

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт машиностроения  
(наименования института полностью)

Кафедра «Промышленная электроника»  
(наименование)

11.03.04 Электроника и наноэлектроника  
(код и наименование направления подготовки, специальности)

Промышленная электроника  
(направленность (профиль) / специализация)

## **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**

на тему Система контроля доступа в корпуса ТГУ

Студент

А.Ю. Данилов

(И.О. Фамилия)

(личная подпись)

Руководитель

к.т.н., доцент, М.В. Позднов

(ученая степень, звание, И.О. Фамилия)

Тольятти 2020

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования  
«Тольяттинский государственный университет»

Институт машиностроения  
(наименование института полностью)

Кафедра Промышленная электроника  
(наименование)

## **ЗАДАНИЕ на выполнение бакалаврской работы**

Студент Данилов Андрей Юрьевич, Элб-1601а

1. Тема Система контроля доступа в корпуса ТГУ
2. Срок сдачи студентом законченной выпускной квалификационной работы \_\_\_\_\_

---

3. Исходные данные к выпускной квалификационной работе:

Первичный источник питания: внешний блок питания с выходом 5В DC

Тип предоставляемых измерений: код с RFID метки

Вторичный источник питания: аккумулятор, встроенный в корпус

Наличие дисплея и графического интерфейса: да

4. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов, разделов)

Аннотация

Введение

1. Состояние вопроса

2. Формулирование цели и задач проекта

3. Аппаратная часть

4. Программная часть

5. Выбор остальных элементов устройства

6. Разработка корпуса устройства

7. Диаграмма работы устройства

8. Изготовление прототипа устройства

Заключение

Список используемой литературы

5. Ориентировочный перечень графического и иллюстративного материала

1. Обзор существующих решений

2. Схема структурная устройства

3. Схема электрическая принципиальная устройства

4. Временные диаграммы работы

5. Изометрия корпуса устройства

6. Структура интерфейса устройства

6. Консультанты по разделам Позднов М.В

7. Дата выдачи задания «26» января 2019 г.

Руководитель выпускной  
квалификационной работы

\_\_\_\_\_

(подпись)

М.В. Позднов

(И.О. Фамилия)

Задание принял к исполнению

\_\_\_\_\_

(подпись)

А.Ю. Данилов

(И.О. Фамилия)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования  
«Тольяттинский государственный университет»

Институт машиностроения  
(наименование института полностью)

Кафедра Промышленная электроника  
(наименование)

**КАЛЕНДАРНЫЙ ПЛАН**  
**выполнения бакалаврской работы**

Студент А.Ю. Данилов

по теме Система контроля доступа в корпуса ТГУ

Наименование раздела (главы) <sup>1</sup>	Плановый срок выполнения раздела (главы) <sup>1</sup>	Фактический срок выполнения раздела (главы) <sup>1</sup>	Отметка о выполнении раздела (главы) <sup>1</sup>
Состояние вопроса	02.03.2020	02.03.2020	Выполнено
Формулирование цели и задач работы	09.03.2020	09.03.2020	Выполнено
Аппаратная часть	15.03.2020	15.03.2020	Выполнено
Программная часть	23.03.2020	23.03.2020	Выполнено
Выбор остальных элементов устройства	01.04.2020	01.04.2020	Выполнено
Разработка корпуса	08.04.2020	08.04.2020	Выполнено
Разработка ПО	26.04.2020	26.04.2020	Выполнено
Выбор остальных элементов	05.05.2020	05.04.2020	Выполнено
Разработка корпуса	15.05.2020	15.05.2020	Выполнено
Диаграмма работы устройства	24.05.2020	24.05.2020	Выполнено
Изготовление прототипа	01.06.2020	01.06.2020	Выполнено
Заключение	07.06.2020	07.06.2020	Выполнено
Список литературы	10.06.2020	10.06.2020	Выполнено

Руководитель бакалаврской работы \_\_\_\_\_

(подпись)

М.В. Позднов  
(И.О. Фамилия)

## Аннотация

Название дипломной работы «Система контроля доступа в корпуса ТГУ».

Данная дипломная работа состоит из пояснительной записки на 64 страницы, введения, 31 рисунка, списка из 21 источников, включая 6 источников на иностранном языке и двух приложений, и чертежей на двух листах А3.

Дипломная работа может быть разделена на следующие логически взаимосвязанные части: состояние вопроса, выбор компонентной базы, разработка программного и аппаратного обеспечения, разработка корпуса и создание прототипа устройства.

Важной проблемой на данный момент является контроль посетителей в организациях с большим количеством людей, поэтому необходимо разработать систему, которая заменит старые способы идентификации людей.

В работе затрагиваются следующие вопросы: как обеспечить контроль посетителей в общеобразовательном учреждении с большим количеством людей, как хранить информацию о посетителях и избежать утечки данных.

Цель работы - разработать прототип устройства, который может считывать код с карты, отправлять запрос на сервер, проверять полученное значение в базе данных, а также отображать информацию из базы данных на экране ПК. В зависимости от результата проверки значения в базе данных контролируют реле и светодиоды.

## **Abstract**

The title of the graduation project is Access Control System in TSU Buildings.

The graduation work consists of an explanatory note on 64 pages, introduction, including 31 figures, the list of 21 references including 6 foreign sources and two appendices, and the graphic part on two A3 sheets.

The graduation work may be divided into several logically connected parts, which are follows: problem state, selection of component base, software and hardware development, corpus and prototype device development.

At the moment the important problem is the control of visitors in organizations with many people. There is a need to develop a system that will replace the old ways of identifying people.

The work touches upon: how to control visitors in the educational institution, how to keep information about visitors and avoid data leakage.

The object of the graduation project is the working prototype, that can read code from card, send request to the server, check received value in the database, as well as display information from the database on the PC' screen. Depending on the result of checking the value in the database, control the relay and LEDs.

## Содержание

Введение.....	3
1 Состояние вопроса .....	4
1.1 Анализ исходных данных.....	4
1.2 Анализ известных решении .....	6
1.3 Выводы по разделу.....	9
2 Формулирование цели и задач проекта .....	10
2.1 Выводы по разделу.....	10
3 Аппаратная часть.....	11
3.1 Первичная структурная схема .....	11
3.2 Подбор компонентной базы устройства .....	12
3.3 Полная структурная схема .....	23
3.4 Разработка электрической принципиальной схемы устройства .....	24
3.5 Выводы по разделу.....	31
4 Программная часть.....	32
4.1 Написание программы для микроконтроллера.....	32
4.2 Написание программы для ПК .....	36
4.3 Выводы по разделу.....	40
5 Выбор остальных элементов устройства .....	41
5.1 Выводы по разделу.....	41
6 Разработка корпуса устройства. ....	42
6.1 Выводы по разделу.....	43
7 Диаграмма работы устройства.....	44
7.1 Вывод по разделу .....	45
8 Изготовление прототипа устройства.....	46
8.1 Выводы по разделу.....	48
Заключение .....	49
Список используемой литературы .....	50
Приложение А Программа для микроконтроллера .....	52
Приложение Б Программа для ПК .....	54

## Введение

Контроль посетителей в организациях, например, в общеобразовательных учреждениях, защита данных от хищений, являются одними из серьезных проблем в наше время.

Традиционные методы идентификации, основывающиеся на паспортах, пропусках или других бумажных носителях не всегда соответствуют современным требованиям безопасности. Также такие методы проверки существенно уменьшают пропускную способность проходной.

На данный момент вводятся системы, которые с помощью датчиков идентифицируют посетителей намного быстрее и точнее, сводя к минимуму участие человека. Целью таких систем является фиксация незаконного доступа на территорию учреждения. Стоит заметить, что такая система может быть не только автономной, но и работать в совокупности с другими охранными системами.

В настоящее время системы управления для большинства СКУД выполняются на основе микроконтроллеров имеющих проприетарную архитектуру, которая обязывает потребителя пользоваться оборудованием и услугами только одного производителя СКУД.

Существуют проекты по созданию микроконтроллеров, которые находятся под лицензией свободного программного обеспечения. Одним из таких проектов является Arduino – платформа, предназначенная для проектирования электронных устройств. Информация о характеристиках и устройстве платы находится в открытом доступе, что позволяет использовать её любому разработчику аппаратного обеспечения. Из-за возможности перепрограммировать плату, а также способности снизить материальные затраты на разработку было принято решение использовать её в качестве основы системы управления.



# 1 Состояние вопроса

## 1.1 Анализ исходных данных

Общие технические требования, классификацию и методы испытания средств и систем контроля и управления доступом регламентируются ГОСТ Р 51241-2008 [1].

Согласно ГОСТ, СКУД классифицируют по:

### 1. По способу управления:

- Автономные;

- Для управления одним или несколькими УПУ без передачи информации на центральное устройство управления и контроля со стороны оператора;

- Централизованные (сетевые) - для управления УПУ с обменом информацией с центральным пультом и контролем и управлением системой со стороны центрального устройства управления;

- Универсальные (сетевые) - включающие в себя функции как автономных, так и сетевых систем, работающие в сетевом режиме под управлением центрального устройства управления и переходящие в автономный режим при возникновении отказов в сетевом оборудовании, центральном устройстве или обрыве связи.

### 2. Число контролируемых точек доступа:

- Малой емкости (не более 64 точек);

- Средней емкости (от 64 до 256 точек);

- Большой емкости (более 256 точек);

### 3. Функциональным характеристикам

- С ограниченными функциями;

- С расширенными функциями;

- Многофункциональные;

### 4. Уровню защищенности системы от несанкционированного доступа к информации:

- Нормальная;
- Повышенная;
- Высокая;

В итоге СКУД можно представить, как совокупность:

- Считывающего устройства: предназначенного для считывания идентификационных признаков.
- Управляемых преграждающих устройств (УПУ): устройства, обеспечивающие физическое препятствие доступу.
- Исполнительного устройства: обеспечивающие работу УПУ.
- Систем управления: программные и аппаратные средства, обеспечивающие прием и обработку информации с носителей, установление режимов доступа, идентификацию и аутентификацию, управление УПУ.

Рассмотрим что должны обеспечивать автономные СКУД:

- Выдачу сигнала на открывание УПУ при считывании зарегистрированного в памяти системы идентификационного признака;
- Запрет открывания УПУ при считывании незарегистрированного в памяти системы идентификационного признака;
- Запись идентификационных признаков в память системы;
- Защиту от несанкционированного доступа при записи кодов идентификационных признаков в память системы;
- Сохранение идентификационных признаков в памяти системы при отказе и отключении электропитания;
- Ручное, полуавтоматическое или автоматическое открывание УПУ для прохода при аварийных ситуациях, пожаре, технических неисправностях в соответствии с правилами установленного режима и правилами противопожарной безопасности;
- Автоматическое формирование сигнала закрытия на УПУ при отсутствии факта прохода;

СКУД с централизованным управлением и универсальные должны соответствовать общим функциональным требованиям для автономных систем и дополнительно обеспечивать:

- Работу в локальной сети контроллеров СКУД;
- Регистрацию и протоколирование тревожных и текущих событий;
- Приоритетное отображение на экране управляющего компьютера тревожных событий;
- Управление работой УПУ в точках доступа по командам оператора;
- Задание временных режимов действия идентификаторов в точках доступа и уровней доступа;
- Защиту технических и программных средств от несанкционированного доступа к элементам управления, к установке режимов и к информации;
- Автоматический контроль исправности средств, входящих в систему, и линий передачи информации;
- Возможность автономной работы контроллеров системы с сохранением контроллерами основных функций при отказе связи с пунктом централизованного управления;
- Установку режима свободного доступа с пункта управления при аварийных ситуациях и чрезвычайных происшествиях (пожар, землетрясение, взрыв и т.п.);
- Блокировку прохода по точкам доступа командой с пункта управления в случае нападения;
- Возможность подключения дополнительных средств специального контроля, средств досмотра;

## **1.2 Анализ известных решений**

Одними из самых популярных производителей СКУД являются такие компании, как BOLID и PERCo.

## СКУД производства компании BOLID

Рассмотрим СКУД компании BOLID на базе интегрированной системы охраны “Орион”.

Система имеет следующие возможности:

- Запрет повторного прохода - используется для того, чтобы одним идентификатором нельзя было воспользоваться повторно для входа в какую-либо зону доступа, предварительно не выйдя из неё;
- Стандартный режим прохода - каждой точки доступа на предприятии, подлежащей контролю, устанавливается контроллер доступа и считывающие устройства. Для того чтобы сотрудники имели возможность прохода через точки доступа, каждому из них выдаётся уникальный идентификатор пользователя;
- Организация учета рабочего времени – сохранение и анализ. Контроль места нахождения сотрудника на объекте с точностью до зоны доступа;

Важным достоинством системы является универсальный контроллер С-2000-2, который можно использовать для обслуживания: турникетов, дверей, шлагбаумов. Модули системы взаимодействуют по сети Ethernet [2].

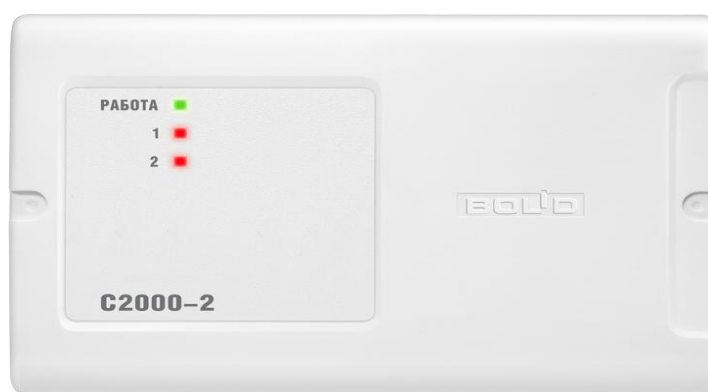


Рисунок 1 - Контроллер С-2000-2

Цена контроллера на 19.05.2020 составляет 3 576 руб. за штуку. Но расходы резко возрастают из-за необходимости использования ПО данной компании, цена которого начинается от 53 тыс. руб.

СКУД от компании PERCo

СКУД от компании PERCo имеют большой ряд преимуществ: учёт рабочего времени сотрудников, различные способы идентификации посетителей, интеграция с различными сервисами, наличие базового пакета ПО [3].

Также у компании есть система PERCo-S-20 для общеобразовательных учреждений, которые позволяют с помощью SMS-сообщения уведомлять родителей о времени прихода и ухода ребенка [4]. У данной системы есть интеграция с различными сервисами, например “Электронный дневник”, в котором сохраняются данные о перемещении ребенка.



Рисунок 2 - Контроллер CL-15

Контроллер CL-15 со встроенным считывателем предназначен для управления одним замком или контроля доступа через турникет в одном направлении прохода. Для контроля прохода в двух направлениях необходимо оборудовать турникет двумя контроллерами CL15.

Одна из самых популярных СКУД данной компании PERCo-Web,

она решает такие задачи как:

- Автоматизация учета рабочего времени;
- Разграничение прав доступа сотрудников и посетителей;
- Контроль нарушений трудовой дисциплины;
- Автоматизация выдачи и сбора гостевых пропусков;

К существенным недостаткам можно отнести высокую стоимость комплексов систем, оборудования и их обслуживание. К примеру, контроллер CL-15 с встроенным считывателем для PERCo-S-20 стоит 338 евро на 19.05.2020.

### **1.3 Выводы по разделу**

В данном разделе были рассмотрены виды систем управления и контроля доступом, а также исследованы самые известные устройства на рынке. В результате исследования существующих аналогов было обнаружено, что современные системы имеют такие недостатки, как: дорогое оборудование, высокая стоимость базового ПО, хранение информации в базе данных производителя и сложность в использовании компонентов систем разных компании. Отсюда следует, что нужно подобрать элементы, выбрать способ хранения большого объема информации, создать устройство и ПО, которые не уступают по базовому функционалу выше рассмотренных систем.

## **2 Формулирование цели и задач проекта**

Целью работы является разработка прототипа устройства и необходимого программного обеспечения, предназначенного для идентификации посетителей образовательного учреждения. Основные преимущества устройства будут заключаться в низкой стоимости компонентов и в хранении данных внутри ТГУ, что позволит исключить утечку информации. Для хранения и проверки информации предполагается использование базы данных, для вывода информации о посетителе будет использоваться экран персонального компьютера, для считывания кода карты - RFID модуль.

Для выполнения цели работы необходимо выполнить следующие задачи:

1. Обзор существующих систем контроля и управления доступом
2. Разработка структурной и электрической принципиальной схемы
3. Разработка программ для микроконтроллера и ПК
4. Разработка корпуса
5. Разработка прототипа устройства

### **2.1 Выводы по разделу**

В данном разделе на основе полученного задания на выполнение бакалаврской работы и проведенного анализа известных решений в первом разделе были сформулированы цель и задача бакалаврской работы. Также были обозначены требования, которые предоставят конкурентное преимущество перед рассмотренными аналогами.

### 3 Аппаратная часть

#### 3.1 Первичная структурная схема

В соответствии с поставленными задачами и выдвинутыми требованиями, разработана первичная структурная схема системы, представленная на рисунке 3.



Рисунок 3 - Первичная структурная схема

Схема представляет собой совокупность блоков:

- Контроля и управления – будет принимать данные от блока ввода и отправлять их с помощью блока передачи данных для проверки в базе данных; в зависимости от результата проверки значения подавать сигнал блокам индикации и исполнительных устройств;
- Ввода – предназначен для считывания кода карты и управления блоком исполнительных устройств;



- Хранения данных – база данных с записями о посетителях;
- Исполнительных устройств – содержит в себе устройства, которые будут осуществлять пропуск или не допуск посетителя;
- Индикации – визуальный сигнал, сообщающий об удачной или неудачной проверке;
- Передачи данных – необходим для передачи данных от устройства к ПК и обратно с помощью Wi-Fi;
- Вывода и проверки данных – в зависимости от результата проверки будет выводить информацию на экран ПК. Например, это может ФИО и фото найденного посетителя или сообщение о том, что карты с таким ID нет в базе данных.

### **3.2 Подбор компонентной базы устройства**

#### **RFID считыватель и RFID метка**

Технология RFID – это технология бесконтактного обмена данными, основанная на использовании радиочастотного электромагнитного излучения. RFID применяется для автоматической идентификации и учета объектов. Типовая RFID-система состоит из 3-х базовых компонентов: RFID-меток, RFID-считывателей и программного обеспечения [5].

Существуют различные типы считывателей, в данной системе используется порталный RFID-считыватель, который предназначен для регистрации RFID-меток в контролируемых проходах.

RFID-RC522 высоко интегрированный считыватель для бесконтактной коммуникации. Считыватель поддерживает интерфейсы SPI, UART и I2C через которые происходит обмен данными с другими приборами [6]. На плате модуля на выводах микросхемы выбран интерфейс SPI. Основа модуля – микросхема MFRC522.

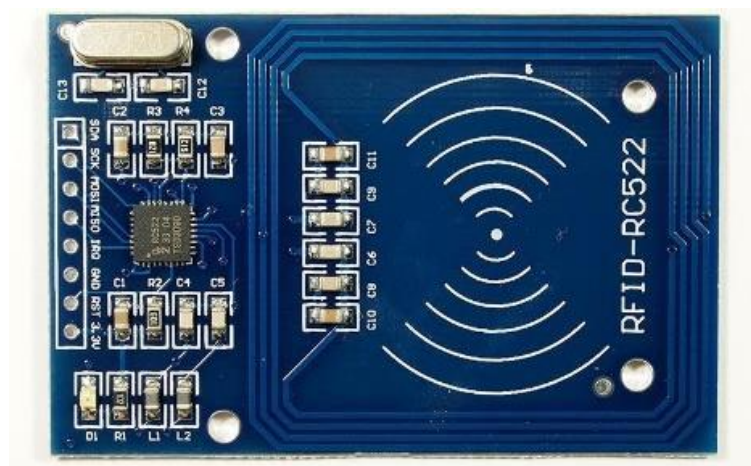


Рисунок 4 - Считыватель RFID-RC522

Считыватель RFID RC522 срабатывает при поднесении метки. Обмен данными происходит через рамочные антенны, находящиеся в карточке (метке) и в модуле. Сигнал модуля служит источником энергии для метки. Он может обрабатывать информацию одновременно от нескольких меток.

Технические характеристики:

1. Напряжение питания: 3,3 В;
2. Ток потребления в режимах:
  - 2.1 Дежурный 80 мкА;
  - 2.2 Ожидания 12 мА;
  - 2.3 Наибольший 30 мА;
3. Рабочая частота: 13,56 мГц;
4. Расстояние считывания 0-25мм;
5. Скорость передачи информации:
  - 5.1 106 кбит/с;
  - 5.2 212 кбит/с;
  - 5.3 424 кбит/с;
  - 5.4 848 кбит/с;

Данному считывателю соответствует RFID-метка в корпусе или в виде ключ-карты.



Рисунок 5 - RFID-метки в корпусах

Брелоки и карты работают на той же частоте, что и считыватели. Внутри них находятся антенна и микросхема Mifare S50, содержащая память, размер которой 1 килобайт, тип EEPROM. Уникальность карточки Mifare обеспечивается присвоением изготовителем номера, который используется в качестве индикатора.

#### Элементы индикации

Для индикации выбраны следующие элементы:

1. Светодиод красный;
2. Светодиод зеленый;

Светодиод – полупроводниковый прибор, создающий оптическое излучение при пропускании через него электрического тока в прямом направлении.

Светодиоды 5 мм F53GD2YG-1 красный и зелёный.



Рисунок 6 - Светодиоды

В разрабатываемой системе используются для сигнала о предоставлении/запрете доступа. Если доступ закрыт загорается красный светодиод, если доступ открыт – зелёный.

Технические характеристики:

1. Красный:

1.1. Рабочее напряжение: от 1,6 до 2,03 В;

1.2. Длина волны: 610-760 нм;

1.3. Прямой ток: 30 мА;

2. Зелёный:

2.1. Рабочее напряжение: от 2,1 до 2,3 В;

2.2. Длина волны: 500-570 нм;

2.3. Прямой ток: 30 мА;

### Аппаратная платформа

Из большого выбора плат Arduino стоит выделить две модели, которые могут подойти в качестве устройства управления:

- Arduino Nano
- NodeMCU

Arduino Nano – это платформа, построенная на основе микроконтроллера ATmega328. Arduino Nano имеет стабилизатор питания на 5.0 В, встроенный программатор с разъёмом USB Micro. Диапазон её предельных входных напряжений это 6-20 В, а диапазон рекомендуемого входного напряжения 7-12 В. Она имеет 14 цифровых входов/выходов (6 из которых могут использоваться как ШИМ) и 8 аналоговых входов. Габаритные размеры печатной платы составляют 18.5 x 4.2 мм [7]. Чтобы Arduino Nano начала функционировать, её необходимо запитать одним из двух возможных способов, а именно:

- через USB-кабель при подключении к компьютеру или другому источнику питания 5В.

- при помощи внешнего стабилизированного источника питания, напряжение которого должно лежать в диапазоне 6 – 20В. Данное напряжение подаётся непосредственно на вход VIN платы Arduino Nano.

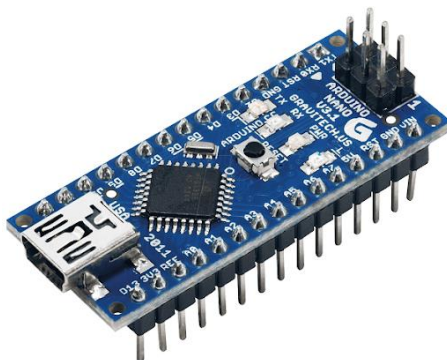


Рисунок 7. Arduino Nano

Технические характеристики:

- Напряжение питания 5В;
- Входное питание 7-12В (рекомендованное);
- Количество цифровых пинов – 14, из них 6 могут использоваться в качестве выходов ШИМ;
- 8 аналоговых входов;
- Максимальный ток цифрового выхода 40 мА;
- Флэш- память 32 Кб;
- ОЗУ 2 Кб;
- EEPROM 1 Кб;
- Частота 16 МГц;
- Размеры 18.5 см x 4.2 см

NodeMCU - это платформа на основе модуля ESP8266. Плата практически имеет схожие характеристики как у Arduino Nano, но функционал этой платформы шире благодаря встроенному Wi-Fi модулю. Wi-Fi модуль можно использовать для управления различными схемами на расстоянии посредством передачи сигнала в локальную сеть или интернет.

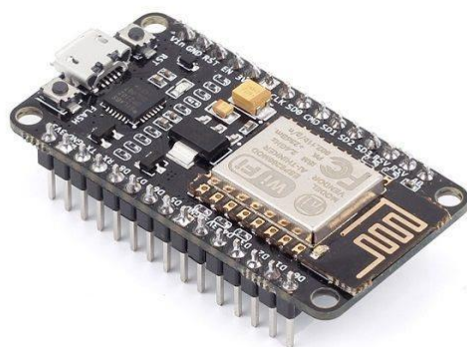


Рисунок 8 - Аппаратная платформа NodeMCU

На лицевой части платы расположен разъем Micro USB, с помощью которого в контроллер записывают программу. Также через этот разъем можно подавать питание. Рядом с разъемом располагаются две кнопки: Flash и Reset. Кнопка Flash используется для отладки, кнопка Reset для перезагрузки платы.

Технические характеристики модуля:

- Поддерживает Wi-Fi протокол 802.11 b/g/n;
- Поддерживаемые режимы Wi-Fi – точка доступа, клиент;
- Входное напряжение 3,7В – 20 В;
- Рабочее напряжение 3В - 3,6В;
- Максимальный ток 220мА;
- Встроенный стек TCP/IP;
- Диапазон рабочих температур от -40С до 125С;
- Частота 80 МГц, 32-битный процессор;
- Время пробуждения и отправки пакетов 22мс;
- Встроенные TR переключатель и PLL;
- Наличие усилителей мощности, регуляторов, систем управления питанием.

В качестве управляющего устройства были рассмотрены именно платы Arduino Nano и NodeMCU, так как данные платы имеют компактные размеры

и удовлетворяют требуемому функционалу. Каждая из плат имеет достаточное количество выводов для подключения модулей. В итоге была выбрана плата NodeMCU, так как с помощью встроенного модуля Wi-Fi можно передавать и принимать данные по сети. В качестве основного способа передачи данных и будет выбран модуль Wi-Fi.

### Питание устройства

Для питания устройства с условием его автономности необходимо добавить аккумуляторную батарею. Для этого было решено использовать Li-ion аккумулятор Samsung ICR18650-22E [8].



Рисунок 9 - Аккумулятор Samsung ICR18650-22E

Технические характеристики:

- Минимальная ёмкость - 2200 mAh;
- Максимальное напряжение - В 4,2;
- Минимальное напряжение - В 2,75;
- Номинальное напряжение - 3.6 В;

Для заряда аккумулятора был выбран модуль HW-357, который является улучшенной версией модуля TP4065. Модуль имеет защиту аккумулятора от перезаряда, перегрузки и короткого замыкания,

обеспечивает полноценную зарядку Li-ion аккумуляторных батарей с возможностью подсоединения нагрузки и с защитой от сгорания и автоматически завершает цикл зарядки при достижении напряжения на нем 4.2 В и снижении тока заряда до 1/10 от запрограммированной величины. Преимущество HW-357 над TP4065 в том, что в плату встроен DC-DC преобразователь, на выходе которого можно получить от 4.3В до 27В.

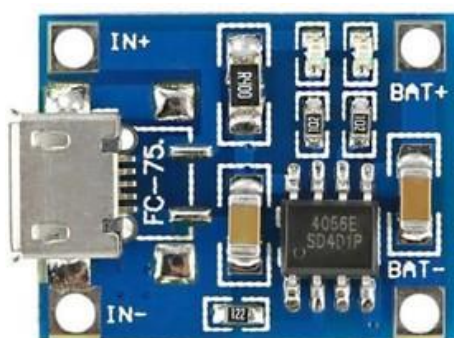


Рисунок 10 - Модуль TP4065

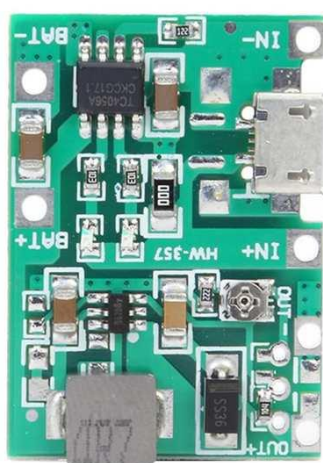


Рисунок 11 - Модуль HW-357

Технические характеристики HW-357:

- Режим зарядки: линейная 1%;
- Ток зарядки: до 1А (настраивается);
- Точность зарядки: 1.5%;
- Входное напряжение: 4.5 - 5.5В;



- Напряжение полного заряда: 4.2В;
- Индикаторы: красный — зарядка, зеленый — заряд окончен;
- Входной разъем: mini USB или контакты для подпайки проводов;
- Диапазон температур: -10 до +85 град.С;
- Защита от переплюсовки: нет;
- Защита от перезаряда:  $4.30 \pm 0.050$  В;
- Защита от переразряда:  $2.40 \pm 0.100$  В;
- Вес: 5 г;
- Размеры платы:  $25 \times 17 \times 4$  mm.

### Управление исполнительным устройством

Рассмотрим два типа реле:

- Твердотельное
- Электромагнитное

Твердотельное реле Omron G3MB-202P.



Рисунок 12 - Модуль реле Omron G3MB-202P

Основной задачей твердотельных реле является – включение цепи управления токами большой нагрузочной величины с помощью симисторов, транзисторов и тиристоров. Принцип работы твердотельных реле: управляющий сигнал подается на светодиод, который при этом передает

сигнал далее – на фотодиодную матрицу, при этом обеспечивается гальваническая развязка коммутируемых и управляющих цепей. Напряжение, которое при этом создается, управляется силовым ключом. Модуль предназначен для замыкания выходной цепи переменного тока (с номинальным напряжением до 240 В), при подаче управляющего сигнала 5 В постоянного тока на вход модуля. Средняя цена модуля 160 руб [11].

Достоинства твердотельного реле:

- Отсутствие электромагнитных помех при включении цепи;
- Высокий уровень быстродействия;
- Контакты не шумят и не дребезжат;
- Гарантированность срабатываний на протяжении большого промежутка времени;
- Входные и коммутируемые цепи надежно заизолированы друг от друга;

Технические характеристики модуля Omron G3MB-202P:

- Напряжение срабатывания на входе (S):  $> 4$  В (номинально 5 В);
- Максимальное напряжение на входе (S):  $< 6$  В;
- Максимальный ток, на входе модуля:  $< 12$  мА, при  $V_{cc} = 5$  В;
- Номинальное напряжение нагрузки (AC): 120 ... 240 В, переменного тока 50/60 Гц;

В качестве аналога Omron G3MB-202P рассмотрим электромагнитный релейный модуль SONGLE SRD-05VDC.



Рисунок 13 - Модуль реле SONGLE SRD-05VDC

Напряжение питания модуля 5В. Управляющий сигнал подается от микроконтроллера. Для переключения контактов реле достаточно подать 3 – 5В, так как управление производится через встроенный ключ. Реле срабатывает при подаче на вход модуля логической единицы. Модуль имеет один нормально замкнутый и один нормально разомкнутый контакт реле. Модуль обеспечивает коммутацию переменного тока с напряжением 220В и постоянного тока до 30В. На плате модуля расположен светодиодный индикатор, сигнализирующий включение реле. Модуль имеет разъем с тремя выводами, два из которых для подачи напряжения питания 5В и один логический пин, который подключается непосредственно к микроконтроллеру [8]. Электромагнитное реле не имеет таких же достоинств как твердотельное, но у электромагнитного реле меньше выделение тепла в рабочем режиме. Средняя цена модуля 70 руб [12].

Технические характеристики SONGLE SRD-05VDC:

- Номинальное рабочее напряжение катушки: 5В;
- Номинальный рабочий ток катушки: 71 мА;
- Сопротивление катушки: 70 Ом;
- Потребляемая мощность: 0,36 Вт;
- Максимальный коммутируемый ток: 10А
- Максимальное коммутируемое напряжение (постоянное): 28 В;

- Максимальное коммутируемое напряжение (переменное): 250 В
- Рабочая температура: -25°C до +70°C;

Оба модуля предоставляют необходимый для разрабатываемого устройства функционал и подходят по характеристикам. Выбор был сделан в пользу электромагнитного реле, так как оно имеет меньшую цену.

### 3.3 Полная структурная схема

После сравнения и выбора компонентов для управляющего устройства была составлена полная структурная схема устройства

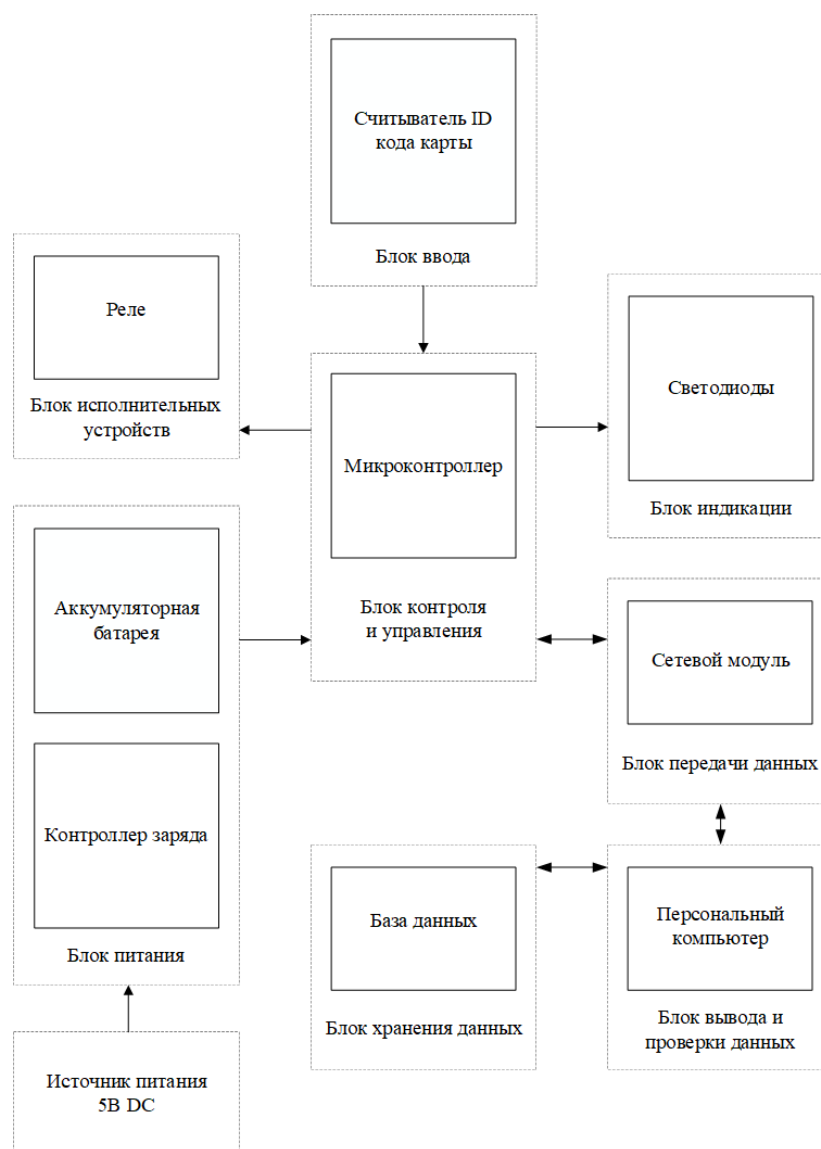


Рисунок 14 - Полная структурная схема

### 3.4 Разработка электрической принципиальной схемы устройства

Устройство на базе платформы NodeMCU состоит из следующих блоков:

- Считыватель RFID-RC522
- Светодиоды
- Аппаратная платформа NodeMCU
- Аккумулятор Samsung ICR18650-26F
- Модуль заряда HP-357
- Модуль реле SONGLE SRD-05VDC

Аппаратная плата NodeMCU

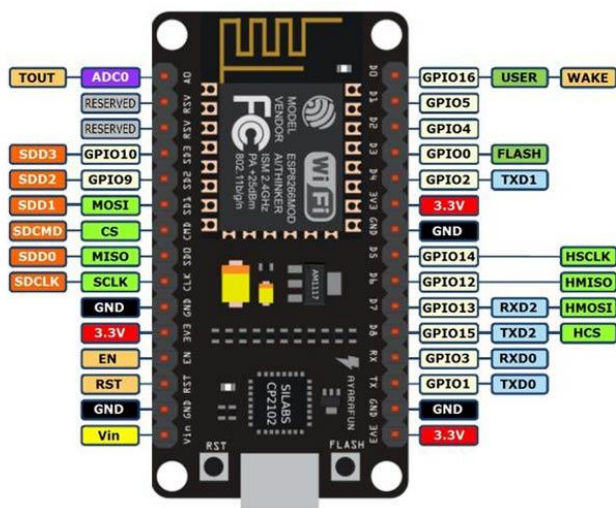


Рисунок 15 - Распиновка аппаратной платы NodeMCU

Вывод Vin предназначен для подключения внешнего источника питания 5V. Стабилизатор AMS1117-3.3 позволяет подавать питание на Vin в широком диапазоне от 5 до 10 V. Выходное напряжение стабилизатора от 3.23 до 3.36 В.



Рисунок 16 - Стабилизатор напряжения AMS1117-3.3

3.3V - контакт выходного напряжения внутрисхемного стабилизатора. Может быть использован для питания подключаемых к плате датчиков. Суммарная максимальная нагрузка всех выводов 3.3V не должна превышать 300мА. Выводы GPIO - контакты общего назначения для ввода/вывода данных. Могут быть сконфигурированы как входы или выходы и назначены в программе на выполнение различных функций. Вывод RST используется для сброса микроконтроллера. Вывод EN - при подаче на вывод сигнала высокого уровня, микроконтроллер ESP8266 переходит в рабочий режим, при сигнале низкого уровня - в режим энергосбережения. Вывод WAKE используется для пробуждения чипа ESP8266 из режима глубокого сна.

ADC0/TOUT - вывод встроенного 10-разрядного аналого-цифрового преобразователя (АЦП). Преобразованные значения лежат в интервале 0-1023. Плата разработки NodeMCU V2 имеет внутренний делитель напряжения, входной диапазон АЦП составляет 0 - 3,3 В. UART - асинхронный последовательный интерфейс устанавливает связь с другими устройствами по шине UART. SPI - последовательный периферийный интерфейс. NodeMCU имеет два SPI (SPI и HSPI) в ведущем и подчиненном режимах. SDIO - интерфейс безопасных цифровых входов/выходов, предназначен для коммутации с внешней флэш-памятью стандарта SD по последовательной шине. Reserved - зарезервированные

выводы. Кнопка Flash на NodeMCU подключает к земле GPIO0. Ее можно использовать как обычную кнопку. Если программно подтянуть вывод GPIO0 с помощью внутреннего подтягивающего резистора к высокому уровню, то появление низкого уровня на этом выводе будет означать, что кнопка нажата. Интерфейс I2C - последовательная асимметричная шина. I2C используется для подключения датчиков и периферийных устройств. NodeMCU не имеет аппаратных выводов I2C, но интерфейс можно реализовать программно [13].

Некоторые пины NodeMCU поддерживают программный ШИМ. ШИМ – широко импульсная модуляция, используется для получения аналогового сигнала с цифрового выхода микроконтроллера. NodeMCU имеет 4 ШИМ вывода – D2, D5, D6, D8.

#### Считыватель RFID-RC522

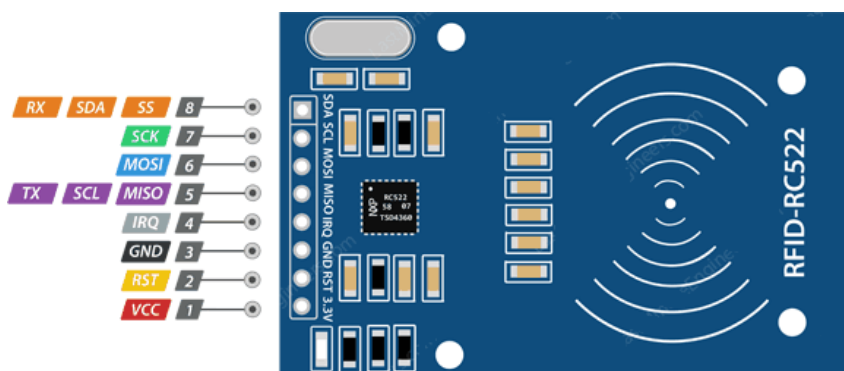


Рисунок 17 - Распиновка считывателя RC522

Питание для считывателя подается на пин VCC. Напряжение питания находится в диапазоне от 2,5 до 3,3 В. VCC следует подключить к пину 3v3, который находится на плате NodeMCU. Пин RST предназначен для сброса и отключения питания. При поступлении сигнала LOW (логический ноль), происходит перезагрузка считывателя. Пин RST нужно подключить к пину D3 платы NodeMCU.

Пин GND – вывод земли, подключается к выводу GND на NodeMCU. IRQ – вывод прерывания, предназначенный для пробуждения устройства,

когда поблизости оказывается RFID метка. Вывод MISO/SCL/TX – служит для коммуникации по протоколу SPI, I2C или UART. Вывод действует либо как MISO (Master In Slave Out) при включенном интерфейсе SPI, либо как последовательный тактовый сигнал при включенном интерфейсе I2C, либо как выход последовательных данных при работающем интерфейсе UART. MOSI (Master Out Slave In) - вход SPI для считывателя RC522. MOSI - служит для передачи данных от ведущего устройства ведомому, подключается к выводу D7 платы NodeMCU; а MISO - служит для передачи данных от ведомого устройства ведущему и подключается к пину D6.

Пин SCK принимает тактовые импульсы, предоставляемые мастером на шине SPI. Служит для синхронизации. Подключается к выводу платы D5. Вывод SS/SDA/Rx действует либо как вход, когда включен интерфейс SPI, либо как линия последовательных данных, когда включен интерфейс I2C, либо как вход последовательных данных, когда включен интерфейс UART. Этот вывод обычно помечается заключением в квадрат, чтобы его можно было использовать в качестве опорной точки для идентификации других выводов. Вывод SDA подключается к выводу платы D1 [14].

#### Модуль зарядки и аккумулятор

Подключение к зарядке через стандартный разъем microUSB (или miniUSB) или через дублирующие контакты «+» и «-». Аккумулятор (без встроенной защиты разряда) подключается к контактам «В+» и «В-». Нагрузка подключается к контактам «OUT+» и «OUT-» (при зарядке батареи нагрузку желательно отключать, если она не рассчитана на напряжение 4.2 В).

#### Светодиоды

При подключении светодиодов необходимо соблюдать полярность. Короткая ножка светодиода является катодом, длинная анодом. Также у светодиода ограничение по протекающему току, следовательно нужно



подобрать резисторы для каждого типа светодиодов. Сопротивление резистора можно высчитать по формуле:

$$I = \frac{U}{R} \quad (1)$$

Рассчитаем напряжение для красного и зеленого светодиода. Возьмем рабочее напряжение красного светодиода 1.6В и зеленого светодиода 2.1В. Потребляемый ток светодиодов равен 30мА. Выходное напряжение платы NodeMCU равно 3.3В.

Расчет сопротивления для красного светодиода:

$$R = \frac{3.3 - 1.6}{0.03} = 56.6 \text{ Ом}$$

Расчет сопротивления для зеленого светодиода:

$$R = \frac{3.3 - 2.1}{0.03} = 40 \text{ Ом}$$

### Электромагнитное реле

Для управления исполнительным устройством использовано реле. Питание модуля составляет 5В, управляющий сигнал подается с микроконтроллера. Вывод IN предназначен для управления замыканием и размыканием цепи; VCC для питания модуля; GND для заземления. Вывод IN соединяем с выводом D2. Для питания используется блок питания на 5В [15].

Проанализировав все выбранные компоненты устройства и способы их подключения между собой, была составлена электрическая принципиальная схема устройства.

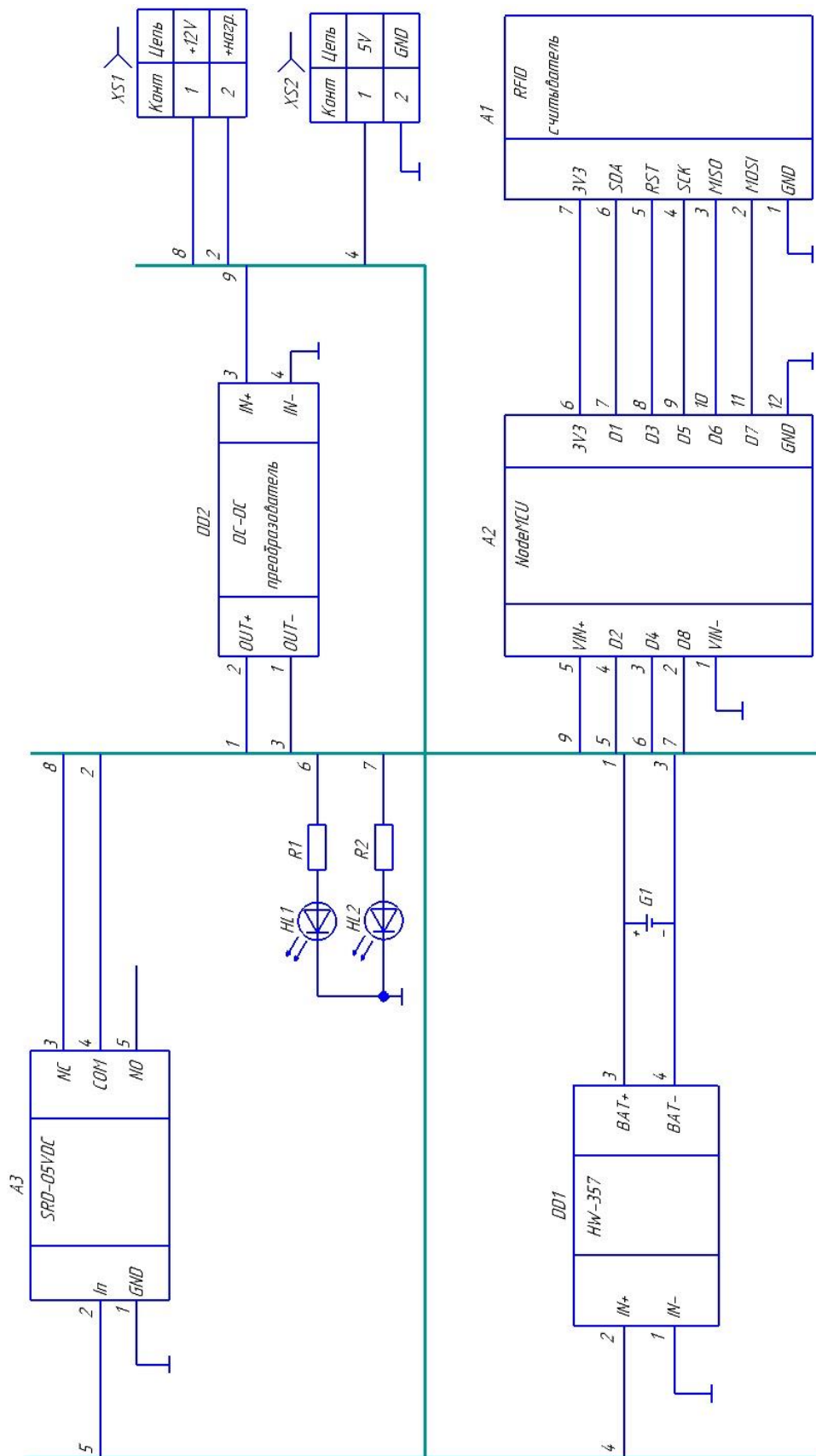


Рисунок 18 - Электрическая принципиальная схема устройства

<i>Поз</i>	<i>Обозначение</i>	<i>Наименование</i>	<i>Кол</i>	<i>Примечание</i>
		<i>Документация</i>		
	<i>ЦУ 01.30.000 СЭ</i>	<i>Схема электрическая принципиальная</i>	<i>1</i>	
		<i>Детали</i>		
<i>1</i>	<i>ЦУ 01.30.001</i>	<i>Модуль RFID-считыватель</i>	<i>1</i>	<i>A1</i>
<i>2</i>	<i>ЦУ 01.30.001</i>	<i>Модуль NodeMCU</i>	<i>1</i>	<i>A2</i>
<i>3</i>	<i>ЦУ 01.30.001</i>	<i>Модуль SRD-05VDC</i>	<i>1</i>	<i>A3</i>
<i>4</i>	<i>ЦУ 01.30.001</i>	<i>Плата HW-357</i>	<i>1</i>	<i>DD1</i>
<i>5</i>	<i>ЦУ 01.30.001</i>	<i>Плата DC-DC преобразователь</i>	<i>1</i>	<i>DD2</i>
<i>6</i>	<i>ЦУ 01.30.001</i>	<i>Аккумулятор Samsung ICR18650-22E</i>	<i>1</i>	<i>G1</i>
		<i>Прочие изделия</i>		
		<i>Резисторы</i>		
<i>7</i>		<i>С1-4-0,25-56 Ом ±5% ШКАБ.434110.005 ТУ</i>	<i>1</i>	<i>R1</i>
<i>8</i>		<i>С1-4-0,25-39 Ом ±5% ШКАБ.434110.005 ТУ</i>	<i>1</i>	<i>R2</i>
		<i>Светодиоды</i>		
<i>9</i>		<i>F53GD2YG-1 TS 62504-2011</i>	<i>2</i>	<i>HL1, HL2</i>

Рисунок 19 - Спецификация

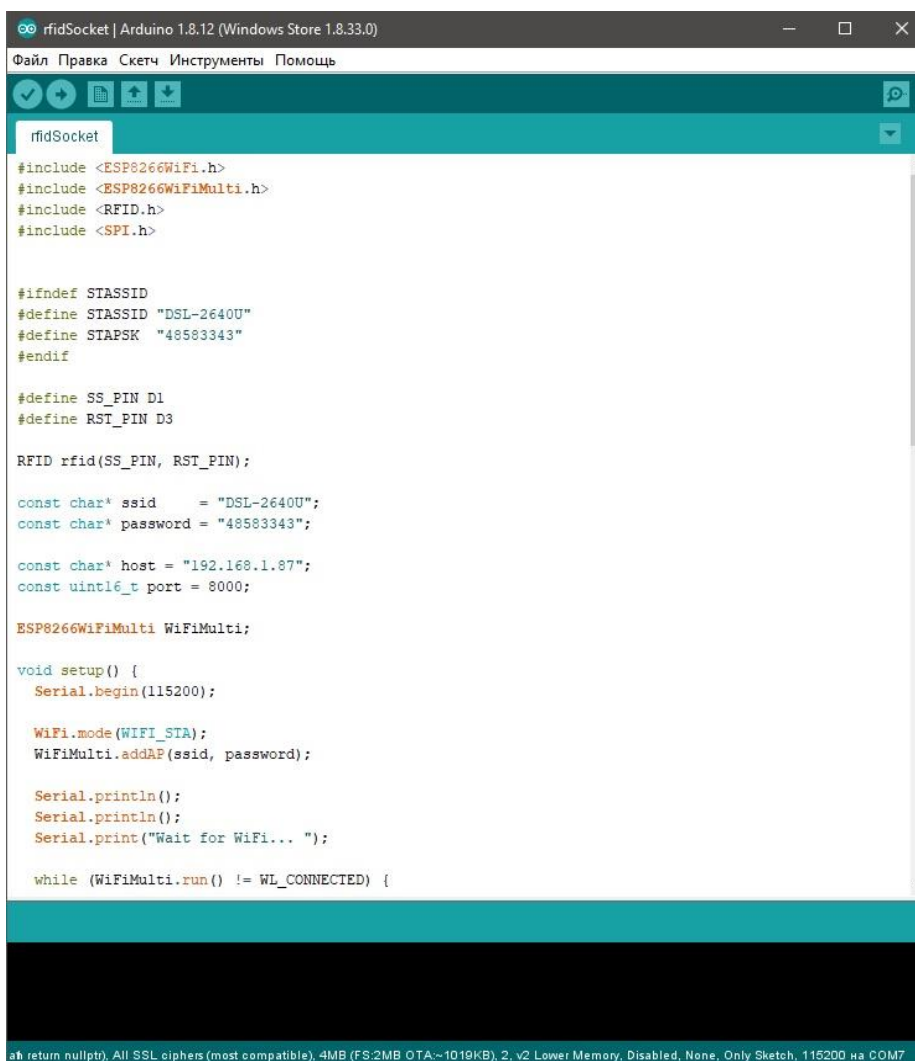
### **3.5 Выводы по разделу**

В данном разделе в соответствии с поставленными требованиями была создана первичная структурная схема устройства, после чего подобраны компоненты, имеющие необходимый для разрабатываемого устройства функционал. На основе выбранных компонентов была разработана полная структурная схема устройства, проанализированы способы подключения выбранных модулей. В результате была создана электрическая принципиальная схема устройства, которая обеспечивает возможность дальнейшей разработки алгоритма работы устройства и реализации программного обеспечения.

## 4 Программная часть

### 4.1 Написание программы для микроконтроллера

Для написания программной части было принято решение использовать интегрированную среду разработки Arduino IDE.



```
rfidSocket | Arduino 1.8.12 (Windows Store 1.8.33.0)
Файл Правка Скетч Инструменты Помощь

rfidSocket

#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <RFID.h>
#include <SPI.h>

#ifndef STASSID
#define STASSID "DSL-2640U"
#define STAPSK "48583343"
#endif

#define SS_PIN D1
#define RST_PIN D3

RFID rfid(SS_PIN, RST_PIN);

const char* ssid = "DSL-2640U";
const char* password = "48583343";

const char* host = "192.168.1.87";
const uint16_t port = 8000;

ESP8266WiFiMulti WiFiMulti;

void setup() {
  Serial.begin(115200);

  WiFi.mode(WIFI_STA);
  WiFiMulti.addAP(ssid, password);

  Serial.println();
  Serial.println();
  Serial.print("Wait for WiFi... ");

  while (WiFiMulti.run() != WL_CONNECTED) {
```

Рисунок 20 - Интерфейс Arduino IDE.

Данная программа является бесплатной и кроссплатформенной. Разработка ПО велась на языке C++, который является стандартным для микроконтроллеров Arduino [16].

Для микроконтроллера требуется написать программу, которая будет реагировать на поднесенную к модулю RFID карту, содержащую уникальный 12-значный код. Отправлять полученный код на ПК микроконтроллер будет

с помощью модуля WIFI, который будет подключен к роутеру. Связь между микроконтроллером и ПК будет установлена с помощью WinSock. Для этого необходимо будет написать отдельную программу для ПК. После того, как будет проведена проверка кода карты микроконтроллер получит ответ от ПК, от которого будет зависеть можно ли замыкать реле или нет. Также будет загораться красный или зеленый светодиод и подаваться звуковой сигнал в зависимости от результата идентификации. После проверки также будет выведена соответствующая информация на экране ПК. Это будет фото и короткая информация о посетителе или предупреждающее сообщение, что данного человека нет в базе данных.



Рисунок 21 - Блок схема алгоритма

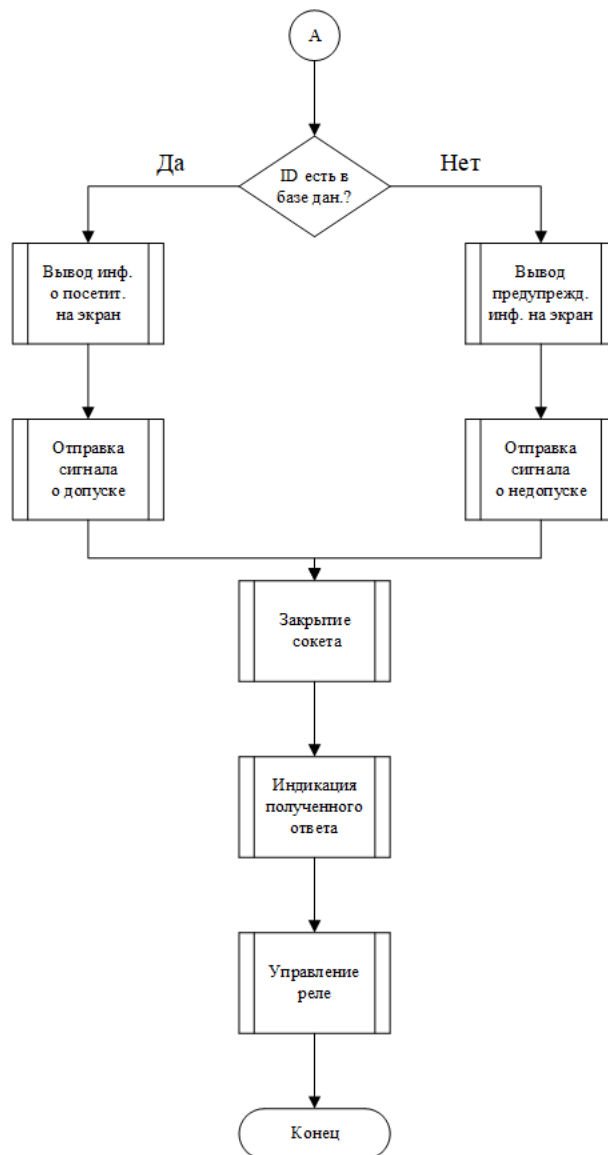


Рисунок 22 - Продолжение блок схемы алгоритма

Чтобы написать программу, необходимо будет подключить библиотеки для упрощения написания кода. Библиотеки, используемые в программе:

```
#include <ESP8266WiFi.h>
```

```
#include <ESP8266WiFiMulti.h>
```

Данные заголовочные файлы содержат все необходимые функции, которые содержат в себе библиотеки для работы сетевого модуля WIFI.

Для соединения RFID модуля и Arduino подключим следующую библиотеку:

```
#include <RFID.h>
```

За создание экземпляра отвечает следующая функция из библиотеки, где аргументами являются пины, к которым подключен модуль:

RFID rfid(SS\_PIN, RST\_PIN)

Инициализация модуля RFID происходит при помощи функции rfid.init().

Для передачи данных между модулем RFID и Arduino используем библиотеку:

```
#include <SPI.h>
```

SPI – последовательный протокол синхронной передачи данных, используемый микроконтроллерами для связи с периферийными устройствами на коротких расстояниях. При использовании SPI всегда есть одно ведущее устройство, которое управляет периферийными устройствами [17]. Инициализация SPI происходит при помощи функции SPI.begin().

После того, как подключили все необходимые библиотеки, можно приступать к написанию программы. Написание программы было разбито на части для удобства.

Для начала напишем код, который будет отвечать за соединение сетевого модуля WIFI непосредственно к роутеру. Чтобы подключить сетевой модуль к роутеру, зададим имя и пароль от точки доступа WIFI в программе и используем функции подключения:

```
const char* ssid = "name"  
const char* password = "password"  
WiFiMulti.addAP(ssid, password)  
WiFiMulti.run()
```

После того, как установилось соединение, микроконтроллер должен ждать поднесения карты к RFID модулю и считать данные с карточки. Для этого используем функции rfid.isCard() и rfid.readCardSerial().

Соединение с ПК и отправка ID карты осуществляется при помощи:

1. client.connect(host, port) – где host является адрес ПК, на котором будет запущена программа с базой данных, а port – порт, на котором эта программа запущена.



2. `client.println(rfid)` – где `rfid` является строковой переменной, содержащая ID карты.

Для управления блоками индикации и исполнительных устройств, зададим режим работы с помощью функции `pinMode`, которая принимает два аргумента: номер пина, к которому подключено устройство и режим работы. Для всех блоков зададим режим работы `OUTPUT`. В ходе программы можно управлять блоками функцией `digitalWrite`, принимающей номер пина и тип управляющего сигнала (логическая единица и логический ноль). Например, чтобы включить светодиод, который подключен к 1 пину, нужно написать `digitalWrite(1, HIGH)`; чтобы выключить - `digitalWrite(1, LOW)`. Готовая программа находится в Приложении А.

## **4.2 Написание программы для ПК**

Для того чтобы обработать запрос с микроконтроллера, проверить ID карты в базе данных и отправить результат обратно – нужно написать программу для ПК на C++, при помощи WinAPI и WinSock. Было принято решение писать программу в Microsoft Visual Studio, так как в нем есть весь необходимый функционал и интерфейс относительно прост для освоения. Для написания программы было необходимо отдельно подключить в настройках проекта файлы библиотек для базы данных и выбрать многобайтовую кодировку проекта.

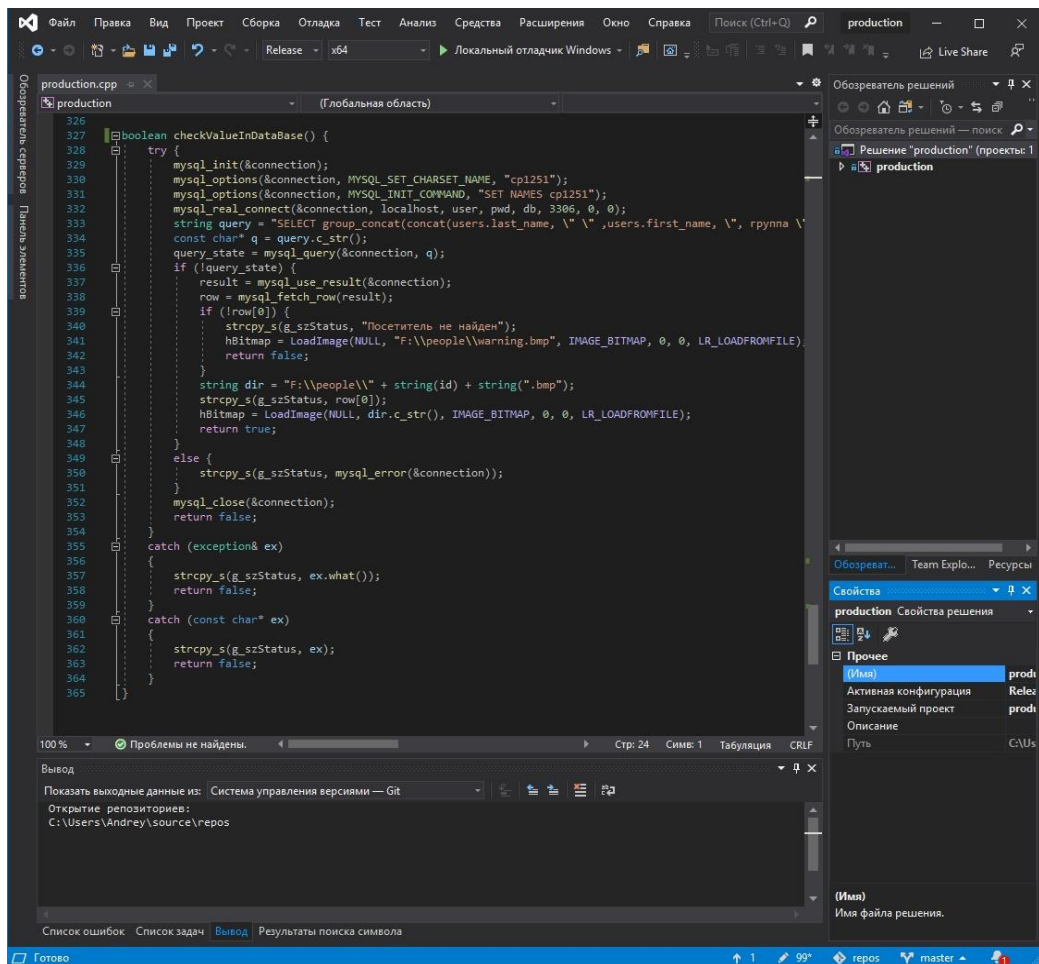


Рисунок 23 - Интерфейс Microsoft Visual Studio.

Для хранения информации базу данных MySQL, потому что уже был опыт в её использовании. MySQL поддерживает SQL (структурированный язык запросов) и может применяться в качестве SQL-сервера. Это означает, что общаться с сервером можно на языке SQL: клиент посылает серверу запрос, тот его обрабатывает и отдает клиенту только те данные, которые были получены в результате этого запроса [18]. Она имеет достаточно понятный интерфейс и практически не возникает трудностей в использовании у неопытных пользователей. Объем хранимой информации ограничивается только памятью компьютера. Информация была разбита на две таблицы, так как эта база данных является реляционной, то можно настроить связи между таблицами и создавать сложные запросы для получения данных. Пример такого запроса будет в Приложении Б.

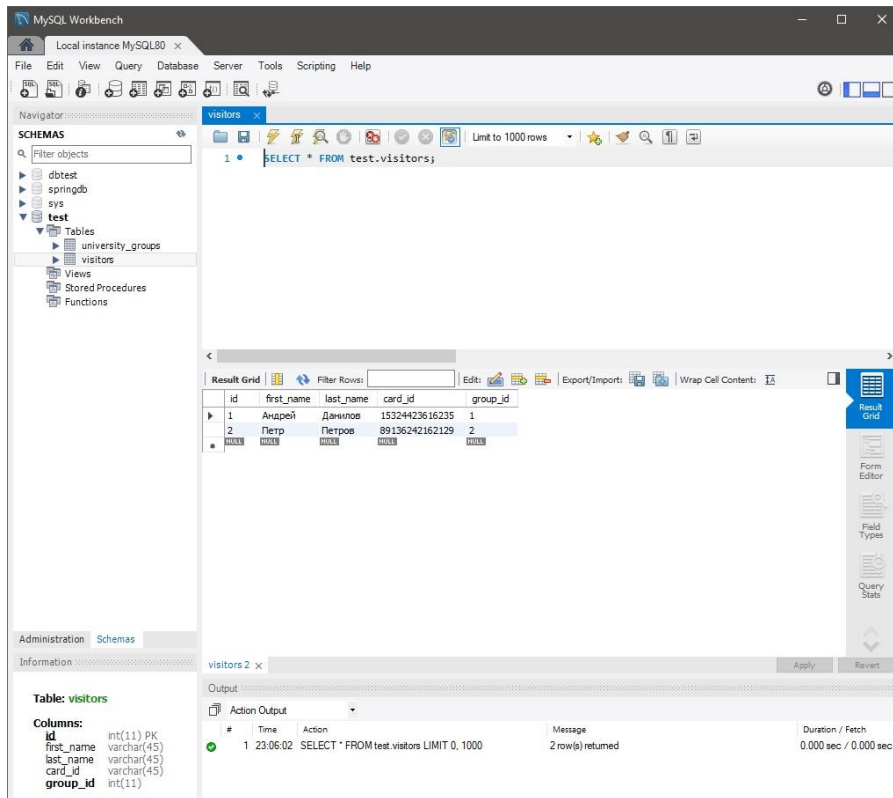


Рисунок 24 - Таблица с посетителями

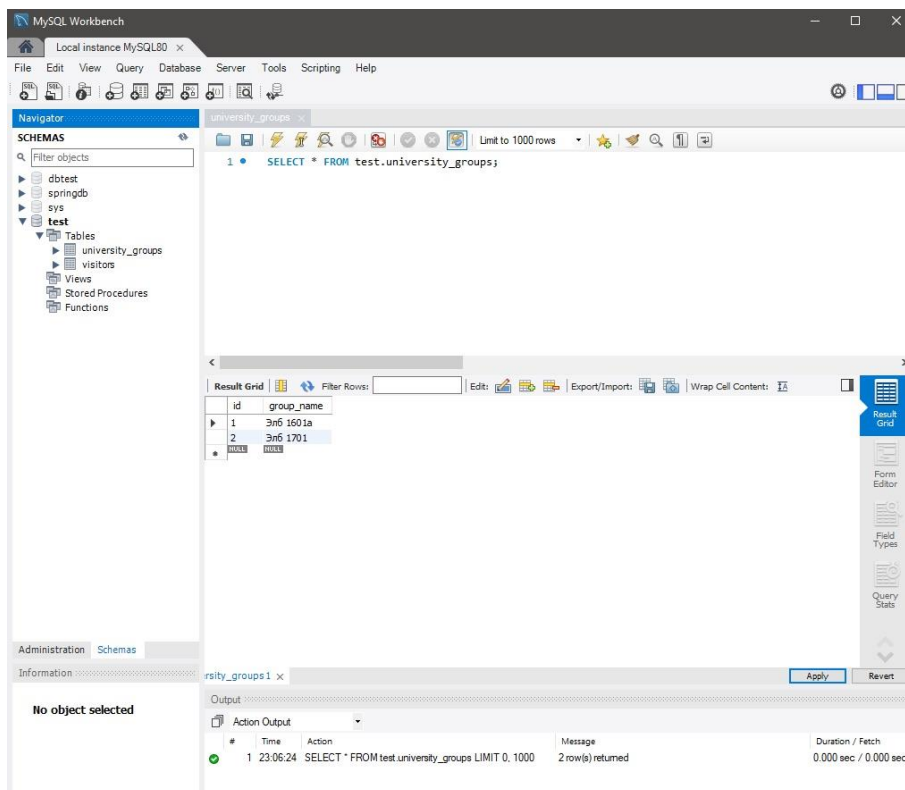


Рисунок 25 - Таблица с группами

WinAPI – набор функции, работающих под управлением ОС Windows. При помощи WinAPI можно создавать оконные процедуры, диалоговые окна, программы и т.д. Этот набор функции используем для создания оконной процедуры для вывода информации о посетителе на экран.

Windows-приложения являются программами, управляемые событиями. Мы не можем заранее знать последовательность вызовов функции, так как она определяется в ходе выполнения кода. Последовательность действия зависит от потока события в системе. Наступление события обозначается поступлением сообщения. Сообщение – информация о том, какое событие должно быть выполнено. Многие сообщения в Windows начинаются с WM\_ (Windows Message). Например, сообщение WM\_CREATE посылается при создании окна программы и можно задать выполнение какого-либо кода для этого события. WM\_RBUTTONDOWN, WM\_LBUTTONDOWN – сообщения о нажатии правой и левой кнопки мыши [19].

Написание программы начинается с подключения библиотек:

1. `#include <windows.h>` - эта библиотека содержит все функции, которые нужны для написания оконного приложения.
2. `#include <winsock.h>` - с помощью этой библиотеки сможем получать запросы с микроконтроллера и отправлять ответ обратно.
3. `#include <iostream>` - библиотека с классами, функциями и переменными для организации ввода-вывода в языке программирования C++.
4. `#include <tchar.h>` - для возможности использования символьного типа TCHAR.
5. `#include <mysql.h>` - библиотека базы данных MySQL.

Основная функция программы - `int WINAPI WinMain()`. Она принимает 4 аргумента:

- 1 `HINSTANCE hInstance` – дескриптор экземпляра окна.
- 2 `HINSTANCE hPrevInstance` - всегда является нулём [20].

3 LPSTR lpCmdLine - нужен для запуска окна в режиме командной строки.

4 int nCmdShow - режим отображения окна. Например, с помощью него мы можем развернуть окно на весь экран, сделать его определенной высоты, прозрачным или поверх остальных.

Связь между микроконтроллером и ПК будет осуществляться при помощи WinSock. WinSock представляет интерфейс программирования приложения, предназначенный для реализации программ в сети на основе протоколов TCP/IP. Для запуска сервера используем функцию Start() и для остановки функцию Stop(), которые принимают дескриптор окна приложения. На подключение клиента срабатывает событие WM\_SERVER\_ACCEPT, в котором зададим функцию OnServerAccept. Данная функция будет обрабатывать запрос от клиента и вызывать последующие функции для проверки полученного ID карты в базе данных MySQL. Проверять ID будет функция checkValueInDataBase, которая инициализирует соединение с базой данных, отправляет запрос с полученным ID и получает ответ в виде строки. Делаем проверку – если получили пустую строку, то выводим предупреждающее сообщение и отправляем микроконтроллеру значение false; если строка не пустая, то выводим ФИО посетителя и его фото и отправляем микроконтроллеру true. Готовая программа находится в Приложении Б.

#### **4.3 Выводы по разделу**

В данном разделе были подобраны средства для разработки программного обеспечения, разработан алгоритм взаимодействия микроконтроллера и ПК. В качестве основного способа передачи данных была выбрана сеть Wi-Fi и как следствие было разработано программное обеспечение для ПК на основе WinAPI и WinSock, которое позволяет передавать данные на расстоянии и выводить информацию на экран.

## 5 Выбор остальных элементов устройства

Для того чтобы устройство полноценно функционировало нужно подобрать недостающие элементы, обеспечивающие питание микроконтроллера и зарядку аккумулятора для него. Для зарядки аккумулятора требуется блок питания с выходным напряжением 5В и выходным током 1А.

Выбор пал на блок питания Robiton TinyCharger. Этот блок питания удовлетворяет поставленным требованиям, а также имеет защиту от перегрузок и короткого замыкания.

Технические характеристики:

- Вход: 100-240В ~ 50/60Гц, 120мА
- Выходное напряжение: 5В
- Выходной ток: 1000 мА (макс)



Рисунок 26 - Блок питания TinyCharger.

### 5.1 Выводы по разделу

В данном разделе был подобран блок питания, который удовлетворяет требованиям из задания на выполнение бакалаврской работы. Таким требованием являлось, чтобы блок питания выдавал 5В DC. Выбранный блок питания обладает также дополнительными преимуществами, такие как защита от перегрузок и короткого замыкания.

## 6 Разработка корпуса устройства.

Проанализировав размеры всех используемых модулей, возможное расположение выводов и проводов был разработан прототип корпуса.

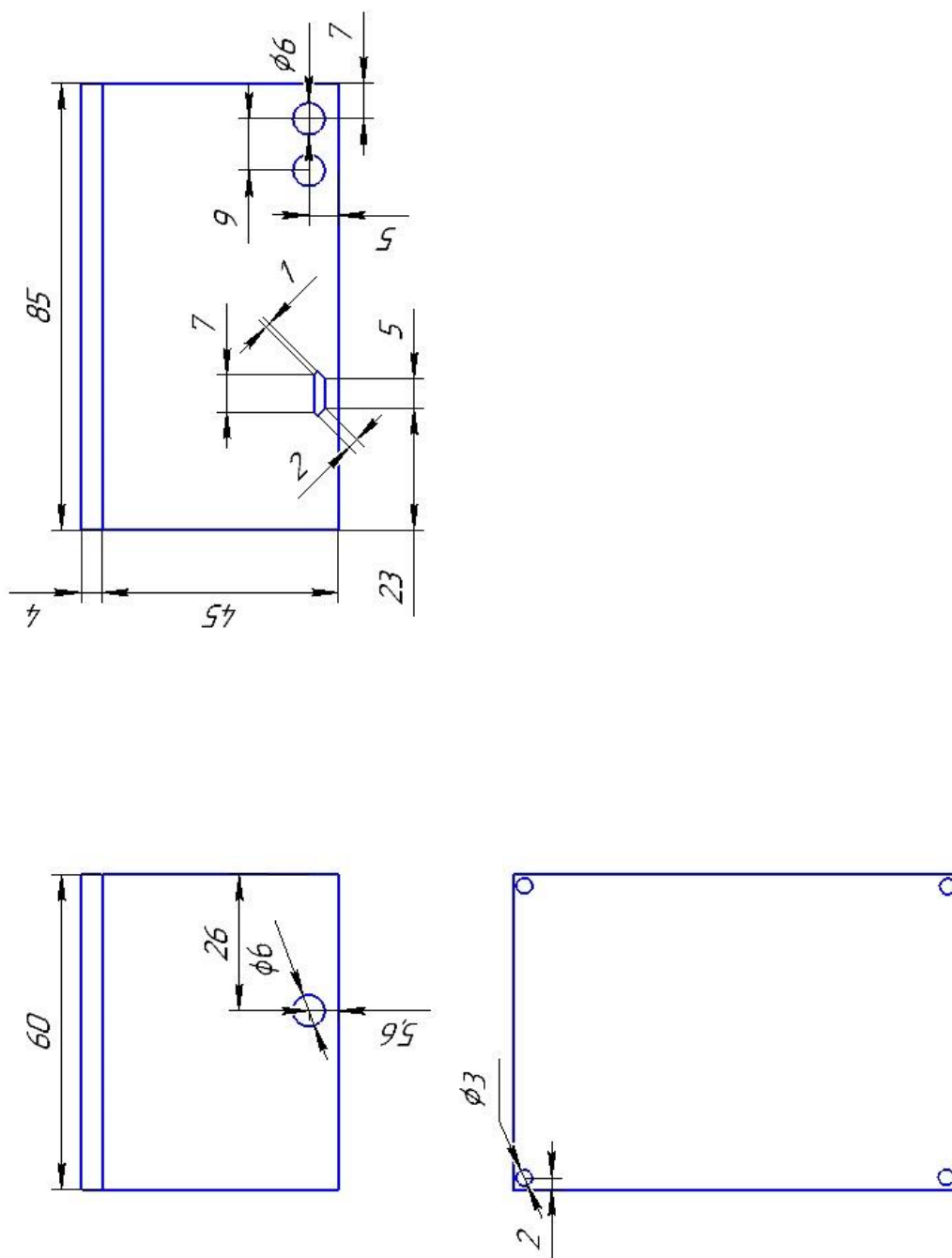


Рисунок 27 - Корпус устройства

## **6.1 Выводы по разделу**

В данном разделе были проанализированы размеры выбранных в третьем разделе модулей и плат, а также продумано их возможное расположение относительно друг друга. На основе этого был разработан корпус устройства, который содержит в себе отверстие под разъем micro-USB, светодиоды и вывод проводов для управления блоком исполнительных устройств. Наличие разъема micro-USB позволяет подавать питание устройству и заряжать аккумуляторную батарею. Для защиты доступа к программе микроконтроллера было принято решение не делать отверстие в корпусе под разъем. Для обновления программы придется снимать крышку корпуса.



## 7 Диаграмма работы устройства

Для более полного понимания работы устройства рассмотрим временную диаграмму работы RFID считывателя. Передача данных от RFID модуля к микроконтроллеру происходит с помощью интерфейса SPI.

SPI – последовательный периферийный интерфейс, позволяющий передавать 1 байт информации за раз между двумя модулями. SPI протокол известен как мастер/ведомый протокол. Это означает, что есть только одно управляющее устройство и несколько управляемых устройств. Этот интерфейс отлично подходит для быстрой передачи данных с датчиков. Для работы этого интерфейса необходимо минимум 4 пина: SCK, MOSI, MISO и SS (на модуле RFID обозначен как SDA).

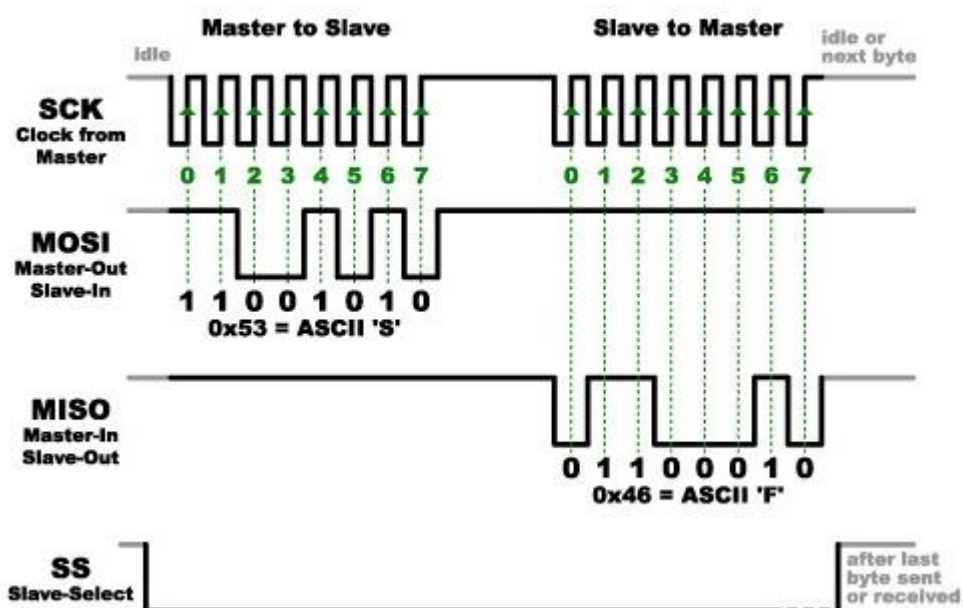


Рисунок 28 - Временная диаграмма обмена данными SPI

SCK это тактовый сигнал, который нужен для синхронизации между двумя устройствами, например как быстро нужно передавать и принимать данные. Скорость передачи данных известна как бод. Без SCK устройства не могли бы получать данные друг от друга.

MOSI отвечает за передачу данных от мастера к ведомому, MISO за передачу информации от ведомого к мастеру. На временной диаграмме

видно, как ведущее устройство отправляет символ “S” ведомому устройству и в ответ получает символ “F”.

SS используется для уведомления определенного ведомого устройства о том, что мастер сейчас отправит данные. После того, как мастер инициализирует связь между устройствами, он будет держать низкий сигнал на SS до тех пор, пока не прекратится передача данных. При низком сигнале на SS мастер начнет переключать тактовый вывод SCK и управлять MOSI для предоставления информации ведомому. Цифры на диаграмме обозначают каждый бит передаваемого байта [21].

## **7.1 Вывод по разделу**

В данном разделе была рассмотрена временная диаграмма работы RFID считывателя. Из-за того что существует несколько типов протоколов типов передачи информации от модуля к микроконтроллеру, был рассмотрен протокол SPI, на котором работает RFID модуль. Были рассмотрены преимущества данного протокола, как происходит синхронизация между модулями, по какому принципу и в какой последовательности происходит передача и прием информации. Также при необходимости подключения к микроконтроллеру нескольких считывателей RFID возможны несколько типов подключения, что позволяет расширить возможный функционал устройства до требуемого.

## 8 Изготовление прототипа устройства

На данном этапе был создан прототип устройства на основе подобранным ранее модулям и написанного ПО, также проверена работоспособность устройства.

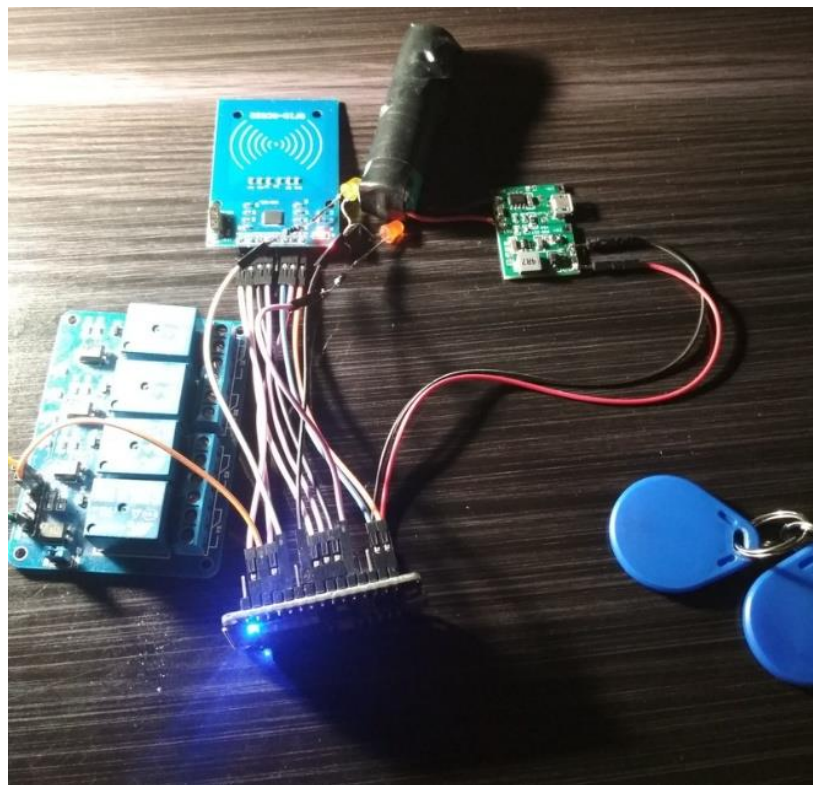


Рисунок 29 - Прототип устройства.

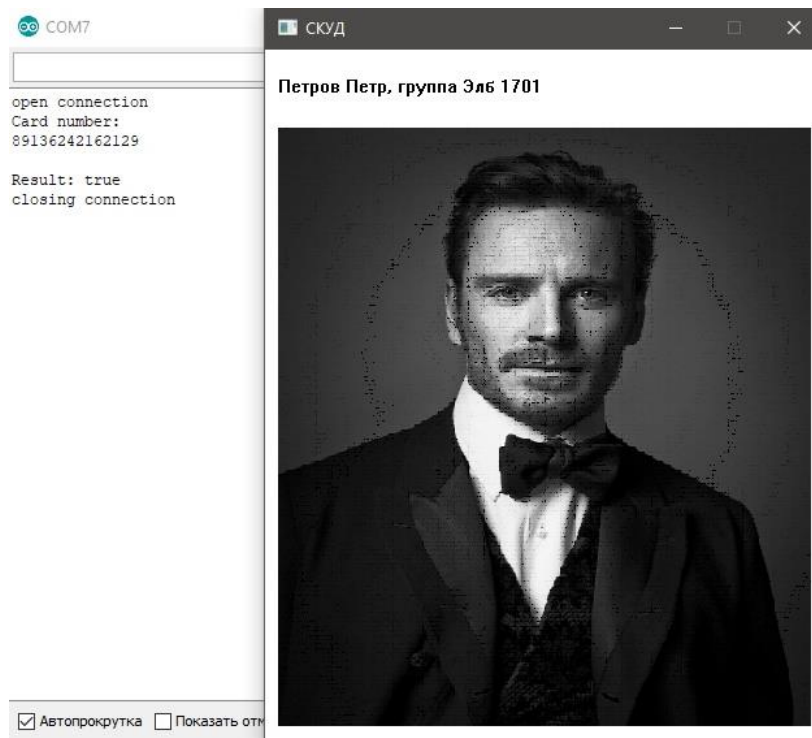


Рисунок 30 - Результат выполнения программ при поднесении карты с записанным в базе данных ID.



Рисунок 31 - Результат выполнения программ при поднесении карты с незаписанным в базе данных ID.

## 8.1 Выводы по разделу

В данном разделе на основе разработанной в третьем разделе принципиальной электрической схемы и подобранным компонентам был собран прототип устройства. В микроконтроллер и ПК были загружены программы, написанные в четвертом разделе. После загрузки программ, был определен IP-адрес и порт компьютера, на котором запускается программа на WinAPI и WinSock. После того, как микроконтроллер успешно подключился к сети Wi-Fi была проверена работоспособность устройства. Работа устройства полностью соответствует алгоритму, разработанному в четвертом разделе. Результатом успешной проверки является вывод фотографии и информации посетителя на экран в противном случае выводится предупреждающая картинка и текст.

## Заключение

В результате выполнения бакалаврской работы было разработано устройство на основе платформы Arduino для идентификации посетителей в образовательное учреждение.

В первом разделе работы была обоснована необходимость в разработке устройства. С этой целью было проведено исследование существующих систем контроля и управления доступом и выявлены функции, которые реализованы в настоящее время в СКУД. В втором разделе на основании анализа известных решений были выдвинуты требования к разрабатываемому устройству и сделан вывод о необходимости разработки устройства с наименьшей экономической стоимостью. В третьем разделе была разработана первичная структурная схема устройства, после чего подобраны необходимые компоненты, проанализированы их технические характеристики и способы подключения. В результате была получена электрическая принципиальная схема устройства. Разработанные структурная схема и принципиальная схема компонентов обеспечили возможность дальнейшей разработки алгоритмов функционирования и реализации программного прототипа. В четвертом разделе был разработан алгоритм работы устройства, выбраны средства для написания программного обеспечения и база данных для хранения информации, после чего написаны программы для микроконтроллера и компьютера. В заключительной части работы был разработан корпус устройства с учетом всех размеров модулей и плат, а также собран рабочий прототип и проверена работоспособность устройства.

Преимуществами разработанного устройства являются хранение информации в учебном заведении, что предотвращает вероятность получения доступа к данным другими лицами, возможность добавлять новый функционал и новые компоненты, благодаря гибкой структуре и возможности обновления кода микроконтроллера и ПК.

## Список используемой литературы

1. ГОСТ Р 51241-2008. Средства и системы контроля и управления доступом. Классификация. Общие технические требования. Методы испытаний: утв. приказом от 17 декабря 2008 г. N 430-ст. URL: <http://docs.cntd.ru/document/1200071688>
2. СКУД компании BOLID [Электронный ресурс]: сайт компании. URL: <https://bolid.ru/projects/iso-orion/access-control/>
3. СКУД компании PERCO [Электронный ресурс]: сайт компании. URL: <https://www.perco.ru/products/sistema-kontrolya-dostupa-perco-web/>
4. Система безопасности для образовательных учреждений PERCO [Электронный ресурс]: сайт компании. URL: <https://www.perco.ru/products/sistema-bezopasnosti-perco-s-20-shkola/>
5. Производитель RFID-меток и RFID-оборудования компания РСТ-Инвент [Электронный ресурс]: сайт компании. URL: <http://www.rst-invent.ru/faq/>
6. RFID считыватели и метки – Режим доступа [Электронный ресурс]: обучающий блог. URL: <https://arduino-kit.ru/product/schityivatel-rfid-rc522-1356mhz-2b-karta-2b-brelak>
7. Микроконтроллер Arduino Nano [Электронный ресурс]: документация. URL: <http://wiki.amperka.ru/%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D1%8B:arduino-nano>
8. Samsung ICR18650-22E Cell Specifications.// Specification of product for Lithium-ion Rechargeable Cell, page 1, 2681 symbols. URL: <http://elemakku.hu/datasheets/Samsung%20ICR18650-22E.pdf>
9. Модуль TP4065 [Электронный ресурс]: интернет-магазин. URL: <https://www.chipdip.ru/product/tp4056-charge-module-with-protection>
10. Модуль HW-357 [Электронный ресурс]: интернет-магазин. URL: <https://chipmaster.pro/shop/product/modul-zaryadki-li-ion-akkumulya-tora-hw-357-s-povyshayushhim-dc-dc-preobrazovatelem/>

11. Твердотельное реле [Электронный ресурс]: интернет-магазин. URL: <https://smdx.ru/g3mb-202p-1ch>
12. Электромагнитное реле [Электронный ресурс]: интернет-магазин. URL: [https://roboshop.spb.ru/1-channel-rele?gclid=CjwKC\\_AjwIzf3BRABEiwA8Q0qq-qyusNSQzKIhmeE40jqjpaajluElnbr-vzm\\_MJvxIHnZjiLNduoAQRoCbEYQAvD\\_BwE](https://roboshop.spb.ru/1-channel-rele?gclid=CjwKC_AjwIzf3BRABEiwA8Q0qq-qyusNSQzKIhmeE40jqjpaajluElnbr-vzm_MJvxIHnZjiLNduoAQRoCbEYQAvD_BwE)
13. Устройство платы NodeMCU [Электронный ресурс]: образовательный блог. URL: <https://myrobot.ru/wiki/index.php?n=Experiences.NodemcuPinout>
14. Распиновка считывателя RC522 [Электронный ресурс]: документация. URL: <https://3d-diy.ru/wiki/arduino-moduli/rfid-modul-rc522/>
15. Подключение реле к Arduino [Электронный ресурс]: образовательный блог. URL: <https://voltiq.ru/relay-and-arduino/>
16. Arduino IDE. // Documentation, symbols 278. URL: [https://en.wikipedia.org/wiki/Arduino\\_IDE](https://en.wikipedia.org/wiki/Arduino_IDE)
17. A Brief Introduction to the Serial Peripheral Interface // Documentation, symbols 720. URL: <https://www.arduino.cc/en/reference/SPI>
18. Getting Started with MySQL // Documentation, symbols 720. URL: <https://dev.mysql.com/doc/mysql-getting-started/en/>
19. Глибин, Е.С. Программирование электронных устройств: электронное учеб. пособие / Е.С. Глибин, А.В. Прядилов. - Тольятти
20. WinMain function. // Documentation, symbols 552. URL: <https://docs.microsoft.com/ru-ru/windows/win32/api/winbase/nf-winbase-winmain?redirectedfrom=MSDN>
21. Communication Protocols. // Documentation. URL: [https://tessel.gitbooks.io/t2-docs/content/Tutorials/Communication\\_Protocols.html#spi](https://tessel.gitbooks.io/t2-docs/content/Tutorials/Communication_Protocols.html#spi)



### Программа для микроконтроллера

```
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <RFID.h>
#include <SPI.h>

#ifndef STASSID
#define STASSID "DSL-2640U"
#define STAPSK "48583343"
#endif

#define SS_PIN D1
#define RST_PIN D3

RFID rfid(SS_PIN, RST_PIN);

const char* ssid = "DSL-2640U";
const char* password = "12345678";

const char* host = "192.168.1.90";
const uint16_t port = 8000;

ESP8266WiFiMulti WiFiMulti;

void setup() {
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFiMulti.addAP(ssid, password);

  Serial.println();
  Serial.print("Wait for WiFi... ");

  while (WiFiMulti.run() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  SPI.begin();
  rfid.init();
  pinMode(D8, OUTPUT); // зеленый светодиод
  pinMode(D2, OUTPUT); // реле
  pinMode(D4, OUTPUT); // красный светодиод
  delay(3000);
}

void loop() {
  if (rfid.isCard()) {
    if (rfid.readCardSerial()) {
      WiFiClient client;
      if (!client.connect(host, port)) {
```

```
Serial.print("connecting to ");
  Serial.print(host);
  Serial.print(':');
  Serial.println(port);
  Serial.println("connection failed");
  Serial.println("wait 2 sec...");
  delay(2000);
  return;
}

Serial.println("open connection");
String rfid = printID();
client.println(rfid);
String line = client.readStringUntil('\n');
Serial.print("Result: ");
if (line.equals("true")){
  digitalWrite(D2, HIGH);
  digitalWrite(D8, HIGH);
  close(1);
} else {
  digitalWrite(D4, HIGH);
  close(2);
}
Serial.println(line);
Serial.println("closing connection");
client.stop();
}
}
rfid.halt();
delay(1000);
}

void close (int num) {
  delay(2000);
  if (num == 1){
    digitalWrite(D2, LOW);
    digitalWrite(D8, LOW);
  } else {
    digitalWrite(D4, LOW);
  }
}

String printID(){
  String address = "";
  Serial.println("Card number:");
  for (int i = 0; i < 5; i++){
    address += String(rfid.serNum[i]);
  }
  address += "\n";
  Serial.println(address);
  return address;
}
```

```
#include <windows.h>
#include <winsock.h>
#include <iostream>
#include <tchar.h>
#include <mysql.h>
#pragma comment(lib, "ws2_32.lib")

using namespace std;

MYSQL connection, mysql;
MYSQL_RES* result;
MYSQL_ROW row;

HANDLE hBitmap;

int query_state;
const char* localhost = "localhost";
const char* user = "root";
const char* pwd = "test";
const char* db = "test";
char id[14];
string res;
char* idStar = id;

boolean checkValueInDataBase();
int printUserID();

const int WINSOCK_VERSION = 0x0101;
const int WM_SERVER_ACCEPT = WM_USER + 1;

// Прототип функции обратного вызова для обработки сообщений
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
void OnServerAccept(WPARAM wParam, LPARAM lParam);

bool Start(HWND);
void Stop();
char g_szStatus[512];
char buffer[512];
SOCKET g_serverSocket;
char response[512];

static TCHAR szWindowClass[] = _T("DesktopApp");
static TCHAR szTitle[] = _T("СКУД");

// Вызов функции при запуске программы
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
    PSTR szCmdLine, int iCmdShow)
{
    HWND hWnd; // дескриптор родительского окна
    MSG msg;
    WNDCLASSEX wndclass;
    // Настройка класса окна
```

## Продолжение приложения Б

```
    wndclass.cbSize = sizeof(WNDCLASSEX);
wndclass.style = CS_HREDRAW | CS_VREDRAW;
    wndclass.lpfWndProc = WndProc;
    wndclass.cbClsExtra = 0;
    wndclass.cbWndExtra = 0;
    wndclass.hInstance = hInstance;
    wndclass.hIcon = LoadIcon(NULL, IDI_APPLICATION);
    wndclass.hCursor = LoadCursor(NULL, IDC_ARROW);
    wndclass.hbrBackground =
        (HBRUSH)GetStockObject(WHITE_BRUSH);
    wndclass.lpszMenuName = NULL;
    wndclass.lpszClassName = szWindowClass;
    wndclass.hIconSm = LoadIcon(NULL, IDI_APPLICATION);
    // Регистрация класса окна
    if (RegisterClassEx(&wndclass) == 0)
    {
        // Ошибка выполнения программы
        return 0;
    }
    // Создание окна
    hWnd = CreateWindowEx(
        WS_EX_OVERLAPPEDWINDOW,
        szWindowClass, // Имя класса
        szTitle, // Текст заголовка
        WS_OVERLAPPED | WS_CAPTION | WS_SYSMENU |
        WS_MINIMIZEBOX,
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        450, // размеры окна по X
        575, // размеры окна по Y
        NULL,
        NULL,
        hInstance,
        NULL);
    // Отображение окна
    ShowWindow(hWnd, iCmdShow);
    // Обработка сообщений, пока программа не будет прервана
    while (GetMessage(&msg, NULL, 0, 0))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }

    return (int)msg.wParam;
}
// Функция обратного вызова для обработки сообщений
LRESULT CALLBACK WndProc(HWND hWnd, UINT iMsg, WPARAM wParam,
    LPARAM lParam)
{
    SOCKET AcceptSocket;
    SOCKADDR_IN socketaddr;
    WSADATA wsaData;
    HDC hDC;
    HDC hCompatibleDC;
```

## Продолжение приложения Б

```
HANDLE hOldBitmap;
RECT Rect;
BITMAP Bitmap;
PAINTSTRUCT ps;
int foo;
char id[512];
int index = 0;
boolean isIdFound = false;

switch (iMsg)
{
    // Вызывается сразу же при создании окна функцией CreateWindow() (или
    CreateWindowEx())
    case WM_CREATE:
        strcpy_s(g_szStatus, "Щелкните левой кнопкой мыши для запуска сервера");
        // g_serverSocket инициализируется значением недействительного сокета
        g_serverSocket = INVALID_SOCKET;
        break;
        // Вызывается, когда пользователь отпускает левую кнопку мыши
    case WM_LBUTTONDOWN:
        // Запуск сервера
        Start(hWnd);
        // Перерисовка окна (генерация сообщения WM_PAINT)
        InvalidateRect(hWnd, NULL, TRUE);
        break;
        // Вызывается, когда пользователь отпускает правую кнопку мыши
    case WM_RBUTTONDOWN:
        // Остановка сервера
        Stop();
        InvalidateRect(hWnd, NULL, TRUE);
        break;
    case WM_SERVER_ACCEPT:
        OnServerAccept(wParam, lParam);
        InvalidateRect(hWnd, NULL, TRUE);
        break;
        // Вызывается, когда окно обновляется
    case WM_PAINT:
        hDC = BeginPaint(hWnd, &ps);
        TextOut(hDC, 10, 20, g_szStatus,
            (int)strlen(g_szStatus));
        if (hBitmap) {
            GetObject(hBitmap, sizeof(BITMAP), &Bitmap);
            /* создать совместимый с контекстом окна контекст в памяти */
            hCompatibleDC = CreateCompatibleDC(hDC);
            /* делаем загруженный битмап текущим в совместимом контексте */
            hOldBitmap = SelectObject(hCompatibleDC, hBitmap);
            /* определить размер рабочей области окна */
            GetClientRect(hWnd, &Rect);
            // версия без прокрутки
            // копировать битмап с совместимого на основной контекст устройства с
масштабированием
            StretchBlt(hDC, 10, 60, 410, 460, hCompatibleDC, 0, 0, Bitmap.bmWidth,
```

```

Bitmap.bmHeight, SRCCOPY);

        SelectObject(hCompatibleDC, hOldBitmap);
        /* удалить загруженный битмап */

        DeleteObject(hBitmap);
        /* удалить совместимый контекст */
        DeleteDC(hCompatibleDC);
    }
    EndPaint(hWnd, &ps);
    break;
    // Вызывается, когда пользователь закрывает окно
case WM_DESTROY:
    // Остановка сервера при закрытии окна
    Stop();
    PostQuitMessage(0);
    break;
default:
    return DefWindowProc(hWnd, iMsg, wParam, lParam);
}
return 0;
}
bool Start(HWND hWnd)
{
    int error;
    SOCKADDR_IN socketaddr;
    WSADATA wsaData;
    if (g_serverSocket != INVALID_SOCKET)
    {
        strcpy_s(g_szStatus, "Сервер уже запущен");
        return false;
    }
    if (WSAStartup(WINSOCK_VERSION, &wsaData))
    {
        strcpy_s(g_szStatus, "Ошибка инициализации Winsock");
        return false;
    }
    g_serverSocket = socket(PF_INET, SOCK_STREAM, 0);
    if (g_serverSocket == INVALID_SOCKET)
    {
        strcpy_s(g_szStatus, "Ошибка создания сокета");
        return false;
    }
    socketaddr.sin_family = AF_INET;
    socketaddr.sin_addr.s_addr = INADDR_ANY;
    socketaddr.sin_port = htons(8000);
    error = bind(g_serverSocket, (LPSOCKADDR)&socketaddr,
        sizeof(socketaddr));
    if (error == SOCKET_ERROR)
    {
        strcpy_s(g_szStatus, "Ошибка связи сокета с портом");
        return false;
    }
}

```

## Продолжение приложения Б

```
error = WSAAsyncSelect(g_serverSocket, hWnd,
    WM_SERVER_ACCEPT, FD_ACCEPT | FD_READ);

if (error == SOCKET_ERROR)
{
    strcpy_s(g_szStatus, "Ошибка связи сокета с окном");
    return false;
}
error = listen(g_serverSocket, 10);
if (error == SOCKET_ERROR)
{
    strcpy_s(g_szStatus, "Ошибка прослушивания сокета");
    return false;
}
strcpy_s(g_szStatus, "Жду посетителя");
return true;
}
void Stop()
{
    if (g_serverSocket == INVALID_SOCKET)
    {
        strcpy_s(g_szStatus, "Сервер не запущен или уже остановлен");
        return;
    }
    closesocket(g_serverSocket);
    g_serverSocket = INVALID_SOCKET;
    if (WSACleanup())
    {
        strcpy_s(g_szStatus, "Ошибка освобождения Winsock");
        return;
    }
    strcpy_s(g_szStatus, "Сервер успешно остановлен");
    return;
}

void OnServerAccept(WPARAM wParam, LPARAM lParam)
{
    int error, length;
    SOCKET clientSocket;
    SOCKADDR socketclientaddr;
    static char positive[] = "true\n";
    static char negative[] = "false\n";
    if (WSAGETASYNCERROR(lParam))
    {
        strcpy_s(g_szStatus, "Ошибка определения клиента");
        return;
    }
    if (WSAGETSELEVENT(lParam) == FD_ACCEPT)
    {
        length = sizeof(SOCKADDR);
        clientSocket = accept(g_serverSocket,
            NULL, NULL);
```

## Продолжение приложения Б

```
if (clientSocket == INVALID_SOCKET)
    {
        strcpy_s(g_szStatus, "Ошибка создания сокета клиента");

        return;
    }
}
else if (WSAGETSELECTEVENT(IPParam) == FD_READ)
    {
        clientSocket = (SOCKET)wParam;

        //ZeroMemory(g_szStatus, sizeof(g_szStatus));
        error = recv(clientSocket, g_szStatus,
            sizeof(g_szStatus), 0);
        if (error == SOCKET_ERROR)
            {
                strcpy_s(g_szStatus, "Ошибка в получении запроса ");
                return;
            }
        printUserID();
        boolean res = checkValueInDataBase();

        if (res) {
            error = send(clientSocket, positive, (int)strlen(positive), 0);
        }
        else {
            error = send(clientSocket, negative, (int)strlen(negative), 0);
        }

        if (error == SOCKET_ERROR)
            {
                strcpy_s(g_szStatus, "Ошибка в ответе клиенту");
                return;
            }
        closesocket(clientSocket);
        return;
    }
return;
}

void printUserID() {
    for (int i = 0; i < 14; i++) {
        id[i] = g_szStatus[i];
    }
}

boolean checkValueInDataBase() {
    try {
        mysql_init(&connection);
        mysql_options(&connection, MYSQL_SET_CHARSET_NAME, "cp1251");
        mysql_options(&connection, MYSQL_INIT_COMMAND, "SET NAMES cp1251");
        mysql_real_connect(&connection, localhost, user, pwd, db, 3306, 0, 0);
    }
```



## Продолжение приложения Б

```
string query = "SELECT group_concat(concat(users.last_name, \" \", users.first_name, \", группа \",
university_groups.group_name)) FROM users, university_groups WHERE users.group_id =
university_groups.id and users.card_id = " + string(id);
    const char* q = query.c_str();
    query_state = mysql_query(&connection, q);

    if (!query_state) {
        result = mysql_use_result(&connection);
        row = mysql_fetch_row(result);
        if (!row[0]) {
            strcpy_s(g_szStatus, "Посетитель не найден");
            hBitmap = LoadImage(NULL, "F:\\people\\warning.bmp",
IMAGE_BITMAP, 0, 0, LR_LOADFROMFILE);
            return false;
        }

        string dir = "F:\\people\\" + string(id) + string(".bmp");
        strcpy_s(g_szStatus, row[0]);
        hBitmap = LoadImage(NULL, dir.c_str(), IMAGE_BITMAP, 0, 0,
LR_LOADFROMFILE);
        return true;
    }
    else {
        strcpy_s(g_szStatus, mysql_error(&connection));
    }
    mysql_close(&connection);
    return false;
}
catch (exception& ex)
{
    strcpy_s(g_szStatus, ex.what());
    return false;
}
catch (const char* ex)
{
    strcpy_s(g_szStatus, ex);
    return false;
}
}
```