**РЕФЕРАТ**

ЭЛЕКТРОННАЯ КНИГА НА БАЗЕ ESP32: дипломный проект / С. А. Каленик. – Минск : БГУИР, 2022, – п.з. – 79 с., чертежей (плакатов) – 6 л. формата А1

Дипломный проект посвящен разработке и реализации электронной книге на базе ESP32. В нем будут рассмотрены такие функции как: вывод текста книги на экран, обработка пользовательского нажатия на сенсорную панель, загрузка и запись книг в память, отображение меню выбора книг на экран, отображение меню настроек для выбора размера шрифта.

Электронная книга на базе ESP32 с помощью встроенного микроконтроллера проверяет SD-карту на наличие книг с поддерживаемым форматом, после чего выводит их названия на экран, чтобы пользователь мог выбрать необходимую ему книгу и далее начать читать выбранную книгу.

Разработанная система может быть рекомендована людям любящим читать книги в электронном формате и желающим иметь портативное устройство, которое легко взять с собой.

Ключевые слова: электронная книга, ESP32, Wi-Fi, экран, сенсорная панель, SD-карта.

CОДЕРЖАНИЕ

[Введение 7](#_Toc103990540)

[1 Обзор аналогичных разработок 8](#_Toc103990541)

[1.1 The Open Book Project 8](#_Toc103990542)

[1.2 ESP32 Based ePub Reader 9](#_Toc103990543)

[1.3 Электронная книга на базе STM32H750 11](#_Toc103990544)

[1.4 Вывод по результатам обзора аналогичных разработок 13](#_Toc103990545)

[2 Анализ технического задания 14](#_Toc103990546)

[3 Разработка структуры электронной книги на базе ESP32 15](#_Toc103990547)

[3.1 Разбиение системы на модули 15](#_Toc103990548)

[3.2 Выбор соотношения между аппаратными программными средствами 18](#_Toc103990549)

[3.3 Построение и описание структурной схемы 18](#_Toc103990550)

[4 Аппартано-программная реализация электроной книги на базе ESP32 20](#_Toc103990551)

[4.1 Разработка алгоритма работы электронной книги на базе ESP32 20](#_Toc103990552)

[4.1.1 Разработка схемы алгоритма работы системы и программы 20](#_Toc103990553)

[4.1.2 Описание алгоритма работы системы и программы 20](#_Toc103990554)

[4.2 Разработка печатного узла электронной книги на базе ESP32 23](#_Toc103990555)

[4.2.1 Особенности применяемой элементной базы 23](#_Toc103990556)

[4.2.2 Особенности применяемых материалов 30](#_Toc103990557)

[4.2.3 Оценка потребляемой мощности 30](#_Toc103990558)

[4.2.4 Оценка токов, протекающих в некоторых цепях 32](#_Toc103990559)

[4.2.5 Расчет минимальной ширины проводника для цепи 3,3 В 33](#_Toc103990560)

[4.2.6 Расчет минимальной ширины проводника для цепи 5 В 33](#_Toc103990561)

[4.2.7 Выбор типоразмера печатной платы 34](#_Toc103990562)

[4.2.8 Расчет печатного монтажа 35](#_Toc103990563)

[4.2.9 Особенности применяемых пакетов САПР 38](#_Toc103990564)

[4.2.10 Решение задачи топологического синтеза печатной платы с помощью применяемого пакета САПР 38](#_Toc103990565)

[4.2.11 Оценка качества разработанной конструкции 39](#_Toc103990566)

[5 Технико-экономическое обоснование эффективности разработки и использования электронной книги на базе ESP32 42](#_Toc103990567)

[5.1 Характеристика электронной книги 42](#_Toc103990568)

[5.2 Формирование отпускной цены электронной книги 42](#_Toc103990569)

[5.2.1 Расчет прямых затрат на комплектующие изделия для электронной книги 42](#_Toc103990570)

[5.2.2 Расчет накладных расходов 43](#_Toc103990571)

[5.2.3 Расчет полной себестоимости 43](#_Toc103990572)

[5.2.4 Расчет плановой прибыли 44](#_Toc103990573)

[5.2.5 Расчет отпускной цены изделия 44](#_Toc103990574)

[5.2.6 Формирование отпускной цены электронной книги на основе полной себестоимости электронной книги 44](#_Toc103990575)

[5.3 Расчет экономического эффекта от производства и реализации новых изделий 45](#_Toc103990576)

[5.4 Расчет инвестиций в производство электронной книги 45](#_Toc103990577)

[5.4.1 Инвестиции в разработку электронной книги 45](#_Toc103990578)

[5.4.2 Инвестиции в прирост основного капитала 47](#_Toc103990579)

[5.4.3 Инвестиции в прирост собственного оборотного капитала 47](#_Toc103990580)

[5.5 Расчет показателей экономической эффективности инвестиций в производство электронной книги 48](#_Toc103990581)

[5.6 Результаты расчета 49](#_Toc103990582)

[6 Анализ результатов проектирования эектронной книги на базе ESP32 50](#_Toc103990583)

[6.1 Макетирование устройства 50](#_Toc103990584)

[6.2 Тестирование устройства 52](#_Toc103990585)

[Заключение 56](#_Toc103990586)

[Список используемых источников 57](#_Toc103990587)

Введение

Электронная книга – это устройство, использующее экран на электронных чернилах для отображения текста. В отличие от обычных матричных экранов, текст не пропадает при подаче питания, а также не требуется свет для отображения. Благодаря этому уменьшается нагрузка на глаза и читателю приятней использовать электронную книгу, чем, например, телефон. Еще одной отличительной чертой данного типа устройств является низкое энергопотребление, поскольку электричество тратиться лишь на смену текста на экране, в отличие от обычных матричных экранов, в которых включенный экран постоянно потребляет много энергии [[1]](#wikiEreader).

В данном дипломном проекте рассматривается задача реализации электронной книги на базе микроконтроллера *ESP32* и *E-ink* дисплея с сенсорной панелью.

Электронная книга выполняет следующие действия:

– отображает текст книги на экране;

– изменяет страницу в зависимости от того, на какую сторону дисплея нажал пользователь;

– визуально отображает номер текущей и последней страницы;

– дает возможность пользователю выбрать необходимую ему книгу из загруженных в памяти книг;

– также присутствует возможность загрузить книгу посредством *telegram* бота;

– возможность изменять размер шрифта.

В дипломном проекта приведена структурная схема устройства и её описание, а также обобщённый алгоритм функционирования устройства. Проверка работоспособности устройства выполнена опытным путём.

Была проведена проверка на плагиат. Индивидуальность данной работы составляет 97,7%. Отчёт представлен в приложении A.

# Обзор аналогичных разработок

## The Open Book Project

*The Open Book* [2] — это совместимая с *E-ink* экранами плата на базе *SAMD51*, основанная на базе *Adafruit PyBadge* [3]. В дополнение к стандартным функциям данная разработка поддерживает различные периферийные устройства, предназначенные для создания доступного универсального устройства для чтения текста.

На рисунке 1.1 приведено изображение внешнего вида *The Open Book Project*.

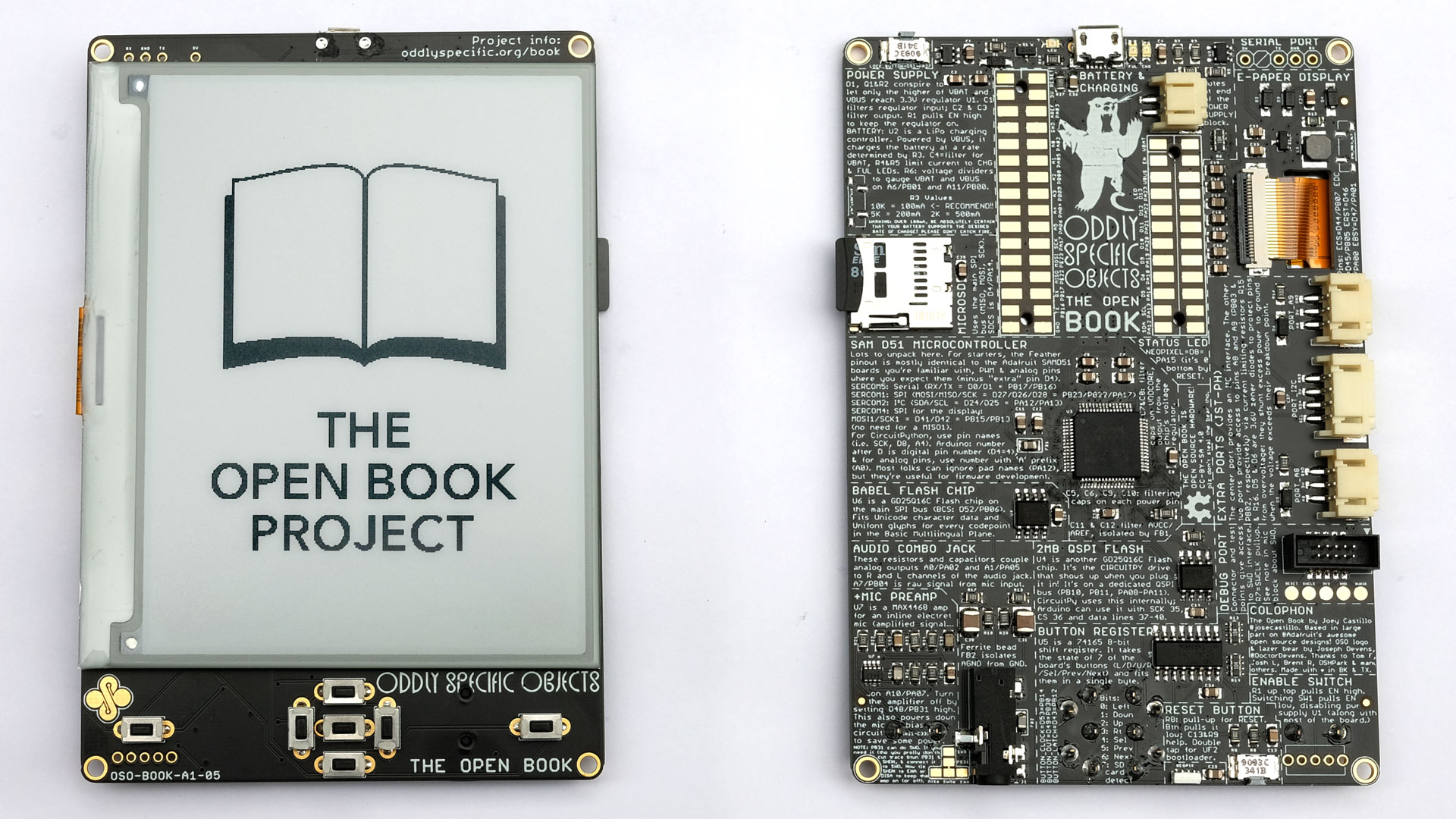


Рисунок 1.1 – Внешний вид The Open Book Project

Основные характеристики:

– 4,2-дюймовый дисплей электронной бумаги с возможностью частичного обновления, управляемый по выделенной шине *SPI*;

– семь кнопок (клавиши направления, кнопка выбора и кнопки перелистывания страниц), доступные через сдвиговый регистр, а также кнопка сброса, подключенная к контакту прерывания на *SAMD51*;

– cлот *MicroSD* на основной шине *SPI*;

– светодиоды для индикации этапов зарядки;

– флэш-чип емкостью 2 МБ;

– доступ к двум цифровым или аналоговым контактам через *STEMMA*-совместимые порты с защитой по току и напряжению;

– доступ к интерфейсу *I2C* через третий *STEMMA*-совместимый порт;

– мониторы напряжения как на аккумуляторе, так и на порте питания *USB*;

– комбинированный аудиоразъем со стереофоническим аудиовыходом и необработанным микрофонным входом;

– микрофонный усилитель *MAX4468* ​​для усиленного микрофонного входа.

Как следует из названия, предназначение описанной платы заключается в создании доступного устройства с открытым исходным кодом для чтения книг. Каждое периферийное устройство было выбрано для выполнения конкретной задачи: экран для отображения слов, *MicroSD* для хранения текстов для чтения; *Flash*-чип для языковой поддержки; разъем для наушников для воспроизведения аудиокниг; вход для микрофона для голосового управления.

К сильным сторонам данной разработки можно отнести следующее:

– поддержка аудиокниг;

– микрофонный вход для голосового управления;

– большое количество кнопок;

– поддержка *SD*-карты;

– поддержка формата *FB2*.

К слабым сторонам данной разработки можно отнести следующее:

– дороговизна аппаратной части;

– маленькая ёмкость памяти;

– маленький экран;

– отсутствие сенсорной панели.

## ESP32 Based ePub Reader

Данная разработка представляет собой электронную книгу с поддержкой формата *ePub* на базе существующих аппаратных платформ, таких как *EPDiy* *V6, M5 Epaper* и *Lilygo EPD* [4].

Проект имеет ограниченную поддержку форматирования — *CSS*-содержимое файла *ePub* не анализируется, поэтому просто используются стандартные теги *HTML*, такие как *<h1>, <b>, <i>*.

Пользователю доступно четыре вида начертания текста: стандартный, полужирный, курсив и полужирный курсив. Шрифты поддерживают только латинские символы и знаки препинания.

В описанной разработке файлы *ePub* рассматриваются как zip-архивы, содержащие несколько файлов. Для чтения файлов используется zip-библиотека, модифицированная для работы на *ESP32* с *PSRAM*.

Для управления устройством пользователю доступно три кнопки:

1. Кнопка «вверх» осуществляет переход вверх по списку файлов или на предыдущую страницу при чтении.
2. Кнопка «вниз» осуществляет переход вниз по списку файловили на следующую страницу при чтении.
3. Кнопка «выбор» открывает текущий выбранный файл *ePub* или возвращается к списку файлов из режима чтения.

Также у пользователя есть возможность выбирать и перелистывать книги при помощи сенсорной панели.

На рисунке 1.2 приведено изображение внешнего вида *ESP32 Based ePub Reader*.

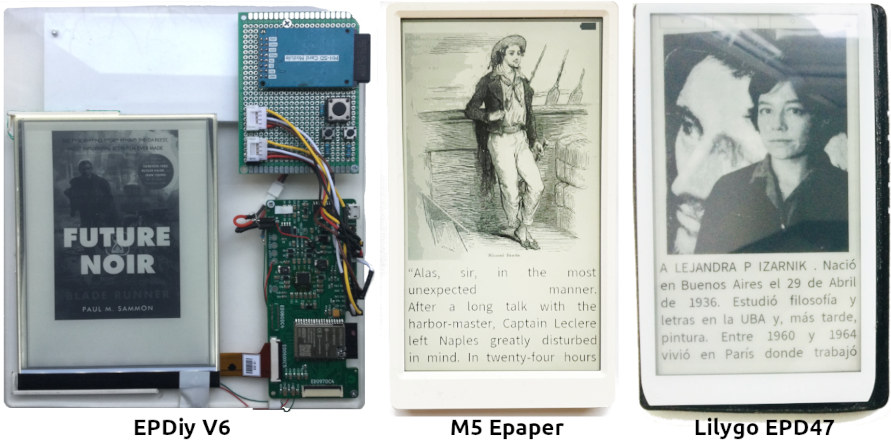


Рисунок 1.2 – Внешний вид ESP32 Based ePub Reader

Данная разработка обладает следующими преимуществами:

– поддержка нескольких аппаратных платформ;

– поддержка формата *ePub*;

– возможность отображения изображений;

– поддержка глубокого сна;

– сенсорное управление.

К недостаткам данной разработки можно отнести следующее:

– поддержка только латинских символов;

– отсутствие поддержки *FB2*;

– долгий рендеринг сложных страниц;

– отсутствие поддержки *SD*-карты.

## Электронная книга на базе STM32H750

Разработанная система предоставляет следующие возможности [5]:

– поддержка формата *FB2*;

– считывание файлов с *SD*-карты;

– поддержка кириллицы;

– поддержка курсивного и полужирного начертания символов;

– для управления используется четыре кнопки: «вперед(вверх)», «назад(вниз)», «меню», «функция»;

– различная обработка длинных и коротких нажатий на кнопки;

– низкое энергопотребление;

– корпус из дерева.

Электронная книга состоит из следующих компонентов:

– экран *E-ink* — *Waveshare* *7.5* *inch E-ink raw display* *800×480*;

– микроконтроллер *STM32H750VBT6*;

– два энергонезависимых хранилища для запоминания служебной информации (*EEPROM* — *AT24C02D* и *FLASH* — *W25Q64)*.

Навигация по пунктам меню осуществляется при помощи светодиодов. То есть отображенные на экране списки для выбора (файлы, список действий) статичными, и при навигации по ним перерисовка экрана не осуществляется, а выбор производится включением светодиода напротив выбранного пункта списка.

Основное состояние электронное книги – выключенное. Включение производится только для обновления экрана, для это реализован менеджер питания, который запускается от внешнего импульса, работает несколько секунд, после чего отключает шину питания. Если внешний импульс повторить до отключения, то работа продолжается на эти самые несколько секунд.

На рисунке 1.3 приведено изображение внешнего вида электронной книги на базе *STM32H750*.

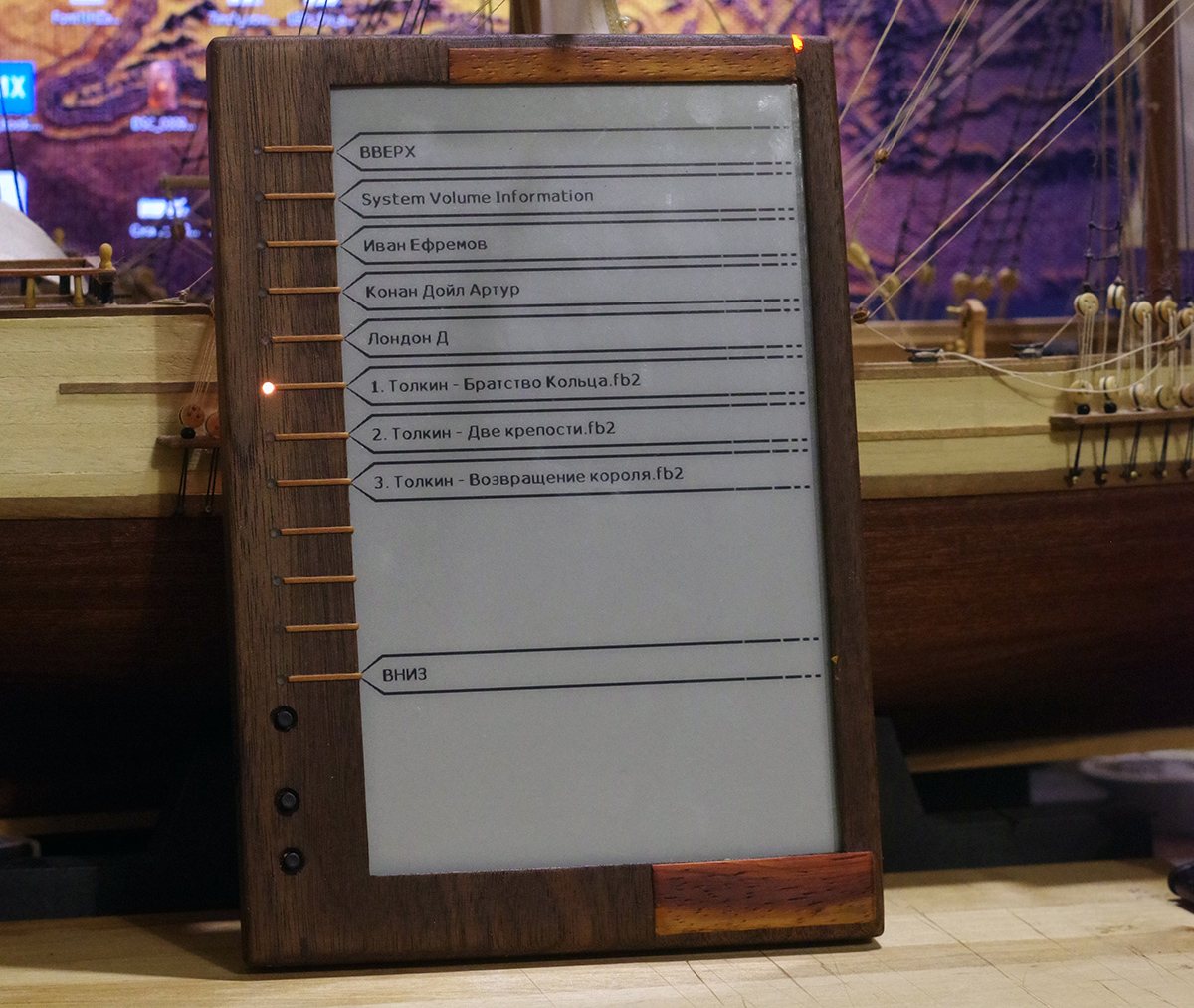


Рисунок 1.3 – Внешний вид электронной книги на базе *STM32H750*

К сильным сторонам данной разработки можно отнести следующее:

– большой экран;

– хороший корпус;

– поддержка кириллицы;

– поддержка формата *FB2*.

К слабым сторонам данной разработки можно отнести следующее:

– отсутствие сенсорной панели;

– низкое разрешение экрана;

– отсутствие серого цвета, есть только черный и белый;

– отсутствие поддержки *SD*-карты.

## Вывод по результатам обзора аналогичных разработок

Итоговые характеристики для сравнения аналогичных разработок приведены в таблице 1.1:

Таблица 1.1 – Сравнение аналогичных разработок

|  |  |  |  |
| --- | --- | --- | --- |
| Характеристика | The Open Book Project | ESP32 Based ePub Reader | Электронная книга на базе STM32H750 |
| Сенсорная панель | Нет | Да | Нет |
| Поддержка SD-карты | Да | Нет | Да |
| Поддержка кириллицы | Нет | Нет | Да |
| Поддержка формата FB2 | Да | Нет | Да |
| Поддержка формата ePub | Нет | Да | Нет |
| Поддержка серого цвета | Да | Да | Нет |

Исходя из данных, приведенных в таблице 1.1, следует, что каждая разработка обладает своими преимуществами и недостатками. В рамках дипломного проектирования были учтены преимущества и недостатки аналогичных разработок, на основании которых были выбраны функциональные и аппаратные возможности для разрабатываемого устройства.

Разработанная электронная книга поддерживает управление при помощи сенсорной панели и использование *SD*-карты. Также она обеспечивает поддержку кириллицы и серого цвета.

# Анализ технического задания

В данном дипломном проекте разработана и реализована система для чтения электронных книг на базе микроконтроллера *ESP32* и *E-ink* экрана с сенсорной панелью.

Электронная книга обладает следующими функциональными возможностями:

– отображение текста книги на экране;

– изменение текущей отображаемой на экране страницы;

– отображение номера текущей страницы, а также общего количества страниц;

– выбор необходимой книги из списка загруженных в память книг;

– загрузка книги в память посредством telegram бота;

– изменение размера шрифта отображаемой книги.

Для выполнения дипломного проекта потребовалось приобрести следующие электрические компоненты:

*–* LilyGo T5-4.7 inch E-Paper ESP32;

– LilyGo T5-4.7 inch Capacitive Touch Cover;

– LilyGo T5-4.7 inch T-FH Interface RF Card Expansion Module;

– аккумулятор 18650.

Все вышеперечисленные электрические компоненты соответствуют условиям эксплуатации, указанным в техническом задании.

На основании требований технического задания составлена функциональная спецификация. Таблица 2.1 представляет собой функциональную спецификацию, разделенную по категориям: входы, выходы и функции.

Таблица 2.1 – Функциональная спецификация электронной книги

|  |  |  |
| --- | --- | --- |
| Входы | Выходы | Функции |
| Wi-Fi модуль | Wi-Fi модуль | Загрузка книг через телеграм бота; вывод состояния книги |
|  | E-ink дисплей | Вывод текста книги и вывод состояния |
| Сенсорная панель |  | Ввод нажатий пользователя |
| SD-карта | SD-карта | Чтение и запись книг |

# Разработка структуры электронной книги на базе ESP32

## Разбиение системы на модули

На основе функциональной спецификации определён набор модулей, реализующих выполняемые в системе функции.

Разработанное устройство содержит следующие аппаратные модули:

– процессорный модуль, используемый для обработки информации;

– модуль памяти микроконтроллера, используемый для хранения памяти программ и данных;

– модули интерфейсов ввода/вывода, используемые для связи процессорного модуля с другими модулями системы;

– модули преобразования входного и выходного сигнала, используемые для обмена входными и выходными сигналами с внешним окружением;

– модуль внешней памяти, используемый для хранения данных пользователя.

На рисунке 3.1 приведена модульная структура аппаратных средств электронной книги.



Рисунок 3.1 – Модульная структура аппаратных средств электронной книги

Также были определены следующие программные модули: исполнительный модуль, модуль ожидания, модуль telegram бота, модуль проверки, модуль рисования, интернет модуль, модуль выключения, модуль памяти, входной и выходной модули.

На рисунке 3.2 приведена функционально-модульная структура электронной книги.



Рисунок 3.2 –Функционально-модульная структура электронной книги

Распределение функций по модулям электронной книги выглядит следующим образом:

1. Исполнительный модуль:

– управление системой.

1. Модуль ожидания:

– ожидание нажатия кнопки включения;

– ожидание нажатия на дисплей;

– ожидание новых сообщений telegram боту.

1. Модуль telegram бота:

– изменение настроек устройства;

– загрузка книг;

– вывод книг присутствующих в памяти устройства посредством общения с пользователем через telegram бота.

1. Модуль проверки:

– проверка на нажатие кнопки включения;

– проверка наличия *SD*-карты;

– проверка входящих данных;

– проверка состояния подключения *Wi-Fi*.

1. Модуль рендеринга:

– рендеринг текста и картинок на экране.

1. Интернет модуль:

– установка связи и обработка *HTTP* запросов для приема и отправки данных.

1. Модуль выключения:

– выключение всех светодиодов;

– выключение дисплея;

– сброс данных в энергозависимой памяти;

– переход микроконтроллера в режим глубокого сна.

1. Входной модуль:

– прием данных по *Wi-Fi*;

– считывание нажатий на дисплей;

– считывание нажатия кнопки включения.

1. Модуль памяти:

– чтение и запись данных в память *SD*-карты или в энергонезависимую память микроконтроллера.

1. Выходной модуль:

– отправление данных по *Wi-Fi*.

На рисунке 3.3 приведена полная функционально-модульная структура электронной книги.

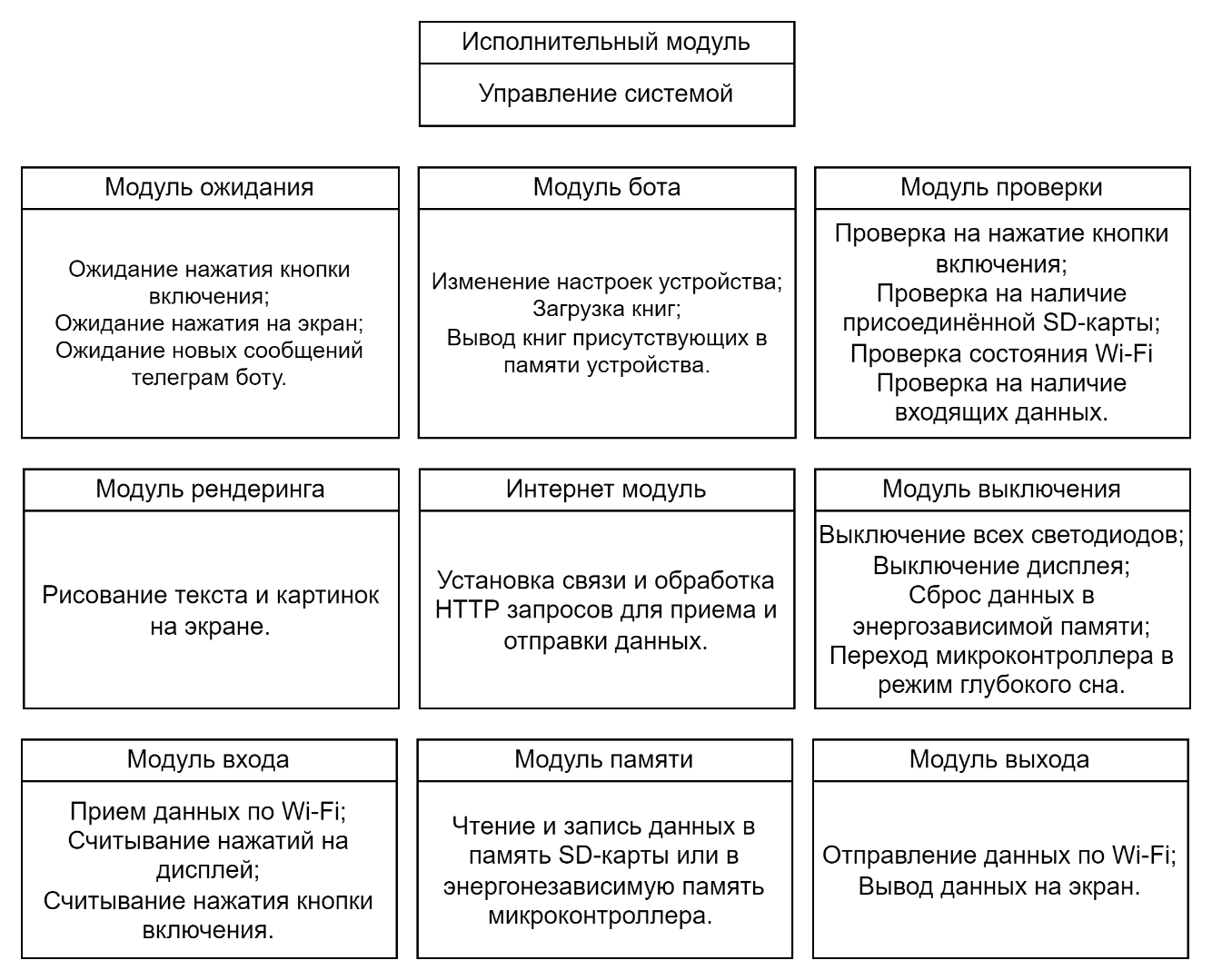


Рисунок 3.3 – Полная функционально-модульная структура электронной книги

## Выбор соотношения между аппаратными программными средствами

Функция управления системы реализуется микроконтроллером в результате выполнения основной (управляющей) программы путем последовательного вызова функций, соответствующих программным модулям системы.

Связь между программными и аппаратными средствами представлена на рисунке 3.4.

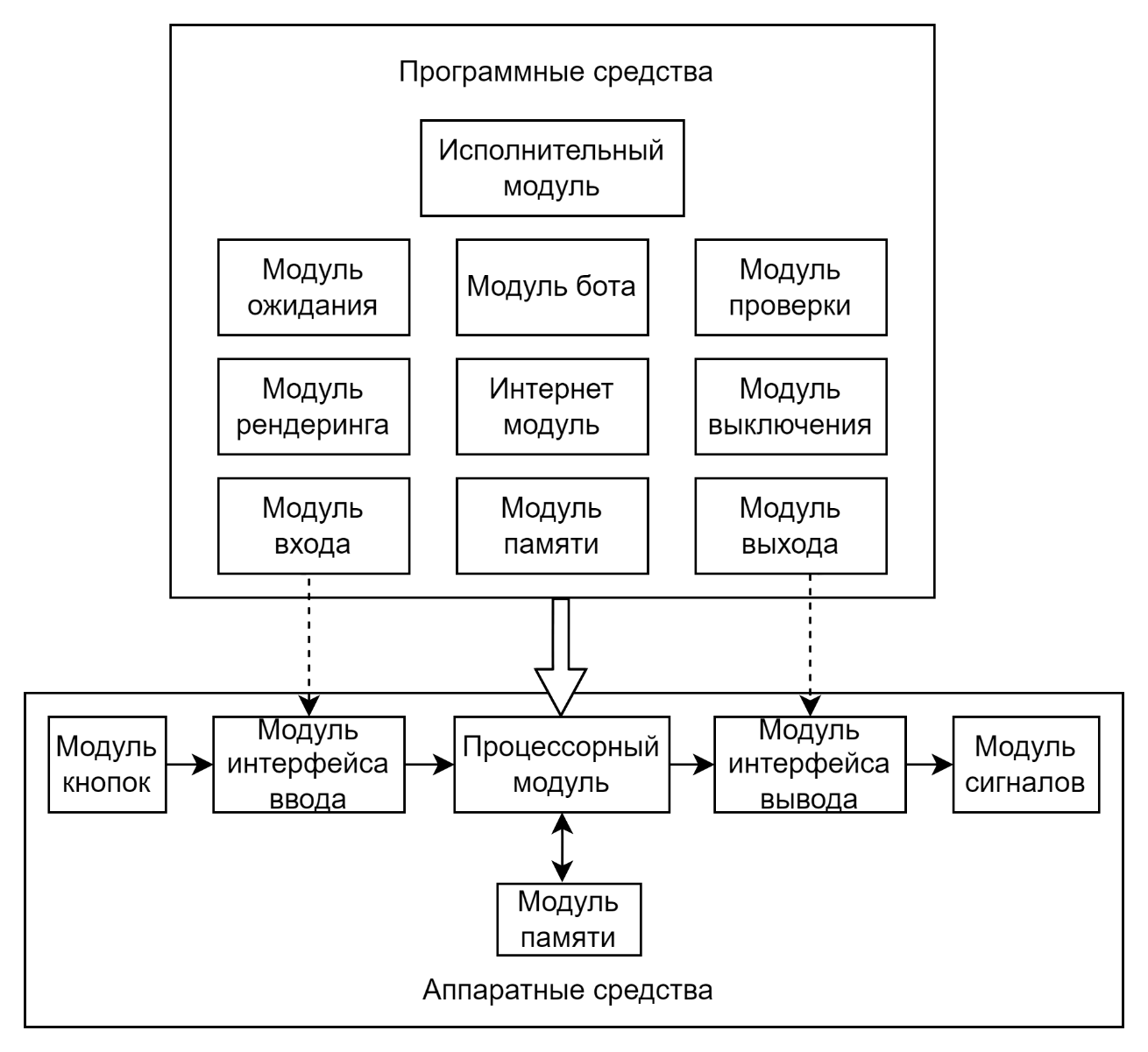


Рисунок 3.4 - Связь между программными и аппаратными средствами

## Построение и описание структурной схемы

На основе выполняемых системой функций разработана структурная схема аппаратной части.

Центральной частью схемы является микроконтроллер. Он обеспечивает взаимодействие всех элементов устройства. Микроконтроллер взаимодействует с *Wi-Fi* модулем для предоставления возможности загружать книги при помощи telegram бота, а также отображать пользователю список книг, находящихся на устройстве, в приложении Telegram. Также микроконтроллер связан с модулем *SD*-карты для чтения и записи файлов. На *E-ink* экране [[6]](#displayDataSheet) отображается различная информация для взаимодействия с пользователем, такая как: меню выбора книг, уровень заряда аккумулятора устройства, текст книги, количество страниц и прочее. Управление электронной книгой осуществляется при помощи нажатий на сенсорную панель. Также для экономии заряда батареи используется кнопка питания для включения/выключения устройства.

Обобщённая структурная схема устройства представлена на рисунке 3.5.

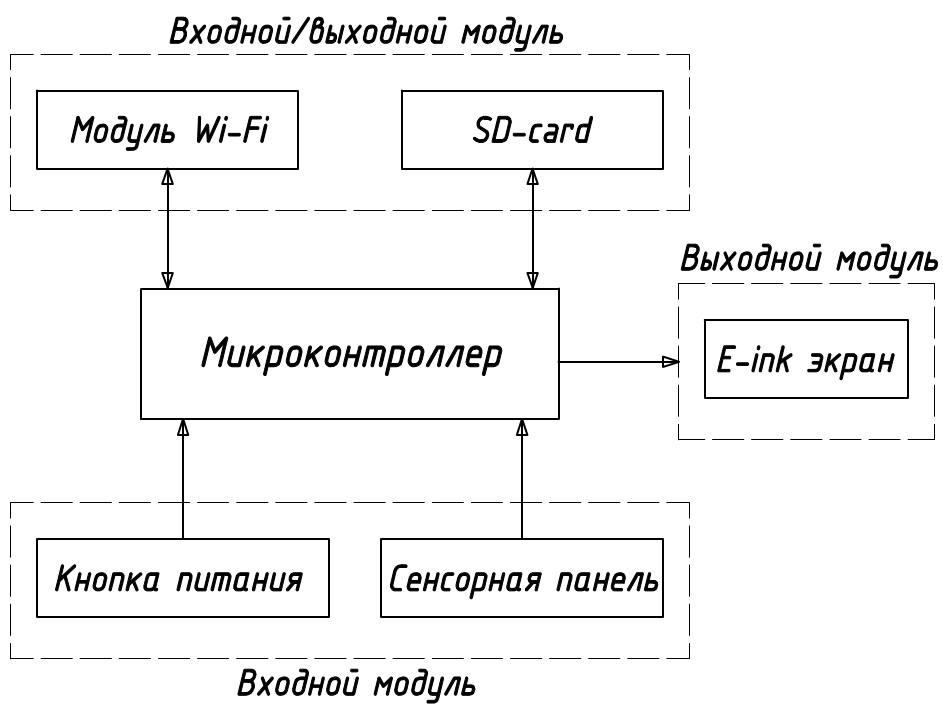


Рисунок 3.5 – Структурная схема электронной книги

Схема электрическая структурная электронной книги представлена на чертеже ГУИР.467849.001 Э1.

# Аппартано-программная реализация электроной книги на базе ESP32

## Разработка алгоритма работы электронной книги на базе ESP32

### Разработка схемы алгоритма работы системы и программы

Алгоритм работы устройства условно можно разделить на две части: инициализация и основная часть (циклическая). Первая выполняется при включении устройства и содержит включение базовых модулей устройства. В циклической части каждый раз происходит загрузка и отображение новых метеоданных.

Схема программы работы устройства приведена на чертеже ГУИР 467849.004 ПД. Код программы приведен в приложении Б.

### Описание алгоритма работы системы и программы

Программа написана при помощи среды разработки *Visual Studio* [7]с использованием расширения *Visual Micro* [8], который позволяет перенести функционал по работе с прошивкой из *Arduino IDE* [9] в *Visual Studio*.

Также для ускорения процесса разработки использовано расширение *GitHub Copilot* [10]. *GitHub Copilot* – это анализатор кода на базе нейронной сети обученной при помощи всего кода загруженного на платформу *GitHub*. *GitHub Copilot* умеет анализировать код программиста, который пользуется данным расширением, и на базе анализа предлагать код, который может решить необходимую проблему. Также нейронная сеть хорошо справляется с условными операторами, где очень умело расставляет все нужные условия. Нейронной сети достаточно комментария в коде, чтобы понять, что хочет написать программист, после чего она предлагает свои варианты решения необходимой задачи. Стоит отметить, что данный проект находится еще в стадии разработки, поэтому при его использовании могут возникать ситуации, когда он предлагает некорректные решения проблемы. На рисунке 4.1 продемонстрированно использование расширения *GitHub Copilot* в среде разработки *Visual Studio*.

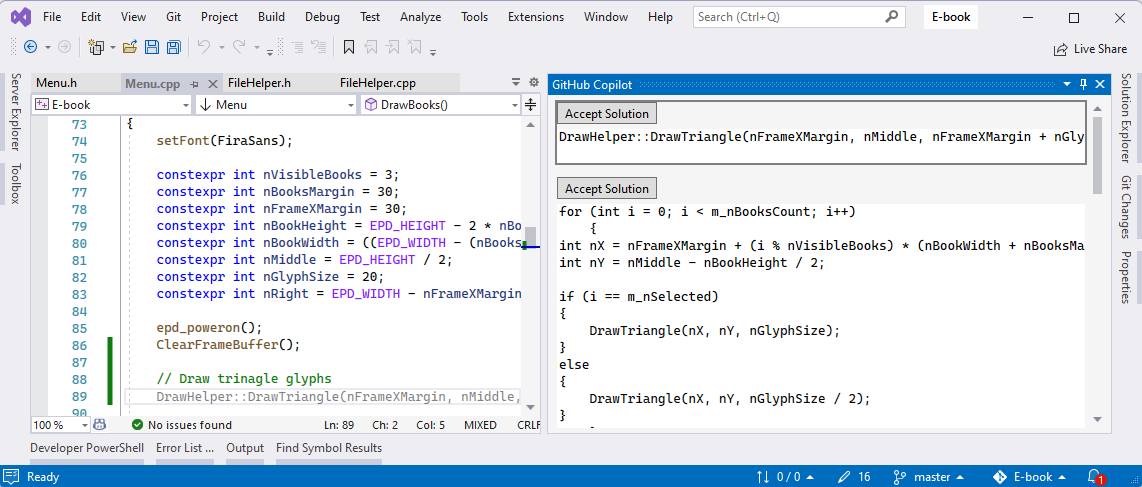


Рисунок 4.1 – Использование GitHub Copilot в среде разработки Visual Studio

Может добавить про ГИТ

Язык программирования *Arduino* называется *Arduino C* и представляет собой язык *C++* с фреймворком *Wiring*, Он имеет некоторые отличия по части написания кода, который компилируется и собирается с помощью компилятора *avr-gcc* [11], с особенностями, облегчающими написание работающей программы — имеется набор библиотек, включающий в себя функции и объекты. При компиляции программы *IDE* создает временный файл с расширением *\*.cpp*.

Программы, написанные программистом *Arduino*, называются наброски или скетчи (транслитерация от англ. *sketch*) и сохраняются в файлах с расширением *\*.ino*. Эти файлы перед компиляцией обрабатываются препроцессором *Arduino*. Также существует возможность создавать и подключать к проекту стандартные файлы *C++*.

Программист должен написать две обязательные для Arduino функции *setup()* и *loop().* Первая вызывается однократно при старте, вторая выполняется в бесконечном цикле.

В текст своей программы (скетча) программист не обязан вставлять заголовочные файлы используемых стандартных библиотек. Эти заголовочные файлы добавит препроцессор Arduino в соответствии с конфигурацией проекта. Однако пользовательские библиотеки нужно указывать.

Менеджер проекта *Arduino IDE* имеет нестандартный механизм добавления библиотек. Библиотеки в виде исходных текстов на стандартном *C++* добавляются в специальную папку в рабочем каталоге *IDE*. При этом название библиотеки добавляется в список библиотек в меню *IDE*. Программист отмечает нужные библиотеки, и они вносятся в список компиляции.

*Arduino IDE* не предлагает никаких настроек компилятора и минимизирует другие настройки, что упрощает начало работы для новичков и уменьшает риск возникновения проблем; но присутствуют директивы препроцессора, такие как *#define*, *#include*, и много других [12].

Также использовались дополнительные библиотеки:

1. *UniversalArduinoTelegramBot* – библиотека от независимого разработчика *witnessmenow* выложенная на платформе *GitHub*. Библиотека поддерживает только микроконтроллер *ESP32* и нужна для использования *Telegram Bot API* [13], который позволяет принимать и отправлять сообщения, выводить кнопки для пользовательского выбора, принимать и отправлять файлы [14];
2. *LilyGo EPD47* – библиотека от производителя аппаратной платформы используемой в данной дипломном проекте. Библиотека содержит драйвер для работы с экраном и драйвер для работы с сенсорной панелью, а также примеры использования функционала, предоставляемого аппаратной платформой [15];
3. *Button2* – библиотека от независимого разработчика *Lennart Hennigs* выложенная на платформе *GitHub*. Библиотека создана с целью упростить работу с кнопками при помощи встроенных в нее обработчиков различных событий, таких как: одиночное нажатие, двойное нажатие, тройное нажатие, нажатие с длительной задержкой [16];
4. *Arduino ESP32* – библиотека от производителя микроконтроллера *ESP32* компании *Espressif* выложенная на платформе *GitHub*. Библиотека обладает огромнейшим функционалом, который в полной мере обеспечивает возможность использовать весь функционал микроконтроллера *ESP32* [17].

Обобщённый алгоритм работы системы приведён ниже:

1. На первом этапе устройство сканирует все файлы в памяти и добавляет их имена в список книг, если формат файла поддерживается.
2. Далее формируется главное меню устройства, в центре которого отображается список из 3 книг, а также указатели для перемещения по списку книг (указатель слева – отображение предыдущих 3 книг из общего списка, справа – следующие 3 книги). Список книг отображается в виде вытянутого по горизонтали прямоугольника, в центре которого указано название книги без учёта формата файла и пути к нему.
3. На следующем шаге пользователь должен при помощи нажатия на соответствующие пункты меню на экране выбрать книгу из отображаемых или сменить страницу для дальнейшего поиска и выбора нужной книги.
4. Далее, когда выбрана необходимая книга, по её полному пути в директории происходит чтение файла.
5. На следующем этапе происходит обработка текста входе которой определяется, сколько слов поместится в одну строку на экране. Определение происходит следующим образом:
   1. За стартовый индекс начала строки принимается ноль.
   2. Выбирается *N* следующих символов. Число *N* задаётся больше максимального количества символов в строке. Индекс конца строки равен сумме стартового индекса и *N*.
   3. Производится проверка поместится ли текст, размещенный между стартовым и конечным индексами.
   4. Если текст не помещается, тогда происходит смещение конечного индекса таким образом, чтобы сместить конечный индекс ровно на длину одного слова с конца. Далее происходит выполняется возвращение к пункту 5.3.
   5. Сохранение индексов начала и конца итоговой строки.
   6. Если индекс конца строки меньше, чем длинна текста книги, тогда стартовый индекс равен конечному. Переход к пункту 5.2.
6. Далее рассчитается итоговое количество страниц книги, которое равно общему количество строк, деленное на количество строк, которые можно отобразить на экране. Результат округляется в большую сторону.
7. Происходит отображение текста текущей страницы при помощи индексов, подсчитанных в 5 пункте. Также отображается номер текущей страницы относительно количества всех страниц в книге.
8. В случае нажатия на экран происходит переход на следующую или предыдущую страницу. Левая сторона сенсора отвечает за переход назад, а правая – вперёд.
9. В случае нажатия на текущий номер страницы или общее количество страниц происходит возвращение к пункту 1.
10. В случае нажатия на кнопку питания происходит отображение рисунка уведомляющего пользователя об включении/выключении устройства.

ДОПОЛНИТЬ СВОИМИ ФУНКЦИЯМИ

## Разработка печатного узла электронной книги на базе ESP32

### Особенности применяемой элементной базы

Правильно выбранная элементная база позволяет обеспечить надежное функционирование всего изделия в целом, снизить вероятность возникновения помех, получить высокие эксплуатационные характеристики, уменьшить энергопотребление.

В общем случае критерием выбора элементной базы является соответствие технологических и эксплуатационных характеристик заданным условиям эксплуатации.

В случае, если на исходной схеме электрической принципиальной указывалось наименование для конкретного элемента, выбирался указанный элемент. Иначе выбирался элемент с техническими характеристиками, совпадающими с указанными на исходной схеме.

В результате для разработки печатного узла была использована элементная база, технические характеристики элементов которой приведены далее в этом подпункте. Наименования элементов, соответствующих позиционным обозначениям приведены в перечне элементов схемы электрической принципиальной (приложение А).

Для разработки печатного узла были выбраны резисторы, представленные в таблице 4.1.

Таблица 4.1 – Характеристики резисторов печатного узла

|  |  |  |  |
| --- | --- | --- | --- |
| Позиционное обозначение | Тип | Характеристики | Кол., шт. |
| R1-R3 | MCR03EZPJ100 | Рабочая температура:  -55°C…155°C  R = 100 кОм | 3 |
| R4-R6 | MCR03EZPJ039 | Рабочая температура:  -55°C…155°C  R = 39кОм | 3 |
| R7 | MCR03EZPJ120 | Рабочая температура:  -55°C…155°C  R = 120кОм | 1 |

Продолжение таблицы 4.1

|  |  |  |  |
| --- | --- | --- | --- |
| Позиционное обозначение | Тип | Характеристики | Кол., шт. |
| R8-R11 | MCR03EZPJ002 | Рабочая температура:  -55°C…155°C  R = 2кОм | 4 |
| R12-R15 | MCR03EZPJ001 | Рабочая температура:  -55°C…155°C  R = 1кОм | 4 |
| R16 | MCR03EZPJ120 | Рабочая температура:  -55°C…155°C  R = 120кОм | 1 |
| R17-R22 | MCR03EZPJ002 | Рабочая температура:  -55°C…155°C  R = 2кОм | 6 |
| R23 | MCR03EZPJ120 | Рабочая температура:  -55°C…155°C  R = 120кОм | 1 |
| R24-R28 | MCR03EZPJ002 | Рабочая температура:  -55°C…155°C  R = 2кОм | 5 |
| R29 | MCR03EZPJ001 | Рабочая температура:  -55°C…155°C  R = 1кОм | 1 |
| R30-R37 | MCR03EZPJ120 | Рабочая температура:  -55°C…155°C  R = 120кОм | 8 |
| R38-R42 | MCR03EZPJ005 | Рабочая температура:  -55°C…155°C  R = 5кОм | 5 |

К метрическому типоразмеру резисторов MCR03 относится 0603. Размеры резисторов с данным типоразмером приведены в таблице 4.2. Посадочное место изображено на рисунке 4.1.

Таблица 4.2 – Характеристики метрического типоразмера 0603

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| L, мм | W, мм | H, мм | D, мм | T, мм |
| 1.6 ± 0.1 | 0.85 ± 0.1 | 0.45 ± 0.05 | 0.3 ± 0.2 | 0.3 ± 0.2 |

|  |
| --- |
|  |
| Рисунок 4. – Посадочное место SMD резисторов 0603 |

Для разработки печатного узла были выбраны конденсаторы, представленные в таблице 4.3.

Таблица 4.3 – Характеристики конденсаторов печатного узла

|  |  |  |  |
| --- | --- | --- | --- |
| Позиционное обозначение | Тип | Характеристики | Кол., шт. |
| C1-C14 | C1608C0G1H102JT | Рабочая температура:  -55°C…125°C  C = 0,1нФ | 14 |
| C15-С20 | C1608C0G1H110JT | Рабочая температура:  -55°C…125°C  C = 1мкФ | 6 |
| C21-C28 | C1608C1G1H185JT | Рабочая температура:  -55°C…125°C  C = 10нФ | 8 |
| C29-С41 | C1608C1G1H285JT | Рабочая температура:  -55°C…125°C  C = 100нФ | 13 |
| C42-С48 | C1608C0G1H147JT | Рабочая температура:  -40°C…85°C  C = 4,7 нФ | 7 |

К метрическому типоразмеру конденсаторов, приведенных в таблице, относится 0603. Размеры резисторов с данным типоразмером приведены в таблице 2.4. Посадочное место изображено на рисунке 4.2.

Таблица 4.4 – Характеристики метрического типоразмера 0603

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| L, мм | W, мм | H, мм | D, мм | T, мм |
| 1.6 ± 0.1 | 0.85 ± 0.1 | 0.45 ± 0.05 | 0.3 ± 0.2 | 0.3 ± 0.2 |

|  |
| --- |
|  |
| Рисунок 4.2 – Посадочное место SMD резисторов 0603 |

Для разработки печатного узла были выбраны диоды, представленные в таблице 4.5.

Таблица 4.5 – Характеристики диодов печатного узла

|  |  |  |  |
| --- | --- | --- | --- |
| Позиционное обозначение | Тип | Характеристики | Кол., шт. |
| VD1 | KM2520SGC01 | Рабочая температура:  -55°C…125°C | 1 |
| VD2-VD4 | 1N5819 | Рабочая температура:  -55°C…125°C | 3 |
| VD5 | KM2520SGC01 | Рабочая температура:  -55°C…125°C | 1 |
| VD6, VD7 | 1N4148 | Рабочая температура:  -55°C…125°C | 2 |
| VD8, VD9 | KM2520SGC01 | Рабочая температура:  -55°C…125°C | 2 |
| VD10, VD11 | *ESD5D5* | Рабочая температура:  -55°C…125°C | 2 |
| VD12 | 1N5819 | Рабочая температура:  -55°C…125°C | 1 |

Для разработки печатного узла были выбраны транзисторы, представленные в таблице 2.6.

Таблица 2.6 – Характеристики транзисторов печатного узла

|  |  |  |  |
| --- | --- | --- | --- |
| Позиционное обозначение | Тип | Характеристики | Кол., шт. |
| VT1 | AO3400 | Рабочая температура:  -40°C…125°C  Vcc = 3В | 1 |
| VT2-VT3 | SI2301 | Рабочая температура:  -40°C…125°C  Vcc = 5В | 2 |
| VT4-VT6 | PMBT3904 | Рабочая температура:  -40°C…125°C  Vcc = 3В | 3 |
| VT7 | SI2301 | Рабочая температура:  -40°C…125°C  Vcc = 5В | 1 |

Размеры транзисторов приведены в таблице 4.7. Данные транзисторы выполнены в корпусах SOT-23, который изображен на рисунке 4.3.

Таблица 4.10 – Размеры транзисторов

|  |  |
| --- | --- |
| Параметр | Размер, мм |
| A | 2.93 ± 0.2 |
| B | 1.30 +0.20/-0.15 |
| C | 1.30 ± 0.05 |
| D | 1.45 +0.15/-0.05 |
| E | 2.40 +0.30/-0.20 |
| G | 1.90 ± 0.05 |
| H | 0.95 ± 0.05 |
| J | 0.13 +0.10/-0.05 |
| K | 0.1 ± 0.05 |
| L | 0.55 ± 0.10 |
| M | ± 0.20 ± 0.05 |
| N | 1.0 +0.20/-0.10 |

|  |
| --- |
|  |
| Рисунок 4.3 – Корпус SOT-23 |

Для разработки печатного узла были выбраны разъемы, представленные в таблице 4.6.

Таблица 4.6 – Характеристики разъёмов печатного узла

|  |  |  |  |
| --- | --- | --- | --- |
| Позиционное обозначение | Тип | Характеристики | Кол., шт. |
| X1 | Connector, 2011P 48PIN P=2.0 | Рабочая температура:  -25°C…85°C | 1 |
| X2, X3 | Connector, 2011P 2PIN P=2.0 | Рабочая температура:  -25°C…85°C | 2 |
| X4… X6 | MOLEX 53015-0410 | Рабочая температура:  -25°C…85°C | 3 |

Для разработки печатного узла были выбраны микросхемы, представленные в таблице 4.7.

Таблица 4.7 – Характеристики микросхем печатного узла

|  |  |  |  |
| --- | --- | --- | --- |
| Позиционное обозначение | Тип | Характеристики | Кол., шт. |
| DA1, DA2 | CJ79L15 | Рабочая температура:  -45°C…85°C | 2 |
| DD1 | LM358DR2G, SOIC-8 | Рабочая температура: -45°C…85°C | 1 |
| DD2 | 74HC4094D, SOIC-16 | Рабочая температура:  -45°C…85°C | 1 |
| DD3 | ESP32­WROVER­E | Рабочая температура:  -40°C…125°C | 1 |
| DD4 | IP3005A, eSOP8 | Рабочая температура:  -45°C…85°C | 1 |
| DD5 | LT1945, MSOP10 | Рабочая температура:  -45°C…85°C | 1 |
| DD6 | AP2112K, SOT25 | Рабочая температура: -45°C…85°C | 1 |
| DD7 | HX6610S, eSOP8 | Рабочая температура:  -45°C…85°C | 1 |
| DD8 | 1-2295018-2 | Рабочая температура:  -40°C…125°C | 1 |
| DD9 | CP2014, SOIC-24 | Рабочая температура:  -45°C…85°C | 1 |
| DD10 | UMH3N, SOT-363 | Рабочая температура:  -45°C…85°C | 1 |

### Особенности применяемых материалов

Поскольку в задании нет специальных требований к материалу печатной платы, в данном дипломном проекте будет использован двухсторонний фольгированный стеклотекстолит СФ-235Г толщиной 1,5 мм.

Для монтажа печатной платы будет использована смешанная технология, представляющая собой сочетание технологии монтажа в сквозные отверстия и поверхностного монтажа на одной плате.

### Оценка потребляемой мощности

Мощность электрического тока – количество электричества, проходящего через поперечное сечение цепи в течение одной секунды, есть не что иное, как сила тока в цепи. Следовательно, мощность электрического тока будет прямо пропорциональна разности потенциалов (напряжению) и силе тока в цепи. Единица измерения мощности – ватт (Вт).

Мощность определяется по формуле 4.1:

, (4.1)

|  |  |
| --- | --- |
|  | (4.1) |

где – падение напряжения в цепи, В; – сила тока, протекающего в цепи, А.

Оценим суммарную потребляемую мощность элементов, используя формулу 4.2:

, (4.2)

где – мощность, рассеиваемая на -ом элементе, Вт; – количество элементов на схеме.

В таблице 4.1 представлены данные о мощности, рассеиваемой на соответствующих элементах. Будем считать, что конденсаторы и кварцевый резонатор являются идеальными и мощности не потребляют. Некоторые значения максимальной рассеиваемой мощности взяты из официальной документации на соответствующие элементы, оставшиеся рассчитаны по формуле 4.8.

Таблица .8 – Расчёт мощностей, потребляемых элементной базой

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Элементы | Позиционные обозначения | Тип | Мощность одного элемента, мВт | Количество, шт. | Общая мощность, мВт |
| 1 | 2 | 3 | 4 | 5 | 6 |
| Диоды | VD2-VD4, VD12 | 1N5819 | 280 | 4 | 1120 |
| Стабилитроны | VD1, VD5, VD8, VD9 | KM2520SGC01 | 1 000 | 4 | 4 000 |

Продолжение таблицы 4.8

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| Диодные оптроны | VD6, VD7 | 1N4148 | 150 | 2 | 300 |
| ESD диоды | VD10, VD11 | ESD5D5 | 750 | 2 | 1500 |
| Транзисторы | VT1 | AO3400 | 1 000 | 1 | 1 000 |
| VT2-VT3, VT7 | SI2301 | 350 | 3 | 700 |
| VT4-VT6 | PMBT3904 | 350 | 3 | 700 |
| Резисторы | R1 – R42 | RC0805 | 125 | 42 | 5 250 |
| Микросхемы | DD1 | LM358DR2G | 830 | 1 | 830 |
| DD2 | 74HC4094D | 300 | 1 | 300 |
| DD3 | ESP32­WROVER­E | 650 | 1 | 650 |
| DD4 | IP3005A | 400 | 1 | 400 |
| DD5 | LT1945 | 500 | 1 | 500 |
| DD6 | AP2112K | 200 | 1 | 200 |
| DD7 | HX6610S | 660 | 1 | 660 |
| DD8 | 1-2295018-2 | 780 | 1 | 780 |
| DD9 | CP2014 | 640 | 1 | 640 |
| DD10 | UMH3N | 350 | 1 | 350 |

Исходя из данных, представленных в таблице 4.8, для данной схемы электрической принципиальной суммарная рассеваемая мощность, рассчитанная по формуле 4.2, составляет:

= 19880 Вт.

На основании полученных расчетов принимаем для схемы источник питания с P = 20 Вт.

### Оценка токов, протекающих в некоторых цепях

Для расчета ширины проводников необходимо оценить токи в наиболее нагруженных цепях. Минимальная ширина проводника рассчитывается по формуле 4.3:

, (4.3)

где – максимальная сила тока, протекающий в цепи, мА; – допустимая плотность тока, мА/мм2; – толщина проводника, мм.

### Расчет минимальной ширины проводника для цепи 3,3 В

Для микросхем DD2, DD5, DD6, DD7 сила тока составляет 20 мА, 100 мА и 30 мА соответственно.

Рассчитаем силу тока для транзистора VT4:

Таким же образом рассчитаем силу тока для транзистора VT5:

Исходя из расчетов, произведенных выше, получаем силу тока в цепи 3,3 В равную 151,2 мА. Рассчитаем минимальную ширину проводника для данной цепи по формуле 4.3:

### Расчет минимальной ширины проводника для цепи 5 В

Для микросхем DD1, DD3, DD4, DD5, DD7, DD10 сила тока составляет 20 мА, 50 мА, 95 мА и 100 мА соответственно.

Рассчитаем силу тока для транзистора VT1:

Таким же образом рассчитаем силу тока для транзистора VT2:

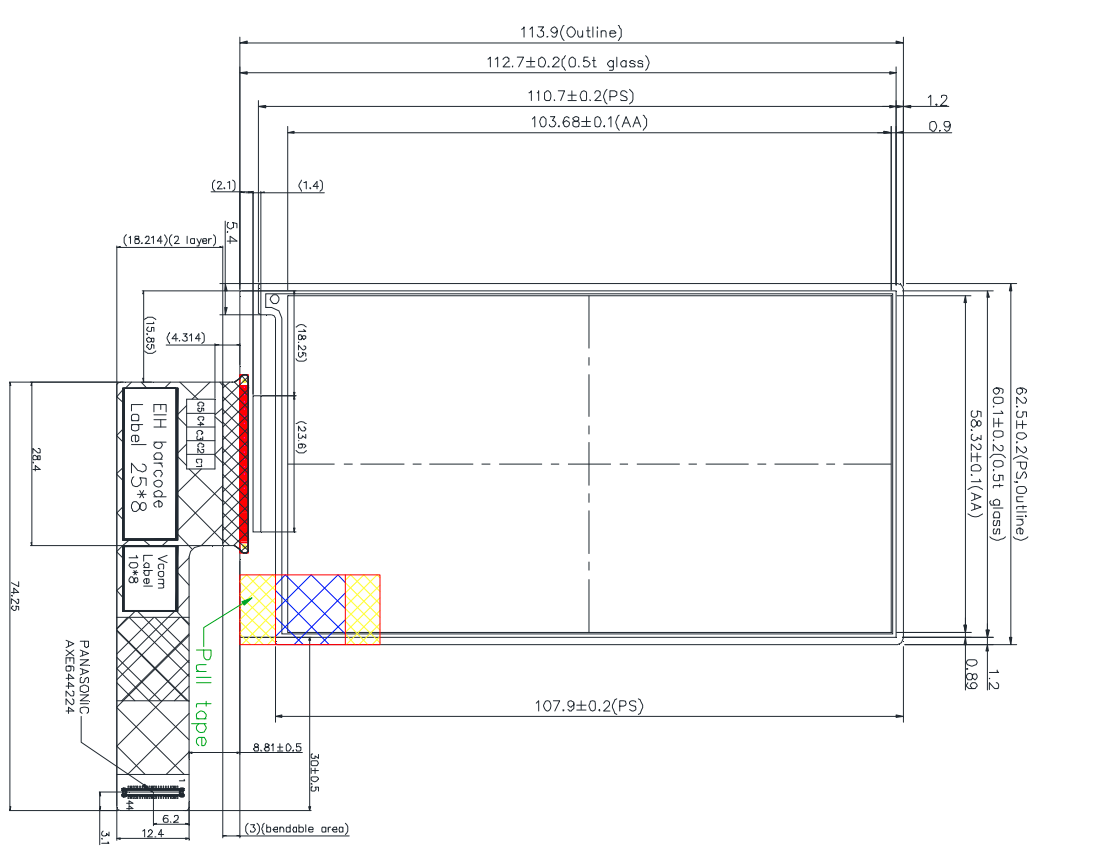
и для транзистора VT3:

Исходя из расчетов, произведенных выше и значений силы тока для цепей 3,3 В получаем силу тока в цепи 5 В равную 533,4 мА. Рассчитаем минимальную ширину проводника для данной цепи по формуле 4.3:

Минимальная ширина проводника цепи земли также составляет 0,76 мм.

### Выбор типоразмера печатной платы

Размер платы выбран с учетом размера дисплея электронной книги. Диагональ дисплея составляет 4.7 дюйма. Экран и его размеры продемонстрированы на рисунке 4.4.



|  |
| --- |
| Рисунок 4.4 – Дисплей электронной книги |

### Расчет печатного монтажа

Исходя из выбранной элементной базы, в данном курсовом проекте будет использоваться смешанная технология монтажа печатных плат. Так как используются как элементы поверхностного монтажа, так и выводные, необходимо произвести ряд конструкторских расчётов для определения параметров контактных площадок и монтажных отверстий для их установки.

В качестве метода изготовления ПП был выбран комбинированный позитивный метод, поскольку при своей относительно небольшой стоимости и простой технологии он обеспечивает создание элементов печатного рисунка высокой точности, высокую надёжность изоляции и хорошую адгезию элементов печатного рисунка и диэлектрического основания платы.

Данный метод применяют для изготовления двусторонних и гибких ПП с металлизированными отверстиями на двустороннем фольгированном диэлектрике. Проводящий рисунок получают субтрактивным методом, а металлизацию отверстий осуществляют электрохимическим методом. После экспонирования рисунка сразу проводят сверление отверстий и металлизацию. Затем рисунок схемы и металлизированные отверстия защищают слоем металла, стойкого к травителю для меди, и травят незащищенную медь.

Из пяти классов точности был выбран четвёртый класс точности, предусмотренных ГОСТ 23751-86, руководствуясь соотношением цены, качества и сложности изготовления печатной платы.

Характеристики размеров элементов конструкций четвертого класса точности приведены в таблицах 4.9 – 4.12:

Талица 4.9 – Предельные отклонения диаметров монтажных и переходных отверстий

|  |  |  |
| --- | --- | --- |
| Диаметр отверстия d, мм | Наличие металлизации | Предельное отклонение диаметра Δd, мм |
| До 1,0 | Без металлизации | ± 0,05 |
| С металлизацией без оплавления | + 0; – 0,10 |
| С металлизацей и с оплавления | + 0; – 0,13 |
| Свыше 1,0 | Без металлизации | ± 0,10 |
| С металлизацией без оплавления | + 0,05; – 0,15 |
| С металлизацией и с оплавления | + 0,05; – 0,18 |

Таблица 4.10 – Предельные отклонения ширины печатного проводника

|  |  |
| --- | --- |
| Наличие металлического покрытия | Предельное отклонение ширины печатного проводника Δt, мм |
| Без покрытия | ± 0,03 |
| С покрытием | ± 0,05 |

Таблица 4.11 – Позиционные допуски расположения осей отверстий

|  |  |
| --- | --- |
| Размер печатной платы по большей стороне, мм | Значение позиционного допуска расположения осей отверстий Td, мм |
| До 180 включительно | 0,05 |
| Свыше 180 до 360 включительно | 0,08 |
| Свыше 360 | 0,10 |

Таблица 4.12 – Дополнительные параметры

|  |  |
| --- | --- |
| Параметр | Значение допуска, мм |
| Допуск на расположение центров контактных площадок TD, при размере платы до 180 мм | 0,10 |
| Допуск на расположение печатного проводника Tl, для ДПП | 0,03 |

Используя данные, приведенные в таблицах 6.1 – 6.4, произведем расчеты печатного монтажа.

Номинальное значение диаметра монтажного отверстия рассчитывается по формуле 4.3:

, (4.3)

где – максимальное значение диаметра вывода навесного элемента, устанавливаемого на печатную плату; *r* – разность между минимальным значением диаметра отверстия и максимальным значением диаметра вывода; – нижнее предельное отклонение номинального значения диаметра отверстия.

Диаметры монтажных отверстий рекомендуется выбирать так, чтобы разница с диаметром вывода ЭРЭ составляла 0,1...0,4 мм. При этом для повышения технологичности необходимо стремиться к сокращению количества различных диаметров.

Диаметр контактных площадок рассчитывается по формуле 4.4:

, (4.4)

где – номинальное значение монтажного отверстия; – верхнее предельное отклонение диаметра отверстия;  *–* гарантийный поясок; – верхнее предельное отклонение диаметра контактной площадки; – величина подтравливания диэлектрика; – позиционный допуск расположения центра контактной площадки; – позиционный допуск расположения оси отверстия; – нижнее предельное отклонение диаметра контактной площадки.

Для выводов с диаметрами 0,5 – 0,7 мм номинальные значения диаметров монтажного отверстия и контактной площадки составляют:

Для выводов с диаметрами 0,78 – 0,9 мм:

Минимальный диаметр переходного отверстия рассчитывается по формуле 4.5:

, (4.5)

где – сила тока цепи, А; – толщина медной фольги, мм; – допустимая токовая нагрузка, А/мм2.

Таким образом минимальный диаметр переходного отверстия составляет:

Размеры контактных площадок для элементов с поверхностным монтажом взяты из технической документации этих элементов.

### Особенности применяемых пакетов САПР

Для проектирования печатной платы были рассмотрены два программных пакета: Altium Designer и OrCAD. Ранее были получены навыки работы в обоих проектах.

Для выполнения данного курсового проекта был выбран пакет Altium Designer. В данном пакете есть возможности создания и редактирования библиотек условно-графических обозначений элементов, которые соответствует используемым стандартам, и посадочных мест компонентов. Также есть возможности создания схемы электрической принципиальной, передачи списка связей со схемы электрической принципиальной на печатную плату. Существует возможность ручной и автоматической трассировок.

В пакете OrCAD также существуют эти возможности, однако Altium Designer более удобен в использовании, поэтому для выполнения курсового проекта был выбран именно он.

Для оформления технической документации был использован пакет AutoCAD. В пакете есть большой функционал возможностей, например:

– автоматическая и быстрая подготовка документации проекта;

– гибкость пользовательского интерфейса.

### Решение задачи топологического синтеза печатной платы с помощью применяемого пакета САПР

Граничные значения основных параметров печатного монтажа, которые могут быть обеспечены при конструировании и производстве для четвёртого класса точности приведены в таблице 4.13.

Таблица 4.13 – Граничные значения основных параметров печатного монтажа

|  |  |  |
| --- | --- | --- |
| Обозначение | Параметр | Значение |
| t, мм | Ширина печатного проводника | 0,15 |
| S, мм | Расстояние между краями соседних элементов проводящего рисунка | 0,15 |
| b, мм | Гарантийный поясок | 0,05 |
| γ | Отношение номинального значения диаметра наименьшего из металлизированных отверстий к толщине печатной платы | 0,25 |

На основании расчётов, были установлены следующие ограничения:

– минимальная ширина проводника: 0,15 мм;

– минимальная ширина проводника для цепи 1,8 В: 0,15 мм;

– минимальная ширина проводника для цепи 3,3 В: 0,25 мм;

– минимальная ширина проводника для цепей 5В и GND: 0,8 мм;

– топология трассировки: самая короткая – соединяет все узлы по кратчайшей траектории;

– минимальный диаметр переходного отверстия: 0,08 мм;

 минимально допустимый зазор между двумя любыми объектами печатного монтажа на сигнальном слое: 0,15 мм;

 минимальное расстояние между двумя контактными площадками: 0,15 мм.

После настройки правил редактора печатных плат производим передачу списка связей из редактора схем в редактор печатных плат и размещаем элементы на печатной плате. Выполняем вручную трассировку цепей питания и земли, оставшиеся цепи трассируем в автоматическом режиме с учетом выставленных конструкторских ограничений. После трассировки производим необходимые корректировки.

### Оценка качества разработанной конструкции

После задания конструкторских ограничений и размещения компонентов была произведена трассировка печатной платы. Цепи питания и земли трассировались вручную, остальные цепи – с использованием автотрассировщика, предоставленного пакетом Altium Designer. Результат проектирования представлен на рисунке 4.5.

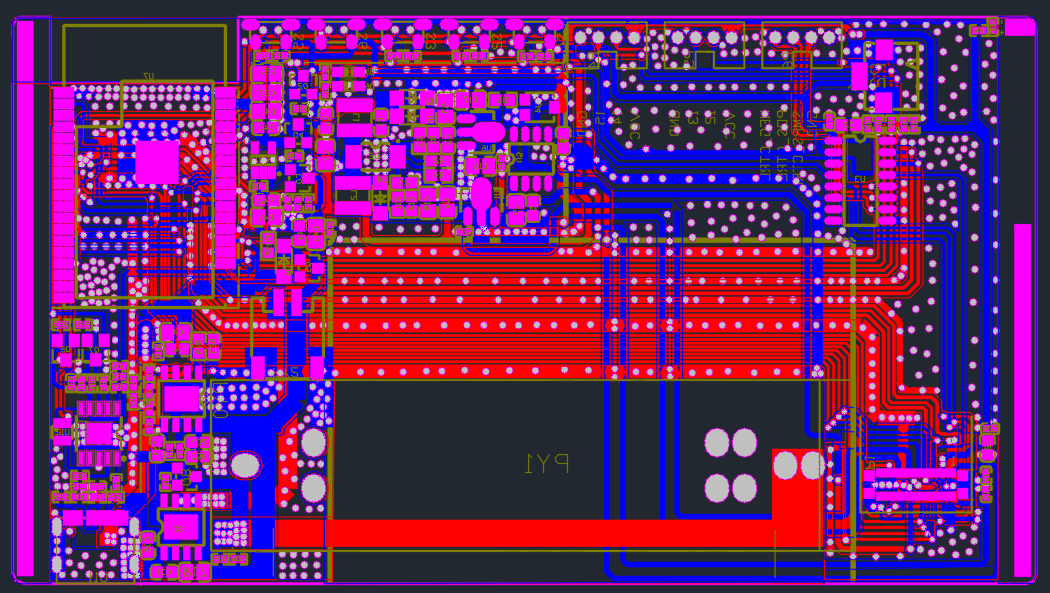


Рисунок 4.5 – Печатная плата

Далее необходимо оценить качество разработанной конструкции. Для этого воспользуемся функцией Design Rule Check, которая позволит нам удостоверится в том, что все выставленные ранее конструкторские ограничения были соблюдены. Результат проверки приведён на рисунке 4.6

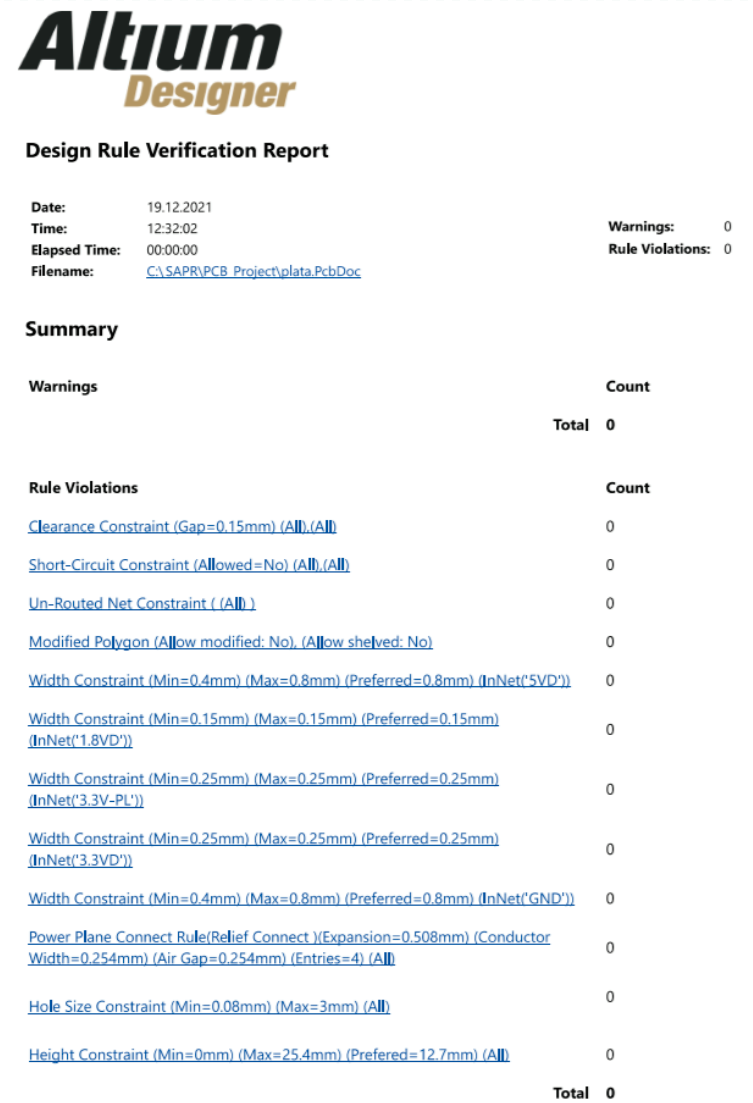


Рисунок 4.6 – Результаты проверки печатной платы

Исходя из результатов проверки, можно сделать вывод, что все конструкторские ограничения были соблюдены, и разработанная печатная плата соответствует техническому заданию.

Чертеж печатной платы представлен в приложении Б. Сборочный чертёж представлен в приложении В.

# Технико-экономическое обоснование эффективности разработки и использования электронной книги на базе ESP32

## Характеристика электронной книги

Разрабатываемая в дипломном проекте электронная книга представляет собой устройство, состоящее из готовой аппаратной платформы, модуля для подключения *SD*-карты, *E-ink* экрана и сенсорной панели.

Тип производства будет серийным с корректировками в зависимости от спроса покупателей, предполагается производить 500 устройств в год – именно такое среднее количество заказов аналогичных устройств на площадке *Wildberries*. Потребитель устройства – это пользователь, часто читающий книги в электронном формате. Пользователь получает компактное и легкое устройство, которое можно использовать как дома, так и брать с собой. Устройство способно держать заряд значительно дольше, чем смартфон, а также имеет E-ink экран, который из-за своей специфики меньше нагружает глаза пользователя.

Пользователь получает электронную книгу, которая позволяет выбирать и читать книги в электронном формате. Загрузка книг доступна при помощи *telegram* бота. Преимущество в том, что пользователь получает недорогое специализированное устройство для чтения книг в электронном формате.

Главным преимуществом разрабатываемого устройства значительно меньшая цена, чем у аналогичных устройств, представленных на рынке на сегодняшний день, и при этом устройство будет иметь более широкий функционал по сравнению с наиболее дешевыми из аналогичных устройств.

## Формирование отпускной цены электронной книги

Производство электронной книги будет производится с использованием готовых модулей, которым не нужны дополнительные материалы.

### Расчет прямых затрат на комплектующие изделия для электронной книги

Расчет прямых затрат на комплектующие изделия для производства аппаратной части комплекса осуществляется в соответствии с количеством комплектующих на изделие и рыночными ценами. Коэффициент транспортных расходов равен 1,1. Результаты расчётов приведены в таблице 5.1.

Таблица 5.1 – Расчет затрат на комплектующие изделия

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование специальных инструментов, приспособлений, приборов, стендов, устройств и другого специального оборудования | Количество, шт. | Цена за единицу, р. | Сумма,  р. |
| 1. Аппаратная платформа LilyGo T5-4.7 inch E-Paper ESP32 | 1 | 89,42 | 89,42 |
| 1. Сенсорная панель LilyGo T5-4.7 inch Capacitive Touch Cover | 1 | 36,82 | 36,82 |
| 1. Разъем для чтения SD-карт LilyGo T5-4.7 inch T-FH Interface RF Card Expansion Module | 1 | 7,89 | 7,89 |
| 1. Аккумулятор Panasonic NCR18650B 3350 mAh | 1 | 10,52 | 10,52 |
| Итого | | | 144,65 |
| Всего с учетом транспортных расходов (Ктр = 1,1) | | | 159,11 |

### Расчет накладных расходов

Расчет накладных затрат выполняется по формуле 5.1:

|  |  |
| --- | --- |
| Рнакл = = = 85,92 р., | (5.1) |

где Рк ‒ расходы на комплектующие изделия, р.; Ннакл – норматив накладных расходов (Ннакл = 54%).

### Расчет полной себестоимости

Полная себестоимость Сп рассчитывается по следующей формуле:

|  |  |
| --- | --- |
| Сп = Рк + Рнакл= 159,11 + 85,92 = 245,03 р. | (5.2) |

### Расчет плановой прибыли

Плановая прибыль Пед рассчитывается по следующей формуле:

|  |  |
| --- | --- |
| Пед = = = 73,50 р., | (5.3) |

где Рпр – рентабельность продукции (Рпр = 30%).

### Расчет отпускной цены изделия

Отпускная цена Цотп рассчитывается по следующей формуле:

|  |  |
| --- | --- |
| Цотп = Сп + Пед = 245,03 + 73,50 = 318,53 р. | (5.4) |

### Формирование отпускной цены электронной книги на основе полной себестоимости электронной книги

Формирование отпускной цены электронной книги на основе полной себестоимости приведены в таблице 5.2.

Таблица 5.2 – Формирование отпускной цены электронной книги на основе полной себестоимости

|  |  |
| --- | --- |
| Наименование статьи затрат | Сумма, р. |
| 1 Покупные комплектующие изделия Рк | 159,11 |
| 2 Накладные расходы Рнакл | 85,92 |
| 3 Полная себестоимость Сп | 245,03 |
| 4 Плановая прибыль Пед | 73,50 |
| 5 Отпускная цена Цотп | 318,53 |

Итоговая отпускная цена электронной книги оказалась ниже, чем стоимость устройств аналогичных характеристик, таких как: PocketBook 628 и Amazon Kindle Paperwhite. В случае выхода продукта на рынок, такая цена будет дополнительным конкурентным преимуществом, которое будет важно для потребителя.

## Расчет экономического эффекта от производства и реализации новых изделий

На основе маркетинговых исследований и заказов потребителей было установлено количество производимых изделий как = 500 изделий в первый год. В последующие годы планируется увеличение выпуска продукции с каждым годом на 20% (для второго года = 600, для третьего года = 720 и т.д.). Исходя из этого, определим чистую прибыль. Ставка налога на прибыль Нп составляет 18%.

Расчет прироста чистой прибыли производится по формуле 5.5:

|  |  |
| --- | --- |
| , | (5.5) |

Расчёт чистой прибыли за 4 года производства:

|  |  |
| --- | --- |
| = = 30 135 р., |  |

|  |  |
| --- | --- |
| = = 36 162 р., |  |

|  |  |
| --- | --- |
| = = 43 394,40 р., |  |

|  |  |
| --- | --- |
| = = 52 073,28 р., |  |

## Расчет инвестиций в производство электронной книги

### Инвестиции в разработку электронной книги

Инвестиции в разработку электронной книги включают:

– инвестиции в разработку программной части;

– инвестиции в разработку аппаратной части.

Для создания и длительной поддержки продукта необходима команда разработчиков из трех человек: бизнес-аналитик, программист, тестировщик.

Расчет затрат на основную заработную плату команды разработчиков осуществляется исходя из состава и численности команды, размера месячной заработной платы каждого участника команды, а также трудоемкости работ, выполняемых при разработке программного средства отдельными исполнителями, по формуле:

|  |  |
| --- | --- |
|  | (5.6) |

где коэффициент премий и иных стимулирующих выплат (;

n категории исполнителей, занятых разработкой программного средства; часовой оклад плата исполнителя -й категории, р.; трудоемкость работ, выполняемых исполнителем -й категории, ч. .

Результаты расчётов затрат на заработную плату команды разработчиков приведены в таблице 5.4.

Таблица 5.4 – Расчет затрат на заработную плату команды разработчиков

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Категории исполнителей | Месячный  оклад, р. | Часовой  оклад, р. | Трудоемкость, ч | Сумма, р. |
| Бизнес-аналитик | 2000 | 11,90 | 40 | 476 |
| Программист | 2450 | 14,58 | 80 | 1 166,40 |
| Тестировщик | 1850 | 11,01 | 80 | 880,8 |
| Итого | | | | 2 523,20 |
| Премия и иные стимулирующие выплаты | | | | 1 261,60 |
| Всего затрат на основную заработную плату разработчиков | | | | 3 784,80 |

Дополнительная заработная плата Зд команды разработчиков рассчитывается по формуле 5.7:

|  |  |
| --- | --- |
| Зд = = = 378,48 р., | (5.7) |

где Зо – основная заработная плата, рассчитанная в таблице 5.2; Нд – норматив дополнительной заработной платы.

Отчисления в фонд социальной защиты населения и по обязательному страхованию от несчастных случаев на производстве вычисляется по формуле 5.8:

|  |  |
| --- | --- |
| Рсоц = = = 1 440,49 р., | (5.8) |

где Нсоц – норматив отчислений в ФСЗН и Белгосстрах (Нсоц = 34,6%).

Результаты расчётов затрат на разработку программного средства приведены в таблице 5.3.

Таблица 5.3 – Расчет затрат на разработку программного средства

|  |  |
| --- | --- |
| Наименование статьи затрат | Сумма, р. |
| 1 Основная заработная плата разработчиков Зо | 3 784,80 |
| 2 Дополнительная заработная плата разработчиков Зд | 378,48 |
| 3 Отчисления на социальные нужды Рсоц | 1 440,49 |
| 4 Инвестиции в разработку электронной книги Ир | 5 603,77 |

Инвестиции в разработку аппаратной части не требуются, поскольку электронная книга построена на базе серийной аппаратной платформы, в которой уже предоставлены все необходимые компоненты.

### Инвестиции в прирост основного капитала

Инвестиции в прирост основного капитала не требуются, т. к. производство программно-аппаратного комплекса планируется осуществлять на действующем оборудовании в связи с наличием на предприятии-производителе свободных производственных мощностей.

### Инвестиции в прирост собственного оборотного капитала

Для производства нового вида продукции требуется прирост инвестиций в собственный оборотный капитал в размере 20% общей годовой потребности в материальных ресурсах.

Годовая потребность в комплектующих изделиях определяется по формуле:

|  |  |
| --- | --- |
| Пк = = = 79 555 р., | (5.9) |

где затраты на комплектующие изделия на единицу изделия, р. ; прогнозируемый годовой объем производства и реализации программно-аппаратного комплекса за первый год (определен на основании анализа рынка продаж в пункте 5.1, 500 шт), шт. .

Инвестиции в прирост собственного оборотного капитала в первый год:

|  |  |
| --- | --- |
| = 0,2 = 15 911 р., | (5.10) |

где годовая потребность в комплектующих ( = 20%).

Инвестиции в собственный оборотный капитал изменяются по годам из-за увеличения производимых устройств на 20% каждый год в отношении предыдущего (). Из этого следует формула расчёта инвестиций в оборотный капитал (при учёте прироста собственного оборотного капитала на 20%):

|  |  |
| --- | --- |
| = 0,2 = 3 182,20 р. |  |

|  |  |
| --- | --- |
| = 0,2 = 3 818,64 р. |  |

|  |  |
| --- | --- |
| = 0,2 = 4 582,36 р. |  |

Общие инвестиции в разработку и в прирост собственного оборотного капитала электронной книги на базе ESP32 за первый год рассчитываются по следующей формуле.

|  |  |
| --- | --- |
| Иобщ = Ир + Ис.о.к.1 = 5 603,77 + 15 911 = 21 514,77 р. | (5.11) |

## Расчет показателей экономической эффективности инвестиций в производство электронной книги

Общие инвестиции в разработку и в прирост собственного оборотного капитала Иобщ = 21 514,77 р. меньше, чем экономический эффект за первый год = 30 135 р., поэтому оценка экономической эффективности инвестиций в производство электронной книги осуществляется на основе расчета рентабельности инвестиций по следующей формуле:

|  |  |
| --- | --- |
| ROI = =  = = 40,07 %, | (5.12) |

Инвестиции в производство электронной книги экономически эффективны, так как рентабельность инвестиций превышает ставку по банковским долгосрочным депозитам, и, следовательно, разработка электронной книги является целесообразной.

## Результаты расчета

Эффективность данного проекта сильно зависит от количества проданных изделий при плановом объеме выпуска присутствует запас прочности в 303 изделия, то есть при продаже 197 изделий проект выйдет на безубыточный уровень.

Расчёт экономической эффективности оказался больше, чем банковский долгосрочный депозит. Но данная экономическая эффективность сильно зависит от двух факторов. Первый фактор, для устройств подобного типа необходимо продаваться большими тиражами для окупаемости и конкурентоспособности, поэтому в случае малого спроса будет сложнее данному проекту окупится. Второй фактор — это высокая волатильность рубля и высокие инфляционные ожидания, из-за чего сложно планировать бюджет и правильно распоряжаться средствами, так как все комплектующие покупаются с Китая в иностранной валюте.

# Анализ результатов проектирования эектронной книги на базе ESP32

## Макетирование устройства

Для аппаратной реализации электронной книги на базе ESP32 была использована аппаратная платформа *LilyGo EPD47*. Данная аппаратная платформа состоит из *USB* интерфейса, *UART* интерфейса, микроконтроллера *ESP32-WROVER-E*, схемы питания, 5 кнопок и 4,7-дюймового *E-ink* дисплея. На рисунке 6.1 приведено изображение аппаратной платформы.



Рисунок 6.1 – Аппаратная платформа LilyGo EPD47

Также для аппаратной реализации была использована сенсорная панель *LilyGo T5-4.7 inch Capacitive Touch Cover*. Размер сенсорной панели соответствует 4,7-дюймовому дисплею. Также для обработки и вывода на шину есть специальный микроконтроллер. На рисунке 6.2 приведено изображение сенсорной панели.

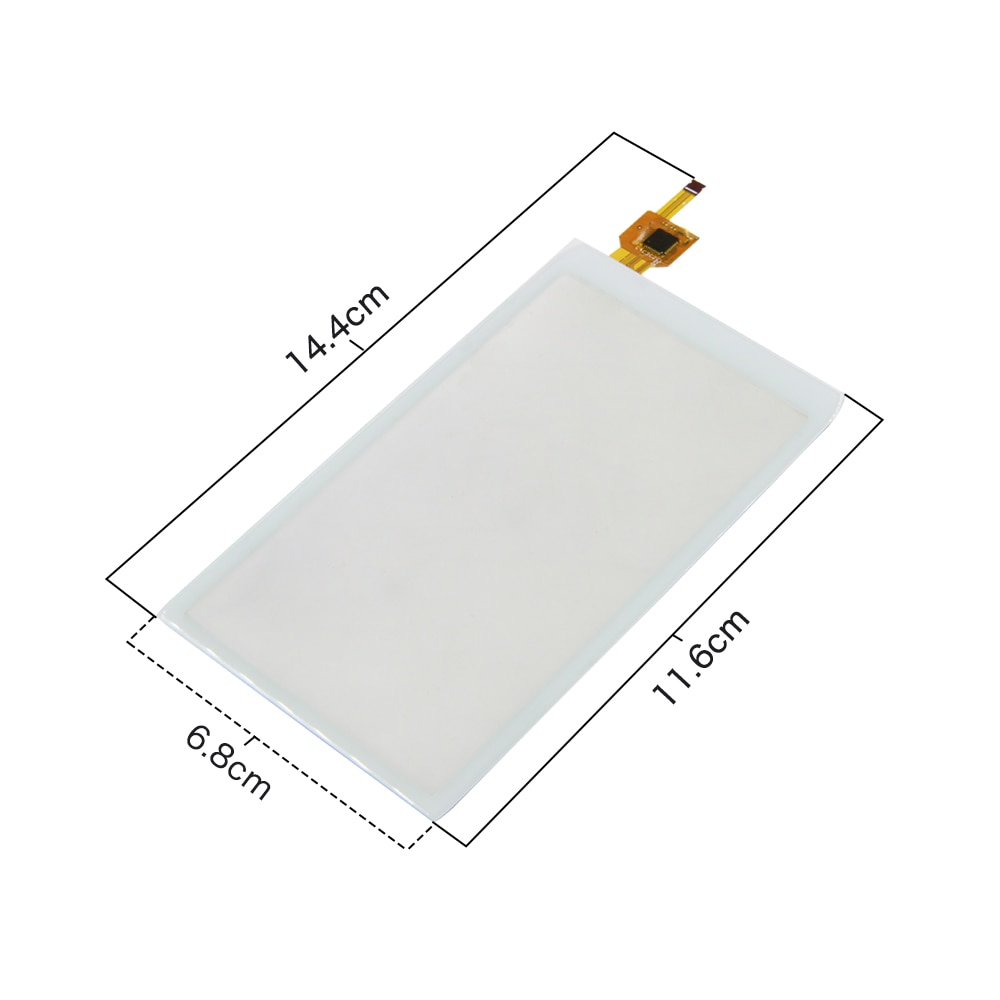


Рисунок 6.2 – Сенсорная панель

Для хранения большого количества книг был использован интерфейсный модуль для чтения *SD*-карт *LilyGo T5-4.7 inch T-FH Interface RF Card Expansion Module*. На рисунке 6.3 приведено изображение интерфейсного модуля для чтения *SD*-карт.

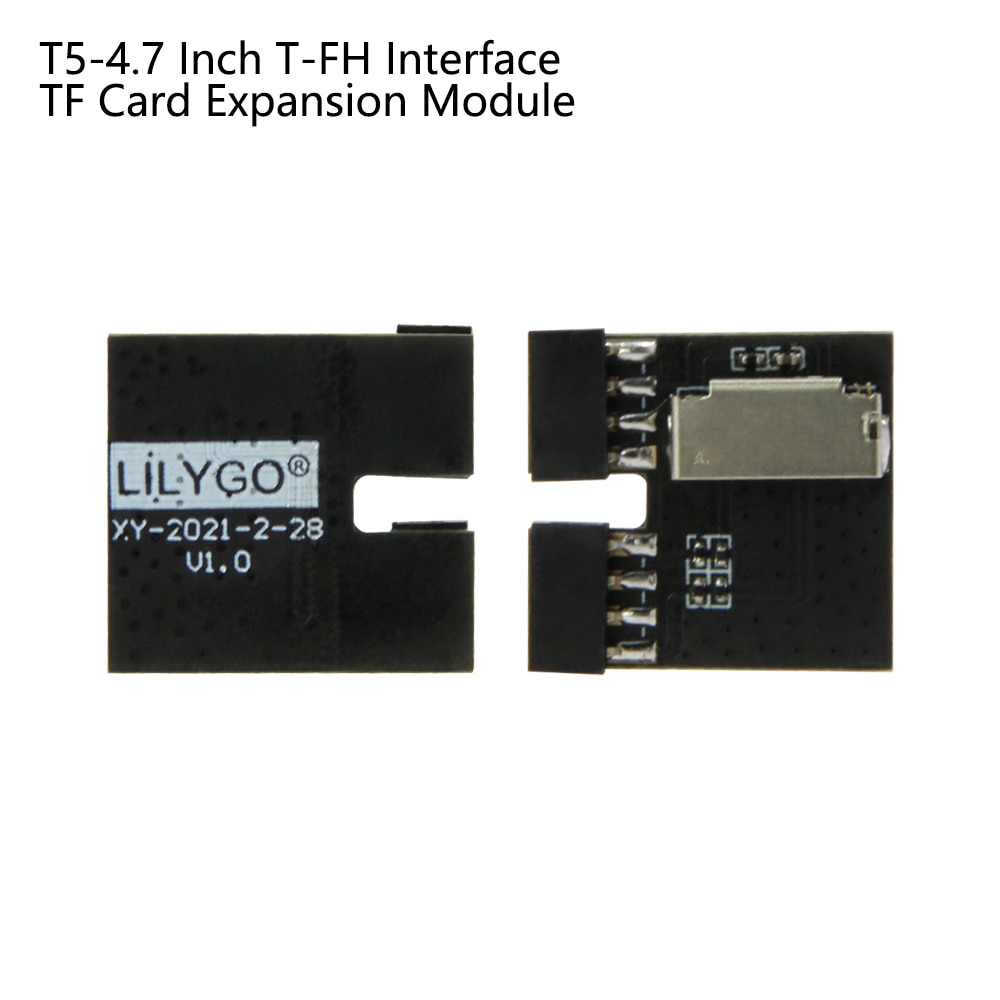


Рисунок 6.3 – Интерфейсный модуль для чтения *SD*-карт

## Тестирование устройства

Для проверки работоспособности электронной книги проведено её макетирование. При тестировании устройства особое внимание уделялось удобству использования, читабельности отображаемой информации, а также его интуитивности для пользователя.

При включении устройства происходит инициализация базовых систем и анализ памяти для нахождения в нем книг с поддерживаемым форматом. На время инициализации отображается загрузочный экран. На рисунке 6.4 приведено изображение загрузочного экрана.



Рисунок 6.4 – Загрузочный экран

После инициализации на экране отображается меню выбора книг, в котором можно перемещаться для нахождения и выбора необходимой книги. Перемещение производится при нажатии на левый или правый указатели, если далее книг нет, то позиция не изменяется. Выбор книги производится нажатием на одну из трех выводимых книг. Если книг меньше, чем 3, то на месте, где должны быть отсутствующие книги, ничего не выводится. На рисунке 6.5 приведено изображение меню выбора книг. А ЧТО ЕСЛИ ДВЕ КНИГИ?

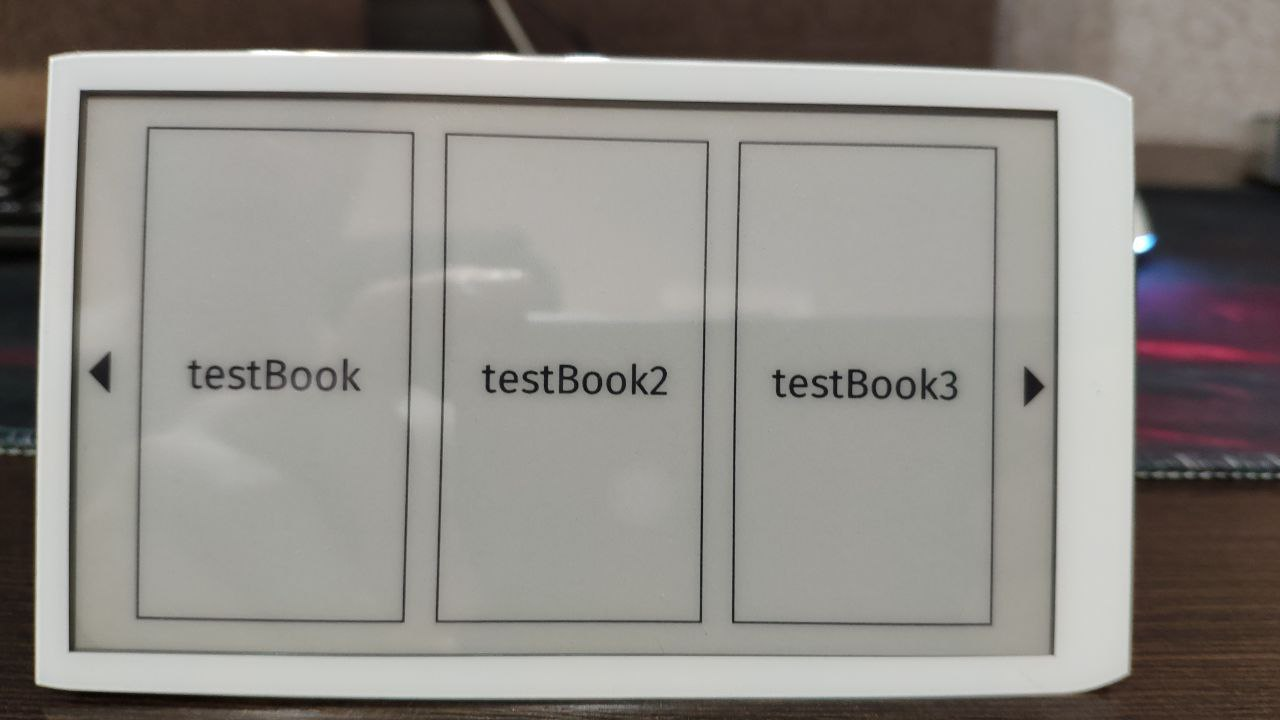


Рисунок 6.5 – Меню выбора книг

После выбора книги устройство переходит в режим чтения. В данном режиме выводится текст текущей страницы книги. При помощи нажатия на левую или правую часть дисплея пользователь возвращается или переходит на следующую страницу соответственно. При нажатии на номер страницы пользователь возвращается в меню выбора книг. На рисунке 6.6 приведено изображение электронной книги в режиме чтения.

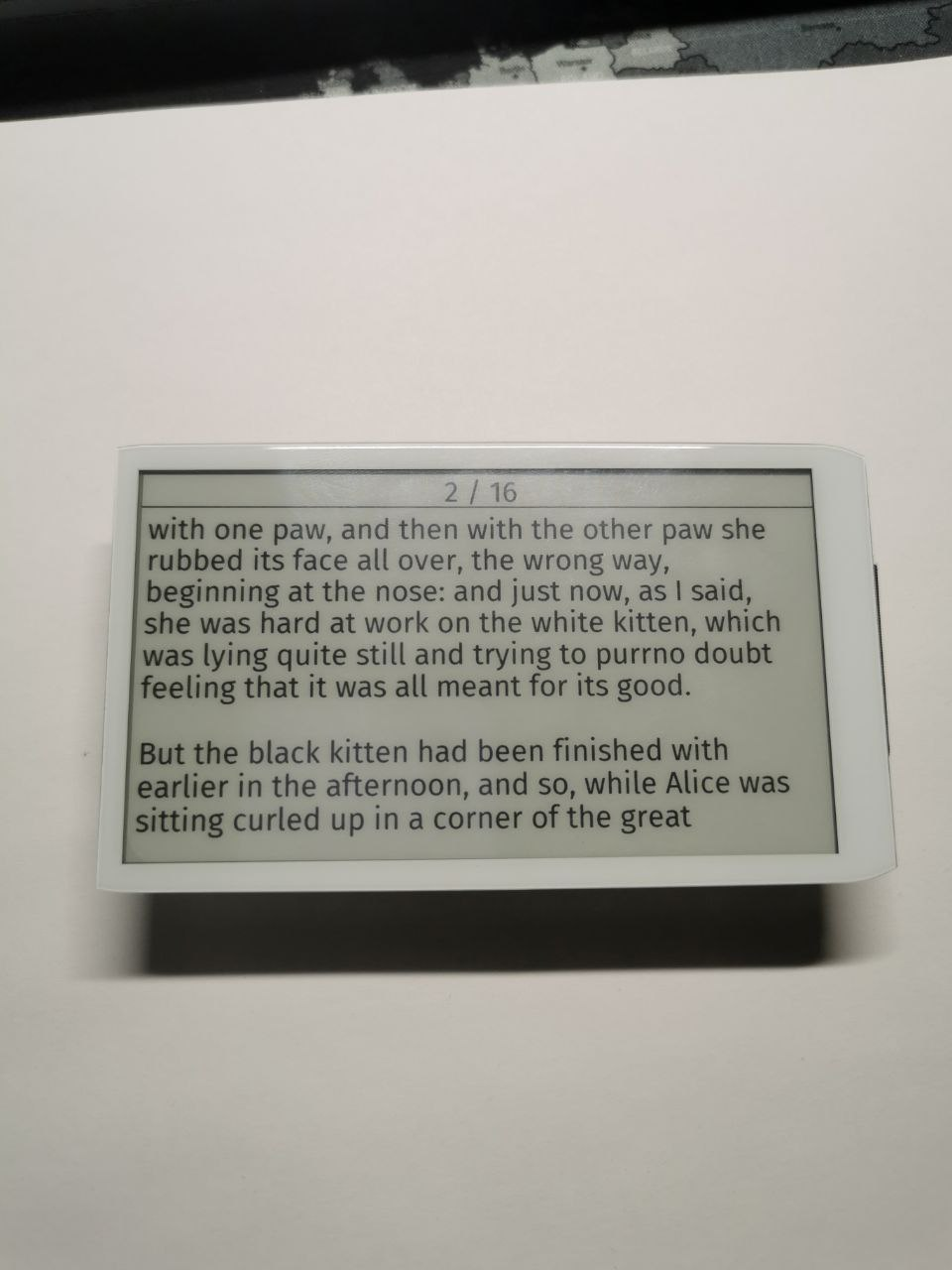


Рисунок 6.6 – Пример отображения текста книги в режиме чтения

Также для экономии заряда аккумулятора предусмотрена возможность выключения устройства, в ходе которого микроконтроллер переходит в режим глубокого сна, а питания дисплея и светодиодов вовсе отключается. Поскольку в данной работе используется E-ink экран, то перед выключением мы можем рисовать картинку для большей наглядности и интуитивности происходящего. На рисунке 6.7 приведено фото в выключенном состоянии.



Рисунок 6.7 – Выключенное устройство

Также была произведена оценка потребляемой мощности устройства в разных режимах работы, таких как: выключенный режим, режим ожидания, выбор книги, обновление экрана.

Пока устройство выключено оно потребляет в среднем около 629 мкВт.

Когда устройство включено и готово обрабатывать нажатия на сенсор или кнопки, то оно находится в режиме ожидания и потребляет в среднем около ??? Вт.

Больше всего устройство потребляет во время обновления экрана, в среднем около ??? Вт.

ТАБЛИЦА С ПОТРЕБЛЕНИЕМ

Заключение

В данной работе приведён процесс разработки электронной книги на базе микроконтроллера *ESP32* с использованием экрана на электронных чернилах и сенсорной панели. Также приведена структурная схема с кратким описанием взаимодействия элементов в ней. Рассмотрен общий алгоритм функционирования устройства. Также выполнена проверка работоспособности устройства опытным путем.

В ходе выполнения, был создан телеграм бот для удобной настройки электронной книги. Так же были получены навыки взаимодействия с *HTTP* запросами к различным сервисам для получения информации.

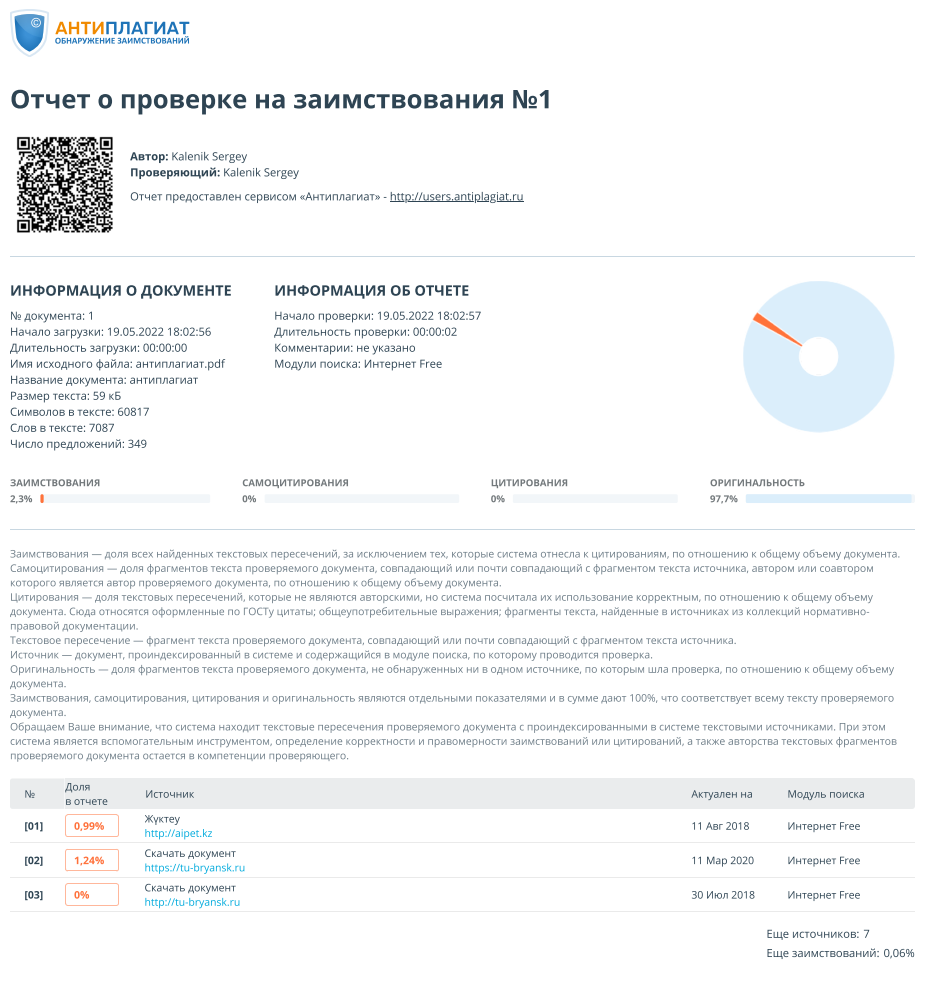
Для большей продолжительности работы устройства была добавлена особенная сборка прошивки *LONG\_LIFE\_MODE* позволяющая более разумно экономить заряд батареи, но данная сборка лишена возможности взаимодействия с пользователем через телеграм бота, а также не обрабатывает нажатие кнопок.

ДОПИСАТЬ ДО КОНЦА СТРАНИЦЫ

Список используемых источников

1. Электронная книга (устройство) [Электронный ресурс]. – Режим доступа: <https://en.wikipedia.org/wiki/E-reader>
2. GitHub: joeycastillo / The-Open-Book [Электронный ресурс]. – Режим доступа: <https://github.com/joeycastillo/The-Open-Book>
3. GitHub: adafruit / Adafruit-PyBadge-PCB [Электронный ресурс]. – Режим доступа: <https://github.com/adafruit/Adafruit-PyBadge-PCB>
4. GitHub: atomic14 / diy-esp32-epub-reader [Электронный ресурс]. – Режим доступа: <https://github.com/atomic14/diy-esp32-epub-reader>
5. Электронная книга своими руками на STM32H750 [Электронный ресурс]. – Режим доступа: https://habr.com/ru/post/528148/
6. Техническое описание E-ink экрана [Электронный ресурс]. – Режим доступа: <https://github.com/Xinyuan-LilyGO/LilyGo-EPD47/blob/master/Display_datasheet.pdf>
7. Visual Studio IDE [Электронный ресурс]. – Режим доступа: <https://visualstudio.microsoft.com/>
8. Visual Micro - Arduino IDE for Visual Studio [Электронный ресурс]. – Режим доступа: <https://www.visualmicro.com/>
9. Arduino IDE [Электронный ресурс]. – Режим доступа: https://www.arduino.cc/en/software
10. GitHub Copilot. Your AI pair programmer [Электронный ресурс]. – Режим доступа: <https://copilot.github.com/>
11. avr-gcc - GCC Wiki [Электронный ресурс]. – Режим доступа: <https://gcc.gnu.org/wiki/avr-gcc>
12. Arduino - Wikipedia [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Arduino>
13. Telegram Bot API [Электронный ресурс]. – Режим доступа: <https://core.telegram.org/bots/api>
14. GitHub: witnessmenow / Universal-Arduino-Telegram-Bot [Электронный ресурс]. – Режим доступа: <https://github.com/witnessmenow/Universal-Arduino-Telegram-Bot>
15. GitHub: Xinyuan-LilyGO / LilyGo-EPD47 [Электронный ресурс]. – Режим доступа: <https://github.com/Xinyuan-LilyGO/LilyGo-EPD47>
16. GitHub: LennartHennigs / Button2 [Электронный ресурс]. – Режим доступа: <https://github.com/LennartHennigs/Button2>
17. GitHub: espressif / arduino-esp32 [Электронный ресурс]. – Режим доступа: <https://github.com/espressif/arduino-esp32>

ПРИЛОЖЕНИЕ А  
(Обязательное)  
Отчет о проверке на заимствования



ПРИЛОЖЕНИЕ Б  
(Обязательное)  
Код программы

**Файл E-book.ino:**

#include "Menu.h"

#include "FileHelper.h"

#include <Arduino.h>

#include <touch.h>

#include "DrawHelper.h"

#include "Button2.h"

#include "Defines.h"

#include <Wire.h>

Button2 btnPower(BUTTON\_POWER);

TouchClass touch;

Menu menu;

void setup()

{

Serial.begin(115200);

InitializeTouch();

InitializeScreen();

fileHelper.InitializeFileSystem();

btnPower.setPressedHandler(buttonPressed);

drawStartImage();

menu.Show();

}

void loop()

{

btnPower.loop();

if (digitalRead(TOUCH\_INT) && touch.scanPoint())

{

uint16\_t x, y;

touch.getPoint(x, y, 0);

menu.HitTest(x, y);

}

}

void buttonPressed(Button2& button)

{

Serial.println("Button attached to pin " + String(button.getAttachPin()));

if (button.getAttachPin() == BUTTON\_POWER)

{

//fileHelper.TryChangeFileSystem();

drawSleepImage();

BookSleep();

}

}

void InitializeTouch()

{

pinMode(TOUCH\_INT, INPUT\_PULLUP);

Wire.begin(15, 14);

if (!touch.begin())

{

Serial.println("start touchscreen failed");

while (1);

}

Serial.println("Started Touchscreen poll...");

}

void BookSleep()

{

touch.sleep();

delay(5);

pinMode(14, INPUT);

pinMode(15, INPUT);

epd\_poweroff\_all();

// Set to wake up by GPIO39

esp\_sleep\_enable\_ext1\_wakeup(GPIO\_SEL\_39, ESP\_EXT1\_WAKEUP\_ALL\_LOW);

esp\_deep\_sleep\_start();

}

**Файл Menu.h:**

#pragma once

#include <Arduino.h>

#include <string.h>

class Menu

{

public:

enum BookState { MENU, BOOK };

void Show();

void FindBooks();

void DrawBooks();

void HitTest(uint16\_t x, uint16\_t y);

protected:

String\*\* m\_pBookNames = nullptr;

int m\_nBooksCount = 0;

int m\_nSelected = 1;

BookState m\_state = MENU;

};

**Файл Menu.cpp:**

#include "Menu.h"

#include "DrawHelper.h"

#include "firasans.h"

#include "FileHelper.h"

#include "Page.h"

PageManager pageManager;

void Menu::Show()

{

FindBooks();

DrawBooks();

}

void Menu::FindBooks()

{

//fileHelper.TryChangeFileSystem();

m\_nBooksCount = 0;

m\_nSelected = 1;

if (m\_pBookNames != nullptr)

{

for (int i = 0; i < m\_nBooksCount; i++)

free(m\_pBookNames[i]);

free(m\_pBookNames);

m\_pBookNames = nullptr;

}

const char\* dirname = "/";

Serial.printf("Listing directory: %s\r\n", dirname);

File root = fileHelper.GetCurrentFileSystem()->open(dirname);

if (!root)

{

Serial.println("- failed to open directory");

return;

}

if (!root.isDirectory())

{

Serial.println(" - not a directory");

return;

}

File file = root.openNextFile();

while (file)

{

if (!file.isDirectory())

{

Serial.print(" FILE: ");

Serial.print(file.name());

Serial.print("\tSIZE: ");

Serial.println(file.size());

m\_nBooksCount++;

constexpr int nBlocksize = 3;

if (m\_nBooksCount % nBlocksize == 1)

{

m\_pBookNames = (String\*\*)realloc(m\_pBookNames, nBlocksize \* ((m\_nBooksCount / nBlocksize) + 1) \* sizeof(String\*));

}

m\_pBookNames[m\_nBooksCount - 1] = new String(file.name());

}

file = root.openNextFile();

}

}

void Menu::DrawBooks()

{

setFont(FiraSans);

constexpr int nVisibleBooks = 3;

constexpr int nBooksMargin = 30;

constexpr int nFrameXMargin = 30;

constexpr int nBookHeight = EPD\_HEIGHT - 2 \* nBooksMargin;

constexpr int nBookWidth = ((EPD\_WIDTH - (nBooksMargin + nFrameXMargin \* 2)) / 3) - nBooksMargin;

constexpr int nMiddle = EPD\_HEIGHT / 2;

constexpr int nGlyphSize = 20;

constexpr int nRight = EPD\_WIDTH - nFrameXMargin;

epd\_poweron();

ClearFrameBuffer();

// Draw triangle glyphs

fillTriangle(nFrameXMargin, nMiddle - nGlyphSize, nFrameXMargin, nMiddle + nGlyphSize, nFrameXMargin - nGlyphSize, nMiddle);

fillTriangle(nRight, nMiddle - nGlyphSize, nRight, nMiddle + nGlyphSize, nRight + nGlyphSize, nMiddle);

// Draw Books

for (int nBook = 0; nBook < nVisibleBooks; nBook++)

{

const int nBookIndex = (m\_nSelected - 1) \* nVisibleBooks + nBook;

if (nBookIndex >= m\_nBooksCount || m\_pBookNames[nBookIndex]->isEmpty())

break;

Serial.println("m\_pBookNames[" + String(nBookIndex) + "] = " + \*m\_pBookNames[nBookIndex]);

int nXOffset = nFrameXMargin + nBooksMargin + nBook \* (nBookWidth + nBooksMargin);

drawRect(nXOffset, nBooksMargin, nBookWidth, nBookHeight, 2);

String strBookName = m\_pBookNames[nBookIndex]->c\_str();

strBookName.replace("/", "");

strBookName.replace(".txt", "");

drawString(nXOffset + nBookWidth / 2, nBooksMargin + nBookHeight / 2 + 15, strBookName, CENTER);

}

epd\_clear();

UpdateScreen();

epd\_poweroff();

}

void Menu::HitTest(uint16\_t x, uint16\_t y)

{

if (m\_state == BOOK)

{

constexpr int nMiddle = EPD\_WIDTH / 2;

constexpr int nHalfWidth = EPD\_WIDTH / 6;

constexpr int nHeight = 100;

if (x > nMiddle - nHalfWidth && x < nMiddle + nHalfWidth && y > EPD\_HEIGHT - nHeight)

{

Show();

m\_state = MENU;

}

else if (x < nMiddle)

pageManager.GoToPreviousPage();

else

pageManager.GoToNextPage();

}

else if (m\_state == MENU)

{

constexpr int nFrameXMargin = 30;

constexpr int nMiddle = EPD\_HEIGHT / 2;

constexpr int nRight = EPD\_WIDTH - nFrameXMargin;

constexpr int nVisibleBooks = 3;

constexpr int nCorrection = 15;

constexpr int nGlyphSize = 20 + nCorrection;

// check click on triangle glyphs

if (x < nFrameXMargin + nCorrection && x > 0 && y > nMiddle - nGlyphSize && y < nMiddle + nGlyphSize) // left

{

if (m\_nSelected > 1)

{

m\_nSelected--;

DrawBooks();

}

}

else if (x < EPD\_WIDTH && x > nRight - nCorrection && y > nMiddle - nGlyphSize && y < nMiddle + nGlyphSize) // right

{

if (m\_nSelected \* nVisibleBooks < m\_nBooksCount)

{

m\_nSelected++;

DrawBooks();

}

}

else // books

{

constexpr int nBooksMargin = 30;

constexpr int nBookHeight = EPD\_HEIGHT - 2 \* nBooksMargin;

constexpr int nBookWidth = ((EPD\_WIDTH - (nBooksMargin + nFrameXMargin \* 2)) / 3) - nBooksMargin;

for (int nBook = 0; nBook < nVisibleBooks; nBook++)

{

if (x >= nFrameXMargin + nBooksMargin + nBook \* (nBookWidth + nBooksMargin) && x <= nFrameXMargin + nBooksMargin + nBook \* (nBookWidth + nBooksMargin) + nBookWidth &&

y >= nBooksMargin && y <= nBooksMargin + nBookHeight)

{

const int nBookIndex = (m\_nSelected - 1) \* nVisibleBooks + nBook;

if (nBookIndex >= m\_nBooksCount || m\_pBookNames[nBookIndex]->isEmpty())

break;

Serial.println("m\_pBookNames[" + String(nBookIndex) + "] = " + \*m\_pBookNames[nBookIndex]);

pageManager.OpenBook(\*m\_pBookNames[nBookIndex]);

m\_state = BOOK;

break;

}

}

}

}

}