МИНИСТЕРСТВО НАУКИ и высшего образования РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

**«тюменский индустриальный университет»**

многопрофильный колледж

Отделение информационных технологий и

вычислительной техники

**курсовОЙ ПРОЕКТ**

на тему **ПРОЕКТИРОВАНИЕ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ КОНТРОЛЯ И УПРАВЛЕНИЯ ДОСТУПОМ**

по ПМ.03 Техническое обслуживание и ремонт компьютерных систем и комплексов

Студент Кулаков М. И. « » 2020 г.

Группа КСт-16-(9)-2

Специальность 09.02.01 «Компьютерные системы и комплексы»

Руководитель Проданчук И. В. « » 2020 г.

2020

МИНИСТЕРСТВО НАУКИ и высшего образования РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

**«тюменский индустриальный университет»**

многопрофильный колледж

Отделение информационных технологий и

вычислительной техники

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ**

**по** ПМ.03 Техническое обслуживание и ремонт компьютерных систем и комплексов

Студента Кулакова М. И. группы КСт-16-(9)-2

Тема курсовой работы: Проектирование интеллектуальной системы контроля и управления доступом

утверждена на заседании ЦК ИТВТ

протокол № \_\_\_ от «\_\_\_» \_\_\_\_\_\_\_ 20\_\_г.

Срок сдачи курсовой работы \_\_\_\_\_\_\_\_

Перечень вопросов, подлежащих исследованию или разработке:

1 Теоретическая часть

1.1 Анализ предметной области

1.2 Анализ технического задания

1.3 Обзор существующих аналогов

2 Практическая часть

2.1 Разработка структурной схемы

2.2 Выбор электронных компонентов

2.3 Разработка принципиальной схемы

2.4 Расчет энергопотребления

2.5 Разработка архитектуры программного обеспечения

2.6 Описание информационного обмена

2.7 Конфигурирование микроконтроллера

2.8 Разработка функционально законченных компонентов программного обеспечения

Руководитель курсового проекта:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /И.В. Проданчук

Заведующий отделением ИТВТ:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Т.А.Петрова

Задание принял к исполнению:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /М.И.Кулаков

Дата:\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc34862447)

[1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 7](#_Toc34862448)

[1.1 Анализ предметной области 7](#_Toc34862449)

[1.2 Анализ технического задания 8](#_Toc34862450)

[1.3 Обзор существующих аналогов 9](#_Toc34862451)

[2 ПРАКТИЧЕСКАЯ ЧАСТЬ 10](#_Toc34862452)

[2.1 Разработка структурной схемы 10](#_Toc34862453)

[2.2 Выбор электронных компонентов 11](#_Toc34862454)

[2.3 Разработка принципиальной схемы 14](#_Toc34862455)

[2.4 Расчет энергопотребления 15](#_Toc34862456)

[2.5 Разработка архитектуры программного обеспечения 17](#_Toc34862457)

[2.6 Описание информационного обмена 18](#_Toc34862458)

[2.7 Конфигурирование микроконтроллера 21](#_Toc34862459)

[2.8 Разработка функционально законченных компонентов программного обеспечения 24](#_Toc34862460)

[ЗАКЛЮЧЕНИЕ 26](#_Toc34862461)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 28](#_Toc34862462)

[ПРИЛОЖЕНИЕ А 30](#_Toc34862463)

[ПРИЛОЖЕНИЕ Б 33](#_Toc34862464)

# ВВЕДЕНИЕ

В нашей жизни существует множество зданий и помещений с ограниченным доступом. Ограничение доступа к ним необходимо для обеспечения безопасности и сохранения различного рода тайн (таких как государственная, военная, коммерческая и т. д.) и для ограничения доступа к различному специальному оборудованию. В случае если нежелательные сторонние люди получат доступ к функционально важному объекту, возможен выход его из строя или другие нежелательные последствия. Для ограничения доступа используется пропускной режим. Основной его целью является исключение несанкционированного проникновения. Для его введения необходимо реализовать КПП, бюро пропусков, введение удостоверений и организацию охраны здания. В целях облегчения реализации всё чаще применяют микроконтроллерные устройства – системы контроля и управления доступом, с помощью которых возможна реализация множества функций и возможностей. Использование СКУД позволяет уменьшить количество задействованного персонала для реализации пропускного режима. Основными функциями является ограничение, идентификация и регистрация перемещений, в также сбор статистики на основе полученных данных.

Актуальность темы в наше время обусловлена постоянной растущей потребностью в устройствах для облегчения введения ограничения доступа, как в коммерческом, так и в государственном секторе. Кроме этого, из-за наличия потенциала развития устройств СКУД растут требования, предъявляемые к данным устройствам. В наше время идёт активная интеграция систем контроля и управления доступом с другими подразделениями бизнеса (например, с административными и бухгалтерскими подразделениями).

Объектом исследования является устройства сферы безопасности, а предметом является система контроля и управления доступом.

Целью данного курсового проекта является разработка интеллектуальной системы контроля и управления доступом.

Для достижения цели необходимо выполнить ряд задач, такие как:

* изучение теоретического материала по изготовлению комбинационных схем и плат печатного монтажа;
* изучение строения систем контроля и управления доступа;
* изучение функциональных особенностей используемых компонентов;
* выбор технического решения, наиболее полно удовлетворяющего заданным требованиям;
* создание принципиальной схемы;
* создание структурной схемы;
* разработка программного обеспечения проектируемого изделия.

Данная курсовая работа состоит из теоретической и практической главы.

В теоретической части производится анализ предметной области, технического задания и обзор существующих аналогов устройства. В практической части данной курсовой работы рассматривается структурная и принципиальная схема устройства, электронные компоненты, энергопотребление, архитектура программного обеспечения, информационный обмен и т. д.

# ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Система контроля и управления доступом представляет из себя совокупность программно-аппаратных технических средств контроля и средств управления, имеющих целью ограничение и регистрацию входа-выхода объектов (людей, транспорта) на заданной территории через «точки прохода». Основной задачей данного устройства является управление доступом на определённой территории, ограничение и идентификация.

## 1.1 Анализ предметной области

Изначально в сфере безопасности для ограничения доступа необходимо было провести ряд мероприятий, включающих в себя создание КПП, бюро пропусков и т. д. После повсеместного внедрения цифровых устройств процесс организации пропускного режима значительно облегчился благодаря внедрению СКУД. В данный момент устройства СКУД находятся в активном развитии благодаря внедрению современных технологий. Так как оно представляет из себя совокупность программно-аппаратных средств, для него характерна модульность. Системы контроля и управления доступом в сфере безопасности относятся к направлению инженерно-технической защиты.

В качестве преграждающего устройства может использоваться электрозащёлка, электромагнитный замок, турникет, шлагбаум и другие средства. Идентификатор является базовым элементом определяющий степень надёжности устройства. Он обычно реализуется с помощью RFID меток, карточки с магнитной полосой, либо с помощью биологических данных (отпечаток пальцев и т. д.). Основное управление выполняется контроллером, он может быть автономный или сетевым. Сетевой контроллер может объединяться с другими контроллерами и подключаться к компьютеру. Коды идентификации хранятся внутри памяти контроллера. В качестве вспомогательного оборудования могут использоваться блоки бесперебойного питания, дверные доводчики, датчики открывания двери, кнопки, провода, видеонаблюдение и т. д.

Сетевые СКУД позволяют реализовать различные алгоритмы для организации допуска в разные зоны, собирать данные в виде статистики и архива событий, организовывать учёт рабочего времени и взаимодействовать с другими подсистемами безопасности. В качестве беспроводной сети может использоваться Bluetooth, Wi-Fi. ZigBee, GSM и т. д.

Основной цикл работы устройства представляет из себя ожидание начала процесса идентификации, после окончания процесса ввода (приёма данных) идёт проверка и в случае удовлетворения условий предоставляется доступ (независимо от результата происходит создание записи в журнале). Кроме этого, устройство может находиться в (кратковременном) режиме обновления прошивки и данных и в других режимах.

СКУД может применяться в бизнес-центрах, банках, учреждениях образования, на охраняемых территориях, в промышленных предприятиях и т. д.

## 1.2 Анализ технического задания

Темой технического задания является: “Проектирование интеллектуальной системы контроля и управления доступом”. Согласно этому заданию, необходимо разработать устройство на основе микроконтроллера, которое позволит управлять, ограничивать и отслеживать доступ к определённым контролируемым объектам. Проектируемая система должна быть универсальной сетевой (способной работать как в автономном режиме, так и в сетевом). Устройство должно соответствовать требованиям ГОСТов Р 51241-2008 и Р 54831-2011. Кроме этого, устройство должно поддерживать идентификацию с помощь ПИН-кода, отпечатка пальцев и RFID-меток. Также в функциональные возможности устройства должна входить поддержка веб интерфейса управления. По ГОСТу СКУД (по числу контролируемых точек система) является системой малой ёмкости [1, стр. 12].

## 1.3 Обзор существующих аналогов

На данный момент на рынке существует огромное количество моделей систем контроля и управления доступом. Данные модели различаются назначением, используемыми модулями и способами реализации. В таблице №1 рассматриваются существующие аналогичные устройства.

Таблица № 1 – Сравнение аналогичных устройств

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Характеристики | Smartec ST-SC110EKF | AccordTec AT-CP | Anviz C2 | BioSmart UniPass |
| Тип | Автономный | Автономный | Сетевой | Сетевой |
| Тип идентификатора | Сканер отпечатка пальцев,  Карты EM-Marin | Карты EM-Marin | Клавиатура, Бесконтактные карты, Сканер отпечатка пальца | Сканер отпечатков пальцев, Считыватель RFID меток |
| Выходные подключения | Реле 12 В для замка, датчика, кнопки выхода, звонок, тревожного устройства | Реле 12 В 60 мА и реле замка 3 А | 8-pin для подключения внешнего сканера отпечатков пальцев и замка | 2 реле, RS-485, дискетные входы и выходы, индикаторы состояния |
| Количество пользователей | 200 отпечатков и 2000 карт | 10000 | 3000 отпечатков 1500 карт | 1000000 |
| Напряжение питания | 12 В | 12 В | 5 В | 12 - 24 В |
| Кол-во подкл. считывателей | 0 | 0 | 1 | 2 |

# ПРАКТИЧЕСКАЯ ЧАСТЬ

## 2.1 Разработка структурной схемы

Пред разработкой структурной схемы необходимо определить основные блоки на основании требований и функциональных возможностей устройства. Из технического задания можно выделить следующие блоки структурной схемы:

* микроконтроллер;
* сетевой адаптер;

Для реализации ряда функций н необходимы следующие блоки:

* сканер отпечатков пальцев;
* считыватель бесконтактных карт;
* кодовая панель;
* дисплей;
* источник бесперебойного питания;
* реле.

Микроконтроллер выполняет управление устройством, а также выполняет переключение реле управления электрозамком. В качестве средств идентификации используются: сканер отпечатков пальцев, считыватель беспроводных карт и кодовая панель. Кроме этого, кодовая панель позволяет выполнять управление устройством с помощью отображаемого на дисплее меню. Для подключения к компьютеру и объединения с другими устройствами используется сетевой адаптер. База данных хранится на внешнем сервере и дублируется в ППЗУ. Источник бесперебойного питания позволяет работать устройству при отключении основного питания. Структурная схема СКУД представлена на рисунке № 1.

Рисунок № 1 – Структурная схема устройства

## 2.2 Выбор электронных компонентов

Выбор электронных компонентов производится на основании ряда характеристик. Он необходимо для достижения оптимального соотношения возможностей и цены. Основными компоненты, которые необходимо выбрать это микроконтроллер, сканер отпечатков пальцев, ППЗУ, дисплей, сетевой адаптер и т. д.

Память микроконтроллера бывает нескольких видов: Flash – это память, в которой хранится прошивка МК. Данная память может быть перезаписываемой, но количество циклов перезаписи у неё меньше, чем у ППЗУ. RAM в отличие от Flash при отключении питания теряет данные, но тем нее менее не имеет ограничений на количество циклов перезаписи. Поэтому RAM в основном используется для хранения переменных.

CPU – это электронный блок МК выполняющий код программы. Самой популярной архитектурой, применяемой микроконтроллерах, является ARM. Частота CPU определяет скорость выполнения команд.

Большинство микроконтроллеров имеют встроенную поддержку различных периферийных интерфейсов и протоколов, таких как таймер, PWM (широтно-импульсная модуляция), АЦП (аналогово-цифровой преобразователь), ЦАП, SPI (последовательный периферийный интерфейс), I²C, JTAG и т.д.

Таблица № 2 – Сравнение микроконтроллеров

|  |  |  |  |
| --- | --- | --- | --- |
| Характеристики | Espressif ESP32 | Texas Instr. CC2650 | STM SPWF01SA |
| Память | 448 Кб (Flash)  520 Кб (SRAM) | 128 Кб (Flash)  20 Кб (RAM)  8 Кб (SRAM) | 1.5 Мб (Flash)  64 Кб (RAM) |
| CPU | Tensilica Xtensa LX6 | ARM Cortex-M3 | ARM Cortex-M3 |
| Частота CPU | 240 МГц | 48 МГц | 72 МГц |
| Периферийные интерфейсы и протоколы | UART, I2C, I2S, RTC, SPI, CAN, SDIO, ETH, IR | UART, I2C, I2S, RTC, SSI, TRNG | UART, I2C, I2S, CAN, SDIO, USB |
| Разрядность АЦП | 12 | 12 | 12 |
| Напряжение | 2.3 – 3.6 В | 1,8 – 3,8 В | 3,1 – 3,6 В |
| Поддержка обновления по сети | + | + | + |
| Кол-во таймеров | 4 | 4 | 11 |
| JTAG | + | + | +/– |
| Кол-во GPON | 36 | 31 | 16 |
| Беспроводные сети | Wi-Fi 802.11 bgn, Bluetooth v4.2, BLE | Bluetooth, ZigBee RF4CE, 6LoWPAN | Wi-Fi 802.11 bgn |
| Цена | 160 ₽ | 820 ₽ | 1540 ₽ |

Для упрощения принципиальной схемы выбираемый микроконтроллер должен содержать в себе интегрированный Wi-Fi контроллер. Для выбора МК необходимо произвести анализ их характеристик (таблица №2). В результате анализа можно сделать вывод, что наиболее подходящим микроконтроллером за счёт цены и возможностей является ESP32.

Таблица № 3 – Сравнение сканеров отпечатков пальцев

|  |  |  |
| --- | --- | --- |
| Характеристики | FPM10A | FZ3387 |
| Питание | 3,6 – 6,0 В | 3,8 – 7,0 В |
| Интерфейсы | UART | UART |
| Тип сканера | Оптический | Емкостный |
| Размер шаблона | 512 байт | 768 байт |
| Максимальное кол-во отпечатков | 150 | 200 |
| Площадь соприкосновения | 2 см \* 1,5 см | 2,5 см \* 1,5 см |
| Цена | 500 ₽ | 1300 ₽ |

Для повышения скорости работы и точности работы устройства необходимо, чтобы обработка данных происходила контроллером сканера отпечатков. Максимальное количество хранимых отпечатков зависит от размера шаблона и памяти устройства. Сравнение модулей происходит на основании характеристик, указанных в таблице №3. Так как устройство малой ёмкости, соотношение цены и максимального количества отпечатков является определяющим, поэтому FPM10A является наиболее подходящим.

Таблица № 4 – Сравнение дисплеев

|  |  |  |  |
| --- | --- | --- | --- |
| Характеристики | LCD1602 | SH1122 | ILI9341 |
| Питание | 2,5 – 6 В | 3,3 В | 3,3 – 5 В |
| Интерфейсы | I2C | I2C | I2S |
| Тип экрана | LCD | OLED | TFT |
| Цветной/Монохромный | М | М | Ц |
| Разрешение | 35 \* 16 \* 2 | 256 \* 64 | 240 \* 320 |
| Размер | 3,6 см \* 8 см | 2,7 см \* 7,2 см | 4 см \* 6,7 см |
| Цена | 100 ₽ | 950 ₽ | 500 ₽ |

Для управления устройством в основном будет использоваться Web-интерфейс. Управление с помощью меню, отображаемого на дисплее, носит второстепенную значимость. Поэтому определяющим фактором при выборе дисплея является его доступность, простота использования и цена. Для уменьшения количества занимаемых портов большинство дисплеев имеют встроенную поддержку интерфейсов передачи данных, таких как I2C, I2S и т. д. Одним из самых распространённых дисплеев является LCD1602. Учитывая сравнение характеристик, указанных в таблице №4, дисплей LCD1602 является самым подходящим в этом случае.

## 2.3 Разработка принципиальной схемы

При разработке принципиальной схемы необходимо выполнять ряд правил и указаний, указанных в стандартах. Одним из стандартов распространяющийся на электрические схемы является ГОСТ 2.701 “Правила выполнения электрических схем”. В данном ГОСТе описаны основные термины, а также правила создания соединений, подключений, общих схем и схем расположения. Все элементы и устройства на схеме изображаются в виде условных графических обозначений, согласно ГОСТу 2.710.

При составлении УГО для микросхем рекомендуется использовать документацию производителя. На обозначении входы и выходы группируются, при этом сохраняют свой порядковый номер согласно реальному расположению. Так как в современных микроконтроллерах один порт может совмещать в себе несколько функций может быть указано обобщённое его название (например IO23).

Для упрощения схемы допустимо её разбиение на связанные блоки. Кроме этого, для повышения читаемости используют метки цепи, позволяющие соединить разные компоненты, находящиеся в разных сторонах листа, либо на разных страницах.

В результате разработки принципиальной схемы получилась схема, содержащая в себе 8 микросхем, 2 модуля, 1 дисплей, 2 разъёма для подключения электрического замка и прочие электрические компоненты. Принципиальная схема представлена в приложении А.

## 2.4 Расчет энергопотребления

При разработке любого устройства одним из важных пунктов является расчёт его энергопотребления. Так как мощность — это произведение силы тока на напряжение, то для расчёта энергопотребления необходимо знать потребляемую силу тока и рабочее напряжение. Кроме микросхем одними из потребителей силы тока являются реле, экран, модули, светодиоды и пьезодинамик. Для упрощения расчётов данные значения указаны в таблице №5. Так как некоторые элементы не работают постоянно, а по мере надобности, полученное значение будет отображать максимальное возможное потребление, а не среднее.

Таблица № 5 – Расчёт энергопотребления устройства

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Элемент цепи | Кол-во | Сила тока | Напряжение | Мощность 1 ед. | Мощность |
| ESP32-WROOM-32 | 1 | 100 мА | 3.6 В | 0.36 Вт | 0.36 Вт |
| Контроллер зарядки TP4056E | 1 | 500 мкА | 5 В | 0.0025 Вт | 0.0025 Вт |
| Защита аккумулятора DW01A | 1 | 6 мкА | 3.3 В | 0.0001 Вт | 0.0001 Вт |
| Стабилизатор напряжения AMS1117-3.3 | 1 | 13 мА | 5 В | 0.065 Вт | 0.065 Вт |
| Преобразователь USB-UART CH340G | 1 | 30 мА | 5 В | 0.15 Вт | 0.15 Вт |
| Часы реального времени DS1307 | 1 | 1.5 мА | 3.3 В | 0.00495 Вт | 0.00495 Вт |
| EEPROM память AT24C512PI27 | 1 | 5 мА | 3.3 В | 0.0165 Вт | 0.0165 Вт |
| Преобразователь IIC PCF8574P | 1 | 100 мА | 3.3 В | 0.33 Вт | 0.33 Вт |
| Дисплей LC1602 | 1 | 1.1 мА | 3.3 В | 0.00363 Вт | 0.00363 Вт |
| Сканер отпечатков пальцев FPM10A | 1 | 150 мА | 3.3 В | 0.495 Вт | 0.495 Вт |
| Считыватель бесконтактных меток RC522 | 1 | 10 мА | 3.3 В | 0.033 Вт | 0.033 Вт |
| Реле | 2 | 71.4 мА | 5 В | 0.36 Вт | 0.72 Вт |
| Карта памяти microSD | 1 | 100 мА | 3.3 В | 0.33 Вт | 0.33 Вт |
| Итого: | | | | | 2.51068 Вт |

В результате расчётов получилось, что максимальная мощность устройства примерно равна 2.5 Вт. Так как устройство может питаться от внутреннего источника питания (в виде литий-ионного аккумулятора LGAAS31865 формата 18650, производства LG) необходимо рассчитать максимальную мощность батареи.

Напряжение аккумулятора согласно информации из документации изготовителя составляет 3.6 В, а максимальная сила тока при разрядке 3225 мА. В результате максимальная мощность батарейки равна 11.61 Вт. Для получения значения силы тока, проходящей через аккумулятор необходимо потребляемую мощность разделить на напряжение, в результате получится примерно 700 мА. Время работы устройства от аккумулятора равно ёмкости разделённой на произведение силы тока и коэффициента. В результате можно сделать вывод, что данный аккумулятор подходит для устройства и устройство без внешнего источника питания сможет проработать на нём 2.2 часа.

## 2.5 Разработка архитектуры программного обеспечения

Программная архитектура Arduino Core базируется на библиотеке платформы Wiring. Именно она определяет основную структуру и последовательность кода, основные функции, и т. д. Кроме этого обеспечена совместимость с основными функциями ESP IDF.

Загрузчик используемый в Arduino Core – optiboot. Данный загрузчик использует специфический протокол Arduino основанный на UART. Загрузчик является аналогом программного программатора stk500. Загрузка кода возможна только во время загрузки микроконтроллера, что позволяет защитить от Flash от действий вредоносного кода. После запуска микроконтроллера загрузчик некоторое время следит за выводами. В случае поступления сигнала скомпилированный файл помещается в PROGMEM. После завершения загрузки чип выполняет сброс. Если сигнал с USB не поступает некоторое время, начинается выполнение пользовательского кода. В основном загрузчик занимает около 512 байт. Кроме этого, обновление возможно через интернет, с помощь OTA update.

Работа с встроенным Wi-Fi модулем осуществляется микроконтроллером (при использовании Arduino Core) с помощью AT-команд. Встроенные функции Wi-Fi представляют из себя отправку и обработку ответа AT команд Wi-Fi контроллера.

Так как разработка ПО для ESP32 ведётся с использованием языка С++, то блоки архитектуры, как и в любом объектном ориентированном языке, могут содержать свои функции, встроенные типы данных, переменные и т.д. Кроме этого, в архитектуре программного обеспечения описывается основная концепция работы алгоритма ПО, методы обработки ошибок, реализация изменения архитектуры, основные требования к интерфейсу и т.д.

## 2.6 Описание информационного обмена

При описании информационного обмена необходимо выделить отдельные блоки ПО и описать ихнее взаимодействие. Для обмена данными между блоками устройства используются различные стандартизированные шины, такие как I2C, I2S, UART и USB. Для обмена с базой данных и обеспечения управления устройством с помощью браузере через локальную сеть используется протокол HTTP.



Рисунок № 2 – Схема взаимодействия блоков устройства

Взаимодействие блоков указано на рисунке № 2. Основными блоками являются:

1. устройства IIC (I2C):
   1. EEPROM – постоянная память, хранит основные параметры;
   2. часы реального времени – передача значений текущего времени для составления записей в журнале;
   3. считыватель RFID меток – передаёт и записывает информацию на RFID карту;
   4. дисплей – вывод информации на экран, управление осуществляется путём передачи команд и значений;
2. карта памяти – основное назначение карты памяти в устройстве хранение журналов и резервной копии базы данных, а также всей информации, которая не имеет постоянного определённого размера, для обмена используется I2S интерфейс;
3. устройства UART:
   1. сканер отпечатков пальцев – используется пакетная передача, принимает команды от микроконтроллера, передаёт номер приложенного отпечатка, скан отпечатка, шаблон и т. д.;
   2. USB to UART – основное назначение данного блока является прямая прошивка МК, с помощью загрузчика;
4. Wi-Fi – данный блок предназначен для организации связи с устройствами сети используя Wi-Fi сеть;
5. HTTP – данный блок выделен в отдельный объект в программном коде, он выполняет функции веб-сервера;
6. кодовая панель – передаёт координаты нажатой кнопки.

В основном все блоки связаны с МК. Основная концепция алгоритма работы МК заключается в проверке наличия новых событий в бесконечном цикле, таких как ввод с клавиатуры, использования RFID метки или сканера отпечатков. Остальные блоки, такие как UART to USB и HTTP, будут выполнять свою работу при возникновении прерывания.

Основной шиной, используемой для связи между блоками устройства, является I2C из-за её простоты и распространённости. Тем не менее одним из её минусов является низкая её скорость. Данная шин требует наличие 2-х линий – SCL и SDA. Тактирование выполняется по шине SCL, а передача данных выполняется в полудуплексном режиме по линии SDA. К обеим шинам подключается подтягивающий резистор. На шине находится 5 устройств: 1 master и 4 slave. Передача начинается с передачи стартового бита, после передаётся 7 битный адрес, 8 бит указывает на необходимость считывания или записи, после генерируется ACT – нулевой бит указывающий на успешное принятие данных. После этого следует передача 8 бит данных и генерация ACT. Сигнал окончания передачи всегда генерирует master. Адреса устройств указаны в таблице №6.

Таблица № 6 – Адреса модулей в шине IIC

|  |  |
| --- | --- |
| DS1307 | 0b1101000 |
| AT24C512PI27 | 0b1010000 |
| PCF8574P | 0b0100000 |
| RC522 | 0b0101000 |

Работа Web сервера основана на использовании HTTP протокола. Запрос инициируется получателем страницы и проходит через множество посредников. Сам протокол не имеет состояния, используется сессия. Основной HTTP:

* открытие TCP соединения;
* отправка HTTP заголовка, например:

GET / HTTP/1.1

Host: yandex.ru

Accept-Language: ru

* чтение ответа сервера, например:

HTTP/1.1 200 OK

Date: Sat, 09 Oct 2010 14:28:02 GMT

Server: Apache

Last-Modified: Tue, 01 Dec 2009 20:18:22 GMT

ETag: "51142bc1-7449-479b075b2891b"

Accept-Ranges: bytes

Content-Length: 29769

Content-Type: text/html

<html>…</html>

## 2.7 Конфигурирование микроконтроллера

Для упрощения конфигурирования при разработке кода используется фреймворк Arduino. Настройка частот и разметки FLASH осуществляется пред загрузкой прошивки. Для конфигурирования МК в Arduino IDE, работающей совместно с текстовым редактором Visual Studio Code, необходимо вызвать окно “Конфигурирования платы” (1). После необходимо выбрать МК (2), в зависимости от выбранного микроконтроллера зависят доступные для конфигурирования параметры (3). Этапы конфигурирования показаны на рисунке №3.

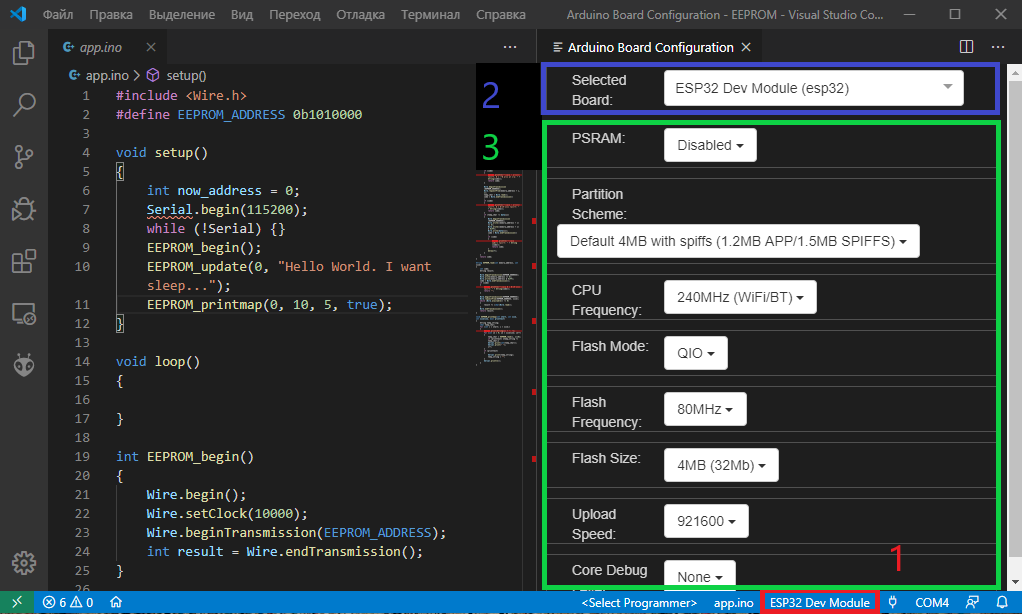


Рисунок № 3 – Конфигурирование микроконтроллера

Частота ядра (CPU Frequency) была выбрана 240 МГц, частота FLASH (Flash Frequency) 80 МГц. Для использования SPIFFS необходимо выделить 1.5 МБ FLASH для файловой системы (Partition Scheme). Настройка портов (таблица №7) осуществляется коде программы, согласно информации производителя и принципиальной схеме.

Таблица № 7 – Конфигурация портов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Порт | № | Доступные функции порта | Режим | Назначенная функция |
| RXD0 | 35 | GPIO3, U0RXD, CLK\_OUT2 | Вывод | U0RXD |
| TXD0 | 34 | GPIO1, U0TXD, CLK\_OUT3, EMAC\_RXD2 | Вход | U0TXD |
| SCS/CMD | 19 | GPIO11, SD\_CMD, SPICS0, HS1\_CMD, U1RTS | Полудуплекс | SD\_CMD |
| SCK/CLK | 20 | GPIO6, SD\_CLK, SPICLK, HS1\_CLK, U1CTS | Вход | SD\_CLK |
| SDO/SD0 | 21 | GPIO7, SD\_DATA0, SPIQ, HS1\_DATA0, U2RTS | Полудуплекс | SD\_DATA |
| SDI/SD1 | 22 | GPIO8, SD\_DATA1, SPID, HS1\_DATA1, U2CTS | Полудуплекс | SD\_DATA1 |
| IO0 | 25 | GPIO0, ADC2\_CH1, TOUCH1, RTC\_GPIO11, CLK\_OUT1, EMAC\_TX\_CLK | Вывод | GPIO0 |
| IO2 | 24 | GPIO2, ADC2\_CH2, TOUCH2, RTC\_GPIO12, HSPIWP, HS2\_DATA0, SD\_DATA0 | Вывод | GPIO2 |
| IO4 | 26 | GPIO4, ADC2\_CH0, TOUCH0, RTC\_GPIO10, HSPIHD, HS2\_DATA1, SD\_DATA1, EMAC\_TX\_ER | Вывод | GPIO4 |
| IO5 | 29 | GPIO5, VSPICS0, HS1\_DATA6, EMAC\_RX\_CLK | Вывод | GPIO5 (PWM) |
| IO12 | 14 | GPIO12, ADC2\_CH5, TOUCH5, RTC\_GPIO15, MTDI, HSPIQ, HS2\_DATA2, SD\_DATA2, EMAC\_TXD3 | Полудуплекс | SD\_DATA2 |
| IO13 | 16 | GPIO13, ADC2\_CH4, TOUCH4, RTC\_GPIO14, MTCK, HSPID, HS2\_DATA3, SD\_DATA3, EMAC\_RX\_ER | Полудуплекс | SD\_DATA3 |
| IO14 | 13 | GPIO14, ADC2\_CH6, TOUCH6, RTC\_GPIO16, MTMS, HSPICLK, HS2\_CLK, SD\_CLK, EMAC\_TXD2 | Вывод | GPIO14 |
| IO15 | 23 | GPIO15, ADC2\_CH3, TOUCH3, MTDO, HSPICS0, RTC\_GPIO13, HS2\_CMD, SD\_CMD, EMAC\_RXD3 | Вывод | GPIO15 |
| IO16 | 27 | GPIO16, HS1\_DATA4, U2RXD, EMAC\_CLK\_OUT | Вход | U2RXD |
| IO17 | 28 | GPIO17, HS1\_DATA5, U2TXD, EMAC\_CLK\_OUT\_180 | Вывод | U2TXD |
| IO21 | 33 | GPIO21, VSPIHD, EMAC\_TX\_EN | Вход | GPIO21 (SDA) |
| IO22 | 36 | GPIO22, VSPIWP, U0RTS, EMAC\_TXD1 | Вывод | GPIO22 (SCL) |
| IO23 | 37 | GPIO23, VSPID, HS1\_STROBE | Вывод | GPIO23 |
| IO25 | 10 | GPIO25, DAC\_1, ADC2\_CH8, RTC\_GPIO6, EMAC\_RXD0 | Вывод | GPIO25 |
| IO26 | 11 | GPIO26, DAC\_2, ADC2\_CH9, RTC\_GPIO7, EMAC\_RXD1 | Вывод | GPIO26 |
| IO27 | 12 | GPIO27, ADC2\_CH7, TOUCH7, RTC\_GPIO17, EMAC\_RX\_DV | Вывод | GPIO27 |
| IO32 | 8 | GPIO32, XTAL\_32K\_P, ADC1\_CH4, TOUCH9, RTC\_GPIO9 | Вход | GPIO32 |
| IO33 | 9 | GPIO33, XTAL\_32K\_N, ADC1\_CH5, TOUCH8, RTC\_GPIO8 | Вход | GPIO33 |
| IO34 | 6 | GPIO34, ADC1\_CH6, RTC\_GPIO4 | Вход | GPIO34 |
| IO35 | 7 | GPIO35, ADC1\_CH7, RTC\_GPIO5 | Вход | GPIO35 |

## 2.8 Разработка функционально законченных компонентов программного обеспечения

Основная структура кода определяется согласно стандартам С++ и документацией Arduino (так как используется фреймворк Arduino). В основном файле должно обязательно находиться два основных метода setup и loop.

Основная инициализация выполняется при запуске МК до и в методе setup. Согласно стандарту С++ в начале кода идут директивы. Директивы #include описывают подключаемые файлы и библиотеки. После идут директивы объявляющие идентификаторы и символы, которые подставляются при обнаружении директивы. Они используются для облегчения читаемости кода и упрощения его изменения. Вначале описываются директивы, используемые в настройке Wi-Fi. После идут директивы используемы для конфигурации EEPROM (адрес памяти в шине IIC, адреса разметки памяти и т. д.).

После идёт объявление глобальных переменных и объектов. В нашем случаем в начале объявляются переменные, отображающие состояние и объекты такие как веб сервер, EEPROM, RC522 и т. д.

Метод setup полностью формируется пользователем, и может быть пустым, так как среда разработки при компиляции файла создаёт метод стандартной инициализации на основе конфигурации платы. В нем содержатся инициализация компонентов, подключение к Wi-Fi сети, настройка маршрутизации Web сервера, включение поддержки обновления прошивки по воздуху и т. д.

Большинство микроконтроллеров имеют бесконечный цикл. В нашем случае данным циклом является метод loop. В этом цикле находятся выполняются такие функции как прослушка Wi-Fi соединения, и проверка на наличие новых попыток авторизации для получения доступа к помещению, ожидание запросов к Web серверу и т. д.

Так как код написан на языке C++, то фреймворк поддерживает объектно-ориентированное программирование. Поэтому основные объекты выделены в отдельные объекты и имеют свои функции. Данные объекты описаны в классах и во внешних подключаемых файлах, такие как библиотеки и заголовочные файлы. Пример кода показан в приложении Б.

# ЗАКЛЮЧЕНИЕ

Сфера безопасности имеет огромное значение в наше время. Для её обеспечения выполняется различные меры, одной из них является ограничение доступа к помещению. Раньше для ограничения доступа к помещению организовывались контрольно-пропускные пункты, которые являлись сложными в организации. Благодаря повсеместному внедрению новых технологий и их постоянному развитию, удалось упростить введение ограничение доступа к различным помещениям, зданиям и т. д. Сейчас все основные функции ограничения доступа реализуются с помощью системы контроля и управления доступом.

Актуальность работы обоснована растущей потребностью в обеспечении безопасности и обеспечения наиболее выгодной минимизации требуемых ресурсов. Целью работы являлась разработка интеллектуальной системы контроля и управления доступом. Для достижения цели был выполнен ряд задач, таких как изучение теоретического материала по изготовлению комбинационных схем и плат печатного монтажа, изучение строения систем контроля и управления доступа, выбор технического решения, наиболее полно удовлетворяющего заданным требованиям, создание принципиальной и структурной схемы и разработка программного обеспечения проектируемого изделия.

Во время работы была проанализирована предметная область, способы идентификации, изучены основные составляющие и возможности СКУД. При анализе технического задания были выделены основные требования к устройству, а также требуемый функционал. В результате обзора существующих аналогов было проведено сравнение текущих аналогов устройства.

В практической части был проведён ряд работ, таких как разработка структурной схемы; сравнение и выбор электронных компонентов, используемых в устройстве; описание информационного обмена; конфигурирование микроконтроллера. Так же был произведен расчёт энергопотребления и и продолжительности работы устройства от аккумулятора. Для разработки программного обеспечения необходимо было разработать и описать архитектуру ПО, а также разработать функционально законченные компоненты программного обеспечения.

В результате можно сделать вывод, что цель была достигнута и все поставленные задачи выполнены. В результате курсового проекта была разработана интеллектуальная система контроля и управления доступом.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. ГОСТ Р 51241-2008. Средства и системы контроля и управления доступом. Классификация. Общие технические требования. Методы испытаний. Введ. – 17.12.2008 – М.: Стандартинформ, 2009. – 52 с. – Текст: непосредственный
2. ГОСТ Р 54831-2011. Системы контроля и управления доступом. Устройства, преграждающие управляемые. Общие технические требования. Методы испытаний. Введ. – 01.09.2011 – М.: Стандартинформ, 2012. – 19 с. – Текст: непосредственный
3. ГОСТ 2.701-2008. Единая система конструкторской документации. Схемы. Виды и типы Общие требования к выполнению. Введ. – 01.07.2009 – М.: Стандартинформ, 2009. – 16 с. – Текст: непосредственный
4. Аполлонский С. М. Электротехника : учебник / С. М. Аполлонский. ⎯ М.: КНОРУС, 2018. ⎯ (Среднее профессиональное образование)
5. Банисла М. Решение задач на современном С++ / пер. с. англ. А. Н. Киселева – М.: ДМК Пресс, 2019. – 302 с.: ил.
6. Бишоп О. Электронные схемы и системы / пер. с англ. к. т. н. Рабодзей А. Н. – М.: ДМК Пресс, 2016. – 516 с.
7. Букатов А. А., Гуда С. А. Компьютерные сети: расширенный начальный курс. Учебник для вузов. ⎯ СПб.: Питер, 2020. ⎯ 496 с.: ил. ⎯ Серия «Учебник для вузов»
8. Лопаткин А. В. Проектирование печатных плат в Altium Designer. учеб. Пособие для практических занятий. 2-е изд., перераб. и доп. – М.: ДМК Пресс, 2017. – 554 с.: ил.
9. Макаров С. Л. Arduino Uno и Raspberry Pi 3: от схемотехники к интернету вещей. ⎯ М.: ДМК Пресс, 2019. – 204 с.: ил.
10. Неировский А. Е. Электроника: учебное пособие / А. Е. Немировский, И. Ю. Сергиевская, О. И. Степанов, А. В. Иванов. – М.: Инфра-Инженерная, 2019. – 200 с.
11. Петин В. А. 77 проектов для Arduino. ⎯ М. ДМК Пресс. 2020. ⎯ 356 с.: ил.
12. Петин В. А. Arduino и Raspberry Pi в проектах Internet of Things. ⎯ СПб.: БХВ-Петебург, 2016. ⎯ 32 с.: ил. ⎯ (Электроника)
13. Робсон Э., Фримен Э. Изучаем HTML, XHTML и CSS. 2-е изд. ⎯ СПб.: Питер, 2019. ⎯ 720 с.: ил. ⎯ (Серия «Head First O’Reilly»)
14. Солтис М. Введение в анализ алгоритмов / пер. с англ. А. В. Логунова. – М.: ДМК Пресс, 2019. – 278 с.: ил.
15. Хрусталева З. А. Источники питания радиоаппаратуры : учебник / З. А. Хрусталёва, С. В. Парфенов. ⎯ М.: КНОРУС, 2019. ⎯ 240 с. ⎯ (Среднее профессиональное образование)
16. Шварц М. Интернет вещей с ESP8266: пер. с англ. ⎯ СПб.: БХВ-Петербург, 2018. ⎯ 192 с.: ил. ⎯ (Электроника)

# ПРИЛОЖЕНИЕ А

# ПРИЛОЖЕНИЕ Б

//Подключаемые бибилиотеки

#include "Arduino.h"

#include "Wire.h"

…

//Предопределённые значения для WPS

#define ESP\_WPS\_MODE      WPS\_TYPE\_PBC

#define ESP\_MANUFACTURER  "ESPRESSIF"

…

//Предопределённые значения для EEPROM

#define EEPROM\_ADDRESS 0b1010000

//Предопределённые значения разметки EEPROM

#define EEPROM\_ADDRESS\_CONF 0  //Адреса памяти

#define EEPROM\_ADDRESS\_WiFi 1

…

#define EEPROM\_SIZE\_CONF 1 //Размер секторов и ед. информации

#define EEPROM\_SIZE\_MDNS 10

…

//Параметры загружаемые с EEPROM

bool NeedAP = false;

bool NeedSerial = true;

…

String ssid;

String password;

String host;

//Переменные состояния компонентов

bool WPSWorking = false;

bool MDSNWorking = false;

…

//Объекты используемые в ходе работы программы

Ticker Timer;

AsyncWebServer server(80);

…

//Прототипы функций

bool EEPROM.begin(int Clock = 10000);

void ConnectToWiFi(bool fromEEPROM = false);

…

//Основной код

void setup()

{

//Инициализация памяти

EEPROM memory (EEPROM\_ADDRESS, 10000);

if (memory.status)

{

//Считывание значений настроек

NeedAP = memory.readbit(1, EEPROM\_ADDRESS\_CONF, true, true);

…

}

//Инициализация компонентов согласно настройкам

if (NeedSerial) {Serial.begin(115200);}

…

//Настройка соединения

WiFi.mode(WIFI\_STA);

WiFi.disconnect();

if (memory.status)

{

WiFi.SetConnection();

}

else

{

ssid = DEFAULT\_SSID;

password = DEFAULT\_PASSWORD;

}

    if (SPIFFSWorking)

    {

//Настройка маршрутизации сервера

        server.on("/", HTTP\_GET, handleNotFound);

        server.onNotFound(handleNotFound);

        server.on("/style.css", HTTP\_GET, [](AsyncWebServerRequest \*request) {

            request->send(SPIFFS, "/style.css", "text/css");

        });

        server.on("/font-awsome all.css", HTTP\_GET, [](AsyncWebServerRequest \*request) {

            request->send(SPIFFS, "/font-awsome all.css", "text/css");

        });

…

        server.begin();

        if (MDSNWorking) {MDNS.addService("http", "tcp", 80);}

    }

    //Обновление по сети

    ArduinoOTA.onStart([]

    {

        String type;

        if (ArduinoOTA.getCommand() == U\_FLASH) {type = "sketch";}

        else {type = "filesystem";}

        if (SPIFFSWorking) {SPIFFS.end();}

    });

    ArduinoOTA.begin();

}

void loop()

{

    ArduinoOTA.handle(); //Проверка на наличие обновлений прошивки

//Опрос клавиатуры

char tempKey = keyboard.readKey(10);

if (!tempKey) {}

else if (tempKey == 13 or tempKey == 14 or tempKey == 15 or tempKey == 16){openMenu();}

else {readPassword();)}

//Опрос сканера отпечатков

int tempNumb = fingerprint.scan(10);

if (tempNumb != 0) {checkRols(tempNumb);}

//Опрос сканера RFID меток

int tempNumb = cardreader.readID();

if (tempNumb != 0) {checkRols(tempNumb);}

}