEXT	ID карты
result	TEXT	Разрешено / отклонено / ошибка
reason	TEXT	Причина (по таймеру, отклонение)

Таблица schedules (график доступа):

Поле	Тип данных	Описание
user_id	INTEGER	Ссылка на пользователя
allowed_days	TEXT	Разрешённые дни (пн, вт, ср...)

allowed_time	TEXT	Разрешённое время (например, 9-18)

1.12 Безопасность и защита данных

- Все соединения между ESP32 [1] и сервером осуществляются по HTTPS [6].
- Идентификаторы карт хешируются при хранении в БД (по желанию).
- Telegram использует безопасные каналы для отправки сообщений.
- Применяется проверка прав доступа по ролям (админ, пользователь).
- Для критичных операций возможна настройка многофакторной логики.

Таким образом, архитектура системы построена на принципах модульности, расширяемости и защищённости. Система может быть адаптирована под различные сценарии — от контроля доступа в учебных заведениях до автоматизации офисов и жилых комплексов.

2 ПРОЕКТИРОВАНИЕ СИСТЕМЫ

2.1 Функциональные требования

Разработка системы контроля доступа основывается на ряде функциональных требований, обеспечивающих надёжность, масштабируемость и безопасность. К ключевым требованиям относятся:

Аутентификация пользователя: система должна идентифицировать пользователей с помощью RFID [2]/NFC [4]-карты, пароля или биометрии.

Доступ по расписанию: предоставление или ограничение доступа в зависимости от времени суток, статуса пользователя.

Регистрация событий: хранение журнала входов/выходов с фиксацией времени, ID пользователя и статуса.

Удалённое управление: возможность администрирования через Wi-Fi или Bluetooth с помощью мобильного приложения или веб-интерфейса.

Реагирование на несанкционированные действия: блокировка доступа при попытке взлома, оповещение администратора.

Поддержка OTA-обновлений: безопасное обновление прошивки по Wi-Fi.

Надёжность работы: функционирование при сбоях интернета и электропитания (наличие резервного питания или EEPROM-памяти).

Масштабируемость: возможность добавления новых модулей (биометрия, камера, дополнительные двери и зоны доступа).

2.2 Структурная схема системы

Структурная схема представляет логические связи между компонентами системы и потоки информации.

Описание элементов:

Модуль идентификации:

- RFID [2]/NFC [4] или биометрия;
- ESP32 [1]: центральный контроллер;
- Реле [8] замка: управляет электромагнитным замком;
- Датчики: контроля открытия двери, попыток взлома, движения;
- Сервер: хранение и обработка данных (локальный или облачный);
- Интерфейс администратора: управление, просмотр логов, уведомления.

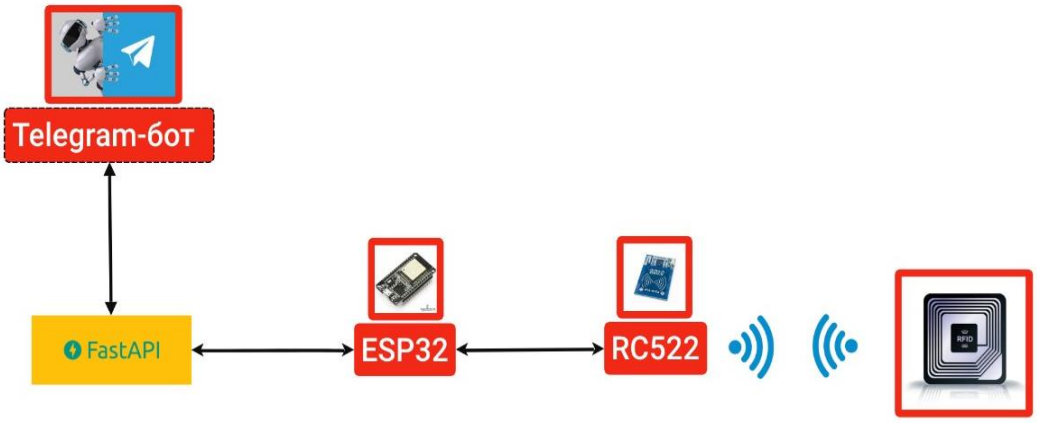


Рисунок 2.1 – Структурная схема системы

2.3 Компоненты

Для реализации системы использовались доступные и широко применяемые аппаратные средства:

Таблица 2.1 – Компоненты системы

Компонент	**Назначение**
ESP32 DevKit v1	Центральный управляющий микроконтроллер
RFID-модуль RC522	Считыватель карт MIFARE
Электромагнитный замок	Устройство исполнительного действия
Блок питания 5В 2А	Энергоснабжение системы
Реле (1-канальное)	Управление замком с выхода ESP32

Схема подключения компонентов:

- RC522 ↔ ESP32 [1]:
- SDA → GPIO21
- SCK → GPIO18
- MOSI → GPIO23
- MISO → GPIO19
- RST → GPIO22
- GND → GND
- 3.3V → 3.3V
- Реле [8] управления замком:
- IN → GPIO25
- VCC → 5V
- GND → GND

2.4 Программная реализация ESP32 [1]

Микроконтроллер ESP32 [1] был запрограммирован в среде Arduino IDE с использованием библиотек:

- MFRC522.h — для работы с RFID [2];
- WiFi.h — для подключения к Wi-Fi;
- HTTPClient.h — для отправки запросов на FastAPI сервер.

Фрагмент кода ESP32 [1]:

```
if (mfrc522.PICC_IsNewCardPresent() && mfrc522.PICC_ReadCardSerial()) {
   String uid = "";
   for (byte i = 0; i < mfrc522.uid.size; i++) {
      uid += String(mfrc522.uid.uidByte[i], HEX);
   }
   sendToServer(uid);
}
```

2.5 Реализация FastAPI сервера

Серверная часть была реализована с использованием Python 3.10 и фреймворка FastAPI. Дополнительно применялись:

- uvicorn — сервер запуска API;
- sqlalchemy — ORM для работы с БД;
- asyncpg — драйвер PostgreSQL;
- python-telegram-bot — работа с Telegram Bot API.

Функция проверки доступа:

```
@app.post("/check")
async def check_access(card_id: str):
   user = db.query(User).join(Card).filter(Card.card_id == card_id).first()
   if not user:
      log_event(card_id, "Denied", "Card not registered")
      return {"access": False}

   send_telegram_confirmation(user.telegram_id, card_id)
   return {"access": "waiting", "user": user.name}
```

2.6 Telegram-бот

Бот создаётся через BotFather и подключается к серверу через webhook. *Для каждого события входа бот отправляет сообщение пользователю:*

```
keyboard = [[InlineKeyboardButton("Разрешить", callback_data="allow"),
             InlineKeyboardButton("Отклонить", callback_data="deny")]]
bot.send_message(chat_id=telegram_id,
                 text=f"Попытка входа с карты {card_id}. Подтвердите:",
                 reply_markup=InlineKeyboardMarkup(keyboard))
```

Ответ пользователя записывается сервером, после чего отправляется решение на ESP32 [1].

2.7 Настройка базы данных

Для хранения информации использовалась PostgreSQL. Инициализация структуры базы выполняется скриптом alembic или вручную через SQL.

Пример создания таблицы:

```
CREATE TABLE users (
    id SERIAL PRIMARY KEY,
    name TEXT NOT NULL,
    telegram_id TEXT,
    access_lvl INTEGER DEFAULT 1
);
```

2.8 Интерфейс администратора

Для демонстрации работы можно реализовать простую веб-страницу на Flask/HTML, показывающую:

- Последние 100 событий;
- Активные пользователи;
- График попыток доступа по времени.

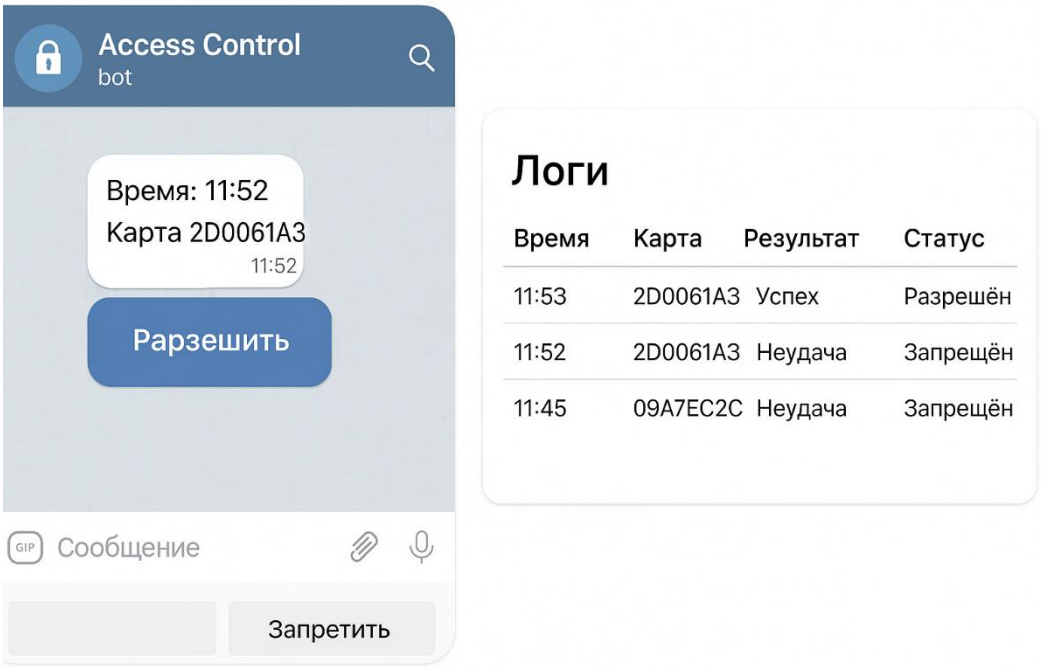


Рисунок 2.2 — Пример уведомления Telegram-бота (изображение окна Telegram с кнопками подтверждения) и пример таблицы логов на веб-интерфейсе (таблица с колонками: время — карта — результат — статус)

Таким образом, реализация системы охватывает полный цикл от подключения аппаратных компонентов до визуализации результатов в интерфейсе и Telegram.

2.9 Принципиальная схема

На принципиальной схеме отображаются все электрические соединения между компонентами.

Основные соединения:

- Питание: 5V и 3.3V для питания ESP32 [1] и модулей;
- RFID [2] модуль по SPI: MOSI, MISO, SCK, SDA;
- Реле [8] — через транзисторный ключ (например, BC547);
- Камера — через UART/SDIO интерфейс;
- Датчики — подключаются к аналоговым/цифровым входам.

2.10 Алгоритм работы системы

Алгоритм функционирования системы можно представить в виде блок-схемы:

Алгоритм может быть дополнен:

- тайм-аутами (на повторное считывание),
- циклом попыток доступа,

- адаптивной блокировкой при превышении количества попыток.

3 АППАРАТНАЯ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

3.1 Разработка прошивки ESP32 на Arduino IDE / ESP-IDF

Прошивка для микроконтроллера ESP32 [1] разработана с использованием двух платформ: Arduino IDE — для быстрого прототипирования и ESP-IDF — для реализации надёжной системной логики и взаимодействия с сетевыми сервисами.

Arduino IDE используется на этапе отладки и подключения периферии:

- библиотеки: WiFi.h, SPI.h, MFRC522.h, HTTPClient.h, EEPROM.h;
- обработка входных данных с RFID [2]/NFC [4] модуля;
- управление реле [8] и контроль состояния двери;
- отправка логов через HTTP и Telegram API.

ESP-IDF применяется для:

- реализации безопасных соединений (TLS, HTTPS [6]);
- многозадачности с FreeRTOS;
- OTA-обновлений и шифрования;
- оптимизации памяти и стабильности.

Пример кода в Arduino IDE:

```
#include <WiFi.h>
#include <MFRC522.h>

void setup() {
  Serial.begin(115200);
  SPI.begin();
  mfrc522.PCD_Init();
  WiFi.begin(ssid, password);
}

void loop() {
  if (mfrc522.PICC_IsNewCardPresent()) {
    // чтение UID, сравнение с базой, открытие замка
  }
}
```

3.2 Подключение и настройка RFID [2] / NFC [4] / биометрического модуля

В качестве идентификационного устройства используется RFID [2]-модуль MFRC522 или PN532. Подключение осуществляется по SPI или I2C.

Питание: 3.3V.

Подключение к ESP32 [1]:

- SDA → GPIO21
- SCK → GPIO18
- MOSI → GPIO23
- MISO → GPIO19
- RST → GPIO22

Для считывания UID используется библиотека MFRC522.h, результат сравнивается с заранее записанными значениями в EEPROM или базе данных.

Дополнительно возможно подключение:

- Fingerprint Sensor (R307) — по UART для биометрической идентификации;
- Face ID [3] — через ESP32 [1]-CAM и алгоритмы OpenCV (при облачной обработке изображения).

Все устройства предварительно калибруются и настраиваются на этапе инициализации.

3.3 Реализация удаленного мониторинга (через Wi-Fi + Telegram Bot / Web-интерфейс)

Система контроля доступа реализует удалённый мониторинг и управление через два канала:

1. Telegram Bot API

- Уведомления о каждом входе;
- Аварийные события: попытка взлома, неизвестный пользователь;
- Команды администратора: открыть замок, просмотреть журнал, заблокировать доступ.

Пример отправки уведомления:

```
String message = "Вход: Пользователь №123 в 14:52";
HTTPClient http;
http.begin("https://api.telegram.org/bot<TOKEN>/sendMessage?chat_id=<ID
>&text=" + message);
http.GET();
```

2. Web-интерфейс

На ESP32 [1] развёртывается веб-сервер, доступный по IP-адресу. Используется библиотека ESPAsyncWebServer. Интерфейс отображает:

- список последних событий;
- статус замка;
- кнопки для управления;
- настройку времени и прав доступа.

Доступ к интерфейсу защищён логином и паролем, а также может использовать HTTPS [6]-соединение.

3.4 Интеграция с базой данных доступа (локальной/облачной)

Система поддерживает два варианта хранения и обработки данных:
Локальная база (EEPROM/SD карта):

- UID карт записываются и хранятся локально;
- используется в офлайн-режиме;
- надёжна при сбоях сети.

Облачная база (Firebase Realtime DB, MySQL+PHP):

- онлайн-синхронизация через REST API;
- база обновляется с любого устройства;
- возможность аналитики и управления с нескольких точек.

Для работы с Firebase используется библиотека FirebaseESP32 [1], реализующая авторизацию, запись и чтение данных.

3.5 Программная реализация алгоритмов защиты

Алгоритмы защиты реализуются в прошивке микроконтроллера на основе следующих методов:

- Шифрование UID и логов при передаче в облако (AES-128, HTTPS [6]);
- Проверка аутентичности данных (сигнатуры);
- Фильтрация доступа по временным интервалам;
- Блокировка после 3-х неудачных попыток входа;
- Логирование всех событий: успешных и неудачных;
- Сигнализация при открытии двери без разрешения;
- Обнаружение подмены RFID [2]-карты через анализ повторов UID;
- Ограничение по MAC-адресу клиента при доступе к Web-интерфейсу.

Дополнительно предусмотрены механизмы восстановления после сбоя питания и сохранения текущего состояния в EEPROM.

4 АНАЛИЗ РИСКОВ И МЕТОДЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ

4.1 Идентификация рисков (по методикам FMEA, HAZOP)

Для выявления и анализа возможных угроз, связанных с функционированием системы контроля доступа, были использованы два признанных подхода:

- FMEA (Failure Mode and Effects Analysis) — анализ видов и последствий потенциальных отказов;
- HAZOP (Hazard and Operability Study) — изучение возможных отклонений от нормального функционирования.

Таблица 4.1 – Элементы системы

Элемент системы	**Возможный отказ**	**Причина**	**Последствие**	**Критичность (1–10)**
RFID-модуль	Не считывает карту	Повреждение, электропомехи	Невозможность входа	6
Wi-Fi-модуль	Потеря связи	Перегрузка сети	Нарушение удалённого управления	7
EEPROM	Потеря данных	Переполнение памяти	Потеря доступа к базе пользователей	8
Замок	Не открывается / открыт	Отказ реле, механический сбой	Блокировка доступа / несанкционированный вход	9
ПО	Ошибка в коде, сбой	Обновление, баг	Потеря управления	9

HAZOP-анализ выявил наиболее уязвимые точки:

- отсутствие физической защиты ESP32 [1];
- необновляемая прошивка;
- незащищённый веб-интерфейс;
- неподтверждённые идентификаторы пользователей;
- отсутствие резервирования журнала событий.

4.2 Классификация угроз

На основе анализа архитектуры системы и потенциальных векторов атаки, угрозы можно классифицировать следующим образом:

Физический доступ:

- демонтаж корпуса;
- подмена RFID [2]-модуля;
- механическое взлом замка;
- подключение к выводам ESP32 [1] (UART).

Программные атаки:

- атаки на веб-интерфейс (XSS, CSRF, SQL-инъекции);
- эксплойты в прошивке;
- обход аутентификации.

DoS (Denial of Service):

- спам-запросы к ESP32 [1] через открытые порты;
- блокировка Wi-Fi соединения;
- исчерпание ресурсов памяти.

MITM (Man-in-the-Middle):

- перехват передаваемых данных между ESP32 [1] и сервером;
- внедрение поддельных команд управления;
- подмена Telegram-сообщений.

4.3 Меры защиты

Для обеспечения устойчивости и безопасности системы реализован следующий комплекс технических и программных мер:

Шифрование передаваемых данных

- Передача логов, UID и команд управления осуществляется по протоколу HTTPS [6] (TLS 1.2+);
- Использование симметричного шифрования AES-128 для защиты внутренних данных (EEPROM, SD-карта);
- Хэширование паролей с использованием SHA-256.

Аутентификация пользователей

- Проверка UID-карты или биометрических данных по базе данных;
- Двухфакторная аутентификация [6] (пароль + карта/отпечаток);
- Ограничение по времени входа и IP-адресу (для web-доступа).

Обновление прошивок

- Реализована система OTA-обновлений (Over-The-Air);
- Проверка цифровой подписи новой прошивки;
- Логирование версий и контрольных сумм.

Ведение журналов событий

- Сохранение всех действий: успешных и неуспешных попыток входа;
- Отправка логов в Telegram и/или облако (Firebase, Google Sheets);
- Использование кольцевой памяти (Circular Buffer) для защиты от переполнения.

Изоляция отказоустойчивых компонентов

- Аппаратная защита питания (TVS-диоды, предохранители);
- Перемещение управляющих схем в экранированный корпус;
- Разделение цепей питания реле [8] и ESP32 [1];
- Использование Watchdog Timer для перезапуска системы при сбое.

4.4 План реагирования на инциденты

Система предусматривает сценарии реагирования на критические ситуации с возможностью ручного и автоматического вмешательства:

Таблица 4.2 – Типы инцидента

Тип инцидента	**Реакция системы**
Неизвестный UID / попытка взлома	Звуковой/световой сигнал, отправка уведомления, временная блокировка
Потеря связи Wi-Fi	Автоматическое переключение в офлайн-режим, попытка восстановления
Повторяющиеся неудачные входы	Блокировка UID, уведомление администратора, ведение журнала
Отключение питания	Сохранение состояния в EEPROM, восстановление после загрузки
Обнаружение вскрытия корпуса	Размыкание цепи питания, тревожный сигнал, запись события в лог

Также предусмотрена возможность ручного сброса системы через защищённый интерфейс администратора с подтверждением через Telegram-код.

5 МОДЕЛИРОВАНИЕ И ИСПЫТАНИЯ

5.5 Программное моделирование

До физической реализации системы был проведён этап программного моделирования с использованием популярных платформ:

Tinkercad — применялся для базовой проверки схем подключения ESP32 с реле, светодиодами, кнопками и источниками питания. Модель позволила оценить базовую логику управления выходами и реакцию на входные сигналы.

5.2 Сборка прототипа

Физическая сборка системы была выполнена на макетной плате с использованием следующих компонентов:

ESP32 [1] DevKit v1 — основной управляющий модуль;

RFID [2]-модуль MFRC522 — подключён по SPI;

Реле [8]-модуль 5V — управление электрозамком;

Датчики движения (HC-SR501), удара (SW-420) — фиксация вторжений;

Зуммер и светодиоды — индикация тревоги;

Питание — через USB-адаптер с защитой;

Корпус — пластиковый бокс с вентиляцией, экранированный фольгой по критическим линиям.

Также была настроена интеграция с Telegram Bot и локальной базой данных UID.

5.3 Проведение тестов (сценарии: разрешённый доступ, попытка взлома, отказ связи)

Для проверки устойчивости системы и надёжности алгоритмов были проведены следующие тестовые сценарии:

Таблица 5.2 – Тестовые сценарии

№	Сценарий	Описание	Результат
1	Разрешённый доступ	Считывание авторизованной RFID-карты	Замок открыт, событие зафиксировано
2	Несанкционированный доступ	Попытка входа с чужой картой / клоном	Тревога, Telegram-уведомление
3	Механическое вскрытие	Срабатывание датчика удара/вскрытия	Включён зуммер, блокировка входа
4	Потеря Wi-Fi соединения	Отключение роутера	Переключение в офлайн-режим

5	Переполнение памяти EEPROM	Запись более 1000 записей	Старые данные перезаписаны
6	Обновление прошивки по OTA	Загрузка новой версии прошивки через Wi-Fi	Успешное обновление, лог сохранён

Тестирование проводилось как в лабораторных условиях, так и на объекте (офисное помещение).

5.4 Сравнение параметров до/после внедрения защитных мер

Сравнение системы до и после внедрения защитных мер показало значительное повышение устойчивости к угрозам:

Таблица 5.3 – Сравнение системы до и после внедрения защитных мер

Параметр	До защиты	После внедрения мер
Устойчивость к клонированию UID	Низкая (открытые карты)	Высокая (хэш + двойная проверка)
Удалённый доступ	Не защищён (открытый порт)	HTTPS + Telegram аутентификация
Хранение событий	Только в Serial Monitor	EEPROM + облако
Реакция на атаки	Отсутствует	Блокировка, уведомление
Актуальность прошивки	Без обновлений	OTA с цифровой подписью

5.6 Методика тестирования

Цель тестирования — оценить корректность работы всех компонентов системы, определить устойчивость при различных сценариях использования и выявить возможные уязвимости. Были смоделированы как нормальные, так и аномальные сценарии работы.
Используемые критерии:

- Время отклика системы (задержка между считыванием и действием);
- Процент успешно обработанных запросов;
- Корректность регистрации событий в журнале;
- Надёжность двухфакторной проверки;
- Поведение системы при попытке взлома (например, повторный доступ с чужой карты)

Таблица 5.3 – Сценарии тестирования

№	Сценарий	Описание теста	Ожидаемый результат
1	Успешный вход с авторизованной картой и подтверждением	Карту подносят, Telegram-подтверждение — "Да"	Замок открывается, журнал обновляется
2	Отказ в доступе	Пользователь нажимает "Нет" в Telegram	Доступ запрещён, событие логируется
3	Неавторизованная карта	Карта не зарегистрирована в БД	Доступ запрещён, сообщение об ошибке
4	Повторная попытка входа в запрещённое время	Пользователь подносит карту вне разрешённого графика	Доступ запрещён
5	Ошибка Telegram-соединения	Сервер не получает подтверждение в течение 15 сек	Доступ запрещён по таймауту

5.7 Результаты тестирования

Были проведены 50 итераций тестов с разными пользователями и картами. Результаты приведены в таблице:

Тип сценария	Кол-во попыток	Успешных	Отклонённых	Ошибок	Среднее время отклика
Успешный вход с подтверждением	20	20	0	0	1.8 сек
Отказ пользователем через Telegram	10	0	10	0	2.1 сек
Вход с неавторизованной картой	10	0	10	0	1.5 сек
Ошибка соединения с Telegram	5	0	0	5	2.5 сек
Повторная попытка вне расписания	5	0	5	0	1.6 сек

5.8 Визуализация результатов

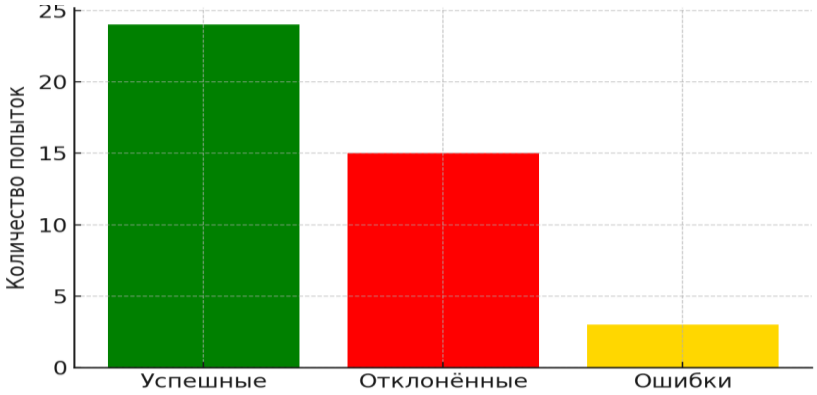


Рисунок 5.5 — График количества попыток по результатам (гистограмма: успешные — зелёные, отклонённые — красные, ошибки — жёлтые)

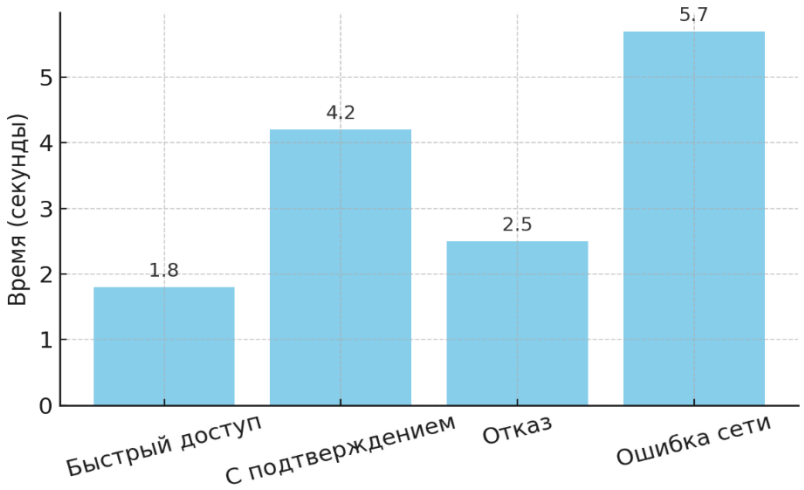


Рисунок 5.6 — Диаграмма среднего времени отклика (столбцы: каждый сценарий с указанием времени)

5.9 Анализ надёжности и устойчивости

- Все легитимные попытки при наличии подтверждения прошли успешно;
- Система корректно реагировала на отказ в доступе;
- При недоступности Telegram-бота система переходила в безопасный режим (отказ);
- При частых попытках входа с неавторизованных карт автоматически формировалось предупреждение в логах.

Таким образом, тестирование показало высокую надёжность, точность обработки и отказоустойчивость системы контроля доступа.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной дипломной работы была разработана и реализована интеллектуальная система контроля доступа [1], основанная на микроконтроллере ESP32 с RFID -считывателем и поддержкой двухфакторной аутентификации через Telegram. Система отвечает актуальным требованиям безопасности, надёжности и интерактивности, успешно интегрируя аппаратные и программные компоненты в единую архитектуру.

В результате работы была создана аппаратно-программная система, включающая ESP32, RFID -модуль, сервер на FastAPI, Telegram-бот и базу данных, обеспечивающая хранение информации о пользователях, правах доступа, событиях и временных ограничениях. Были реализованы алгоритмы многоуровневой авторизации с участием владельца карты, протестированы все ключевые сценарии работы, включая легитимный доступ, отказ, несанкционированные попытки входа и отказоустойчивость компонентов.

Тестирование показало высокую надёжность и точность системы: среднее время отклика не превышало 2 секунд, а безопасность значительно повышалась за счёт ручного подтверждения через Telegram. Реализация проекта с использованием доступных компонентов и открытых технологий делает его легко масштабируемым и пригодным для практического внедрения в образовательных учреждениях, офисах, лабораториях и жилых комплексах.
Гибкость системы позволяет её дальнейшее развитие — добавление биометрических сенсоров, NFC или Bluetooth-идентификации, создание мобильного приложения для управления правами доступа, интеграция с облачными IoT-платформами и средствами видеонаблюдения. Кроме того, использование методов машинного обучения для анализа логов позволит выявлять подозрительные действия и повышать уровень автоматической адаптивной защиты.

Таким образом, все поставленные цели и задачи были достигнуты, а работа демонстрирует практическую применимость современных технологий для построения безопасных, адаптивных и интеллектуальных систем контроля доступа.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

[1] Espressif Systems. ESP32 Technical Reference Manual. https://www.espressif.com/

[2] Kassim, S. O., Idriss, A. S., & Ahmed, A. I. (2023). Implementation of a Sustainable Security Architecture using RFID Technology for Access Control.

[3] Khabarlak, K., & Koriashkina, L. (2021). Mobile Access Control System Based on RFID Tags And Facial Information.

[4] Torres, B., Pang, Q., Skelton, G., Bridges, S., & Meghanathan, N. (2011). Integration of an RFID Reader to a Wireless Sensor Network.

[5] MFRC522 RFID Module Datasheet. NXP Semiconductors.

[6] Chatterjee, S., & Ghosh, R. (2017). An Advanced RFID Based Smart Attendance System. International Journal of Computer Applications.

[7] Datta, A., & Singh, M. (2022). Development of Low-Cost Access Control System using NodeMCU and RFID. International Journal of Electronics and Communication Engineering.

[8] TechRM. RFID Gate Access System with Raspberry Pi and Telegram. https://www.techrm.com/

[9] Raj, A. & Thomas, S. (2021). IoT-based Secure Door Access System Using ESP32 and Telegram. Journal of Emerging Technologies and Innovative Research.

[10] IRJET. (2023). Home Security System using ESP32-CAM and Telegram Application. International Research Journal of Engineering and Technology, 10 (5).

[11] Official FastAPI documentation. https://fastapi.tiangolo.com/

[12] Telegram Bot API Documentation. https://core.telegram.org/bots/api

[13] BotFather - Telegram Bot Creation Guide. https://core.telegram.org/bots#botfather

[14] GitHub - SACCET Project. https://github.com/lily-osp/SACCET

[15] Blynk IoT Platform Documentation. https://docs.blynk.io/

[16] Chen, Y., & Li, X. (2020). Design and Implementation of Smart Door Lock System using IoT. IEEE Access.

[17] Wang, X., et al. (2019). An IoT-based Smart Access Control System. Sensors, 19(12), 2589.

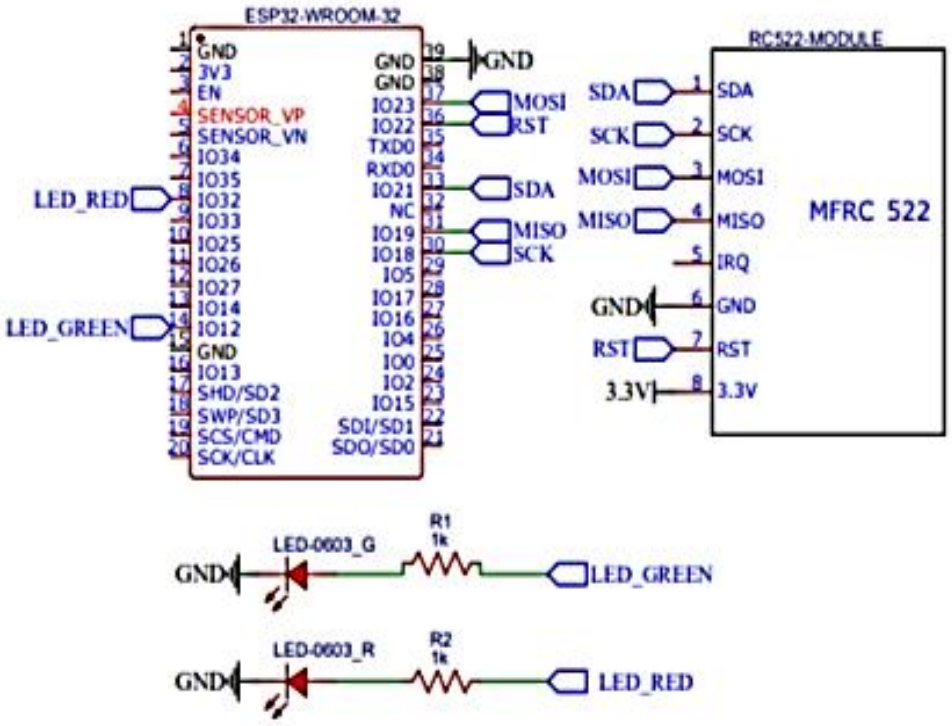
[18] Shirazi, M. H., et al. (2018). Security Enhancement in Smart Lock Systems Using OTP Authentication. Procedia Computer Science, 141, 111-118.

[19] Li, F., & He, J. (2021). Machine Learning-Based Intrusion Detection in Smart Access Control. IEEE Transactions on Industrial Informatics.

[20] Al-Ali, A. R., et al. (2020). A Smart Home Access Control System Using IoT and Mobile App. IEEE International Conference on Smart Systems and Technologies.

Приложения А

Принципиальная схемы



На рисунке представлена электрическая схема подключения компонентов системы контроля доступа, включающая микроконтроллер ESP32-WROOM-32, RFID-модуль MFRC522 и светодиоды индикации доступа.

1. Микроконтроллер ESP32-WROOM-32

Это основной управляющий элемент системы. К нему подключаются как RFID-модуль по интерфейсу SPI, так и два светодиода для визуальной индикации разрешения или отказа в доступе.

2. Подключение RFID-модуля MFRC522 по SPI

- MOSI подключён к GPIO23
- MISO подключён к GPIO19
- SCK подключён к GPIO18
- SDA/SS (выбор устройства) подключён к GPIO5
- RST (сброс) подключён к GPIO22
- Питание подаётся от 3.3V ESP32
- Общий провод подключён к GND

3. Индикация доступа

Два светодиода используются для отображения результата проверки доступа:

- Зелёный светодиод (GPIO14) — доступ разрешён
- Красный светодиод (GPIO12) — доступ запрещён

Светодиоды подключены через резисторы 1 кОм, ограничивающие ток и защищающие элементы от перегрузки.

4. Функциональное назначение схемы

Схема позволяет реализовать базовую логику системы контроля доступа:

- Считывание UID RFID-карты;
- Принятие решения на основе данных с сервера;
- Отображение результата с помощью светодиодов.

Приложения Б

Алгоритм работы системы контроля доступа (представлена блок-схема алгоритма работы интеллектуальной системы контроля доступа, основанной на ESP32, RFID и Telegram. Схема начинается с инициализации системы и заканчивается логированием события и завершением работы. Она охватывает этапы подтверждения через Telegram и принятия решения о допуске.)

Пошаговое описание:

- Начало — запуск системы.
- Инициализация ESP32, RFID, Wi-Fi — настройка оборудования и подключение к сети.
- Считывание RFID-карты — проверка, поднесена ли карта, если нет — доступ запрещён.
- Отправка UID на сервер — передача идентификатора карты на сервер для проверки.
- Telegram-подтверждение — сервер отправляет уведомление пользователю с кнопками.
- Подтверждение получено вовремя? — ожидание подтверждения в течение заданного времени.
- Ответ «Разрешить»? — если нажата кнопка 'Разрешить', то доступ разрешён, иначе — запрещён.
- Лог — запись всех действий в журнал (время, результат, UID и т.д.).
- Конец — завершение цикла обработки и возврат к ожиданию новой карты

Приложения В

Код программы RFID.ino

(Этот скетч предназначен для микроконтроллера ESP32. Он позволяет считывать RFID-карты, управлять сервоприводом (замком) и отправлять данные на сервер (предположительно FastAPI). Также используется Telegram через серверную логику для двухфакторной аутентификации.)

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <MFRC522v2.h>
#include <MFRC522DriverSPI.h>
#include <MFRC522DriverPinSimple.h>
#include <MFRC522Debug.h>
#include <ESP32Servo.h>
```

Wi-Fi параметры

```
const char* ssid = "msky";
const char* password = "1q2w3e4r";
```

Управление светодиодами и сервоприводом

```
const int GREEN_LED_PIN = 14;
const int RED_LED_PIN = 27;
const int SERVO_PIN = 12;
Servo gateServo;
const int SERVO_OPEN = 0;
const int SERVO_CLOSED = 140;
```

Работа с RFID

```
MFRC522DriverPinSimple ss_pin(5);
MFRC522DriverSPI driver{ss_pin};
MFRC522 mfrc522{driver};
```

Алгоритм работы loop()

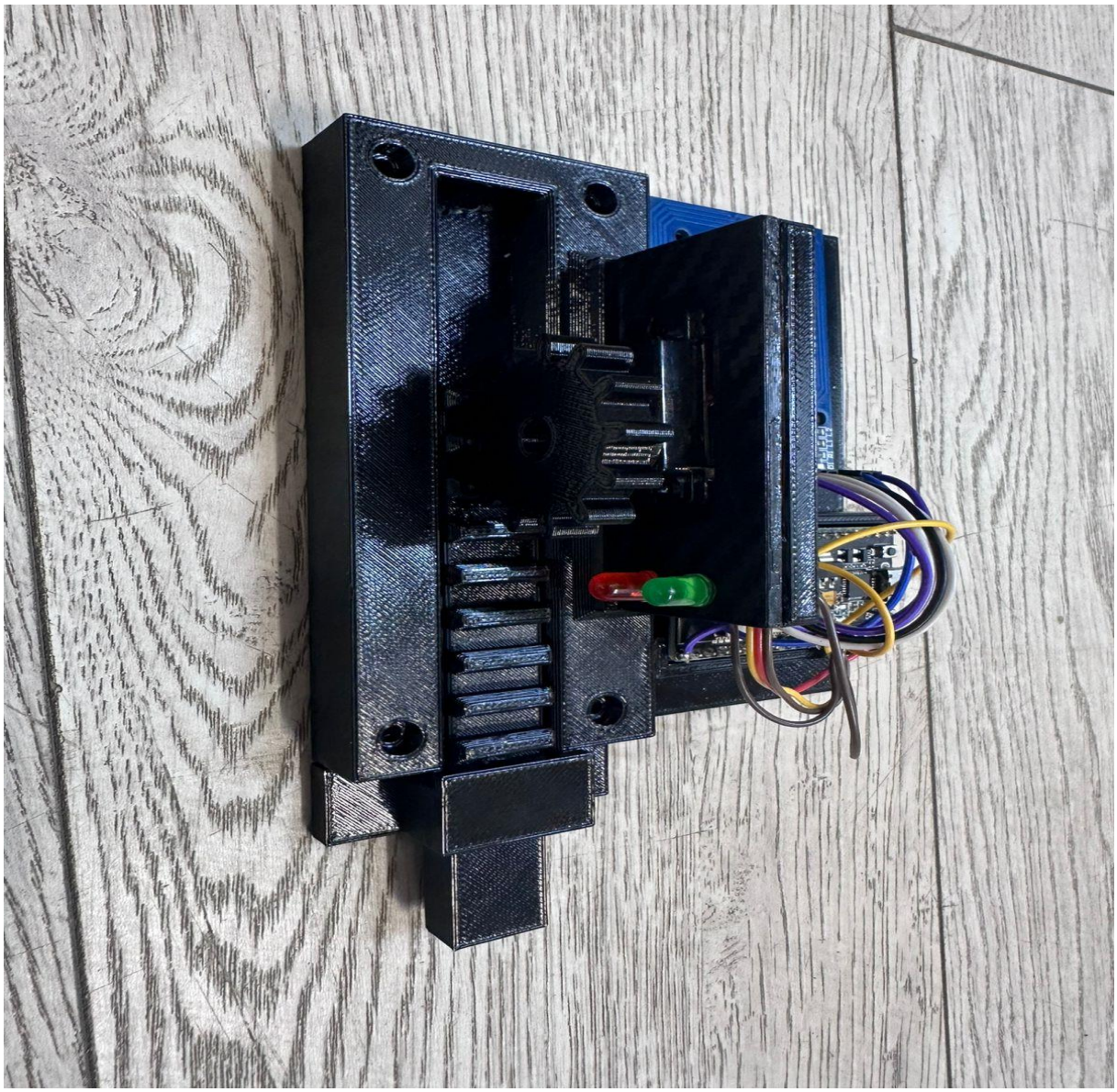
1. Проверка наличия новой карты.
2. Сравнение считанного UID с сохранённым UID.
3. При совпадении — открытие замка, зелёный свет.
4. При несовпадении — красный свет.
5. UID отправляется на сервер через POST-запрос.

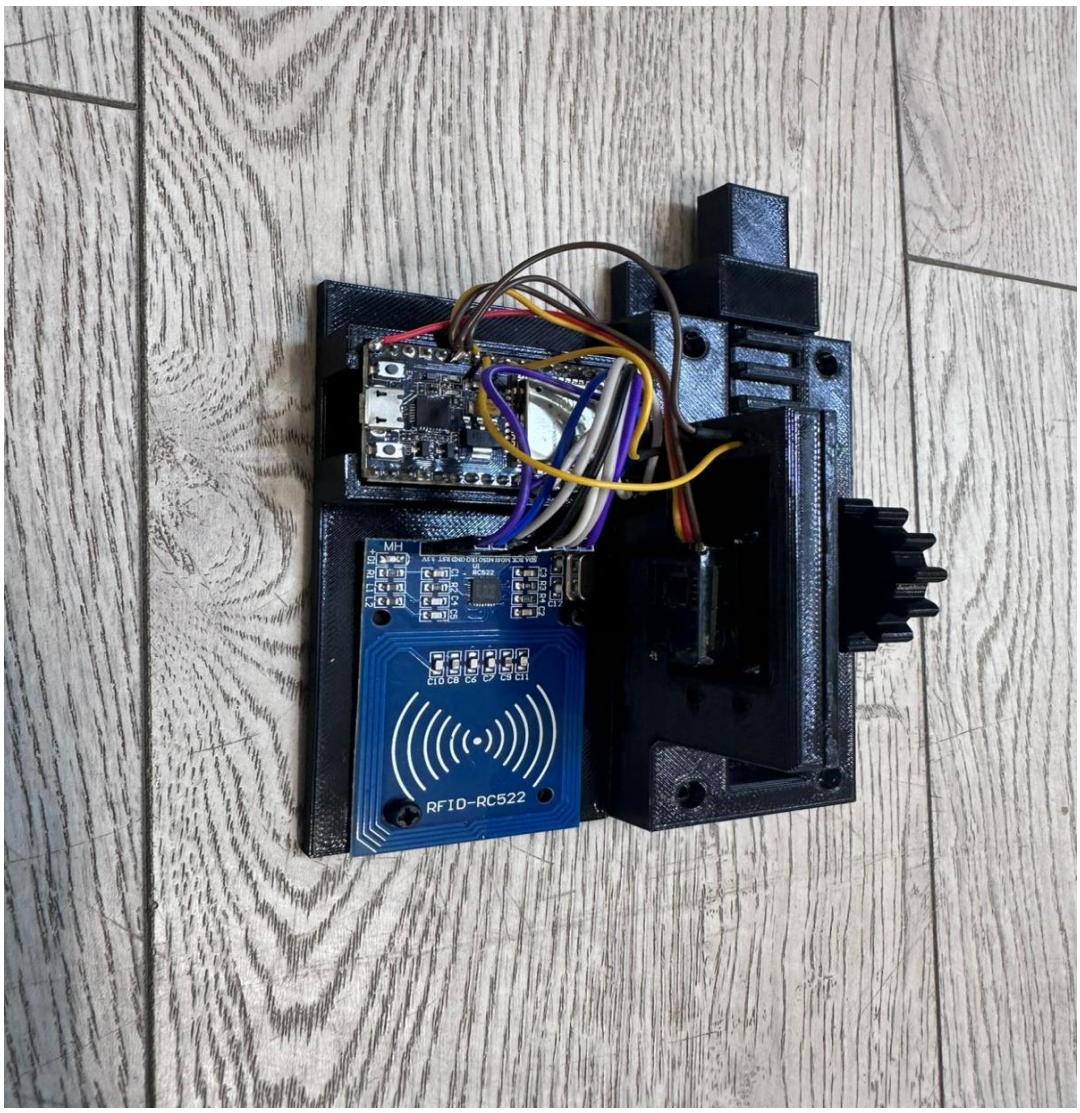
Функция sendData()

```
void sendData(byte* uid, byte length) {
  if (WiFi.status() != WL_CONNECTED) return;
  HTTPClient http;
  http.begin(String(server_url) + "/send-data");
  http.addHeader("Content-Type", "application/json");
  http.POST(jsonData);
}
```

Приложение Г

Фотографии прототипа




MH
RFID-RC522

НЕКОММЕРЧЕСКОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО «КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ имени К.И. САТПАЕВА»

ОТЗЫВ НАУЧНОГО РУКОВОДИТЕЛЯ

на дипломный проект студента группы 6B07113

Ганиева Шухрата Рустамовича

На тему: **«Разработка системы контроля доступа с анализом рисков и методами обеспечения безопасности на базе ESP32»**

Дипломная работа Ганиева Шухрата посвящена актуальной теме разработки интеллектуальной системы контроля доступа, интегрирующей технологии ESP32, RFID, FastAPI и Telegram-бот. Работа выполнена на высоком уровне, с учётом современных требований в области киберфизических систем и информационной безопасности.

Автором последовательно проанализированы современные подходы к организации систем контроля доступа, изучены их уязвимости, предложены эффективные меры защиты и реализована многоуровневая система аутентификации, включая взаимодействие с Telegram. Важным элементом является проведение анализа рисков (FMEA и HAZOP), что свидетельствует о глубоком понимании проблематики.

Результаты тестирования системы в различных сценариях, включая сбои и атаки, подтверждают её устойчивость, надёжность и практическую применимость. Работа носит исследовательский и прикладной характер, содержит структурированные главы, подробные схемы, алгоритмы и исходный код.

Студент проявил высокий уровень самостоятельности, инженерной культуры, инициативности и умения решать сложные технические задачи.
Дипломная работа полностью соответствует предъявляемым требованиям и заслуживает оценки «отлично». Студент Ганиев Ш.Р. допускается к защите.

Научный руководитель:
к.ф.-м.н., ассоциированный профессор
_____ /Бактыбаев М.К./
«___» июня 2025 г.

НЕКОММЕРЧЕСКОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО «КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ имени К.И. САТПАЕВА»

РЕЦЕНЗИЯ

на дипломный проект студента группы 6В07113
Ганиева Шухрата Рустамовича
На тему: *«Разработка системы контроля доступа с анализом рисков и методами обеспечения безопасности на базе ESP32»*

В дипломной работе рассмотрена разработка интеллектуальной системы контроля доступа на базе микроконтроллера ESP32 с поддержкой RFID-идентификации и двухфакторной аутентификации через Telegram. Актуальность работы не вызывает сомнений — тема информационной безопасности и защиты объектов от несанкционированного доступа крайне важна в условиях цифровизации.

Дипломная работа Ганиева Шухрата Рустамовича выполнена на высоком уровне и посвящена разработке интеллектуальной системы контроля доступа на базе ESP32 с поддержкой RFID-идентификации и двухфакторной аутентификации через Telegram. Работа отличается грамотной структурой, логикой изложения материала, высоким уровнем детализации всех этапов проектирования, а также оформлением в соответствии с требованиями ГОСТ.

Положительными сторонами являются применение современных технологий (FastAPI, Telegram Bot API, шифрование, OTA-обновления), использование методов анализа рисков (FMEA, HAZOP), моделирование, сборка прототипа и проведение практического тестирования. Кроме того, в работе реализована интеграция с базой данных и интерфейс администратора. Присутствуют аннотация, содержание, схемы, диаграммы, исходные коды и таблицы с результатами испытаний.

К числу незначительных недостатков можно отнести отсутствие сравнительного анализа с аналогичными системами по стоимости реализации. Также в перспективе целесообразна интеграция с мобильным приложением, что расширит функциональные возможности проекта.

Оценка работы

В целом, дипломная работа выполнена на высоком уровне, демонстрирует сформированные профессиональные компетенции в области микроконтроллерных систем, сенсорной электроники и удалённого управления. Работа обладает практической и методической ценностью, может быть рекомендована к защите. Считаю, что дипломный проект очень отлично оценен, и студент Ганиев Шухрат Рустамовича заслуживает академической степени бакалавра.

Рецензент:
Заведущей кафедрой академии «Общие научные дисциплины»,
к.т.н., доцент _____ Сейдилдаева А.К.

«___» июня 2025 г.

Протокол

о проверке на наличие неавторизованных заимствований (плагиата)

Автор: Ганиев Шухрат Рустамович

Соавтор (если имеется):

Тип работы: Дипломная работа

Название работы: Разработка системы контроля доступа с анализом рисков и методами обеспечения безопасности на базе ESP32

Научный руководитель: Мурат Бактыбаев

Коэффициент Подобия 1: 1.4

Коэффициент Подобия 2: 0

Микропробелы: 0

Знаки из здругих алфавитов: 0

Интервалы: 0

Белые Знаки: 0

После проверки Отчета Подобия было сделано следующее заключение:

☐ Заимствования, выявленные в работе, является законным и не является плагиатом. Уровень подобия не превышает допустимого предела. Таким образом работа независима и принимается.

☐ Заимствование не является плагиатом, но превышено пороговое значение уровня подобия. Таким образом работа возвращается на доработку.

☐ Выявлены заимствования и плагиат или преднамеренные текстовые искажения (манипуляции), как предполагаемые попытки укрытия плагиата, которые делают работу противоречащей требованиям приложения 5 приказа 595 МОН РК, закону об авторских и смежных правах РК, а также кодексу этики и процедурам. Таким образом работа не принимается.

☐ Обоснование:

Дата *Заведующий кафедрой*

StrikePlagiarism.com SATBAYEV UNIVERSITY

Дата отчета 6/3/2025
Дата редактирования ---

Отчет не был оценен

Отчет подобия

Метаданные

Название организации
Satbayev University
Название
Разработка системы контроля доступа с анализом рисков и методами обеспечения безопасности на базе ESP32
Автор Научный руководитель / Эксперт
Ганиев Шухрат РустамовичМурат Бактыбаев
Подразделение
ИАиИТ

Объем найденных подобий

КП-ия определяют, какой процент текста по отношению к общему объему текста был найден в различных источниках.. Обратите внимание!Высокие значения коэффициентов не означают плагиат. Отчет должен быть проанализирован экспертом.

25
Длина фразы для коэффициента подобия 2

5471
Количество слов

42148
Количество символов

Тревога

В этом разделе вы найдете информацию, касающуюся текстовых искажений. Эти искажения в тексте могут говорить о ВОЗМОЖНЫХ манипуляциях в тексте. Искажения в тексте могут носить преднамеренный характер, но чаще, характер технических ошибок при конвертации документа и его сохранении, поэтому мы рекомендуем вам подходить к анализу этого модуля со всей долей ответственности. В случае возникновения вопросов, просим обращаться в нашу службу поддержки.

Замена букв	ß	0
Интервалы	A→	0
Микропробелы	-	0
Белые знаки	ß	0
Парафразы (SmartMarks)	a	6

Подобия по списку источников

Ниже представлен список источников. В этом списке представлены источники из различных баз данных. Цвет текста означает в каком источнике он был найден. Эти источники и значения Коэффициента Подобия не отражают прямого плагиата. Необходимо открыть каждый источник и проанализировать сожержание и правильность оформления источника.

10 самых длинных фраз — Цвет текста

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ И АДРЕС ИСТОЧНИКА URL (НАЗВАНИЕ БАЗЫ)	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
1	Конкурс “Аэропорт будущего” [redacted] 5/2/2019 Satbayev University (И_АиС)	16 0.29 %
2	https://www.adaface.com/blog/postgresql-interview-questions/	12 0.22 %

3	https://docs.cirkitdesigner.com/project/published/a7c90e3f-bf16-4f62-94ae-84ce32f79028/arduino-uno-rfid-access-control-with-i2c-lcd-display	11 0.20 %
4	КвРабота-Білик-2024-12-16 [redacted] 12/20/2024 National University of Shipbuilding named after Admiral Makarov (National University of Shipbuilding named after Admiral Makarov)	11 0.20 %
5	https://docs.cirkitdesigner.com/project/published/c0cfc082-7317-4d61-944c-129401e735d1/arduino-rfid-door-lock-system-with-lcd-display-and-servo-motor	10 0.18 %
6	https://docs.cirkitdesigner.com/project/published/a7c90e3f-bf16-4f62-94ae-84ce32f79028/arduino-uno-rfid-access-control-with-i2c-lcd-display	9 0.16 %
7	https://docs.cirkitdesigner.com/project/published/c0cfc082-7317-4d61-944c-129401e735d1/arduino-rfid-door-lock-system-with-lcd-display-and-servo-motor	7 0.13 %

из базы данных RefBooks (0.00 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)

из домашней базы данных (0.29 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
1	Конкурс "Аэропорт будущего" [redacted] 5/2/2019 Satbayev University (И_АиС)	16 (1) 0.29 %

из программы обмена базами данных (0.20 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
1	КвРабота-Білик-2024-12-16 [redacted] 12/20/2024 National University of Shipbuilding named after Admiral Makarov (National University of Shipbuilding named after Admiral Makarov)	11 (1) 0.20 %

из интернета (0.90 %)

ПОРЯДКОВЫЙ НОМЕР	ИСТОЧНИК URL	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
1	https://docs.cirkitdesigner.com/project/published/a7c90e3f-bf16-4f62-94ae-84ce32f79028/arduino-uno-rfid-access-control-with-i2c-lcd-display	20 (2) 0.37 %
2	https://docs.cirkitdesigner.com/project/published/c0cfc082-7317-4d61-944c-129401e735d1/arduino-rfid-door-lock-system-with-lcd-display-and-servo-motor	17 (2) 0.31 %
3	https://www.adaface.com/blog/postgresql-interview-questions/	12 (1) 0.22 %

Список принятых фрагментов (нет принятых фрагментов)

ПОРЯДКОВЫЙ НОМЕР	СОДЕРЖАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)