

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

**Учреждение образования
«Гомельский государственный университет
имени Франциска Скорины»**

Факультет физики и информационных технологий
Кафедра общей физики

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
на тему**

МОБИЛЬНЫЙ ДЕКОДЕР ШТРИХ-КОДОВ

ГГУ КП 1-39 03 02 12 ПЗ

Исполнитель:

Студент группы МС-32:

Сырников А. С.

Научный руководитель:

Старший преподаватель

Кафедры общей физики

Подалов М. А.

Гомель 2023

РЕФЕРАТ

Курсовой проект: 36 страниц, 3 главы, 24 рисунка, источников.

ESP32, C++, GM65, I2C, МИКРОКОНТРОЛЛЕР, ДЕКОДЕР, КОД, СКАНЕР, ARDUINO

Объектом и предметом исследования является создание мобильного декодера штрих-кодов.

Цель работы: разработать электронное мобильное устройство, которое позволит сканировать, декодировать штрих-коды и выводить полученные коды на LCD-дисплей. Изучить особенности интерфейса USB и структуру пакетов. Изучить принципы создания структурных, принципиальных электронных схем в процессе проектирования устройства. Изучить особенности архитектуры и написать программное обеспечение для микроконтроллера.

В процессе выполнения курсового проекта были решены следующие задачи:

1. Изучение технологии сканирования штрих-кодов и способов их декодирования.
2. Изучение аппаратных и программных средств, необходимых для реализации мобильного декодера штрих-кодов.
3. Разработка структурной электрической схемы для подключения камеры и дисплея к микроконтроллеру ESP32.
4. Разработка принципиальной электрической схемы для подключения камеры и дисплея к микроконтроллеру ESP32.
5. Разработка программного обеспечения для работы мобильного декодера штрих-кодов на базе ESP32.
6. Тестирование мобильного декодера штрих-кодов на различных типах штрих-кодов и в разных условиях использования.
7. Оформление пояснительной записки по результатам курсового проекта.

Тема курсового проекта "Мобильный декодер штрих-кодов" является актуальной, так как в настоящее время штрих-коды широко используются в различных отраслях, включая торговлю, логистику, медицину и промышленность. Мобильные устройства, такие как смартфоны и планшеты, также широко распространены, что позволяет использовать их как инструменты для сбора и обработки данных, включая данные из штрих-кодов.

Разработка мобильного декодера штрих-кодов позволит упростить и автоматизировать процессы сбора и обработки данных, а также повысить их точность и скорость. Кроме того, использование мобильного декодера штрих-кодов может улучшить производительность и эффективность работы в различных отраслях, где применение штрих-кодов является обычной практикой.

Таким образом, тема курсового проекта актуальна и имеет практическую значимость в различных отраслях и сферах деятельности.

СОДЕРЖАНИЕ

Введение.....	4
1 Анализ предметной области	5
1.1 Описание предметной области	5
1.2 Обзор платформ для разработки.....	6
1.3 Выбор отладочной платформы для разработки	9
1.4 Выбор сред разработки.....	11
1.5 Языки, выбранные для написания программного обеспечения.....	13
2 Разработка электрических схем мобильного декодера штрих-кодов	15
2.1 Разработка структурной схемы проигрывателя аудио файлов	15
2.2 Разработка принципиальной схемы декодера штрих-кодов	22
3 Разработка алгоритма функционирования мобильного декодера штрих-кодов	31
3.1 Описание алгоритма функционирования мобильного декодера штрих-кодов.....	31
3.2 Описание функций алгоритма	31
Заключение	35
Список используемых источников	36
Приложение А	37
Структурная схема мобильного декодера штрих-кодов.....	37
Приложение Б	38
Блок-схема алгоритма мобильного декодера штрих-кодов	38
Приложение В.....	39
Принципиальная схема мобильного декодера штрих-кодов	39
Приложение Г	40
Код алгоритма программы мобильного декодера штрих-кодов	40
Спецификация к курсовому проекту.....	42
Ведомость документов к курсовому проекту.....	43

Введение

Штрих-коды используются в различных отраслях промышленности, торговли, логистики и других областях для быстрого и точного идентифицирования товаров и инвентаризации. Для этой цели часто используются специализированные устройства, называемые декодерами штрих-кодов. С развитием технологий и увеличением числа мобильных устройств, возросла потребность в мобильных декодерах штрих-кодов, которые могут быть использованы в любом месте и в любое время для быстрого и удобного сканирования штрих-кодов.

Проблематика мобильного декодера штрих-кодов связана с необходимостью быстрого и точного считывания информации из штрих-кодов на товарах, упаковках и других объектах. В современном мире, где торговля и логистика становятся все более автоматизированными, использование штрих-кодов является необходимым условием для оптимизации бизнес-процессов и увеличения эффективности работы предприятий.

Однако, для того чтобы штрих-коды могли быть использованы для автоматического считывания, требуется специальное оборудование – декодеры штрих-кодов. В ряде случаев, такие декодеры являются дорогостоящими и сложными в использовании, что ограничивает их применение в некоторых областях.

Мобильный декодер штрих-кодов является более доступным и удобным решением, поскольку он позволяет использовать для считывания информации обычные смартфоны или планшеты, что значительно упрощает процесс работы с штрих-кодами. Однако, несмотря на все преимущества, существует ряд проблем, связанных с использованием мобильных декодеров, таких как ограниченная производительность устройств и возможность искажения информации в случае неправильного считывания. Решение этих проблем является актуальной задачей и требует дальнейшего изучения данной области.

1 Анализ предметной области

1.1 Описание предметной области

Мобильные декодеры штрих-кодов широко применяются в различных отраслях, таких как розничная торговля, логистика, производство и др., где точность и скорость считывания информации из штрих-кодов являются критически важными для эффективного управления бизнесом.

Разработка мобильного декодера штрих-кодов включает в себя проектирование аппаратных и программных компонентов, которые обеспечивают точное и быстрое считывание информации из штрих-кодов и передачу ее на мобильное устройство для дальнейшей обработки. Также необходимо учитывать факторы, такие как портативность, легкость использования, поддержка различных форматов штрих-кодов.

Разработка мобильного декодера штрих-кодов имеет практическую значимость для бизнеса и может быть использована для автоматизации процессов сканирования и идентификации товаров, ускорения процессов логистики и инвентаризации, повышения качества обслуживания клиентов и т.д.

Ниже представлены основные аспекты анализа мобильного декодера штрих-кодов:

Преимущества:

- Портативность и легкость использования в различных условиях;
- Возможность считывания как одномерных, так и двумерных штрих-кодов;
- Быстрое и точное считывание информации;

Недостатки:

- Ограниченный дальностью считывания;
- Ограниченная поддержка некоторых форматов штрих-кодов;
- Требуется дополнительная обработка данных после считывания, что может занять дополнительное время;
- Некоторые модели могут иметь низкую производительность или долгое время работы от аккумулятора.

Таким образом, мобильный декодер штрих-кодов является удобным и эффективным инструментом для считывания и передачи информации, закодированной в штрих-кодах.

1.2 Обзор платформ для разработки

1.2.1 Arduino Uno

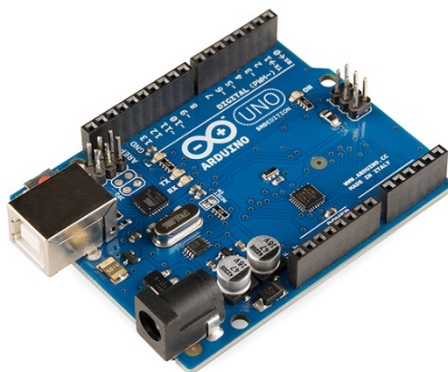


Рис. 1.2.1 – Arduino Uno

Arduino Uno - это микроконтроллерная плата, которая используется для создания электронных проектов. Она была разработана в Италии и представляет собой плату, основанную на микроконтроллере ATmega328P, который обеспечивает ее функционирование.

Одним из основных преимуществ Arduino Uno является простота в использовании. Она имеет простой интерфейс, который позволяет программировать и контролировать устройства с помощью языка программирования C++.

Кроме того, Arduino Uno имеет множество входов и выходов (GPIO), которые можно использовать для подключения различных датчиков, управления двигателями и другими электронными компонентами. Она также имеет возможность подключения к компьютеру через USB, что позволяет загружать программы на микроконтроллер и отладку кода.

1.2.2 Arduino Nano

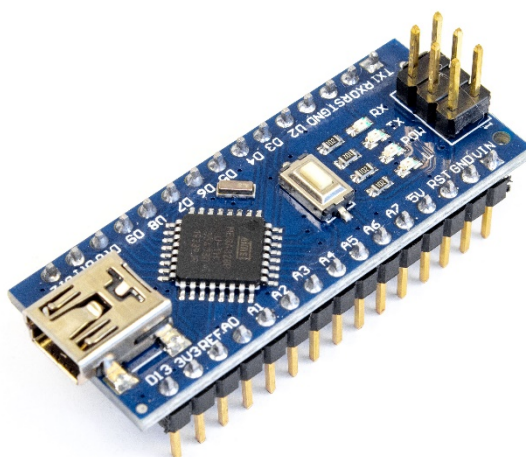


Рис. 1.2.2 – Arduino Nano

Arduino Nano - это компактная микроконтроллерная плата, которая является одним из самых маленьких и экономичных устройств Arduino. Она была разработана на основе той же технологии, что и Arduino Uno, но имеет более компактный размер и меньший набор входов/выходов.

Arduino Nano имеет микроконтроллер ATmega328P, такой же, как и в Arduino Uno, и имеет 14 цифровых входов/выходов и 8 аналоговых входов. Она также имеет возможность подключения к компьютеру через USB, так же как и Arduino Uno, что позволяет загружать программы на микроконтроллер и отладку кода.

Одним из преимуществ Arduino Nano является ее компактный размер, что делает ее идеальным выбором для проектов, где малый размер имеет значение, например, для использования в более маленьких устройствах, включая моделирование и прототипирование, робототехнику, автоматизацию домашнего хозяйства и многие другие.

Кроме того, Arduino Nano является экономичной альтернативой Arduino Uno, поскольку ее цена обычно ниже, чем у Arduino Uno.

1.2.3 ESP32-WROOM-32D

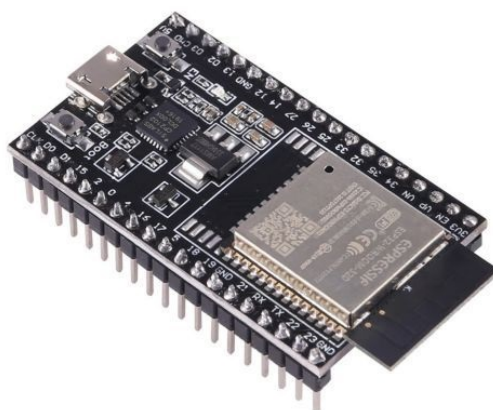


Рис. 1.2.3 - ESP32-WROOM-32D

ESP32-WROOM-32D - это Wi-Fi и Bluetooth модуль, основанный на системе-на-чипе ESP32. Он является одним из самых продвинутых модулей ESP32, который обеспечивает высокую производительность и широкий набор функций.

Модуль ESP32-WROOM-32D имеет два ядра процессора Tensilica Xtensa LX6, работающих на частоте 240 МГц, и может быть запрограммирован через Arduino IDE или с использованием платформы Espressif IoT Development Framework (ESP-IDF). Он также имеет 4 МБ встроенной памяти и поддерживает подключение к сети Wi-Fi 802.11 b/g/n и Bluetooth 4.2 BLE.

ESP32-WROOM-32D также поддерживает множество периферийных устройств, таких как GPIO, SPI, I2C, UART, ADC и DAC, что позволяет создавать разнообразные устройства и приложения. Кроме того, этот модуль может работать как в режиме клиента, так и в режиме точки доступа, что обеспечивает гибкость и разнообразие в использовании.

Модуль ESP32-WROOM-32D также имеет низкое энергопотребление, что делает его идеальным для разработки устройств на батарейной основе или для проектов, требующих длительной автономной работы.

1.2.4 ESP32-WROOM-32U

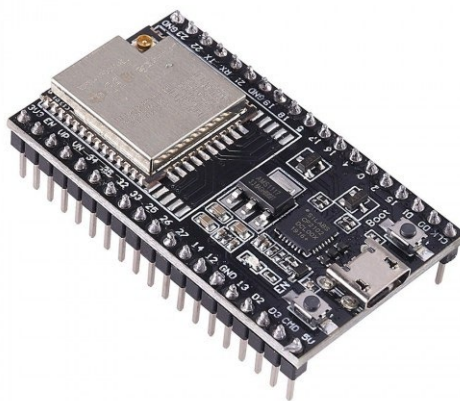


Рис. 1.2.4 - ESP32-WROOM-32U

В отличие от модуля ESP32-WROOM-32D, ESP32-WROOM-32U не имеет встроенного антенного разъема, и вместо этого требует внешней антенны для подключения к Wi-Fi и Bluetooth сетям. Это позволяет устройствам быть более компактными и легкими, так как антенна может быть размещена в другом месте устройства.

1.3 Выбор отладочной платформы для разработки

По требованиям проекта и представленным характеристиками была выбрана отладочная плата ESP32-WROOM-32D.

Во-первых, ESP32-WROOM-32D имеет встроенный антенный разъем, который позволяет легко подключить внешнюю антенну и обеспечить более сильный и стабильный сигнал Wi-Fi и Bluetooth. Это особенно важно для проектов, связанных с мобильными устройствами, которые могут использоваться в различных условиях и местах.

Во-вторых, ESP32-WROOM-32D имеет меньший размер, чем ESP32-WROOM-32U, что делает его более компактным и легким для интеграции в мобильные устройства, такие как мобильные декодеры штрих-кодов.

В-третьих, ESP32-WROOM-32D имеет достаточно мощную вычислительную мощность для обработки сигналов штрих-кодов и отправки данных через Wi-Fi или Bluetooth. Он имеет два ядра процессора Tensilica LX6, которые работают на частоте 240 МГц, и может быстро обрабатывать данные.

В-четвертых, ESP32-WROOM-32D имеет низкое энергопотребление, что является важным фактором для мобильных устройств, таких как декодеры

штрих-кодов. Он поддерживает режим Deep Sleep, в котором потребление энергии очень низкое, что увеличивает время автономной работы устройства на батареях.

В-пятых, ESP32-WROON-32D имеет встроенную поддержку интерфейса I2C, что позволит использовать LCD дисплей с этим интерфейсом.

I2C (Inter-Integrated Circuit) - это протокол передачи данных между устройствами, который использует две линии - SDA (Serial Data) и SCL (Serial Clock) - для передачи информации. Он был разработан компанией Philips (теперь NXP Semiconductors) и широко используется во многих электронных устройствах.

Интерфейс I2C позволяет подключать несколько устройств к одной шине, используя одинаковый адрес на шине для каждого устройства. Это обеспечивает экономию пинов на микроконтроллере и упрощает подключение множества устройств к одной шине.

Каждое устройство на шине I2C имеет свой уникальный адрес, который используется для идентификации устройства в процессе передачи данных. Шина I2C может подключать до 127 устройств, используя разные адреса.

Кроме того, интерфейс I2C поддерживает передачу данных в двух направлениях: от мастера к слейву (write) и от слейва к мастеру (read). Передача данных начинается с генерации стартового сигнала на шине мастером, затем мастер отправляет адрес устройства, с которым он хочет общаться, и выбирает режим передачи данных. После этого начинается передача данных.

Интерфейс I2C очень удобен для подключения различных устройств, таких как датчики, дисплеи, память и т.д. Он является широко распространенным в электронике и используется в многих микроконтроллерах, включая ESP32.

В данном проекте интерфейс IC2 используется для связи LCD-дисплея и управляющего микроконтроллера ESP32-WROOM-32D.

Модуль камеры GM65 - это небольшой модуль камеры, который может быть использован для захвата изображений или видео на микроконтроллерах и одноплатных компьютерах, таких как ESP32 или Raspberry Pi. Модуль имеет высокое разрешение - 5 мегапикселей, и использует интерфейс передачи данных UART для передачи данных. Он также может поддерживать формат JPEG для сжатия изображений и позволяет настраивать различные параметры, такие как экспозиция, баланс белого и контрастность.

Для работы с модулем камеры GM65 необходимо подключить его к микроконтроллеру через интерфейс UART и установить необходимые настройки. Модуль также может быть управляем через команды AT, что упрощает процесс разработки.

Для сканирования штрих-кодов, можно использовать модуль камеры GM65 совместно с программным обеспечением, которое обрабатывает изображения и распознает штрих-коды. Для этого можно написать скрипт на языке программирования, который будет выполнять такую обработку, например, на C++ или Python. В таком случае, модуль GM65 будет использоваться как источник изображений для анализа и распознавания штрих-кодов.

1.4 Выбор сред разработки

1.4.1 Arduino IDE

Arduino IDE (Integrated Development Environment) - это интегрированная среда разработки, которая предоставляет пользователю все необходимые инструменты для разработки приложений на Arduino. Она представляет собой простой и удобный инструмент для написания, отладки и загрузки программного кода на платформу Arduino.

Среда Arduino IDE поставляется со встроенным текстовым редактором, который позволяет создавать и редактировать скетчи (коды программ) на языке Arduino, который основан на языке C++. В IDE есть библиотеки и примеры, которые можно использовать для разработки проектов.

Arduino IDE поддерживает множество платформ, включая саму платформу Arduino, а также многие другие платформы, которые используют микроконтроллеры ATMELE AVR, ARM и т.д. Arduino IDE также поддерживает множество различных библиотек и скетчей, которые могут быть использованы для создания различных проектов.



Рис. 1.4.1 – Интерфейс Arduino IDE

Основные функции Arduino IDE:

- Подсветка синтаксиса для языка Arduino и C/C++;
- Автоматическое форматирование кода;
- Средства отладки;
- Встроенный монитор порта;
- Возможность загрузки скетча на Arduino плату;
- Возможность управления библиотеками и установка новых библиотек.

Arduino IDE - это отличный выбор для начинающих разработчиков и электронщиков, которые хотят создавать свои проекты с использованием платформы Arduino. Среда разработки имеет простой и интуитивно понятный интерфейс, а также множество руководств и примеров, которые помогают быстро начать работу.

Для разработки программного обеспечения для ESP32 в среде Arduino IDE необходимо установить дополнительный пакет инструментов для работы с этой платформой. Этот пакет содержит необходимые файлы заголовков и библиотек для ESP32, а также утилиты для загрузки прошивки в устройство.

Чтобы установить пакет инструментов для ESP32 в Arduino IDE, нужно выполнить следующие шаги:

1. Открыть настройки Arduino IDE (File -> Preferences).
2. В поле "Additional Boards Manager URLs" добавить ссылку на репозиторий ESP32.
3. Открыть менеджер плат (Tools -> Board -> Boards Manager).
4. Найти и установить пакет инструментов для ESP32 (например, "ESP32 by Espressif Systems").
5. Выбрать плату ESP32 в меню инструментов (Tools -> Board).

После установки пакета инструментов для ESP32 в Arduino IDE будут доступны все необходимые функции и библиотеки для разработки программного обеспечения для этой платформы.

1.5 Языки, выбранные для написания программного обеспечения

1.5.1 C++

C++ - это один из языков программирования, которые поддерживаются для программирования на микроконтроллерах ESP32. ESP32 предоставляет набор инструментов для разработки на C++, включая библиотеки и компиляторы.



Рис. 1.5.1 – Логотип C++

Для начала работы с ESP32 и C++ необходимо установить необходимые инструменты, такие как Arduino IDE и компилятор C++. Arduino IDE содержит все необходимые библиотеки и инструменты, а также документацию и примеры кода для разработки приложений на ESP32.

Для написания кода на C++ для ESP32 можно использовать любой редактор кода, и подключить его к Arduino IDE. После написания кода можно использовать инструменты компиляции и загрузки Arduino IDE для загрузки приложения на ESP32.

Пример программы на C++ для ESP32:

```
#include <Arduino.h>

void setup() {
  Serial.begin(115200);
  Serial.println("Hello, ESP32!");
}

void loop() {
  Serial.println("ESP32 is running...");
  delay(1000);
}
```

Рис. 1.5.2 – Пример программы на C++

Эта программа выводит сообщение "Hello, ESP32!" в монитор порта во время настройки и выводит сообщение "ESP32 is running..." каждую секунду в функции loop().

2 Разработка электрических схем мобильного декодера штрих-кодов

2.1 Разработка структурной схемы проигрывателя аудио файлов

2.1.1 Обоснование базовых блоков структурной электрической схемы мобильного декодера штрих-кодов

Рассмотрим каждый из представленных блоков структурной электрической схема из приложения А:

1. ESP32 – отладочная плата с микроконтроллером ESP32-WROOM 32D используется для декодирования штрих-кодов и QR-кодов и вывода их на LCD-дисплей. Питание будет осуществляться от microUSB.



Рис. 2.1.1 - ESP32-WROOM-32D

Характеристики ESP32:

- 32-битный двух ядерный микропроцессор - Xtensa LX6.
- Частота процессора - 160-240 МГц
- ОЗУ — 520 Кб
- ПЗУ — 448 Кб
- RTC таймер с 16 Кб ОЗУ

- Внешняя флешь память 4-16 Мб
- Питание 2.2 В — 3.6 В

Периферия:

- 18 x 12 битный АЦП
- 2 x 8 битный ЦАП
- 10 x портов в режиме сенсорных кнопок
- 1 x Встроенный температурный датчик
- 4 x SPI
- 2 x IS1
- 2 x I2C
- 3 x UART
- 1 x Host (SD/eMMC/SDIO)
- 1 x slave (SDIO/SPI)
- 1 x Ethernet MAC interface with dedicated DMA and IEEE 1588 support
- 1 x CAN 2.0
- 1 x IR (TX/RX)
- 16 x LED PWM
- Motor PWM
- Hall sensor
- Ultra-low-noise analog pre-amplifier

Энергопотребление:

- Максимальный ток при передаче WiFi — 160-260 мА
- Потребление без включенного WiFi и Bluetooth — 20 мА
- LightSleep — 0.8 мА
- DeepSleep — до 10 мкА
- Режим гибернации с активным RTC — 5 мкА

WiFi и Bluetooth:

- WiFi 802.11n 2.4 Гц с максимальной скоростью 150 Мбит/сек
- Шифрование WPA/WPA2/WPA2-Enterprise/Wi-Fi Protected Setup WPS
- Bluetooth v4.2 BR/EDR and BLE

2. Интерфейс I2C (Inter-Integrated Circuit) - это последовательный двухпроводной интерфейс, который используется для подключения LCD дисплея с микроконтроллером ESP32-WROOM-32D.

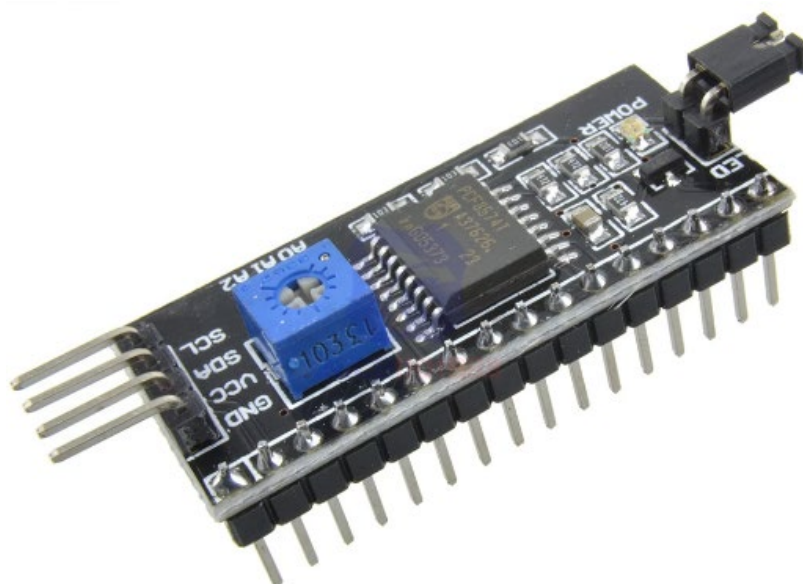


Рис. 2.1.2 – I2C интерфейс

Некоторые из основных характеристик интерфейса I2C включают в себя:

- Скорость передачи данных: Существует несколько стандартных скоростей передачи данных, включая стандартную (100 кбит/с), быструю (400 кбит/с) и высокую (3,4 Мбит/с) скорости.
- Адресация: Каждое устройство на линии I2C имеет уникальный 7 битный адрес, который используется для определения и выбора нужного устройства для обмена данными.
- Формат сообщений: Сообщения в линии I2C представляются в виде кадров, которые могут быть адресованными или неадресованными. В адресованных кадрах содержится адрес получателя и отправителя, а также данные для передачи. В неадресованных кадрах передаются данные без указания адресов.
- Режимы работы: I2C поддерживает несколько режимов работы, включая Master и Slave. В режиме Master, устройство управляет передачей данных и генерирует тактовые импульсы. В режиме Slave, устройство принимает данные и использует тактовые импульсы от устройства Master.
- Надежность: I2C обеспечивает надежность передачи данных путем использования контрольных сумм и повторной передачи данных в случае ошибок.

В рамках данного курсового проекта будет использован I2C коннектор модели PCF8574.

Общее описание:

PCF8574 – кремневая ИС выполненная по CMOS-технологии. Посредством двухлинейной двунаправленной шины (I2C-шина) она обеспечивает дистанционное расширение порта ввода - вывода общего назначения в большинстве серий микроконтроллеров. Устройство состоит из 8-битного квазипорта ввода-вывода и интерфейса I2C-шины. PCF8574 имеет низкий потребляемый ток и выходы с регистром-защелкой с высокими характеристиками по току для прямой передачи сигнала на светодиоды. Дополнительно, в устройстве установлена линия прерывания (INT), которая может быть подключена к логике прерывания микроконтроллера. Посылая сигнал прерывания по этой линии, дистанционный ввод - вывод сообщает микроконтроллеру о поступающих на его порты данных, без необходимости поддерживать связь через I2C-шину. Это значит, что PCF8574 может оставаться простым "подчиненным" устройством. PCF8574 и PCF8574A версии отличаются только адресами.

Отличительные особенности:

- Рабочий режим питания - от 2.5 до 6 В
- Низкий ток покоя – максимум 10 мА
- I2C-шина для расширителя параллельного порта
- Выход прерывания с открытым стоком
- Дистанционный 8-битный расширитель ввода – вывода I2C-шины
- Совместимый с большинством микроконтроллеров
- Выходы с регистром-защелкой с высокими характеристиками по току для прямой передачи сигнала на светодиоды
- Адресация на 3 вывода аппаратных адресов для использования до 8 устройств (до 16 устройств при использовании PCF8574A)
- DIP16 или компактный SO16, а также SSOP20-корпуса.

3. Модуль камеры GM-65, сканирующий штрих-коды и QR-коды.

Используя интеллектуальный алгоритм распознавания изображений, этот модуль может легко декодировать все виды распространенных 1D/2D кодов. Через встроенный интерфейс USB и UART его можно подключить напрямую к компьютеру или интегрировать в различные устройства. Модуль конфигурируется с помощью сканирования специально предусмотренных 2D кодов, которые приведены в приложенном PDF файле. Кроме того, благодаря встроенному источнику света этот сканер может хорошо работать даже в темноте.



Рис. 2.1.3 – модуль камеры GM65

Технические характеристики:

- Оптическая система: CMOS
- Разрешение: 648*488
- Источники освещения:
 - LED белый 6500K
 - LED 617 нм
- Угол зрения камеры (FoV): 34° (по горизонтали), 26° (по вертикали)
- Углы сканирования: вращение 360°; тангаж $\pm 60^\circ$; перекося $\pm 65^\circ$
- Минимальная контрастность: 30%
- Напряжение: 5 VDC
- Ток: 120 мА (сканирование), 30 мА (в режиме ожидания)
- Окружающий свет: 0...86000 люкс
- Интерфейс: USB, UART
- Поддерживаемые коды 1D:
 - Codebar
 - Code 11
 - Code 39 / Code 93
 - UPC / EAN
 - Code 128 / EAN128
 - Interleaved 2 of 5
 - Matrix 2 of 5
 - MSI Code
 - Industrial 2 of 5

- GS1 Databar (RSS)
- Поддерживаемые 2D-коды:
 - QR-код
 - матрица данных
 - PDF417
- Рабочая температура: 0...50°C
- Температура хранения: -40...70°C
- Влажность при эксплуатации: 10...80% RH

4. LCD-дисплей будет использован для вывода информации. В качестве дисплея была выбран LCD-дисплей 1602. LCD1602 - это жидкокристаллический дисплей, который состоит из 16 символов в 2 строках. Он может отображать буквы, цифры и символы. Этот дисплей широко используется в микроконтроллерных проектах для отображения информации, такой как статусы, результаты измерений и другие данные.



Рис. 2.1.4 – LCD-дисплей

Характеристики дисплея LCD1602 могут варьироваться в зависимости от конкретной модели, но в целом они включают в себя:

- Разрешение: 16 символов в 2 строках
- Интерфейс: параллельный или последовательный (обычно используется параллельный)
- Размер: примерно 80 мм x 35 мм
- Тип выводимой информации: символьный.
- Язык в ПЗУ дисплея: латиница, японский.
- Возможность загрузки собственных символов: есть.
- Формат выводимой информации: 16×02 символов;
- Тип дисплея: LCD.
- Технология дисплея: STN.
- Угол обзора: 180°.

- Тип подсветки: LED.
- Цвет подсветки: синий.
- Цвет символов: белый.
- Контроллер: HD44780.
- Интерфейс: синхронный, 8-битный, параллельный.
- Напряжение питания 5 В.
- Рабочая температура: -20 ... +70 °С.
- Температура хранения -30 ... +80 °С.
- Габариты: 80х36 мм.

Для управления дисплеем LCD1602 с помощью ESP32 можно использовать библиотеки Arduino IDE, такие как LiquidCrystal, которые облегчают работу с дисплеем, предоставляя простой интерфейс для отправки символов на дисплей и управления подсветкой.

2.1.2. Обоснование связей структурной электрической схемы логического анализатора.

Структурная электрическая схема представляет собой графическое изображение устройства или системы, показывающее взаимосвязь между компонентами и их функциональную схему. В такой схеме компоненты обычно обозначаются простыми геометрическими фигурами, а связи между ними - линиями, отражающими электрические соединения.

Одна из основных задач при разработке структурной электрической схемы - минимизировать количество компонентов и связей между ними, при этом обеспечивая нужную функциональность устройства. Это может быть достигнуто за счет использования multifunctional компонентов, таких как микроконтроллеры, и соединение их пинов с нужными компонентами.

Также при проектировании структурной электрической схемы необходимо учитывать потенциальные электрические помехи и шумы, которые могут возникнуть на связях между компонентами. Для этого используются методы экранирования, фильтрации и правильного размещения компонентов.

Во-первых, нам необходимо связать микроконтроллер и устройство сканирования штрих-кодов. Данная связь является главной, и при ее отсутствии сканер штрих-кодов и QR-кодов не сможет выполнить свой функционал. Данная связь реализована протоколом UART последовательной

связи, который используется для обмена данными между устройствами. Он используется для передачи данных в двух направлениях: от микроконтроллера к устройству (TX) и от устройства к микроконтроллеру (RX). Обмен данными происходит в байтовом формате и синхронизируется по тактовому сигналу, который генерирует микроконтроллер.

UART работает в режиме полудуплекса, что означает, что обмен данными происходит в обе стороны, но только в одном направлении в каждый момент времени. В UART используются две линии связи: TX (transmit) и RX (receive). TX-линия используется для передачи данных, а RX-линия - для их приема.

UART работает на основе тактовой частоты, называемой baudрейтом. Бaudрейт определяет скорость передачи данных. При использовании UART микроконтроллер может быть настроен на определенный baudрейт для обмена данными с другими устройствами.

UART используется для передачи текстовой и бинарной информации, такой как строки текста, числа, символы, команды и т.д. Он широко применяется в различных устройствах, включая компьютеры, телефоны, модемы, GPS-приемники и микроконтроллеры.

В современных микроконтроллерах GPIO-порты могут выполнять различные функции, например, они могут использоваться для работы с UART, SPI, I2C и другими интерфейсами связи. Также порты могут быть настроены на работу с различными типами сигналов, например, аналоговыми или цифровыми.

Во-вторых, необходимо реализовать связь ESP32 и I2C коннектора. Для этого нам потребуется задействовать GPIO-порт ESP32, который мы будем использовать для работы с интерфейсом связи I2C.

В-третьих, I2C коннектор будет связан с LCD-дисплеем, связь будет происходить с помощью I2C протокола.

2.2 Разработка принципиальной схемы декодера штрих-кодов

2.2.1 Рассмотрение различных САПР и обоснование выбора определённой САПР

При выборе программного обеспечения для разработки принципиальной электрической схемы необходимо учитывать несколько

факторов. В первую очередь, важно определить функциональные требования, которые должна удовлетворять выбранная САПР. Также следует учитывать опыт работы и знания разработчика в использовании данного ПО.

В качестве САПР для разработки принципиальной электрической схемы можно выбрать, например, Eagle PCB Design, Altium Designer, KiCAD или Proteus. Каждая из этих программ имеет свои преимущества и недостатки, и выбор конкретного ПО будет зависеть от конкретных требований проекта и предпочтений разработчика.

Eagle PCB Design - это инструмент для разработки электронных схем и печатных плат. Он предоставляет пользователю удобный и интуитивно понятный интерфейс, а также множество возможностей для создания проектов любой сложности.

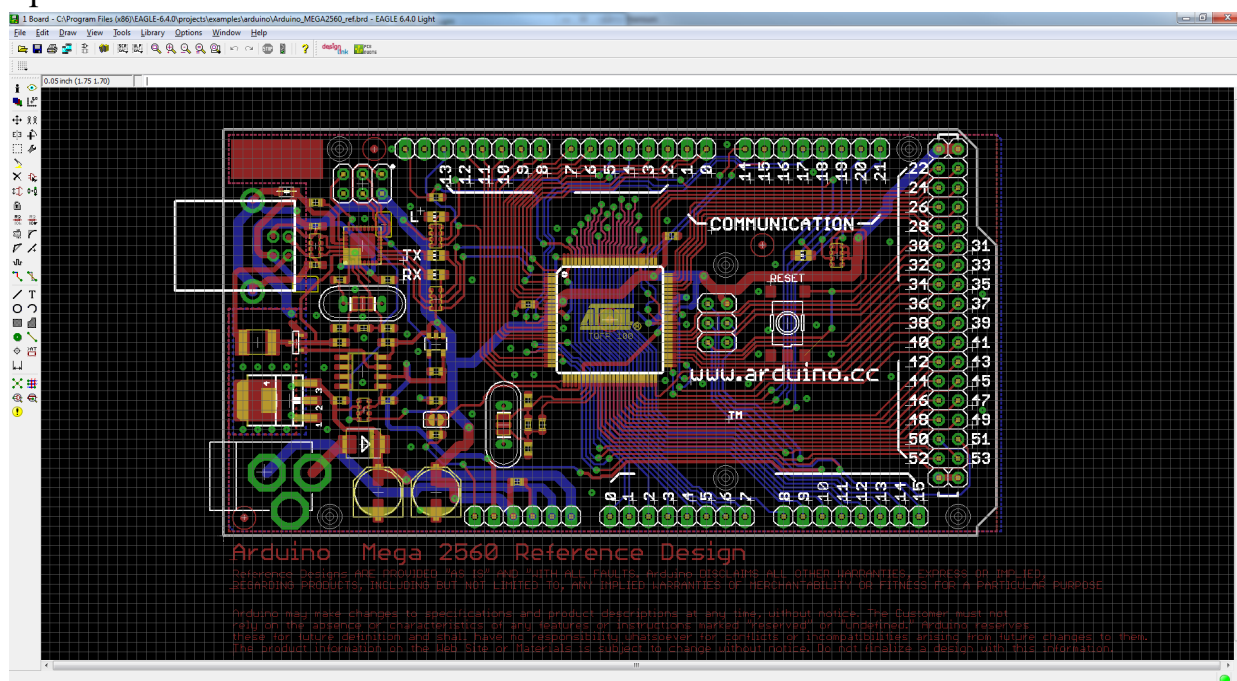


Рис. 2.2.1 – интерфейс Eagle PCB Design

Программа Eagle PCB Design позволяет создавать электрические схемы, размещать компоненты на плате, проводить трассировку печатных проводников, создавать библиотеки компонентов, генерировать файлы для производства печатных плат, а также выполнять другие задачи, связанные с разработкой электронных устройств.

Основные плюсы Eagle PCB Design:

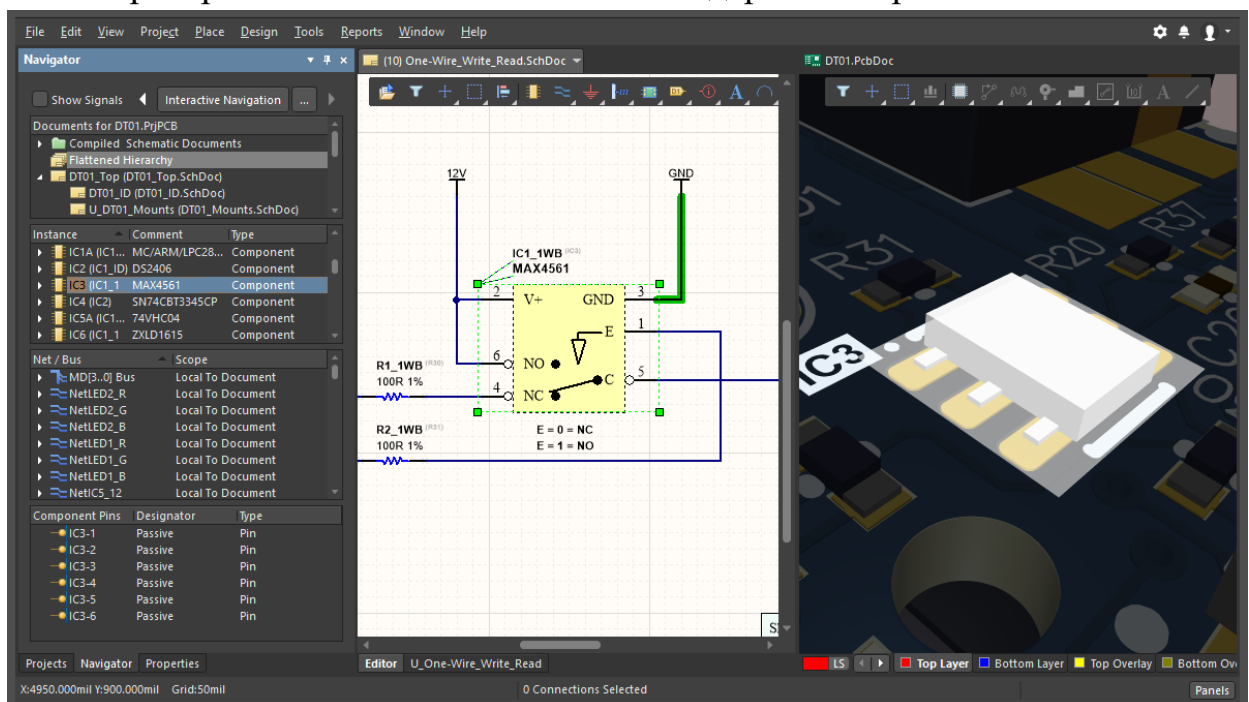
- Простота использования. Интерфейс программы интуитивно понятен и легко настраивается.

- Большое количество библиотек. В программе доступно множество стандартных и пользовательских библиотек, что упрощает работу с различными компонентами.
- Гибкость настроек. Возможность настройки параметров трассировки позволяет удовлетворять требованиям производства.
- Удобство работы с графикой. Программа позволяет легко создавать и редактировать графические элементы.

Основные минусы Eagle PCB Design:

- Платная лицензия. Для полноценного использования программы требуется покупка лицензии.
- Некоторые ограничения бесплатной версии. Бесплатная версия программы ограничена по размеру платы и количеству слоев.
- Ограниченный выбор экспортных форматов. Программа поддерживает только ограниченное количество экспортных форматов файлов.
- Необходимость настройки параметров. Для корректной работы программы требуется настройка ряда параметров, что может быть не очень удобно для начинающих пользователей.
- Отсутствие автоматического анализа ошибок.

Altium Designer – это комплексное программное обеспечение для разработки электронных схем и создания печатных плат. Он предоставляет инструменты для создания принципиальных электрических схем, размещения и трассировки компонентов, создания 3D-моделей плат и компонентов, а также проверки схемы на соответствие стандартам и ограничениям.



Некоторые из основных возможностей Altium Designer:

- Создание принципиальных электрических схем. Altium Designer предоставляет библиотеку компонентов с более чем миллионом элементов, которые можно использовать для создания схемы.
- Размещение и трассировка компонентов. С помощью Altium Designer можно разместить компоненты на плате и соединить их трассами. Программа имеет инструменты для автоматической трассировки, а также возможность ручной трассировки.
- Создание 3D-моделей. Altium Designer позволяет создавать 3D-модели плат и компонентов, что позволяет проводить визуальную проверку совместимости компонентов на плате.
- Проверка на соответствие стандартам и ограничениям. Altium Designer позволяет проверять схему на соответствие стандартам и ограничениям, таким как минимальный зазор между трассами, минимальный размер отверстий и т.д.
- Импорт и экспорт файлов. Altium Designer поддерживает импорт и экспорт файлов из других программных продуктов, таких как Eagle PCB Design, P-CAD, Protel и другие.

Плюсы Altium Designer:

- Комплексность и широкий функционал для разработки электронных схем и создания печатных плат.
- Большая библиотека компонентов.
- Возможность создания 3D-моделей плат и компонентов.
- Проверка на соответствие стандартам и ограничениям.
- Гибкая настройка пользовательского интерфейса.

Минусы Altium Designer:

- Высокая стоимость.
- Высокие требования к аппаратному обеспечению компьютера.
- Сложность использования для начинающих пользователей.
- Большой размер программы.

KiCAD - это свободно распространяемая САПР для проектирования электронных схем и печатных плат. Эта система была создана в 1992 году и разрабатывается сообществом разработчиков по всему миру. KiCAD предоставляет мощные инструменты для создания схематических и графических компонентов, размещения и маршрутизации соединений на печатной плате, а также создания файлов производственного процесса.

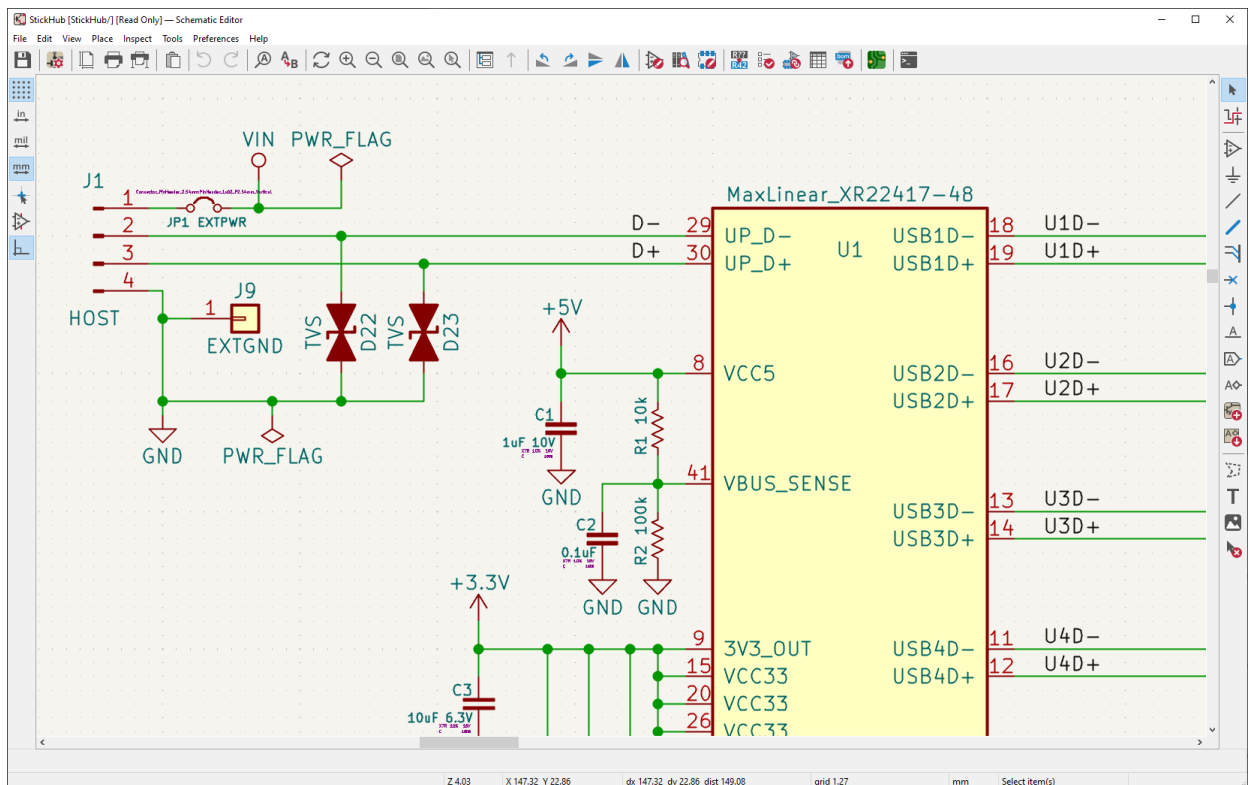


Рис. 2.2.3 – интерфейс KiCAD

Основные модули KiCAD включают в себя:

- Eeschema - модуль для создания схем и управления символами компонентов. Eeschema поддерживает различные виды символов, включая графические и текстовые, а также позволяет создавать свои символы.
- PCBnew - модуль для размещения компонентов и маршрутизации соединений на печатной плате. PCBnew имеет мощный автоматический маршрутизатор и позволяет создавать многоплоскостные платы.
- Gerbview - модуль для просмотра файлов производственного процесса, таких как файлы гербер и сверловки.
- Cvrpcb - модуль для связывания компонентов из Eeschema с физическими пинами компонентов в PCBnew.
- Libraries - библиотеки символов и корпусов компонентов, доступные для использования в проектах.

KiCAD имеет ряд преимуществ и недостатков:

Плюсы:

- Бесплатность и открытый исходный код.
- Мощные инструменты для создания схем и печатных плат.
- Возможность создания многоплоскостных печатных плат.
- Поддержка импорта и экспорта файлов различных форматов.

- ### Минусы:

- В целом, KiCAD - это мощный и гибкий инструмент для разработки электронных схем и печатных плат, который может быть особенно полезен для любительских целей.

Proteus - это программа для симуляции электронных схем, разработки печатных плат и программирования микроконтроллеров. Она позволяет инженерам разрабатывать, отлаживать и проверять свои электронные схемы в виртуальной среде до их физической реализации.

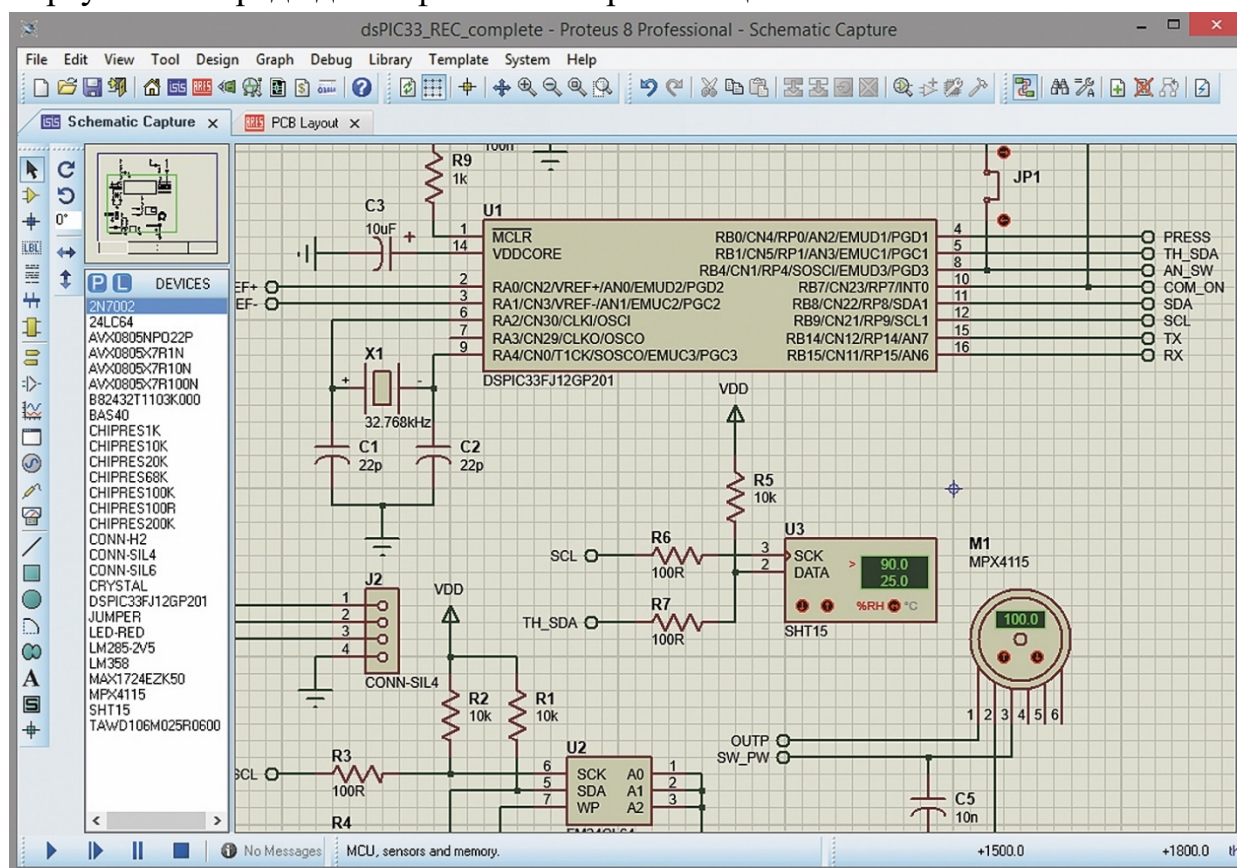


Рис. 2.2.4 – интерфейс Proteus

Основные возможности Proteus:

- Симуляция различных электронных компонентов и устройств;
- Проектирование печатных плат с автоматической трассировкой;
- Работа с микроконтроллерами, включая программирование на языке C;

- Поддержка многопользовательской работы и возможность совместной разработки проектов;
- Возможность интеграции с другими САПР, такими как Altium Designer и Eagle PCB Design.

Proteus имеет удобный и интуитивно понятный пользовательский интерфейс, позволяющий легко создавать и моделировать электронные схемы, а также проектировать печатные платы с помощью многочисленных инструментов. Кроме того, Proteus обладает богатыми возможностями по настройке симуляции и отладки схем, включая возможность анализа временных характеристик, отображение сигналов на осциллографе и многие другие функции.

Среди недостатков Proteus можно отметить высокую стоимость пакета, особенно для коммерческого использования, а также некоторые ограничения в работе с некоторыми типами компонентов. Однако, Proteus остается одной из самых популярных программ для симуляции электронных схем и разработки печатных плат благодаря своей высокой надежности и многофункциональности.

Для курсового проекта был выбран САПР KiCAD.



Рис. 2.2.4 – логотип KiCAD

Вот несколько причин, почему для курсового проекта был выбран KiCAD:

1. Бесплатность: KiCAD доступен бесплатно для скачивания и использования, что делает его доступным для любого пользователя с ограниченным бюджетом.
2. Открытый исходный код: KiCAD является программным обеспечением с открытым исходным кодом, что означает, что пользователи могут настраивать его под свои нужды и добавлять свои собственные функции.
3. Широкий функционал: KiCAD включает в себя инструменты для создания схем, размещения компонентов, трассировки печатных плат, проверки ошибок и экспорта проекта в различные форматы.

4. Большое сообщество: KiCAD имеет большое сообщество пользователей, которые делятся своим опытом и создают библиотеки компонентов, что упрощает процесс проектирования.

В целом, KiCAD является отличным выбором для создания электронных проектов любой сложности, особенно для начинающих и людей с ограниченным бюджетом. Он предоставляет все необходимые инструменты для разработки электронных устройств и обладает высокой гибкостью, что позволяет создавать собственные решения и адаптировать программу к своим потребностям.

2.2.2 Описание используемых библиотечных элементов

Элемент ESP32-WROOM-32D из библиотеки элемент ESP32-WROOM-32. ESP32-WROOM-32D - это Wi-Fi и Bluetooth модуль на базе микроконтроллера ESP32, который разработала компания Espressif Systems. В KiCAD доступна библиотека ESP32-WROOM-32, содержащая различные элементы, символы, макеты и модели печатных плат.

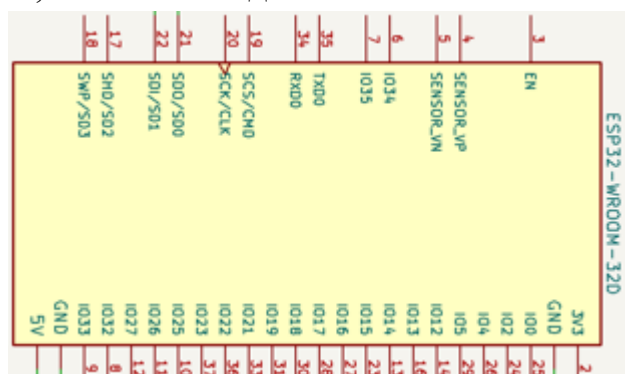


Рис. 2.2.5 – Элемент ESP32-WROOM-32D

Следующий элемент LCD I2C был создан в KiCad конкретно для данного проекта.



Рис. 2.2.6 – Элемент LCD I2C

Аналогично предыдущему элемент GM65 был создан конкретно для данного проекта.

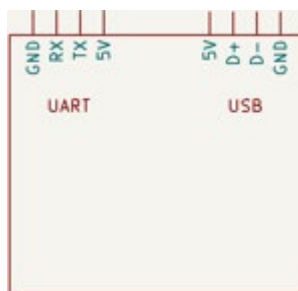


Рис. 2.2.7 – Элемент модуля камеры GM65

3 Разработка алгоритма функционирования мобильного декодера штрих-кодов

3.1 Описание алгоритма функционирования мобильного декодера штрих-кодов

Алгоритм функционирования - это последовательность действий, которые необходимо выполнить для достижения определенной цели. Он может быть описан словесно или в виде блок-схемы, которая представляет собой графическое представление последовательности действий.

Алгоритм функционирования:

1. Включить устройство.
Подключить устройство к питанию посредством использования провода USB.
2. Считать данные штрих-кода.
Навести модуль камеры устройства на штрих-код или QR-код, дождаться окончания считывания, которое обозначится звуковым сигналом из модуля камеры.
3. Обработать полученные данные.
Получить считанные данные из модуля камеры в микроконтроллер, после чего обработать их.
4. Отобразить результат на LCD-дисплее.
Вывести полученные и обработанные данные на LCD-дисплей, при этом данные для вывода с длиной более 16 символов выводить в прокруткой.
5. Выключить устройство.
По завершению всех работ с устройством, отсоединить кабель питания.

3.2 Описание функций алгоритма

Сперва подключим необходимые библиотеки:

```
1  #include <Wire.h>
2  #include <hd44780.h>
3  #include <hd44780ioClass/hd44780_I2Cexp.h>
```

Рис. 3.2.1 – Подключение библиотек

Далее определим переменные и константы:


```

5   hd44780_I2Cexp LCD;
6
7   #define QR Serial1
8
9   String inputString = "";

```

Рис. 3.2.2 – Объявление констант

Далее опишем первоначальную загрузочную функцию, в которой определим подключенный LCD-дисплей, поставим курсор на нулевые координаты, установим соединение с монитором порта, а также определим и настроим порт для сканера:

```

24  void setup()
25  {
26      LCD.begin(16, 2);
27      LCD.setCursor(0, 0);
28      Serial.begin(115200);
29      QR.begin(9600, SERIAL_8N1, 26, 27); //Определяем порт QR
30  }

```

Рис. 3.2.3 – Описание первичной функции

Теперь же бесконечно повторяющуюся функцию, в которой будет происходить получение данных с модуля камеры:

Для начала определим получены ли данные для обработки. Если данные получены, то обнулим нашу переменную `inputString`, после этого очистим экран и отобразим на нём информацию о том, что процесс сканирования запущен. Модуль камеры GM65 предоставляет данные побитово, поэтому создадим цикл, в котором каждый бит поочередно записывается в переменную `inputString`. После того, как код получен, уберём лишний для нас символ, который модуль камеры передал по завершению своей работы. Далее определим размер полученного штрих-кода или QR-кода, и если полученный код меньше 16 символов, то очистим экран, выставим курсор и выведем полученный код на экран. Если же полученный код имеет длину более 16 символов, то запустим функцию `scrollMessage`, в которую передадим в качестве параметров наш полученный код и время задержки в миллисекундах.

```

32 void loop()
33 {
34     if (QR.available()) // Проверяем, есть ли входящие данные в последовательном буфере
35     {
36         inputString = "";
37         LCD.clear();
38         LCD.print("Scanning...");
39         delay(1000);
40         while (QR.available()) // Читаем байт за байтом из буфера, пока буфер не станет пустым
41         {
42             char input = QR.read();// Читаем 1 байт данных и сохраняем его в переменную
43             inputString += input;
44         }
45         inputString = inputString.substring(0, inputString.length() - 1);
46         if (inputString.length() <= 16) {
47             LCD.clear();
48             LCD.setCursor(0, 0);
49             LCD.print(inputString);
50         }
51     }
52     if (inputString.length() > 16) {
53         scrollMessage(inputString, 400);
54     }
55 }

```

Рис. 3.2.4 – Описание бесконечной функции

Теперь опишем функцию `scrollMessage`. Задача этой функции – скролл полученных кодов длиной более 16 символов для корректного отображения их на LCD-дисплее. Принимает данная функция строковую переменную `inputString`, в которую записан полученный ранее код, а также время задержки `delayTime` в миллисекундах для настройки скорости скролла.

В функции создадим цикл, имеющий длину кода с вычитанием 16 символов, которые может сразу отобразить дисплей. В нём создадим условие, проверяющее ведётся ли получения данных с модуля камеры в данный момент, и, если таковое имеется, моментально возвращающее нас в бесконечную функцию.

Для объяснения необходимости данного условия представим ситуацию, в которой пользователь случайно просканировал не тот код, который хотел, тогда, чтобы не ждать вывода нежеланного кода на экран с последующей прокруткой, мы вводим данное условие, отправляющее программу обратно в бесконечную функцию, что позволяет пользователю сканировать желаемый код не дожидаясь вывода его с последующим скроллингом на экран.

Далее по циклу следует очистка экрана и выставления курсора на нулевые координаты. После это начинается вывод штрих-кода или QR-кода со скроллингом имеющим ранее выставленную задержку `delayTime`.

```

11 void scrollMessage(String inputString, int delayTime)
12 {
13   for(int position = 0; position <= (inputString.length() - 16); position++) {
14     if (QR.available()) {
15       return;
16     }
17     LCD.clear();
18     LCD.setCursor(0, 0);
19     LCD.print(inputString.substring(position, position + 16));
20     delay((position == 0 || position == inputString.length() - 16 ? 3 : 1) * delayTime);
21   }
22 }

```

Рис. 3.2.5 – Описание функции скроллинга выводимого кода

Заключение

В ходе разработки курсового проекта был произведён анализ предметной области. Изучены особенности написания программного обеспечения для микроконтроллера ESP32-WROOM-32D. Рассмотрены и описаны различные отладочные платы, которые могли бы использоваться в данном проекте.

Были рассмотрены различные виды САПР, таких как Eagle PCB Design, Altium Designer, KiCAD или Proteus. Так же изучена САПР KiCad для проектирования принципиальных электрических схем. Разработаны структурная, принципиальная электрические схемы мобильного декодера штрих-кодов и QR-кодов.

Была составлена блок-схема алгоритма работы программы декодера штрих-кодов и QR-кодов. На основе блок-схемы был разработан софт, который далее был загружен в отладочную плату ESP32. На основании структурной и принципиальной электронных схем было функционирующее устройство

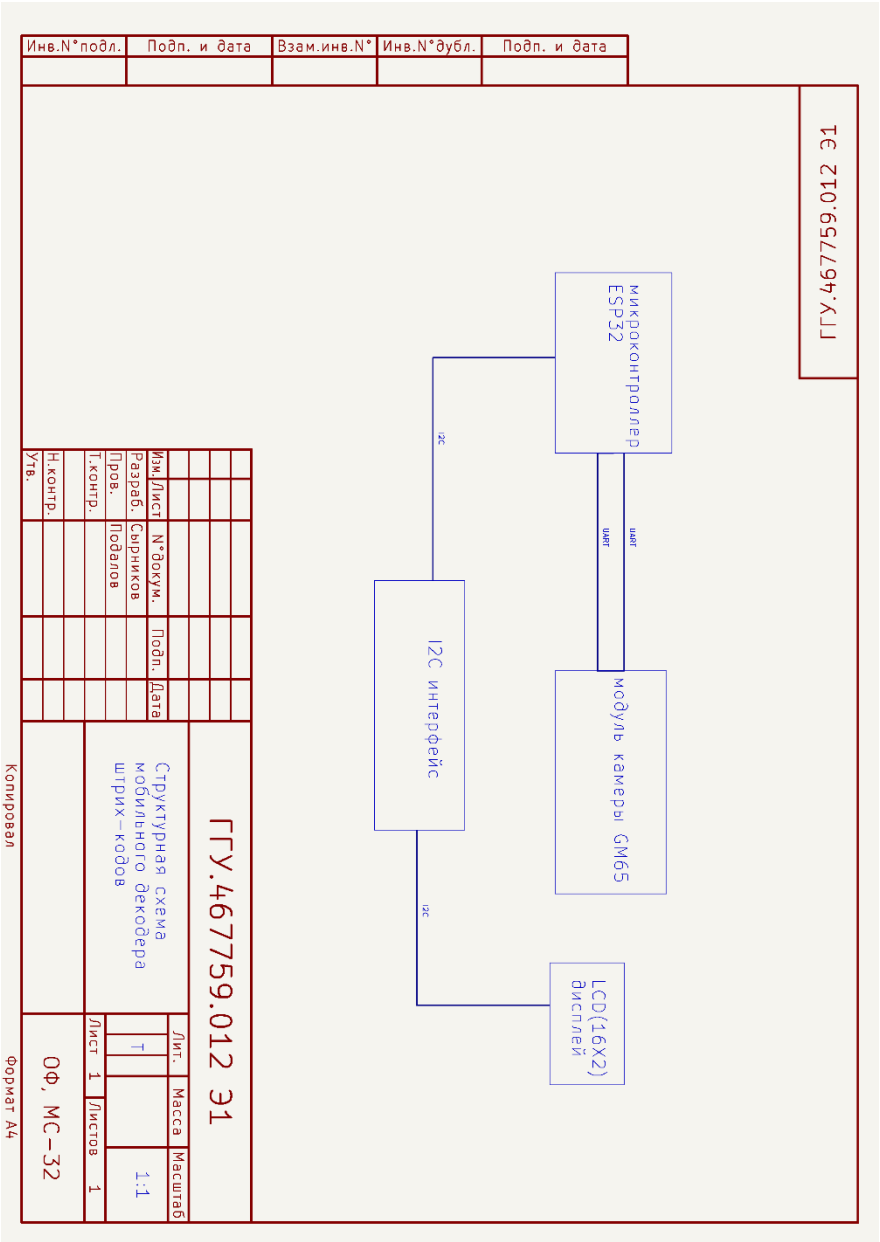
Таким образом, был разработан мобильный декодер штрих-кодов и QR-кодов, позволяющий сканировать и выводить на LCD-дисплей декодированные штрих-коды и QR-коды.

Список используемых источников

1. GM65 [Электронный ресурс] // User manual gm65. – URL: http://www.microtechnica.tv/support/manual/brm65_man.pdf. – Дата доступа: 10.04.2023.
2. KiCAD [Электронный ресурс] // KiCAD. – URL: <https://www.kicad.org/>. – Дата доступа: 11.04.2023.
3. Arduino IDE [Электронный ресурс] // Arduino IDE software. – URL: <https://www.arduino.cc/en/software>. – Дата доступа: 11.04.2023.
4. C++ [Электронный ресурс] // Свободная энциклопедия Википедия. – URL: <https://ru.wikipedia.org/wiki/C%2B%2B>. – Дата доступа: 11.04.2023.
5. I2C [Электронный ресурс] // Свободная энциклопедия Википедия. – URL: <https://ru.wikipedia.org/wiki/I%C2%B2C>. – Дата доступа: 10.04.2023.
6. Arduino [Электронный ресурс] // Свободная энциклопедия Википедия. – URL: <https://ru.wikipedia.org/wiki/Arduino>. – Дата доступа: 04.04.2023.
7. Github [Электронный ресурс] // Github – URL: <https://github.com/Asyrnikov/KYRS>. – Дата доступа: 16.05.2023.

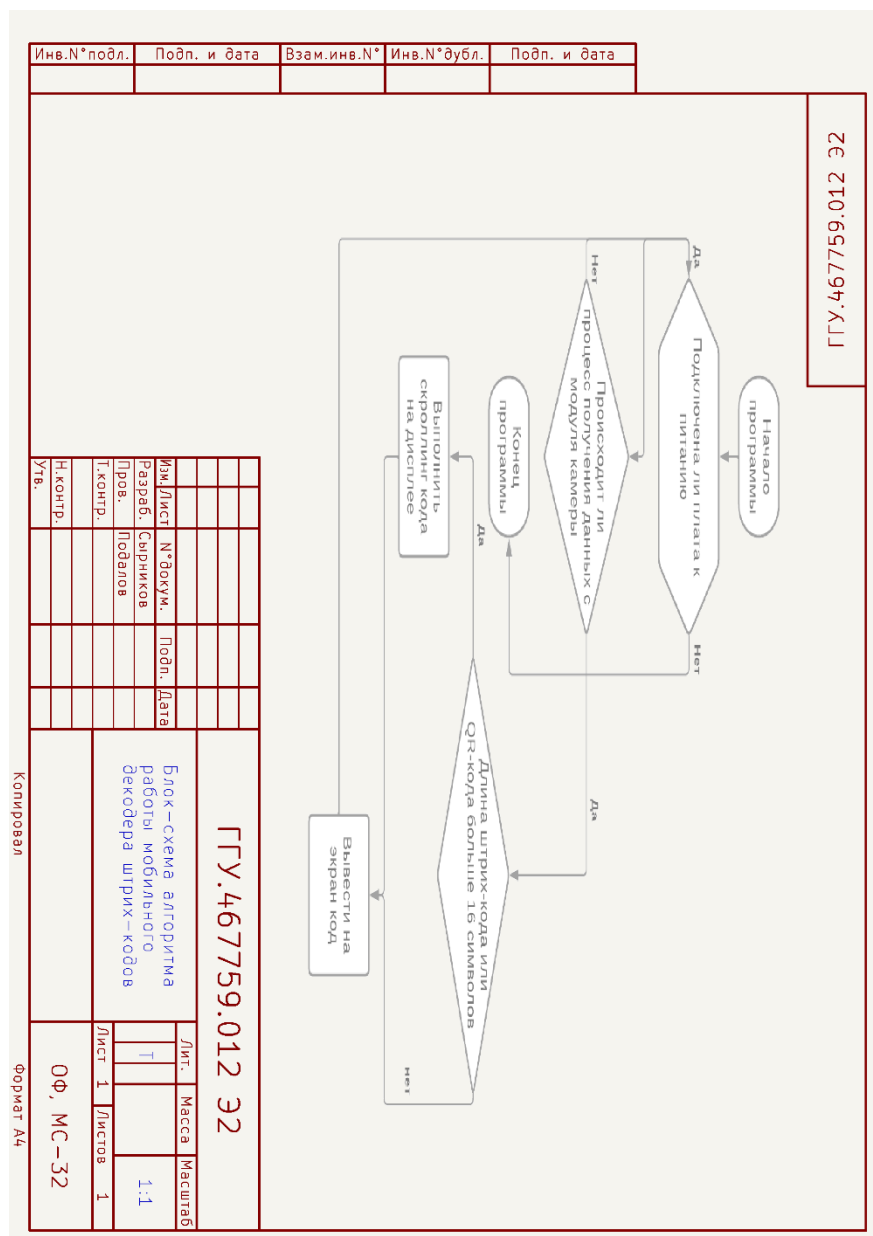
Приложение А

Структурная схема мобильного декодера штрих-кодов



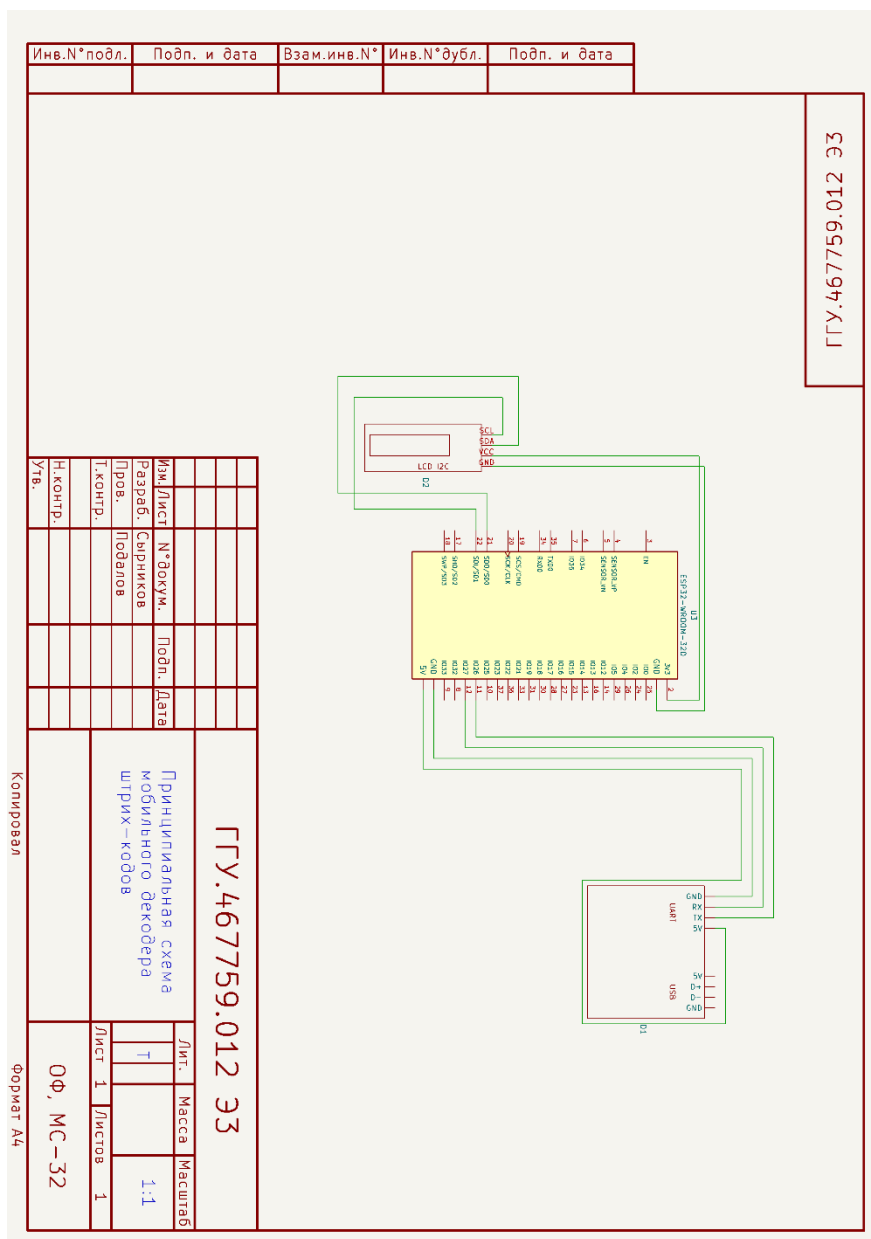
Приложение Б

Блок-схема алгоритма мобильного декодера штрих-кодов



Приложение В

Принципиальная схема мобильного декодера штрих-кодов



Приложение Г

Код алгоритма программы мобильного декодера штрих-кодов

```
#include <Wire.h>
#include <hd44780.h>
#include <hd44780ioClass/hd44780_I2Cexp.h>

hd44780_I2Cexp LCD;

#define QR Serial1

String inputString = "";

void scrollMessage(String inputString, int delayTime)
{
    for(int position = 0; position <= (inputString.length() - 16); position++) {
        if (QR.available()) {
            return;
        }
        LCD.clear();
        LCD.setCursor(0, 0);
        LCD.print(inputString.substring(position, position + 16));
        delay((position == 0 || position == inputString.length() - 16 ? 3 : 1) *
delayTime);
    }
}

void setup()
{
    LCD.begin(16, 2);
    LCD.setCursor(0, 0);
    Serial.begin(115200);
    QR.begin(9600, SERIAL_8N1, 26, 27); //Определяем порт QR
}

void loop()
{
    if (QR.available()) // Проверяем, есть ли входящие данные в последовательном
буфере
    {
        inputString = "";
        LCD.clear();
        LCD.print("Scanning...");
        delay(1000);
    }
}
```

```

    while (QR.available()) // Читаем байт за байтом из буфера, пока буфер не
станет пустым
    {
        char input = QR.read();// Читаем 1 байт данных и сохраняем его в
переменную
        inputString += input;
    }
    inputString = inputString.substring(0, inputString.length() - 1);
    if (inputString.length() <= 16) {
        LCD.clear();
        LCD.setCursor(0, 0);
        LCD.print(inputString);
    }
}
if (inputString.length() > 16) {
    scrollMessage(inputString, 400);
}
}

```

Спецификация к курсовому проекту

Перв. примен.	Справ. №	Формат	Зона	Поз.	Обозначение	Наименование	Кол.	Примечание
ГГУ.467759.012								
		A4		ГГУ.467759.012 31	Схема электрическая структурная	1		
		A4		ГГУ.467759.012 33	Схема электрическая принципиальная	1		
		A4		ГГУ.467759.012 32	Блок-схема алгоритма	1		
Справ. №								
Подп. и дата								
Инв. № дубл.								
Взам. инв. №								
Подп. и дата								
Инв. № подл.								
Инв. № подл.								
Инв. № подл.								
Инв. № подл.								
Инв. № подл.								
Инв. № подл.								
Инв. № подл.								
Инв. № подл.								
Инв. № подл.								
Инв. № подл.								
Инв. № подл.								
Инв. № подл.								
Инв. № подл.								
Инв. № подл.								
Инв. № подл.								
Инв. № подл.								
Инв. № подл.								
Инв. № подл.								
Инв. № подл.								
Инв. № подл.								
Инв. № подл.								

Ведомость документов к курсовому проекту

[illegible]