

Министерство образования Республики Беларусь

Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Факультет компьютерных сетей и систем

Кафедра электронных вычислительных средств

Дисциплина: Микропроцессорные средства и системы

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
на тему
**РАЗРАБОТКА ЦИФРОВОГО УСТРОЙСТВА НА БАЗЕ
МИКРОКОНТРОЛЛЕРА**
БГУИР КП 1-40 02 02 012 ПЗ

Студент гр. 150701 Кривальцевич Е.А.

Руководитель: Порхун М.И.

Минск 2024

Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
Факультет компьютерных систем и сетей

УТВЕРЖДАЮ
Заведующий кафедрой

(подпись)
«_____» _____ 2024 г.

ЗАДАНИЕ

по курсовому проектированию

Студенту гр.150701 Кривальцевичу Егору Александровичу

1. Тема проекта Разработка цифрового устройства на базе микроконтроллера

2. Срок сдачи студентом законченного проекта 13 декабря 2024г.

3. Исходные данные к проекту _____

1. Разработать фотоаппарат на базе микроконтроллера.

2. Фотоаппарат должен выполнять следующие действия:

1) фотографировать по нажатию кнопки;

2) преобразовывать изображение в формат JPEG с разрешением 1024 на 768 пикселей;

3) сохранять изображение на sd карту.

4. Содержание расчетно-пояснительной записки (перечень вопросов, которые подлежат разработке)

Введение. Постановка задачи.

1. Анализ задачи. Функциональная спецификация системы.

2. Предварительное проектирование системы. Разбиение системы на модули, выбор соотношения между аппаратными и программными средствами, построение структурной схемы аппаратной части системы. Описание структурной схемы.

3. Проектирование аппаратных средств системы. Выбор типа микроконтроллера. Разработка принципиальной схемы системы. Описание работы системы по принципиальной схеме.

4. Проектирование программного обеспечения. Разработка схемы алгоритма работы системы и программы. Описание алгоритма работы системы и программы.

5. Моделирование работы системы. Выбор среды моделирования. Описание процесса моделирования
Заключение

5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)

1. Схема электрическая структурная

2. Схема электрическая принципиальная

3. Схема алгоритма работы системы

6. Консультант по проекту (с обозначением разделов проекта) Качинский М.В.
Порхун М.И.

7. Дата выдачи задания 13.09.2024

8. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и трудоемкости отдельных этапов)

Введение, раздел 1 27.09.2024 г. – 10%

Раздел 2 11.10.2024 г. – 10%

Раздел 3 01.11.2024 г. – 25%

Раздел 4, раздел 5 29.11.2024 г. – 40%

Оформление пояснительной записки и графического материала

13.12.2024 г. – 15%

Защита курсового проекта 19.12 – 27.12.2024 г.

РУКОВОДИТЕЛЬ

(подпись)

Задание принял к выполнению

(дата и подпись студента)

СОДЕРЖАНИЕ

Введение.....	5
1 Анализ технического задания.....	6
1.1 Анализ технического задания	6
1.2 Функциональная спецификация системы	6
2 Предварительное проектирование системы.....	8
2.1 Разбиение системы на модули	8
2.2 Разработка структурной схемы устройства	8
3 Проектирование аппаратных средств системы.....	10
3.1 Выбор аппаратных модулей	10
3.2 Разработка принципиальной схемы устройства	11
4 Проектирование программного обеспечения.....	14
4.1 Разработка схемы алгоритма работы системы	14
4.2 Разработка схемы алгоритма работы программы	14
5 Моделирование работы системы.....	16
5.1 Описание процесса моделирования	16
Заключение	22
Список использованных источников	23
Приложение А (Обязательное) Схема электрическая структурная	24
Приложение Б (Обязательное) Схема электрическая принципиальная	26
Приложение В (Обязательное) Схема алгоритма работы устройства	29
Приложение Г (Обязательное) С описание устройства	32
Приложение Д (Обязательное) HTML и CSS описание страниц	39

ВВЕДЕНИЕ

Курсовой проект посвящён разработке цифрового устройства на базе микроконтроллера с использованием систем автоматизированного проектирования и тестирование на микроконтроллере.

Цифровые устройства на базе микроконтроллеров находят широкое применение в различных областях, от бытовой электроники до промышленных систем автоматизации. Микроконтроллеры являются ключевыми элементами управления многими электронными устройствами благодаря их универсальности, компактности и низкой стоимости. Разработка таких устройств требует знания как программирования, так и основ аппаратного обеспечения.

Целью курсового проекта является получение практических навыков проектирования цифровых устройств на базе микроконтроллеров.

В рамках данного курсового проекта по дисциплине «Микропроцессорные средства и системы» рассматривается реализация простого фотоаппарата на базе микроконтроллера ESP32-CAM. Этот микроконтроллер оснащен встроенной камерой и поддерживает программирование через Arduino IDE. Проект направлен на изучение процессов проектирования, программирования и тестирования цифровых устройств, а также на закрепление навыков работы с микроконтроллерами.

Реализация фотоаппарата включает в себя захват изображений, их обработку и сохранение. Итогом работы станет устройство, способное выполнять функции цифровой камеры с базовыми возможностями.

1 АНАЛИЗ ЗАДАЧИ. ФУНКЦИОНАЛЬНАЯ СПЕЦИФИКАЦИЯ СИСТЕМЫ

1.1 Анализ технического задания

В данном курсовом проекте реализуется цифровое устройство в виде фотоаппарата на базе микроконтроллера ESP32-CAM.

Фотоаппарат построен на базе микроконтроллера ESP32-CAM, который представляет собой мощное и универсальное решение для разработки встроенных систем и устройств интернета вещей (IoT). Этот микроконтроллер интегрирует в себе множество функций, что делает его идеальным для реализации фотосъемки и хранения данных. В частности, ESP32-CAM оснащен модулем Wi-Fi, что позволяет осуществлять беспроводное управление устройством. Эта функция открывает возможности для удаленного доступа к камере, управления съемкой, а также передачи изображений на внешние устройства или в облачные сервисы.

Одной из ключевых особенностей этого устройства является возможность подключения камеры OV2640. Эта камера обеспечивает высокое качество изображений и поддерживает различные разрешения, что позволяет адаптировать устройство под разные задачи.

Еще одной важной характеристикой ESP32-CAM является поддержка работы с microSD-картой. Это позволяет сохранять полученные изображения на внешнюю память, что не только увеличивает объем доступного хранилища, но и упрощает процесс передачи данных. Таким образом можно легко извлекать карты памяти для переноса данных на другие устройства, что особенно полезно в условиях ограниченного доступа к сети.

В ходе реализации проекта предполагается разработка и интеграция программного и аппаратного обеспечения. Программное обеспечение будет включать в себя прошивку для управления камерой и Wi-Fi, а также интерфейсы для пользователя, обеспечивающие простоту и удобство в использовании. Аппаратное обеспечение, в свою очередь, будет включать в себя схемы подключения всех компонентов, необходимых для корректной работы фотоаппарата.

Проект нацелен на создание функционального устройства, которое не только отвечает требованиям технического задания, но и предоставляет пользователю широкий спектр возможностей для фотосъемки, хранения и передачи изображений. Все эти аспекты будут проработаны в рамках проекта, чтобы обеспечить надежность и эффективность работы фотоаппарата на базе ESP32-CAM.

1.2 Функциональная спецификация системы

Технические требования к данному устройству определяют его основные функциональные возможности. Фотоаппарат должен выполнять следующие задачи:

1. Фотографирование по нажатию кнопки. Устройство должно уметь обрабатывать сигнал от кнопки, инициирующей процесс захвата изображения с камеры ESP32-CAM,

2. Преобразование изображения в формат JPEG с разрешением 1024x768 пикселей. После захвата изображения оно должно быть преобразовано в формат JPEG для эффективного хранения и последующей обработки. Разрешение 1024x768 пикселей выбрано для обеспечения баланса между качеством изобра-

жения и размером файла,

3. Сохранение изображения на SD-карту. После преобразования изображение должно быть записано на карту памяти формата SD, подключённую к микроконтроллеру, что позволит пользователю сохранять фотографии для дальнейшего просмотра или переноса на другие устройства.

2 ПРЕДВАРИТЕЛЬНОЕ ПРОЕКТИРОВАНИЕ СИСТЕМЫ

2.1 Разбиение системы на модули

Для обеспечения надёжной работы системы фотоаппарата на базе микроконтроллера ESP32-CAM был разбит на несколько функциональных модулей. Каждый модуль отвечает за выполнение определённых задач, что упрощает разработку и тестирование устройства. Основные модули системы:

1 Модуль инициализации камеры. Этот модуль отвечает за конфигурацию и запуск камеры ESP32-CAM. В нём задаются параметры камеры, такие как частота тактирования (XCLK), разрешение изображения (1024x768 пикселей) и формат данных (JPEG). Также модуль управляет буферами кадров для хранения изображений, полученных с камеры.

2 Модуль управления веб-сервером. Данный блок реализует веб-сервер на основе ESP32s, который предоставляет пользователю доступ к функционалу устройства через веб-браузер. Он обрабатывает запросы на:

- Отображение видео потока с камеры в режиме реального времени.
- Съёмку фотографий и сохранение их на SD-карту.
- Возврат страницы с подтверждением успешного выполнения операции.

3 Модуль захвата и сохранения изображений. После получения команды от веб-сервера этот модуль захватывает кадр с камеры, конвертирует его в JPEG и сохраняет на SD-карту. Каждое изображение получает уникальное имя, основанное на текущем времени, чтобы избежать перезаписи существующих файлов.

4 Модуль управления Wi-Fi. Этот модуль настраивает и управляет точкой доступа Wi-Fi, создаваемой микроконтроллером ESP32-CAM. Он позволяет пользователю подключаться к устройству через беспроводную сеть и взаимодействовать с веб-интерфейсом. Для этого задаются параметры SSID и пароля для доступа.

5 Модуль работы с файловой системой. Данный блок отвечает за работу с файловой системой SD-карты. Он инициализирует SD-карту, открывает и закрывает файлы для записи, а также проверяет наличие свободного места для сохранения изображений.

2.2 Разработка структурной схемы устройства

Устройство, реализованное в данном курсовом проекте, состоит из трёх основных аппаратных блоков:

- 1 Модуль камеры OV2640.
- 2 Память на основе SD-карты.
- 3 Микроконтроллер ESP32-cam.

Модуль OV2640 обеспечивает захват изображения с последующим преобразованием его в цифровой формат для дальнейшей обработки микроконтроллером ESP32-cam.

Для долговременного хранения изображений используется SD-карта на 4Гб, которая отформатирована в файловой системе FAT32. Этот формат широко поддерживается и позволяет легко работать с большими файлами. В данном проекте SD-карта служит для хранения снимков, захваченных камерой, в формате JPEG.

Микроконтроллер ESP32-sam — это основной вычислительный блок системы, который управляет работой камеры, обработкой изображений, сохранением данных на SD-карту и организацией взаимодействия с пользователем через веб-интерфейс.

Для удобства разработки и упрощения структуры программного обеспечения микроконтроллер ESP32-sam разделён на несколько модулей, каждый из которых выполняет свои специфические функции.

Схема электрическая структурная представлена в Приложении А.

3 ПРОЕКТИРОВАНИЕ АППАРАТНЫХ СРЕДСТВ СИСТЕМЫ

3.1 Выбор аппаратных модулей

При выборе аппаратных модулей для реализации устройства было учтено несколько ключевых факторов: совместимость с микроконтроллером ESP32S, энергоэффективность, функциональные возможности и простота интеграции.

Камера OV2640 была выбрана за её гибкость и широкую популярность в проектах с микроконтроллерами. OV2640 поддерживает различные разрешения, что позволяет легко настраивать качество изображения. Она имеет встроенную функцию аппаратного сжатия JPEG, что облегчает работу микроконтроллера, позволяя ему сосредоточиться на других задачах, таких как обработка данных и управление беспроводной связью. Камера легко интегрируется с ESP32S благодаря совместимости с интерфейсами микроконтроллера.

Формат JPEG (Joint Photographic Experts Group) — это один из наиболее популярных форматов для сжатия изображений с потерями, широко используемый в цифровой фотографии и веб-дизайне. Основная особенность JPEG заключается в том, что он использует сжатие с потерями, что позволяет значительно уменьшить размер изображения за счёт отбрасывания малозаметной информации. JPEG использует дискретное косинусное преобразование (DCT), которое разделяет изображение на блоки, а затем кодирует каждый блок в частотной области, уменьшая объём данных за счёт отбрасывания менее значимых деталей.

Формат был разработан группой JPEG в 1992 году и с тех пор стал стандартом для хранения и передачи растровых изображений, благодаря высокому уровню сжатия при приемлемом уровне качества изображения[1].

Для длительного хранения данных используется SD-карта. В проекте применяется SD-карта, отформатированная в файловой системе FAT32, что обеспечивает стабильную работу с файлами большого размера, такими как изображения в формате JPEG. Выбор данного модуля обусловлен необходимостью хранения значительных объёмов данных и удобством доступа к ним.

Основным вычислительным блоком системы является ESP32S, который сочетает в себе возможности работы с камерой, Wi-Fi подключение и работу с файловой системой. ESP32s был выбран из-за его интегрированной поддержки работы с изображениями и возможностью быстрой передачи данных через беспроводные сети, что упрощает взаимодействие устройства с пользователем.

Система FAT берет своё начало в конце 1970-х годов, когда IBM и Microsoft начали сотрудничество над созданием операционной системы для первых персональных компьютеров. Первоначальная версия была представлена как FAT12. Она позволила организовывать информацию на дисках емкостью до 16 МБ, что являлось значительным шагом в то время.

FAT12: Первая версия файловой системы, разработанная для поддержки небольших носителей данных, таких как гибкие диски. Основное новшество заключалось в таблице размещения файлов (File Allocation Table, FAT), которая обеспечивала контроль за свободными и занятыми блоками на диске.

FAT16: С развитием технологий и увеличением объема жестких дисков появилась необходимость в расширении возможностей файловой системы. FAT16 позволила работать с дисками объемом до 2 ГБ. Это нововведение стало значительным прогрессом и обеспечило дальнейшее развитие компьютерных технологий.

Файловая система FAT32 была представлена в 1996 году с релизом Windows 95 OSR2. Она была ответом на вызовы времени, связанные с потребностью хранения больших объемов данных на компактных носителях. Главными особенностями системы стали:

- Увеличение максимального размера дискового раздела до 2 ТБ;
- Совместимость с предыдущими версиями FAT, что обеспечивало плавный переход для пользователей;
- Улучшенная управляемость файлами, благодаря экономному использованию пространства[2].

3.2 Разработка принципиальной схемы устройства

В проекте была разработана принципиальная схема устройства, которая включает ключевые аппаратные компоненты, такие как микроконтроллер ESP32S, камера OV2640 и слот для SD-карты. Эти элементы образуют систему, обеспечивающую захват изображений, их обработку и хранение. Далее подробно описаны функции и подключения каждого из элементов схемы.

3.2.1 Микроконтроллер ESP32S:

Микроконтроллер ESP32S является центральным узлом системы, который управляет всеми процессами и обеспечивает взаимодействие между периферийными устройствами. ESP32S обладает встроенным модулем Wi-Fi, что позволяет устройству поддерживать беспроводную связь для передачи данных пользователю через веб-интерфейс. На рисунке 3.1 представлена распиновка ESP32S[3].

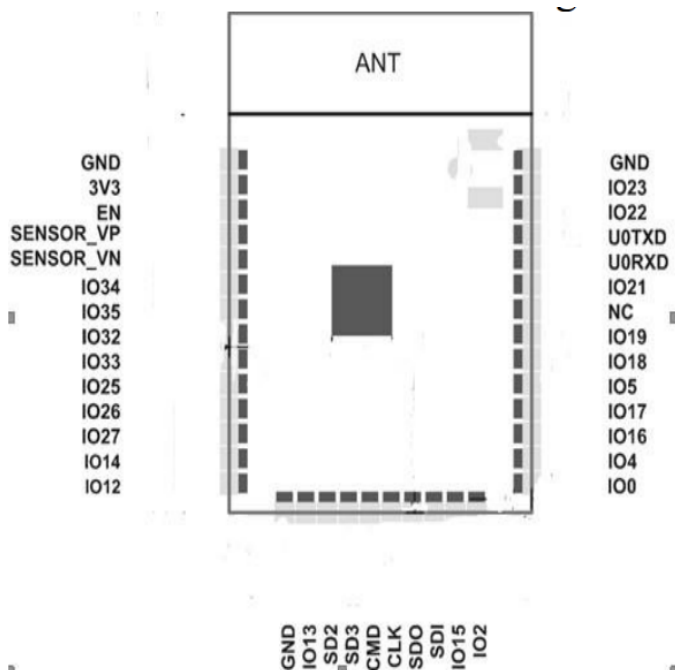


Рисунок 3.1 – Микроконтроллер ESP32S

- Питание: Напряжение питания подаётся на выводы VDD33 и GND, обеспечивая стабильную работу ESP32S.
- Подключение к камере: Для взаимодействия с камерой OV2640 используются специальные пины ESP32S:
- Линии данных камеры (Y7–Y9 и Y2–Y6) подключаются к соответствующим цифровым выводам микроконтроллера, что позволяет получать цифровое

изображение. - Сигнальные линии VSYNC и HREF служат для синхронизации кадров и строк, упрощая процесс передачи данных.

- Тактовый сигнал (PCLK) подаётся на камеру для обеспечения её синхронизации с микроконтроллером.

Подключение к SD-карте: ESP32S поддерживает интерфейс SDIO для взаимодействия со с лотом для SD-карты: - Линии данных (DATA0–DATA3) обеспечивают двунаправленную передачу информации между микроконтроллером и картой памяти.

- Линия команд (CMD) используется для передачи управляющих сигналов, таких как команды чтения и записи.

- Тактовый сигнал (CLK) синхронизирует обмен данными между ESP32S и SD-картой, позволяя достигать высокой скорости передачи.

3.2.2 Камера OV2640 отвечает за захват изображений, которые затем передаются на микроконтроллер для дальнейшей обработки и сохранения. Эта камера способна генерировать изображения в различных разрешениях, и она часто используется в IoT-проектах благодаря своей компактности и поддержке стандарта JPEG для сжатия изображений. На рисунке 3.2 представлена распиновка OV2640[4].

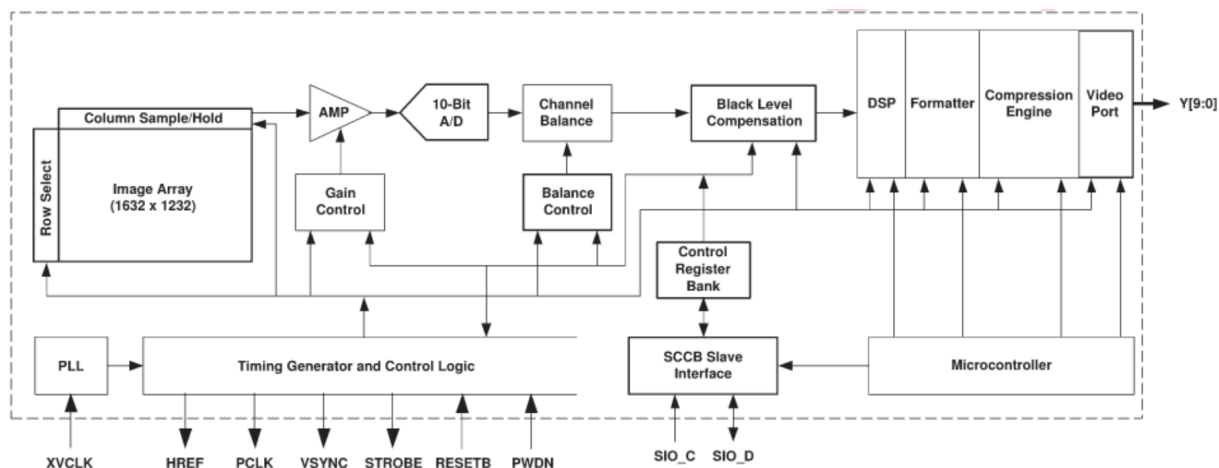


Рисунок 3.2 – Камера OV2640

Камера подключается к ESP32S через цифровые линии данных и сигнальные линии:

- Линии данных (Y7–Y9 и Y2–Y6) используются для передачи каждого пикселя изображения.

- Линии VSYNC и HREF обеспечивают синхронизацию, указывая начало и окончание кадра и строки.

- Линия тактового сигнала (PCLK) контролирует синхронизацию передачи данных, определяя скорость обмена данными между камерой и микроконтроллером.

- Питание камеры: Для работы камеры необходимо подключение к выводам DOVDD и DVDD. Питание камеры стабилизируется с помощью фильтрующих конденсаторов, таких как C1 и C3, что уменьшает влияние высокочастотных шумов и обеспечивает стабильную работу устройства.

3.2.3 Слот для SD-карты служит для длительного хранения изображений, захваченных камерой. Использование SD-карты в качестве хранилища

позволяет сохранять изображения в формате JPEG, что обеспечивает эффективное использование памяти и возможность последующего доступа к данным. На рисунке 3.3 представлена распиновка слота для SD-карты[5].

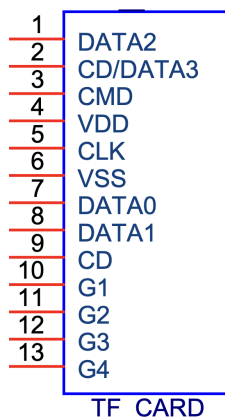


Рисунок 3.3 – Слот для SD-карты

Для взаимодействия с картой памяти применяется интерфейс SDIO, который состоит из нескольких линий:

- Линии данных (DATA0–DATA3) позволяют передавать данные между ESP32S и SD-картой.
- Линия команд (CMD) используется для отправки команд на SD-карту, таких как команды на запись или чтение.
- Линия тактового сигнала (CLK) синхронизирует передачу данных, что позволяет микроконтроллеру быстро и точно обмениваться информацией с картой памяти.
- Форматирование SD-карты: SD-карта отформатирована в файловую систему FAT32, что обеспечивает совместимость с микроконтроллером и позволяет хранить файлы различных размеров. FAT32 также поддерживает долговременное хранение данных и высокую скорость доступа, что важно для эффективной работы устройства.

3.2.4 Для обеспечения стабильной работы всех активных элементов схемы были добавлены пассивные компоненты, такие как конденсаторы и резисторы.

Конденсаторы (C1, C3, C5, C6): Фильтрующие конденсаторы номиналом 0.1 мкФ используются для сглаживания высокочастотных шумов на линиях питания. Они улучшают стабильность и надёжность устройства, предотвращая скачки напряжения, которые могут повлиять на работу чувствительных компонентов, таких как микроконтроллер и камера.

Резисторы (R1, R2, R3): Резисторы с номиналами 1 кОм, 4.7 кОм и 10 кОм обеспечивают согласование уровней сигналов и защиту компонентов от перегрузки. Они также используются для ограничения тока на линиях связи между ESP32S и периферийными устройствами, предотвращая возможное повреждение компонентов при высоких токах.

Схема электрическая принципиальная представлена в Приложении Б.

4 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

4.1 Разработка схемы алгоритма работы системы

В данной части проекта разработан алгоритм работы системы в целом, который включает основные этапы взаимодействия компонентов устройства. Алгоритм работы системы описывает порядок выполнения операций по захвату изображения, его обработке и сохранению на SD-карту, а также организацию взаимодействия с пользователем через веб-интерфейс.

Сначала система выполняет инициализацию микроконтроллера ESP32, настройку Wi-Fi соединения и модулей камеры и SD-карты. Далее происходит захват изображения с камеры, его обработка и сохранение на SD-карту. В завершение система переходит в режим ожидания команд от пользователя через веб-интерфейс, что позволяет загружать и просматривать сохранённые изображения.

Алгоритм работы системы описывает общий процесс, от инициализации до выполнения команд.

Выполняется настройка ESP32, проверка подключения камеры и SD-карты.

Переход в режим точки доступа, установка SSID и пароля. Проверка успешности подключения.

Затем осуществляется подключение и настройка камеры OV2640. Проверка наличия и правильности подключения SD-карты.

Далее система ожидает запросов от пользователя через веб-интерфейс.

Если пришла команда — «Take photo», перейти к процессу захвата, обработки и сохранения изображения.

А далее осуществляется переход к режиму обновления страницы. При получении команды — «Go Back to Main Page» происходит обновление и возврат в режим ожидания.

Схема алгоритма работы системы представлена в Приложении В.

4.2 Разработка схемы алгоритма работы программы

На данном этапе была разработана схема алгоритма работы программы, управляющей микроконтроллером ESP32. Основной алгоритм программы включает следующие этапы:

1. Инициализация системы: Программа выполняет начальную настройку микроконтроллера, устанавливает соединение по Wi-Fi и инициализирует модули камеры и SD-карты.

2. Захват изображения: После инициализации программа отправляет команду на захват изображения с камеры OV2640.

3. Обработка изображения: Программа выполняет сжатие изображения в формат JPEG, что позволяет уменьшить размер файла для хранения.

4. Сохранение на SD-карту: Обработанное изображение сохраняется на SD-карту в формате JPEG.

5. Организация веб-интерфейса: Программа активирует веб-сервер, через который пользователь может получить доступ к сохранённым изображениям.

6. Ожидание команд пользователя: Программа переходит в режим ожидания и обрабатывает команды пользователя, такие как перезагрузка страницы и снимок.

Алгоритм работы программы обеспечивает надёжное выполнение задач по

захвату, обработке и хранению изображений, а также удобное взаимодействие пользователя с системой через веб-интерфейс.

Алгоритм работы программы детализирует действия, выполняемые программой при захвате и сохранении изображений, а также при взаимодействии через веб-интерфейс.

Схема алгоритма работы программы представлена в Приложении В.

5 МОДЕЛИРОВАНИЕ РАБОТЫ СИСТЕМЫ

5.1 Описание процесса моделирования

В процессе моделирования была выполнена проверка работы устройства на реальном оборудовании. Для этого требуемые аппаратные блоки были подключены к микроконтроллеру ESP32S, после чего на него была загружена разработанная прошивка.

Для реализации процесса моделирования были использованы следующие устройств:

- 1 Микроконтроллер ESP32S;
 - 2 Камера OV2640;
 - 3 microSD-карта для хранения изображений;
 - 4 Программатор;
 - 5 Картридер для считывания данных с SD-карты.
- На рисунке 5.1 представлены все использованные устройства.



Рисунок 5.1 – Используемые устройства

Все аппаратные модули, такие как камера OV2640 и слот для SD-карты, были подключены к соответствующим выводам ESP32S в соответствии с принципиальной схемой. Также было подано питание на все компоненты для их корректной работы.

На рисунке 5.2 представлена собранная установка.



Рисунок 5.2 – Собранная установка

Далее было необходимо написать прошивку для микроконтроллера, которая будет выполнять корректное взаимодействие между блоками, а также описать веб-страницу[6], для удобства работы с камерой.

Описанная прошивка микроконтроллера ESP32-cam и файлы веб-страницы представлены в Приложении Г. А файлы веб-страниц описаны в приложении Д.

На ESP32 была загружена прошивка, реализующая функции захвата и сохранения изображения, а также управления устройством через веб-интерфейс. Для этого устройство через microUSB разъем программатора соединяется с компьютером. Прошивка микроконтроллера осуществляется через ArduinoIDE. Для этого необходимо выбрать следующие настройки среды:

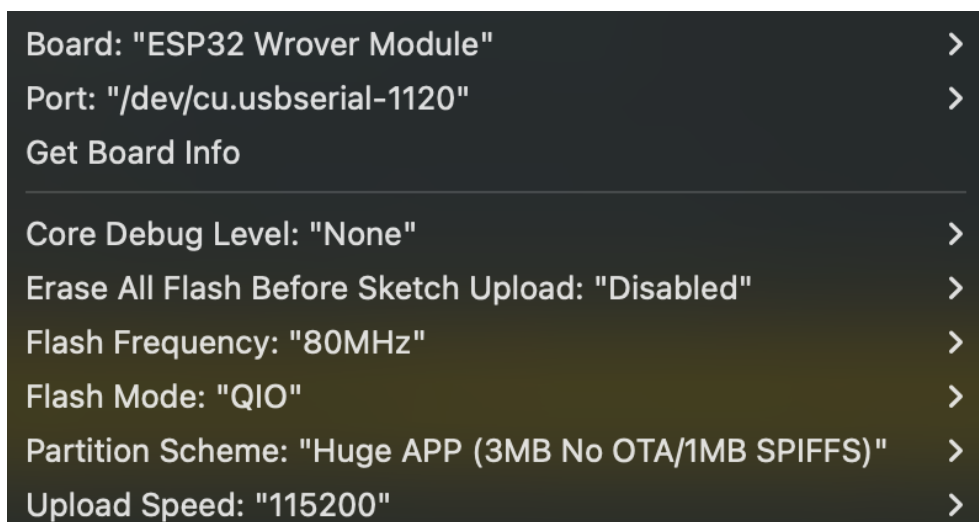


Рисунок 5.3 – Настройка ArduinoIDE

На рисунке 5.4 представлен принцип подключения устройства к компьютеру.

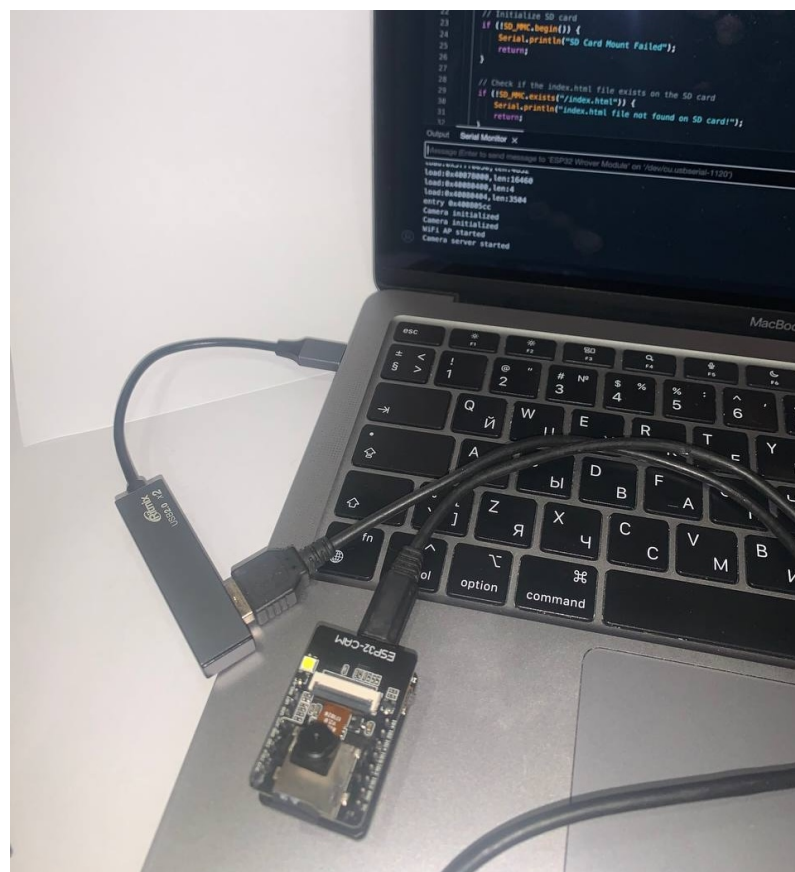


Рисунок 5.4 – Подключение к компьютеру

После загрузки прошивки была проверена базовая работоспособность устройства, включая инициализацию камеры, настройку Wi-Fi соединения и работу с SD-картой. В результате были получены соответствующие сообщения в консоль.

```
Camera initialized
WiFi AP started
Camera server started
```

Рисунок 5.5 – Консоль ArduinoIDEy

Для перехода на веб страницу необходимо в браузере вписать адрес 192.168.4.1, что показано на рисунке 5.6.

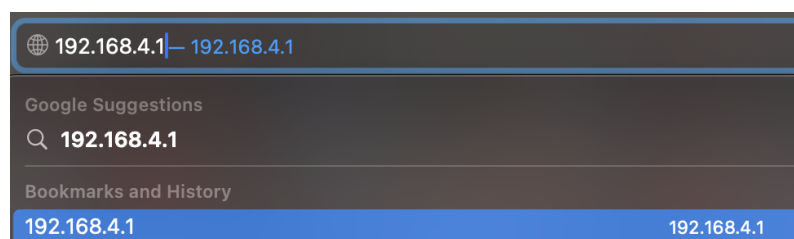


Рисунок 5.6 – Поиск в веб-странице

На этапе тестирования были проверены основные функции устройства:

- Захват изображения при получении команды через веб-интерфейс.
- Сжатие изображения в формат JPEG и сохранение его на SD-карту.
- Проверка сохранения данных с использованием картридера.

Для тестирования необходимо было перейти на веб-страницу. Далее перед пользователем появляется прямая-трансляция, благодаря которой он может отслеживать какое изображение будет сфотографировано. Для того чтобы осуществить съемку необходимо нажать на кнопку «Take Photo».

На рисунке 5.7 представлена основная страница с прямой-трансляцией и используемой кнопкой.

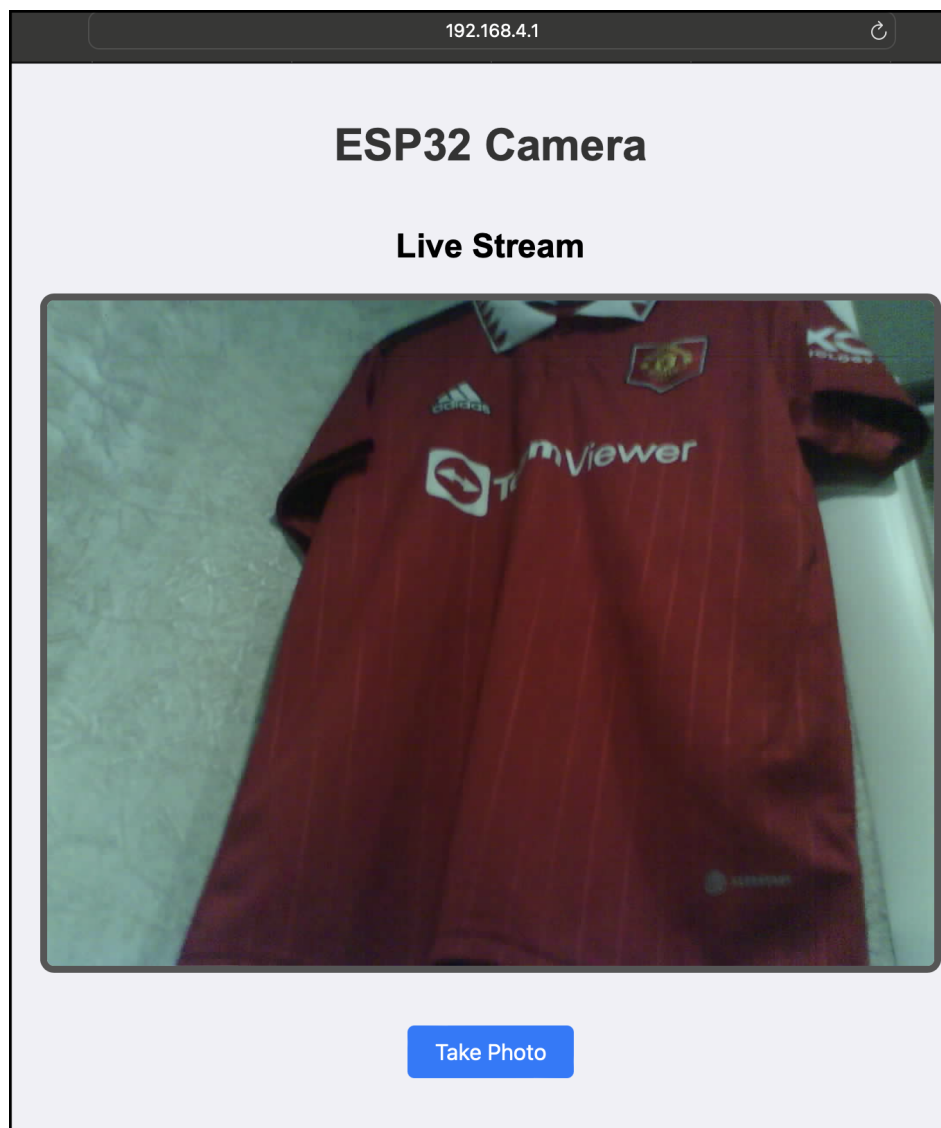


Рисунок 5.7 – Основная страница с прямой-трансляцией

После нажатия на кнопку «Take Photo», осуществляется захват кадра, а затем оно преобразуется к требуемому формату и разрешению и сохраняется на SD-карту. После успешного захвата и сохранения изображения прямая трансляция останавливается. Этот шаг необходим для освобождения ресурсов камеры и буфера, поскольку трансляция требует постоянного доступа к камере. Пользователь получает на странице сообщение о том, что фотография успешно

сохранена. Затем он получает возможность вернуться на начальную страницу через кнопку «Go Back to Main Page».

На рисунке 5.8 представлена основная страница после нажатия на кнопку «Take Photo».

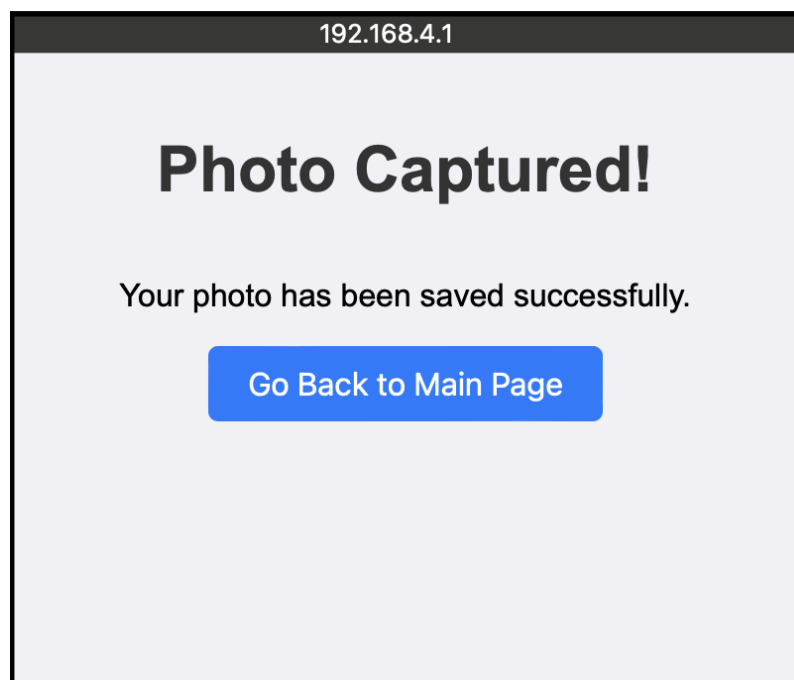


Рисунок 5.8 – Страница после нажатия на кнопку «Take Photo»

Было сделано несколько фотографий. Полученные изображения сохраняются на SD-карте. Для того, чтобы проверить корректность снимков, а также соответствие требованиям необходимо вытащить SD-карту из разъема микроконтроллера и подключить через картридер к компьютеру.

На рисунке 5.9 представлены результаты фотографирования изображения.



Рисунок 5.9 – Результаты фотографирования

Фотографии были сохранены корректно. Для того, чтобы проверить соответствие параметрам необходимо перейти в информацию о изображении, где показано разрешение и формат изображения.

На рисунке 5.10 представлены параметры одного из изображений.

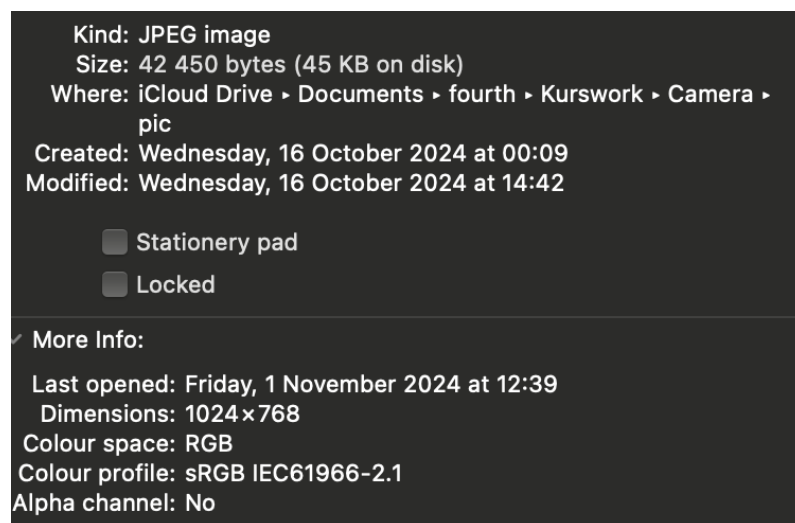


Рисунок 5.10 – Параметры изображения

Таким образом, система отработала корректно. Мы получили изображение в требуемом разрешении(1024x768) и формате(JPEG).

ЗАКЛЮЧЕНИЕ

В процессе выполнения курсового проекта по дисциплине «Микропроцессорные средства и системы» был аппаратно реализован фотоаппарат на базе микроконтроллера ESP32-sam. Для этого была составлена программа на языке С в соответствии данного алгоритма, включающее в себя html описание веб-страницы и её стилистическое оформление через CSS описание.

Получены навыки моделирования устройств на элементной базе и разработки в среде ArduinoIDE. А также отладка работы устройства на аппаратуре.

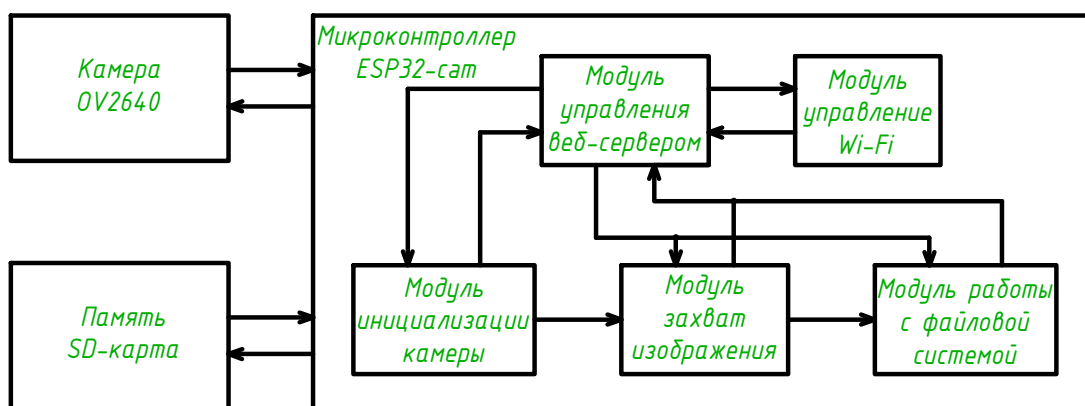
В соответствии с полученными результатами тестирования на микроконтроллере ESP32-sam можно сделать вывод, что система отрабатывает корректно. На SD-карте сохраняются изображения с требуемым разрешением и форматом.

Также были разработаны несколько схем, а именно: схема алгоритма работы устройства, схема электрическая структурная, схема электрическая принципиальная.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

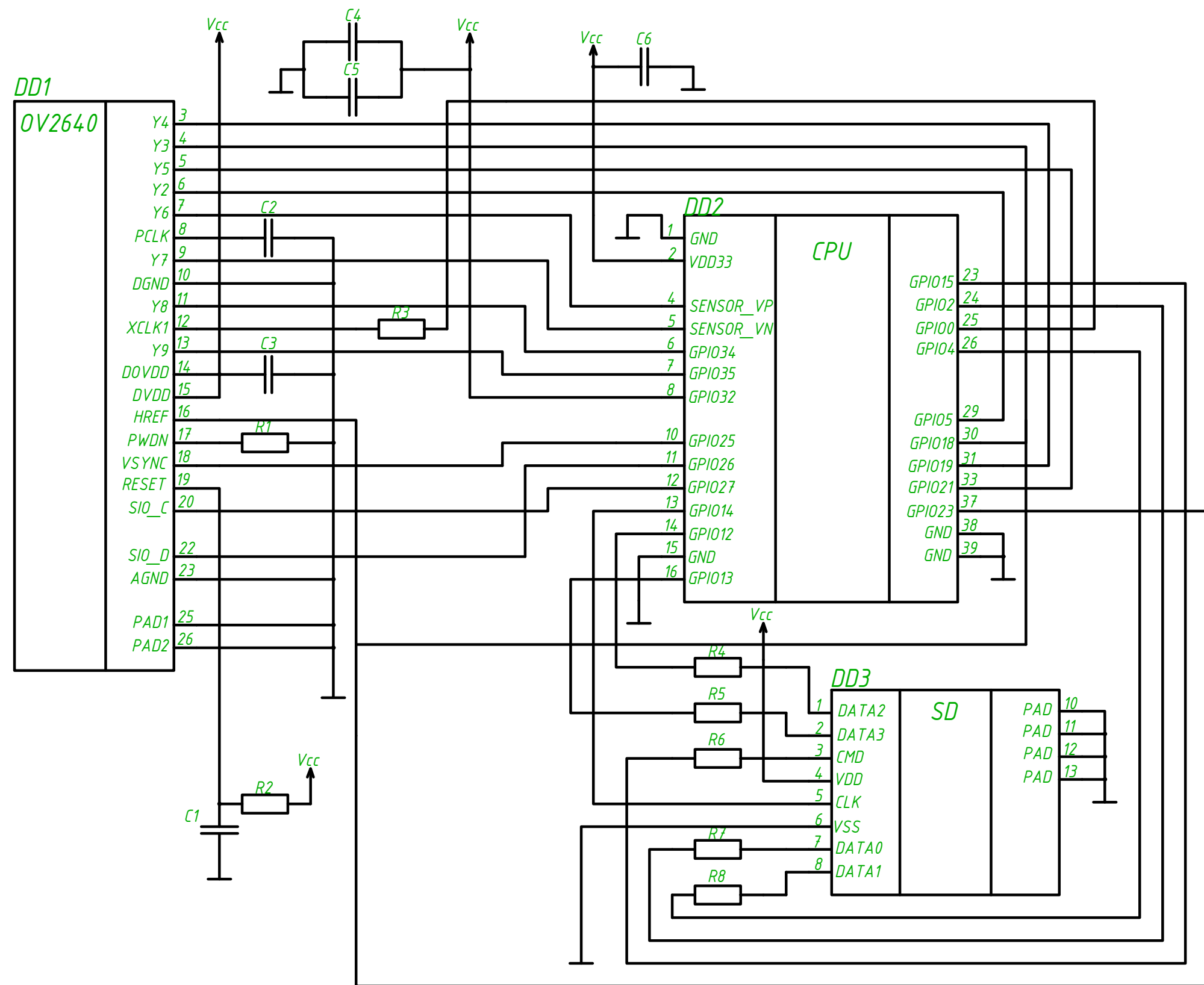
- [1] Гонсалес Р., Вудс Р. Цифровая обработка изображений Издание 3-е, исправленное и дополненное Москва: Техносфера, 2012. – 1104 с.
- [2] Fat32 — что такое [Электронный ресурс] – Электронные данные – Режим доступа: <https://skyeng.ru/magazine/wiki/it-industriya/chto-takoe-fat32/>
- [3] ESP32-S Specification [Электронный ресурс] – Электронные данные – Режим доступа: <https://agelectronica.lat/pdfs/textos/E/ESP-32S.PDF>
- [4] OV2640 Color CMOS UXGA(2.0 MegaPixel) [Электронный ресурс] – Электронные данные – Режим доступа: https://www.waveshare.com/w/upload/9/92/Ov2640_ds_1.8_.pdf
- [5] ESP-WROVER-KIT_20161026A [Электронный ресурс] – Электронные данные – Режим доступа: https://dl.espressif.com/dl/schematics/ESP-WROVER-KIT_SCH-2.pdf
- [6] ESP32-CAM Video Streaming Web Server [Электронный ресурс] – Электронные данные – Режим доступа: <https://randomnerdtutorials.com/esp32-cam-video-streaming-web-server-camera-home-assistant/>

ПРИЛОЖЕНИЕ А
(Обязательное)
Схема электрическая структурная



					ГЧИР.431295.001. Э1					
					Структурная схема фотоаппарата на базе ESP32-сат Схема электрическая структурная			Лит.	Масса	Масштаб
Изм	Лист	№ докум.	Подп.	Дата						
Разраб.	Кривальцевич				Т					
Пров.	Порхун									
								Лист	Листов	1
								ФКСИС, зр.150701		

ПРИЛОЖЕНИЕ Б
(Обязательное)
Схема электрическая принципиальная

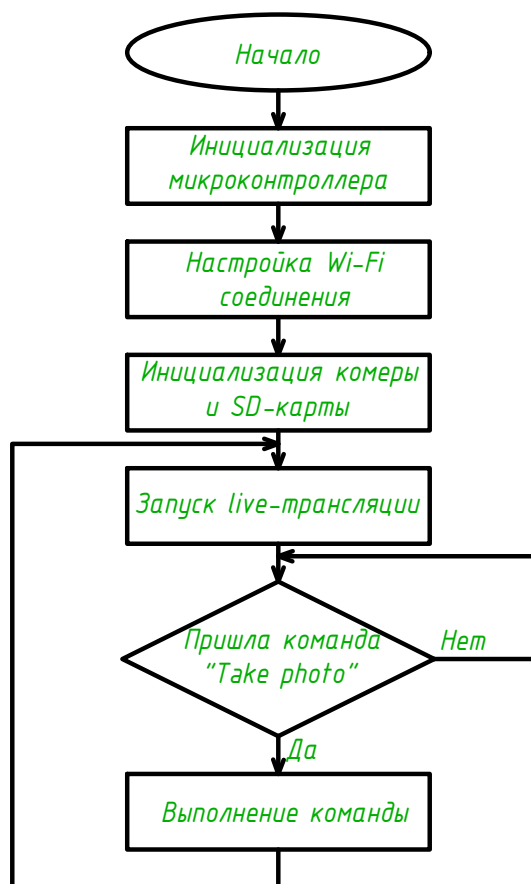


DD1 – OV2640
 DD2 – ESP32S
 DD3 – MicroSD Socket
 Вывод земли – 1, 6, 10, 15, 23, 38, 39
 Вывод питания – 2, 4, 14, 15,
 C1, C3, C5, C6 = 0.1 мкФ
 C2 = 15 пФ
 C4 = 10 мкФ
 R1 = 1 кОм
 R2 = 4.7 кОм
 R3 = 10 кОм
 R4, R5, R6, R7, R8 = 47 кОм

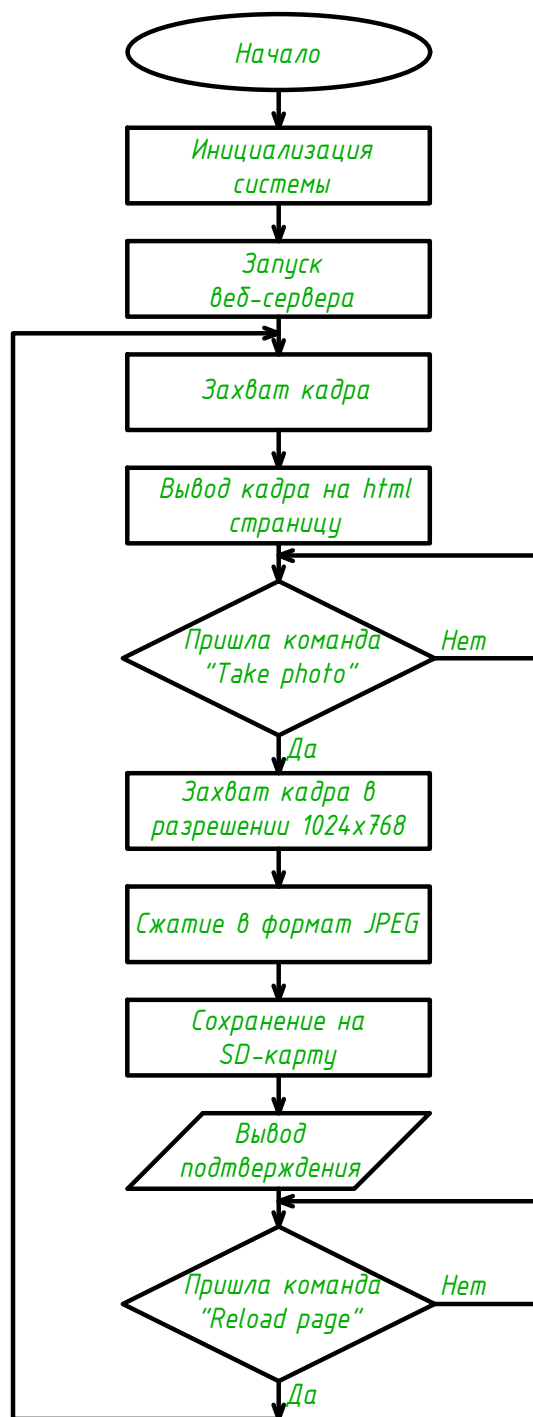
ГЧИР.431295.002. ЭЗ					Принципиальная схема подключения ESP32S			Лит.	Масса	Масштаб
Изм.	Лист	№ докум.	Подп.	Дата	Схема электрическая принципиальная			Т		
Разраб.	Кривальцевич							Лист	Листов	1
Пров.	Порхун							ФКСИС, гр.150701		

[illegible]

ПРИЛОЖЕНИЕ В
(Обязательное)
Схема алгоритма работы слояустройства



					ГЧИР.431295.004. ПД			
Изм.	Лист	№ докум.	Подп.	Дата	Алгоритм работы системы Схема алгоритма работы устройства	Лит.	Масса	Масштаб
Разраб.	Кривальцевич					Т		
Пров.	Порхун					Лист 1	Листов 2	
					ФКСИС, гр.150701			



					ГУИР.431295.005. ПД					
					Алгоритм работы программы Схема алгоритма работы программы	Лит.		Масса	Масштаб	
Изм	Лист	№ докум.	Подп.	Дата						
Разраб.	Кривальцевич					Т				
Пров.	Порхун									
						Лист 1		Листов 2		
						ФКСИС, гр.150701				

ПРИЛОЖЕНИЕ Г

(Обязательное)

С описание устройства

Модуль инициализации камеры:

```
#ifndef CAMERA_SETUP_H
#define CAMERA_SETUP_H

#include "esp_camera.h"
#include "FS.h"
#include "SD_MMC.h"

// Camera pin configuration
#define PWDN_GPIO_NUM    32
#define RESET_GPIO_NUM  -1
#define XCLK_GPIO_NUM    0
#define SIOD_GPIO_NUM    26
#define SIOC_GPIO_NUM    27
#define Y9_GPIO_NUM      35
#define Y8_GPIO_NUM      34
#define Y7_GPIO_NUM      39
#define Y6_GPIO_NUM      36
#define Y5_GPIO_NUM      21
#define Y4_GPIO_NUM      19
#define Y3_GPIO_NUM      18
#define Y2_GPIO_NUM       5
#define VSYNC_GPIO_NUM   25
#define HREF_GPIO_NUM    23
#define PCLK_GPIO_NUM    22

// Initialize the camera with the specified configuration
void startCamera() {
    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d2 = Y4_GPIO_NUM;
    config.pin_d3 = Y5_GPIO_NUM;
    config.pin_d4 = Y6_GPIO_NUM;
    config.pin_d5 = Y7_GPIO_NUM;
    config.pin_d6 = Y8_GPIO_NUM;
    config.pin_d7 = Y9_GPIO_NUM;
    config.pin_xclk = XCLK_GPIO_NUM;
    config.pin_pclk = PCLK_GPIO_NUM;
    config.pin_vsync = VSYNC_GPIO_NUM;
    config.pin_href = HREF_GPIO_NUM;
    config.pin_sscb_sda = SIOD_GPIO_NUM;
    config.pin_sscb_scl = SIOC_GPIO_NUM;
    config.pin_pwdn = PWDN_GPIO_NUM;
```



```

    config.pin_reset = RESET_GPIO_NUM;
    config.xclk_freq_hz = 20000000;
    config.pixel_format = PIXFORMAT_JPEG;
    config.frame_size = FRAMESIZE_XGA;
    config.jpeg_quality = 12;
    config.fb_count = 2;

    // Camera initialization
    esp_err_t err = esp_camera_init(&config);
    if (err != ESP_OK) {
        Serial.printf("Camera_init_failed_with_error_0x%x",
            err);
        return;
    }
    Serial.printf("Camera_initialized\n");
}

#endif

```

Модуль выполнения основных функций:

```

#ifndef WEB_SERVER_H
#define WEB_SERVER_H

#include "esp_http_server.h"
#include "FS.h"
#include "SD_MMC.h"

extern String lastPicturePath;

// Function to read the HTML file from SD card
String loadHtmlFile(const char* path) {
    File file = SD_MMC.open(path);
    if (!file) {
        Serial.printf("Failed_to_open_file:_%s\n", path);
        return String();
    }

    String htmlContent;
    while (file.available()) {
        htmlContent += (char)file.read();
    }
    file.close();
    return htmlContent;
}

// handler for css
esp_err_t css_handler(httpd_req_t *req) {
    File file = SD_MMC.open("/styles.css");
    if (!file) {
        Serial.println("Failed_to_open_styles.css");
        httpd_resp_send_404(req);
    }
}

```

```

        return ESP_FAIL;
    }

    httpd_resp_set_type(req, "text/css");
    String cssContent;
    while (file.available()) {
        cssContent += (char)file.read();
    }
    file.close();

    httpd_resp_send(req, cssContent.c_str(), cssContent.
        length());
    return ESP_OK;
}

// Handler for the index page
esp_err_t index_handler(httpd_req_t *req) {
    String htmlPage = loadHtmlFile("/index.html");

    if (htmlPage.isEmpty()) {
        httpd_resp_send_500(req);
        return ESP_FAIL;
    }

    httpd_resp_set_type(req, "text/html");
    httpd_resp_send(req, htmlPage.c_str(), htmlPage.length())
        ;
    return ESP_OK;
}

// Handler for video streaming
esp_err_t stream_handler(httpd_req_t *req) {
    camera_fb_t * fb = NULL;
    esp_err_t res = ESP_OK;
    size_t _jpg_buf_len;
    uint8_t * _jpg_buf;
    char part_buf[64];

    static const char* _STREAM_CONTENT_TYPE =
        "multipart/x-mixed-replace;boundary=frame";
    static const char* _STREAM_BOUNDARY = "--frame";
    static const char* _STREAM_PART =
        "Content-Type: image/jpeg\r\nContent-Length: %u\r\n\r\n";

    res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
    if (res != ESP_OK) {
        return res;
    }
}

```

```

while (true) {
    fb = esp_camera_fb_get();
    if (!fb) {
        Serial.println("Camera capture failed");
        res = ESP_FAIL;
    } else {
        _jpg_buf_len = fb->len;
        _jpg_buf = fb->buf;

        res = httpd_resp_send_chunk(req, _STREAM_BOUNDARY
            ,
            strlen(_STREAM_BOUNDARY)
            );

        if (res == ESP_OK) {
            size_t hlen = snprintf(part_buf, 64,
                _STREAM_PART,
                _jpg_buf_len
            );

            res = httpd_resp_send_chunk(req, part_buf,
                hlen);
        }
        if (res == ESP_OK) {
            res = httpd_resp_send_chunk(req, (const char
                *)_jpg_buf,
                _jpg_buf
            );
        }
        if (res != ESP_OK) {
            break;
        }
        esp_camera_fb_return(fb);
    }
}

return res;
}

// Handler for capturing a photo
esp_err_t capture_handler(httpd_req_t *req) {
    camera_fb_t * fb = esp_camera_fb_get(); // Capture frame
    if (!fb) {
        Serial.println("Camera capture failed");
        httpd_resp_send_500(req);
        return ESP_FAIL;
    }

    // Save frame to SD card
    String path = "/photo" + String(millis()) + ".jpg";

```

```

File file = SD_MMC.open(path, FILE_WRITE);
if (!file) {
    Serial.println("Failed to open file in write mode");
    esp_camera_fb_return(fb);
    return ESP_FAIL;
}

file.write(fb->buf, fb->len); // Write JPEG buffer to SD
card
file.close();
Serial.printf("Photo saved to: %s\n", path.c_str());

lastPicturePath = path;

esp_camera_fb_return(fb);

// Reinitialize the camera after capture
esp_camera_deinit();
startCamera();

// Reload page
httpd_resp_set_type(req, "text/html");
String htmlPage = loadHtmlFile("/reload.html");
httpd_resp_send(req, htmlPage.c_str(), htmlPage.length())
    ;

return ESP_OK;
}

// Start the HTTP server
void startCameraServer() {
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();

    httpd_uri_t index_uri = {
        .uri          = "/",
        .method        = HTTP_GET,
        .handler        = index_handler,
        .user_ctx       = NULL
    };

    httpd_uri_t stream_uri = {
        .uri          = "/stream",
        .method        = HTTP_GET,
        .handler        = stream_handler,
        .user_ctx       = NULL
    };

    httpd_uri_t capture_uri = {
        .uri          = "/capture",
        .method        = HTTP_GET,

```

```

        .handler    = capture_handler,
        .user_ctx   = NULL
    };

    httpd_uri_t css_uri = {
        .uri         = "/styles.css",
        .method      = HTTP_GET,
        .handler     = css_handler,
        .user_ctx    = NULL
    };

    httpd_handle_t server = NULL;
    if (httpd_start(&server, &config) == ESP_OK) {
        httpd_register_uri_handler(server, &index_uri);
        httpd_register_uri_handler(server, &stream_uri);
        httpd_register_uri_handler(server, &capture_uri);
        httpd_register_uri_handler(server, &css_uri);
    }

}

#endif

```

Модуль сборки всего проекта:

```

#include <Arduino.h>
#include "WiFi.h"
#include "camera_setup.h"
#include "web_server.h"

String lastPicturePath = "";

const char* ssid = "ESP32-CAM-Network";
const char* password = "123456789";

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

    // Initialize the camera
    startCamera();
    Serial.println("Camera initialized");

    // Connect to Wi-Fi
    WiFi.softAP(ssid, password);
    Serial.println("WiFi AP started");

    // Initialize SD card
    if (!SD_MMC.begin()) {
        Serial.println("SD Card Mount Failed");
        return;
    }
}

```

```
// Check if the index.html file exists on the SD card
if (!SD_MMC.exists("/index.html")) {
    Serial.println("index.html_file_not_found_on_SD_card!");
    return;
}

// Start the web server
startCameraServer();
Serial.println("Camera_server_started");
}

void loop() {
}
```

ПРИЛОЖЕНИЕ Д (Обязательное) HTML и CSS описание страниц

HTML основной страницы:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <link rel="stylesheet" href="styles.css">
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
    initial-scale=1.0">
  <title>ESP32-CAM Live Stream</title>
</head>
<body>
  <h1>ESP32 Camera</h1>

  <h2>Live Stream</h2>
  
  <br/>
  <button onclick="location.href='/capture'">Take Photo</
    button>
</body>
</html>
```

HTML страницы обновления:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <link rel="stylesheet" href="styles.css">
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
    initial-scale=1.0">
  <title>Photo Captured</title>
</head>
<body>
  <h1>Photo Captured!</h1>
  <p>Your photo has been saved successfully.</p>
  <button onclick="location.href='/'">Go Back to Main Page<
    /button>
</body>
</html>
```

CSS описание страницы:

```
body {
  display: flex;
  flex-direction: column;
  align-items: center;
  font-family: Arial, sans-serif;
```

```
        background-color: #f0f0f5;
        margin: 0;
        padding: 20px;
    }

    h1 {
        color: #333;
        margin-bottom: 20px;
    }

    img {
        border: 5px solid #555;
        border-radius: 10px;
        margin-bottom: 20px;
    }

    button {
        padding: 10px 20px;
        font-size: 16px;
        color: #fff;
        background-color: #007bff;
        border: none;
        border-radius: 5px;
        cursor: pointer;
        transition: background-color 0.3s ease;
    }

    button:hover {
        background-color: #0056b3;
    }
```


Обозначение					Наименование	Дополнительные сведения		
					<u>Текстовые документы</u>			
БГУИР КР 1-40 02 02 012 ПЗ					Пояснительная записка	36 с.		
					<u>Графические документы</u>			
ГУИР 431295.001 Э1					Схема электрическая структурная	Формат А4		
ГУИР 431295.002 Э3					Схема электрическая принципиальная	Формат А3		
ГУИР 431295.003 ПД					Схема алгоритма работы устройства	Формат А4		
ГУИР 431295.004 ПД					Схема алгоритма работы программы	Формат А4		