

Содержание

Введение	3
1 Техническое задание проекта.....	5
1.1 Постановка задачи и исходные данные.....	5
1.2 Используемые сокращения и условные обозначения.....	5
1.3 Описание этапов проектирования	5
1.4 Требования к устройству	7
1.4.1 Требования к функциональным характеристикам.....	7
1.4.2 Требования к надежности.....	8
1.4.3 Требования к составу и параметрам технических средств	8
1.4.4 Требования к информационной и программной совместимости	9
2 Технический проект	10
2.1 Общие сведения.....	10
2.2 Используемые обозначения и сокращения.....	10
2.3 Функциональное назначение.....	11
2.4 Описание логической структуры.....	12
2.5 Используемые технические и программные средства	13
2.6 Вызов и загрузка.....	17
2.7 Входные данные	18
2.8 Выходные данные.....	20
3 Руководство пользователя	22
3.1 Общие сведения об устройстве.....	22
3.2 Технические характеристики устройства	23
3.3 Настройка программы.....	25
3.4 Инструкция по эксплуатации	28
Заключение.....	30
Список использованных источников	32
Приложения.....	33

					<i>КП.0714.0500.006.40.00.00 ПЗ</i>		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.					<i>Содержание</i>		
Провер.							
Н. Контр.							
Утв.							
						<i>Лист</i>	<i>Лист</i>
						<i>Листов</i>	
						2	1
						<i>ККАТ зр.4ВМ-12</i>	

Введение

Современный мир характеризуется стремительным развитием технологий, которые коренным образом трансформируют различные аспекты человеческой деятельности. Технологический прогресс проникает во все сферы жизни, от бытовых устройств до сложных промышленных систем, упрощая выполнение рутинных задач, повышая точность производственных процессов и улучшая качество жизни. Важнейшей составляющей этого прогресса является создание автоматизированных систем управления, способных работать автономно или с минимальным вмешательством человека, что делает их незаменимыми в условиях возрастающей сложности современных задач.

Одним из ключевых направлений таких разработок стало сочетание интеллектуальных алгоритмов, высокотехнологичной электроники и мощного программного обеспечения. Автоматизированные системы управления позволяют достигать высокой степени точности, оперативности и надежности при выполнении самых разнообразных задач. Особенно актуальным становится использование лазерных технологий, которые нашли широкое применение благодаря своей универсальности и эффективности. Лазерные системы используются для наведения, целеуказания, измерений, визуализации данных и выполнения сложных операций с высокой степенью детализации.

Автоматическое наведение, основанное на обработке изображений, анализе данных и взаимодействии с сенсорными устройствами, становится неотъемлемой частью современных решений. Такие системы успешно применяются в оборонной промышленности, научных исследованиях, медицине, робототехнике, космических миссиях, а также в образовательных и демонстрационных целях. Благодаря способности к высокой точности и быстродействию, эти технологии находят применение в задачах автоматического сопровождения целей, управлении дронами, построении систем слежения и обеспечения безопасности.

					<i>КП.07140500.006.40.00.00 ПЗ</i>			
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	<i>Введение</i>	<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
<i>Разраб.</i>								
<i>Провер.</i>							3	2
<i>Н. Контр.</i>								
<i>Утв.</i>								
						<i>ККАТ гр.4ВМ-12</i>		

Кроме того, развитие лазерных технологий открывает новые горизонты в обучении и исследовании. Автоматизированные системы наведения представляют собой идеальный инструмент для студентов и исследователей, предоставляя возможность изучать и применять передовые методы обработки данных, компьютерного зрения и управления движением. Такие устройства являются отличной платформой для изучения искусственного интеллекта, алгоритмов машинного обучения и автоматизации, что делает их незаменимыми в современном образовательном процессе.

Проектирование и разработка систем, основанных на автоматическом наведении и лазерных технологиях, отражают актуальные запросы времени. Они не только позволяют создавать инновационные решения, но и служат двигателем дальнейшего прогресса, способствуя интеграции автоматизированных процессов в различные области. Это направление открывает широкие перспективы, предлагая эффективные и практичные решения для задач высокой сложности и значимости.

					<i>КП.07140500.006.40.00.00 ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>4</i>

1 Техническое задание проекта

1.1 Постановка задачи и исходные данные

Система автоматического наведения направлена на создание и реализацию установки, способной автоматически определять и точно следовать за объектом подсвеченным лазерным указателем. Все компоненты и модули системы должны работать синхронно, образуя единую экосистему, где данные передаются и обрабатываются в реальном времени. Основной задачей проекта должно являться разработка высокоточной и надежной системы, способной обрабатывать визуальную информацию, определять координаты цели и автоматически управлять движением механизма по двум осям вертикальной и горизонтальной. Реализация проекта предусматривает использование современных микроконтроллеров для анализа изображения, передачи данных и управления сервоприводами, обеспечивая тем самым эффективное выполнение работы системы.

1.2 Используемые сокращения и условные обозначения

В данном техническом проекте предоставлены следующие сокращения в таблице 1.1.

Таблица 1.1 Используемые сокращения

Сокращения	Описание
LAN	Local Area Network
UART	Universal Asynchronous Receiver–Transmitter
I2C	Inter–Integrated Circuit

1.3 Описание этапов проектирования

Проектирование устройства, должно осуществляется в несколько этапов:

- На первом этапе должен проводится анализ требований к устройству и

					<i>КП.07140500.006.40.00.01 ПЗ</i>		
Изм.	Лист	№ докум.	Подпись	Дата	<i>Техническое задание проекта</i>		
Разраб.							
Провер.							
Н. Контр.							
Утв.							
						Лист	Листов
						5	5
					<i>ККАТ зр.4ВМ-12</i>		

разработка общей концепции системы. Определяются ключевые функции устройства: обнаружение лазерной точки, обработка изображения, передача данных и управление сервоприводами. Составляется предварительный перечень необходимых модулей и их взаимодействие. Изучаются возможные решения и обосновывается выбор оборудования;

– На втором этапе выполняется выбор всех необходимых компонентов системы. Подбираются камера для захвата изображения, микроконтроллер для обработки данных, сервоприводы для управления конструкцией. Должно быть особое внимание на совместимость компонентов и их размещению, чтобы обеспечить стабильную работу системы и минимизацию помех;

– Третий этап включает проектирование схемы подключения всех компонентов. Разрабатывается схема подключения камеры, микроконтроллера и сервоприводов. Компоненты размещаются таким образом, чтобы обеспечить их эффективное взаимодействие. На этом этапе проводится монтаж устройства и подключение всех элементов в единую систему;

– Четвертый этап включает в себя разработку программного обеспечения для управления системой, где разрабатываются сложные алгоритмы для анализа изображения, обработки координат и управления сервоприводами. В этот этап интегрируется алгоритм передачи данных и Wi-Fi, что позволяет системе эффективно взаимодействовать с внешними компонентами. Особое внимание уделяется корректности обработки данных, чтобы система могла точно и своевременно реагировать на изменения, а также точности выполнения задач;

– На пятом этапе проводится тестирование и отладка системы. Проверяется работа камеры, алгоритмов обнаружения и обработки лазерной точки, передача данных, а также управление сервоприводами. Тестирование проводится в условиях, приближенных к реальным, чтобы оценить эффективность работы системы.

– Шестой этап включает финальную сборку устройства и тестирование в реальных условиях эксплуатации. Устройство проверяется на соответствие функциональным требованиям, проводится настройка алгоритмов для достижения максимальной точности и стабильности.

					<i>КП.07140500.006.40.00.01 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

1.4 Требования к устройству

1.4.1 Требования к функциональным характеристикам

Устройство должно быть спроектировано для эффективного и точного выполнения основных задач, связанных с автоматическим наведением на объект, используя лазер. Основной функционал включает в себя – определение положения лазерной точки, анализ изображения с помощью камеры и управление конструкцией для точного слежения за целью. Все действия должны выполняться с высокой скоростью и минимальными задержками, что обеспечит надежность работы системы.

Одним из главных требований является распознавание красной области на изображении, что достигается с помощью специализированных алгоритмов обработки. Эти алгоритмы должны быть устойчивы к шумам, изменениям освещения и другим внешним факторам, чтобы устройство могло эффективно идентифицировать лазерную точку даже в сложных условиях. Точность распознавания напрямую влияет на эффективность всего устройства.

При получении координат лазерной точки система должна обрабатывать данные и управлять сервоприводами, регулируя их движение по вертикальной и горизонтальной оси. Управляющий микроконтроллер выполняет расчёты углов поворота для обеспечения точного позиционирования конструкции. Управление должно быть плавным и стабильным, чтобы исключить резкие движения и рывки, которые могут ухудшить работу системы или повлиять на износ компонентов.

Также важным аспектом является гладкость и стабильность движения конструкции. Все механизмы должны быть тщательно сбалансированы, чтобы минимизировать вибрации и обеспечить плавное и точное следование за лазером. Это не только повысит точность работы устройства, но и продлит его срок службы.

Кроме того, устройство должно быть универсальным, что позволит использовать его в различных проектах и задачах. Конструкция и программное обеспечение должны быть модульными, чтобы их легко можно было доработать, а именно, для работы с другими цветами или добавления новых функций, что обеспечит гибкость и долгосрочную применимость устройства.

					<i>КП.07140500.006.40.00.01 ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>7</i>

1.4.2 Требования к надежности

Устройство должно соответствовать следующим требованиям надежности:

- Устройство должно быть устойчиво к кратковременным сбоям в питании, помехам в беспроводной связи и ошибкам в алгоритмах обработки данных. Система должна обеспечивать стабильную работу даже в условиях кратковременных сбоев;
- Для создания устройства должны применяться качественные и проверенные компоненты, обеспечивающие стабильность работы системы и минимизирующие вероятность отказов;
- Система должна включать механизмы проверки корректности данных, поступающих от всех входных модулей и отправляемых к исполнительным элементам, чтобы исключить ошибки на этапе обработки и передачи информации;
- Все основные операции, такие как обнаружение цели, расчет координат и управление исполнительными модулями, должны корректно выполняться, чтобы исключить необходимость постоянного вмешательства специалиста и снизить риск ошибок.

1.4.3 Требования к составу и параметрам технических средств

Устройство должно состоять из следующих компонентов:

- Модуль камеры с поддержкой беспроводной связи: устройство должно включать компонент для захвата изображений, распознавания объектов по цвету, и передачи обработанных данных на управляющий модуль через беспроводное и проводное соединение;
- Устройство должно содержать микроконтроллер, который получает данные от камеры, выполняя необходимые вычисления для управления устройством, и передача управляющих сигналов исполнительным механизмам;
- Для точного перемещения камеры по горизонтальной и вертикальной оси, должны быть два сервопривода с хорошими показателями точности и скорости. Они должны быть способны принимать сигналы от управляющего микроконтроллера и корректно выполнять движения;
- Источник питания должен быть предусмотрен стабильный и надежный, который соответствует потреблению компонентов устройства. Он должен

					<i>КП.07140500.006.40.00.01 ПЗ</i>	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

обеспечивать подачу необходимого напряжения и тока для обеих сервоприводов.

1.4.4 Требования к информационной и программной совместимости

Для обеспечения корректной работы устройства необходимо соответствовать следующим требованиям:

– Разработка программного обеспечения должна осуществляться с использованием современных инструментов разработки и отладки, совместимых с выбранными микроконтроллерами. Поддержка языков программирования, таких как С или С++. Среда разработки должна обеспечивать удобство написания, тестирования и загрузки программ на устройство;

Передача данных между модулями должна выполняться в формате, обеспечивающем минимизацию ошибок и удобство обработки. Протоколы передачи данных должны быть выбраны с учетом стабильности и надежности работы с прямым подключением, например UART или I2C, а также должен быть веб-интерфейс с LAN, без необходимости подключения к внешней сети. Установленные драйверы для корректного взаимодействия с оборудованием. Устройства должны поддерживать беспроводное взаимодействие, обеспечивая стабильный обмен данными. Программное обеспечение должно быть адаптировано под выбранные компоненты и обеспечивать их корректное функционирование в рамках системы. Должны быть разработаны алгоритмы обработки изображений, вычисления координат и управления исполнительными механизмами.

					<i>КП.07140500.006.40.00.01 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

2 Технический проект

2.1 Общие сведения

Проект представляет собой небольшую конструкцию, оснащенную умной системой, предназначенной для автоматического обнаружения и наведения на объект, помеченным лазерной точкой. Основу системы составляют камера, способная фиксировать изображение и вычислять координаты объекта, и механизм управления, обеспечивающий движение платформы по двум осям – горизонтальной и вертикальной, для точного наведения на объект.

Система автоматического наведения разработано с целью изучения и демонстрации технологий визуального распознавания объектов и автоматизированного управления. Особенно будет полезно в исследовательских и образовательных проектах, позволяя на практике изучать принципы работы современных систем компьютерного зрения.

Кроме того, устройство обладает высоким потенциалом для адаптации в различных сферах. В частности, система может быть использована как основа для разработки автоматических турелей, систем слежения или противовоздушной обороны. Проект может быть масштабирован и добавлен в более сложные системы, обеспечивая возможность применения в реальных технических и инженерных задачах.

Система также отличается и экономичностью, что делает её доступной для реализации в условиях ограниченного бюджета. Простота конструкции и использование распространённых компонентов позволяют облегчить процесс сборки и наладки.

2.2 Используемые обозначения и сокращения

В данном техническом проекте применяются следующие сокращения, предоставленные в таблицы 2.1.

					<i>КП.07140500.006.40.00.02 ПЗ</i>		
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>			
<i>Разраб.</i>					<i>Технический проект</i>		
<i>Провер.</i>							
<i>Н. Контр.</i>							
<i>Утв.</i>							
					<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
						10	12
					<i>ККАТ зр.4ВМ-12</i>		

Таблица 2.1 Описание сокращений

Сокращение	Значение
ШИМ	Широтно Импульсная Модуляция
IoT	Internet of Things
IDE	Integrated Development Environment
SRAM	Static Random–Access Memory
GPIO	General Purpose Input/Output
DAC	Digital–to–Analog Converter
PWM	Pulse Width Modulation
ADC	Analog–to–Digital Converter
I2C	Inter–Integrated Circuit
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver–Transmitter
SCCB	Serial Camera Control Bus
CMOS	Complementary Metal–Oxide Semiconductor
DC	Direct Current
VGA	Video Graphics Array
USB	Universal Serial Bus

2.3 Функциональное назначение

Устройство выполняет функции автоматического наведения на объект, обозначенный лазерной точкой. Камера, будет делать снимки каждую секунду. Эти снимки будут анализироваться алгоритмом обработки изображения, который будет выделять красный цвет на изображении.

На основании обработки изображения вычисляются координаты обнаруженной лазерной точки. Для визуализации результатов алгоритм будет обводить найденную точку на изображении, для упрощения проверки работы.

Устройство обеспечивает передачу полученных данных с камеры на управляющий микроконтроллер через Wi-Fi и UART соединение. Управляющий модуль будет выполнять роль точки доступа, предоставляя возможность прямого соединения с камерой и обработки координат.

					<i>КП.07140500.006.40.00.02 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

Микроконтроллер принимает координаты лазерной точки и выполнять необходимые вычисления для определения углов поворота по двум осям. Управляя двумя сервоприводами, система изменяет своё положение таким образом, чтобы точно нацелиться на обнаруженную точку. Для этого используются алгоритмы обработки данных, обеспечивающие плавное и точное наведение. Этот процесс происходит автоматически, обеспечивая динамическое слежение за объектом, даже если его положение меняется, что делает систему устойчивой к резким изменениям координат и позволяет эффективно работать в режиме реального времени.

2.4 Описание логической структуры

Логическая структура устройства основана на взаимодействии двух ключевых модулей, отражающей взаимодействия компонентов системы, представлена в Приложении 2. Которые совместно выполняют задачи обнаружения цели и автоматического наведения.

Логика работы устройства:

- Камера устройства делает фотографию каждую секунду. Полученное изображение анализируется встроенным алгоритмом, который обрабатывает каждый пиксель на соответствие заданному диапазону красного цвета. В случае обнаружения красной точки определяется её центр, который вычисляется относительно размеров изображения (1024x768). Для визуализации алгоритм выделяет область белым кругом, содержащую красную точку. Координаты центра красной точки отправляются на управляющий микроконтроллер через Wi-Fi и UART соединение. Этот процесс позволяет эффективно отслеживать и локализовать красные объекты в реальном времени;

- Управляющий микроконтроллер создает локальную беспроводную сеть, через которую он принимает координаты от камеры, обрабатывает их и преобразует в углы поворота для двух сервоприводов, отвечающих за горизонтальную и вертикальную ось. Эти углы рассчитываются на основе пропорции координат изображения относительно диапазона поворота сервоприводов и углов обзора камеры, как показано в формуле (2.1). Такой подход обеспечивает точное и плавное

					<i>КП.07140500.006.40.00.02 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

наведение устройства на заданную точку;

$$a_{ServoX} = a_{ServoX} + \Delta a_x * \left(\frac{180}{a_x}\right) \quad (2.1)$$

где

a_{ServoX} – основная формула для расчета угла сервопривода;

Δa_x – рассчитанные изменения угла;

a_x – угол обзора камеры по горизонтальной (вертикальной) оси.

– Рассчитанные значения углов передаются на сервоприводы. Первый сервопривод управляет поворотом конструкции по горизонтальной оси, а второй, установленный на первом, регулирует движение по вертикальной оси. При этом, последние полученные координаты сохраняются, если не были получены новые, тем самым углы сервоприводом не меняются.

Процесс обработки изображения, передачи координат и управления сервоприводами повторяется циклично, каждую секунду.

2.5 Используемые технические и программные средства

В устройстве используются следующие технические средства:

– ESP32–CAM: это плата с интегрированным модулем ESP32, который включает в себя встроенный чип Wi-Fi и Bluetooth обеспечивая возможность беспроводной связи. Плата оснащена камерой OV2640, предназначенной для захвата изображений, она отличается своим качеством и компактностью. ESP32–CAM (см. рисунок 2.1), используется для обработки изображений и видео с возможностью их передачи через сеть Wi-Fi, технические характеристики представлены в таблице 2.2;



Рисунок 2.1 – ESP32–CAM с камерой OV2640

					КП.07140500.006.40.00.02 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		13

Таблица 2.2 Техническая характеристика микроконтроллера ESP32–CAM

Характеристика	Описание
Процессор	ESP32 (двухъядерный процессор, 32 бита)
Частота процессора	До 240 МГц
Память	520 КБ SRAM, до 16 МБ Flash
Wi-Fi	802.11 b/g/n
Bluetooth	Bluetooth v4.2
Разъемы	9 GPIO, UART, SPI, I2C, PWM, ADC, DAC
Камера	OV2640 (2 МП)
Рабочая температура	От –40°С до +85°С
Питание	5 В, 250 мА

– Камера OV2640: это CMOS–камера с разрешением 2 МП, которая используется в различных устройствах для захвата изображений. Эта камера поддерживает различные разрешения, включая VGA и более высокие разрешения до 1600x1200 пикселей. Камера обычно используется в устройствах, таких как ESP32–CAM (см. Рисунок 2.1), для захвата изображений и видео. технические характеристики представлены в таблице 2.3;

Таблица 2.3 Техническая характеристика камеры OV2640

Характеристика	Описание
Тип сенсора	CMOS
Разрешение	2 МП (1600x1200 пикселей)
Видеоформаты	YUV422, JPEG, RGB565
Скорость кадрирования	До 30 кадров в секунду
Питание	3.3 В – 5 В
Размеры	27.3 мм x 27.3 мм
Интерфейс	SCCB
Рабочая температура	От –20°С до +70°С

– NodeMCU v3 Lua: это компактный и мощный Wi-Fi модуль, используемый в проектах IoT для организации беспроводной связи. Он

					КП.07140500.006.40.00.02 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

поддерживает подключение к сети Wi-Fi и предоставляет возможности для обработки данных, управления устройствами и взаимодействия с внешними модулями. Модуль ESP8266 (см. рисунок 2.2) может использоваться как самостоятельный микроконтроллер или как модем Wi-Fi в составе более сложных систем. технические характеристики представлены в таблице 2.4;



Рисунок 2.2 – NodeMCU v3 Lua

Таблица 2.4 Техническая характеристика микроконтроллера NodeMcu

Характеристика	Описание
Процессор	Tensilica L106 (32–разрядный)
Частота процессора	80–160 МГц
Память	64 КБ SRAM, до 4 МБ Flash
Wi-Fi	802.11 b/g/n
Поддерживаемые режимы	Клиент, точка доступа
Порты	GPIO, UART, SPI, I2C, PWM
Напряжение питания	3.3 В – 5 В
Потребление энергии	до 250 мА
Рабочая температура	От –40°С до +125°С
Размеры	24 мм x 16 мм

– Сервопривод SG90: это микро–сервопривод, который широко используется в робототехнике, системах автоматизации и других инженерных проектах (см. рисунок 2.3). Его конструкция включает металлические шестерни, обеспечивающие долговечность и высокую точность позиционирования, что делает его оптимальным решением для управления движением различных механизмов. Благодаря низкому энергопотреблению и возможности работы от стандартного

					<i>КП.07140500.006.40.00.02 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		15

напряжения 5 В, он легко интегрируется в проекты с микроконтроллерами, такими как ESP32 или Arduino. Технические характеристики представлены в таблице 2.5;



Рисунок 2.3 – Сервопривод SG90

Таблица 2.5 Техническая характеристика сервопривода SG90

Характеристика	Описание
Тип привода	Аналоговый сервопривод
Угол вращения	0–180°
Питание	4.8–6 В
Ток потребления (покой)	10 мА
Ток потребления (рабочий)	До 700 мА
Крутящий момент	2.2 кг·см
Скорость вращения	0.1 с/60°
Масса	13.4 г
Материал шестерен	Пластик (Нейлон)
Рабочая температура	От –30°С до +60°С
Размеры	22.8 x 12.2 x 28.5 мм

– Блок питания: это ключевой элемент системы, обеспечивающий стабильное питание всех компонентов проекта, включая контроллеры ESP32–CAM, NodeMCU, сервоприводы и другие исполнительные устройства. Основная функция блока питания заключается в преобразовании переменного напряжения сети в постоянное, которое требуется для корректной работы электронных компонентов. Блок питания (см. рисунок 2.4) играет важную роль в защите системы от перегрузок. Техническая характеристика предоставлена в таблице 2.6.



Рисунок 2.4 – Блок питания

Таблица 2.6 Техническая характеристика блока питания

Характеристика	Описание
Тип	Адаптер питания
Входное напряжение	0–100–240 В AC, 50/60 Гц°
Выходное напряжение	5 В DC
Максимальный выходной ток	2 А
Мощность	10 Вт
Разъем	USB или DC–коннектор
Защита	От короткого замыкания и перегрузки

2.6 Вызов и загрузка

При подаче питания устройство запускается, и микроконтроллеры ESP32–CAM и NodeMCU v3 Lua активируют встроенные программные обеспечения, подготавливая систему к выполнению основных функций.

После инициализации выполняется проверка всех периферийных устройств, включая сервоприводы, камеру и другие компоненты, чтобы убедиться в их корректной работе.

ESP32–CAM устанавливает соединение к точке доступа NodeMCU, используя встроенные модули и соответствующие библиотеки. Это необходимо для передачи данных между устройствами.

Для работы ESP32–CAM и камеры OV2640 используются следующие библиотеки:

					КП.07140500.006.40.00.02 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		17

- HTTPClient.h – позволяет осуществлять управление HTTP–запросами, передавая координаты и данные на NodeMCU;
- esp_camera.h – отвечает за настройку параметров камеры (разрешение, экспозиция) и управление съемкой;
- camera_pins.h – определяет пины, необходимые для подключения камеры к микроконтроллеру;
- WiFi.h – Обеспечивает подключение к локальной сети. Поддерживает как режим клиента.

Для работы NodeMCU v3 Lua используются следующие библиотеки:

- ESP8266WiFi.h – Используется для подключения NodeMCU к Wi-Fi сети, а именно создание точки доступа;
- ESPAsyncWebServer.h – для создания асинхронного веб-сервера, позволяющего обрабатывать запросы и управлять устройством через веб-интерфейс;
- ArduinoJson.h – для работы с полученными данными в формате JSON;
- Servo.h – для управления сервоприводами SG90, используемыми для позиционирования по горизонтальной и вертикальной оси (X и Y).

Устройство обрабатывает данные, собранные с камеры, передает их по сети Wi-Fi, а также управляет сервоприводами.

Данные передаются в формате JSON через HTTP–запросы, что обеспечивает удобное управление системой через веб-интерфейс. Веб-страница, интегрированная с устройством, позволяет получать актуальную информацию о положении сервоприводах.

2.7 Входные данные

Система автонаведения использует управляющий микроконтроллер для получения данных с камеры, их обработки и управления сервоприводами, обеспечивая позиционирование на цель. Для получения данных с камеры, их анализа и последующей передачи на управляющий микроконтроллер необходимо соединить модуль OV2640 с микроконтроллером, создавая единую конструкцию, которая функционирует как интегрированное устройство (см. рисунок 2.4).

					<i>КП.07140500.006.40.00.02 ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>18</i>



Рисунок 2.4 – ESP32–CAM и камера OV2640

Камеру необходимо подключить к питанию, а именно контакты 5V и GND (см. рисунок 2.5). И подключить ESP32–CAM к локальной сети управляющего микроконтроллера. Для этого используется следующая строка кода:

```
#include <WiFi.h>
```

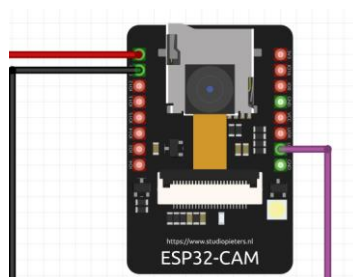


Рисунок 2.5 – Подключение ESP32–CAM к питанию

Потом задаются параметры подключения, такие как название сети и пароль. Используются следующие строки кода:

```
const char *ssid = "SystemAutoDetected";  
const char *password = "ESP8266ESP32CAM";
```

После выполнения алгоритма поиска красной точки вычисленные координаты передаются на управляющий микроконтроллер через локальную сеть. Используется следующая строка кода:

```
#include <HTTPClient.h>
```

Для подключения управляющего микроконтроллера к системе необходимо выполнить следующие действия. Сначала нужно подключить микроконтроллер к питанию, используя пины Vin и GND. Далее подсоединить два сервопривода, используя пины D5 и D6 для управления, и подключить им питание (см. рисунок 2.6).

					КП.07140500.006.40.00.02 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		19

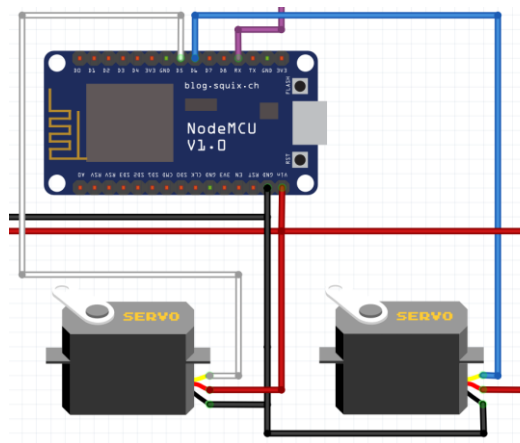


Рисунок 2.6 – Подключение NodeMCU

Для работы сервоприводов требуется подключить соответствующую библиотеку. Для этого нужно добавить следующую строку кода:

```
#include <Servo.h>
```

После подключения библиотеки нужно инициализировать сервоприводы, используя следующий код:

```
servoX.attach(SERVO_X_PIN);  
servoY.attach(SERVO_Y_PIN);
```

После инициализации нужно задать им начальное положение. Для этого используется следующие строки кода:

```
servoX.write(0);  
servoY.write(0);
```

2.8 Выходные данные

Система автонаведения предоставляет два типа данных на веб-интерфейс: обработанное изображение, выводимое на камере, и рассчитанные углы для управления сервоприводами, выводимые на управляющем микроконтроллере. Эти данные представлены как в визуальной, так и в числовой форме.

В программной части камеры первым шагом осуществляется инициализация беспроводной сети с использованием библиотеки:

```
#include <WiFi.h>
```

Далее камера отправляет обработанное изображение в формате JPEG на свой веб-интерфейс. Для этого используются следующие строки кода:

```
client.println("HTTP/1.1 200 OK");
```

					КП.07140500.006.40.00.02 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		20

```

client.println("Content-Type: image/jpeg");
client.println("Content-Length: " + String(jpeg_len));
client.println("Connection: close");
client.println();
client.write(jpeg_buf, jpeg_len);
free(jpeg_buf);

```

После этого камера отправляет координаты, полученные в процессе обработки изображения, на управляющий микроконтроллер через HTTP–запрос. Данные передаются в формате JSON. Для реализации используется следующий код:

```

HTTPClient http;
http.begin(serverUrl);
http.addHeader("Content-Type", "application/json");
String payload = "{\"x\":" + String(x) + ", \"y\":" + String(y) +
"}";
http.end();

```

Управляющий микроконтроллер выводит в числовом формате актуальные углы на своем веб–интерфейсе.

					<i>КП.07140500.006.40.00.02 ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>21</i>

3 Руководство пользователя

3.1 Общие сведения об устройстве

Система автонаведения представляет собой устройство (см. рисунок 3.1), созданное для автоматизированного наведения на объекты, помеченные лазерным целеуказателем. Это компактное и высокотехнологичное решение объединяет возможности современных микроконтроллеров и компонентов для достижения высокой точности и эффективности в работе.



Рисунок 3.1 – Система автонаведения

Устройство оснащено камерой, которая выполняет функцию захвата и анализа изображения. Камера фиксирует текущую обстановку, обрабатывает изображение и обнаруживает лазерную метку. После этого координаты обнаруженной точки передаются на управляющий микроконтроллер, который выполняет сложные вычисления для точного управления сервоприводами. Эти сервоприводы обеспечивают плавное и точное движение, позволяя системе эффективно наводиться на цель по горизонтальной и вертикальной осям.

Основное назначение системы автонаведения заключается в автоматизации процессов наведения с применением визуального анализа. Устройство может применяться в образовательной среде для демонстрации принципов работы

					<i>КП.07140500.006.40.00.03 ПЗ</i>			
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>				
<i>Разраб.</i>					<i>Лит.</i>		<i>Лист</i>	<i>Листов</i>
<i>Провер.</i>							22	8
<i>Н. Контр.</i>					<i>Руководство пользователя</i>			
<i>Утв.</i>								
					<i>ККАТ гр.4ВМ-12</i>			

современных технологий, в робототехнике, а также в задачах, где требуется автономное управление движением на основе визуальных данных.

Помимо основных функций, устройство также поддерживает возможность использования веб-интерфейса. Пользователь может наблюдать обработанное изображение, на котором четко выделена лазерная метка, а также отслеживать углы текущего положения сервоприводов.

Компактный дизайн, легкость в настройке и широкий спектр возможностей делают данное устройство универсальным инструментом. Устройство подходит для начинающих пользователей, которые хотят освоить работу с современными микроконтроллерами, а также для профессионалов, использующих такие решения в своих проектах.

Эlegantное сочетание технологий и простоты использования превращает устройство в идеальный выбор, кто ценит точность, надежность и удобство работы.

3.2 Технические характеристики устройства

Технические характеристики устройства системы автонаведения представлены в таблице 3.1

Таблица 3.1 Технические характеристики системы автонаведения

Характеристика	Описание
1	2
Модель камеры	OV2640
Разрешение	1024x768 пикселей
Обработка изображения	Выделение красного лазера белой обводкой
Модель микроконтроллера	ESP32
Связь	Wi-Fi, IEEE 802.11b/g/n
Формат передаваемых данных	UART (прямое соединение)
Модель сервоприводов	SG90
Диапазон углов	0° – 180°
Управление осями	Горизонтальная и вертикальная
Основное питание	Блок питания (5V)

Продолжение таблицы 3.1

1	2
Альтернативное питание	USB на микроконтроллере NodeMCU
Длина	10 см
Ширина	10 см
Высота	10 см
Вывод данных	Изображение и углы сервоприводов

Особенности работы устройства:

- Режим работы камеры: Камера ESP32–CAM работает с задержкой в 1 секунду, фиксируя изображения с высокой точностью. Алгоритм обрабатывает изображение для нахождения объектов, выделенной красной лазерной точки, добавляя белую обводку для визуализации цели;
- Модуль связи: благодаря NodeMCU является точкой доступа и поддерживает беспроводную передачу данных через Wi-Fi. Позволяет просматривать камеру, что удобно для задач, связанных с автоматизацией;
- Корректность работы и само восстанавливаемость: Устройство автоматически проверяет поступающие данные на корректность формата. В случае обнаружения ошибок модуль связи отправляет запрос на повторную передачу данных;
- Энергопотребление: Использование блока питания на 5 В обеспечивает автономность работы устройства. Дополнительно предусмотрено подключение через USB порт на NodeMCU, что расширяет варианты использования устройства;
- Контроль углов сервоприводов: Веб–интерфейс позволяет проверять углы сервоприводов на управляющем микроконтроллере, а также позволяет посмотреть актуальное обработанное изображение, полученное камерой. Это делает работу с устройством более комфортным.
- Компактные размеры: благодаря своим небольшим габаритам устройство легко сочетается с любыми другими устройствами, включая учебные стенды, робототехнику или автоматизированные системы.

					КП.0714.0500.006.40.00.03 ПЗ	Лист
						24
Изм.	Лист	№ докум.	Подпись	Дата		

3.3 Настройка программы

Для корректной работы устройства, необходимо выполнить настройку используя следующие шаги:

Шаг 1: Подключение питания

Подключите USB-кабель к блоку питания 5 В и убедитесь, что он надежно закреплен в разъеме. Затем включите блок питания в сеть и проверьте, что питание подается стабильно – это можно определить по индикаторным светодиодам на управляющем микроконтроллере, и посмотрев на состояние сервоприводов. Если индикаторы не загорелись и сервоприводы не изменили угол, проверьте целостность кабеля, плотность соединения и исправность блока питания. Убедитесь, что питание подается без перебоев, так как нестабильное напряжение может привести к сбоям в работе всех элементов устройства.

Шаг 2: Подключение к сети Wi-Fi

Для работы системы микроконтроллеры взаимодействуют друг с другом через Wi-Fi. В данном случае ESP32-CAM создаёт точку доступа с заданными параметрами, к которой подключается управляющий микроконтроллер NodeMCU

Система работает в режиме точки доступа с следующими параметрами:

- Название WiFi: SystemAutoDetected ;
- Пароль: ESP8266ESP32CAM.

Управляющий микроконтроллер использует статический IP-адрес 192.168.4.2, а ESP32-CAM имеет IP 192.168.4.1. Чтобы подключиться к Wi-Fi необходимо найти на компьютере или телефоне данную сеть – «SystemAutoDetected» (см. рисунок 3.2).



Рисунок 3.2 – Сеть Wi-Fi

					<i>КП.07140500.006.40.00.03 ПЗ</i>	Лист
						25
Изм.	Лист	№ докум.	Подпись	Дата		

После чего необходимо ввести пароль для подключения к локальной сети (см. Рисунок 3.3).

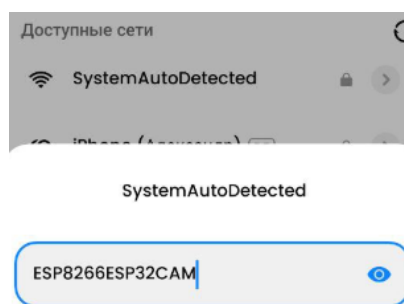


Рисунок 3.3 – Подключение к сети

Шаг 3: Проверка камеры и сервоприводов

После подключения к сети SystemAutoDetected можно проверить работу камеры и сервоприводов через веб-интерфейс:

– Проверка камеры. Откройте веб-браузер (Chrome, Firefox, Edge и др.). Введите IP-адрес 192.168.4.1 (камера) в адресную строку и нажмите Enter (см. рисунок 3.4). В веб-интерфейсе откроется изображение, передаваемое с камеры (см. рисунок 3.5). В любой момент можно обновить страницу, чтобы просмотреть новый снимок. При наличии лазерной метки она будет автоматически выделена белым контуром (см. рисунок 3.6);



Рисунок 3.4 – Ввод IP-адреса



Рисунок 3.5 – Просмотр изображения

					<i>КП.07140500.006.40.00.03 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		26



Рисунок 3.6 – Выделенная лазерная точка

– Проверка сервоприводов. В браузере перейдите на IP–адрес 192.168.4.2 – управляющий микроконтроллер (см. рисунок 3.7). Откроется веб–интерфейс с текущими углами сервоприводов по осям X и Y (см. рисунок 3.8). Если камера зафиксировала лазерную метку, сервоприводы должны будут плавно двигаться, направляя камеру с сервоприводами в сторону лазерной точки. Убедитесь, что значения углов в интерфейсе изменяются при перемещении лазера.

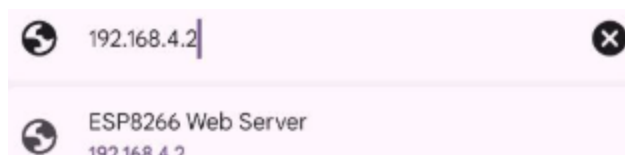


Рисунок 3.7 – Ввод IP–адреса управляющего микроконтроллера

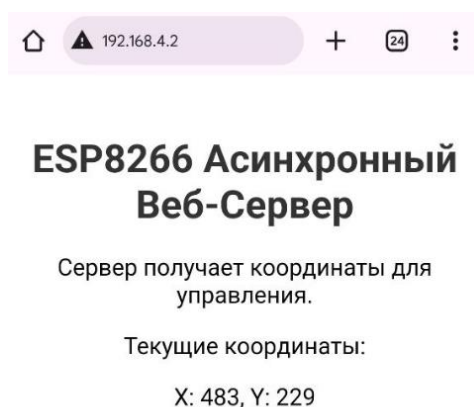


Рисунок 3.8 – Просмотр углов сервоприводов

					<i>КП.07140500.006.40.00.03 ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>27</i>

Шаг 5: Финальная проверка

Убедитесь, что устройство корректно реагирует на изменение координат лазерной точки.

Сервоприводы должны плавно изменять углы по осям X и Y, поворачивая камеру к подсвеченной точке.

Текущие углы сервоприводов отображаются в веб–интерфейсе, доступном по IP–адресу NodeMCU.

После выполнения этих шагов устройство полностью готово к работе.

3.4 Инструкция по эксплуатации

Принцип действия:

- Камера ESP32–CAM фиксирует изображение и анализирует его для обнаружения красной лазерной метки;
- Координаты лазерной метки передаются на NodeMCU прямым подключением и по беспроводной связи, где происходит вычисление углов поворота сервоприводов;
- Сервоприводы управляют движением камеры, точно направляя ее на лазерную метку;
- Контроль состояния и параметров осуществляется через веб–интерфейс.

Составные части устройства:

- модуль ESP32–CAM: фиксирует изображение и обрабатывает его в реальном времени каждую секунду;
- Модуль NodeMCU: принимает актуальные координаты лазерной точки от камеры, выполняет расчеты углов и отправляет сигналы на сервоприводы;
- Сервоприводы SG90: обеспечивают плавное движение камеры по горизонтальной и вертикальной осям;
- Элемент питания: Блок питания или альтернативный – USB кабель на NodeMCU;
- Корпус и крепления: обеспечивают устойчивость и защиту устройства.

Рекомендации:

					КП.0714.0500.006.40.00.03 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		28

- Установка: установите устройство на ровную и устойчивую поверхность, чтобы избежать сбоев в работе и влияния вибраций;
- Электропитание: убедитесь, что блок питания имеет 5 Вольт и 2–3 Ампер;
- Обслуживание камеры: регулярно очищайте объектив от пыли мягкой тканью без использования агрессивных чистящих средств;
- Контроль работы: используйте веб–интерфейс для наблюдения за обработанным изображением, координатами метки и текущими углами сервоприводов.

Запрещенные действия:

- Не погружайте устройство в воду и не подвергайте воздействию высоких температур;
- Не подключайте источник питания, превышающий указанные параметры, чтобы избежать перегрева и поломки.

Техническое обслуживание:

- Регулярная проверка соединений проводов;
- Осмотр корпуса на наличие механических повреждений.

При необходимости ремонта обратитесь к специалисту. Следующие признаки могут указывать на необходимость ремонта:

Отсутствие изображения с камеры в веб–интерфейсе;

Некорректное движение сервоприводов или их полная остановка;

Необходимость замены элементов питания и проводов.

Заключение

В ходе выполнения проекта была разработана система автоматического наведения, которая сочетает в себе технологии компьютерного зрения и автоматизированного управления. Основным функционал устройства заключается в распознавании лазерной точки на изображении и последующем наведении на неё с высокой точностью. Разработанная конструкция продемонстрировала устойчивую и стабильную работу, позволяя системе динамически отслеживать перемещение лазерной отметки в режиме реального времени.

Одним из ключевых преимуществ устройства является его универсальность и возможность адаптации к различным сценариям использования. Система способна выполнять автоматическое наведение и слежение за объектами, что делает её перспективной для применения в широком спектре задач. Например, она может быть использована в образовательных и исследовательских целях для изучения принципов компьютерного зрения и автоматизированного управления, а также в сфере безопасности – для создания интеллектуальных охранных систем, автоматизированных турелей или систем слежения за движущимися объектами.

Техническая реализация проекта основана на взаимодействии камеры и управляющего микроконтроллера, которые обеспечивают сбор, обработку и передачу данных. Камера фиксирует изображение и анализирует его, выделяя лазерную точку, после чего передает координаты обнаруженного объекта на управляющий модуль. Контроллер, в свою очередь, рассчитывает необходимые углы поворота платформы и управляет сервоприводами, обеспечивая точное и плавное наведение. Использование беспроводного соединения и статических IP-адресов позволяет не только эффективно передавать данные, но и удаленно отслеживать активные параметры системы в реальном времени.

Система обладает компактными размерами, простотой конструкции и высокой надежностью соединения между компонентами. Алгоритмы обработки

					<i>КП.07140500.006.40.00.00 ПЗ</i>		
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>			
<i>Разраб.</i>					<i>Заключение</i>		
<i>Провер.</i>							
<i>Н. Контр.</i>							
<i>Утв.</i>							
					<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
						30	2
					<i>ККАТ зр.4ВМ-12</i>		

изображений обеспечивают быструю и точную идентификацию лазерной точки, а использование сервоприводов с оптимизированными углами поворота позволяет минимизировать задержки при наведении. Благодаря этим особенностям устройство демонстрирует стабильную работу даже при резком изменении положения лазерного указателя, оперативно корректируя своё направление.

Помимо традиционного применения, система может быть адаптирована для выполнения дополнительных функций. Например, она может использоваться в автоматизированных системах маркировки объектов, в интерактивных экспозициях, где наведение на определенные точки активирует различные сценарии, а также в робототехнических решениях, где требуется точное следование за обозначенной целью. Возможны варианты использования в аэрокосмической отрасли, системах автономной навигации или даже в игровых технологиях, где устройство может выполнять роль интерактивного элемента, реагирующего на лазерные метки.

Таким образом, созданная система является перспективной разработкой, демонстрирующей возможности современных технологий в области автоматизированного управления и компьютерного зрения. Её гибкость и многофункциональность открывают широкие перспективы для дальнейшего развития и практического применения в самых разных сферах.

					<i>КП.07140500.006.40.00.00 ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>31</i>

Список использованных источников

- 1 Документация ESP32–CAM – <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/index.html>
- 2 Документация NodeMCU – <https://docs.espressif.com/projects/esp8266-rtos-sdk/en/latest/>
- 3 Асинхронный веб сервер ESP32/ESP8266/NodeMCU – <https://randomnerdtutorials.com/esp32-esp8266-asynchronous-web-server/>
- 4 Работа с Wi-Fi ESP32 – <https://github.com/espressif/arduino-esp32/blob/master/libraries/WiFi/src/WiFi.h>
- 5 Использование WebSocket для ESP32 – https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API
- 6 Работа с UART ESP32/ESP8266 – <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/uart.html>
- 7 Работа с ArduinoJson для ESP32/ESP8266 – <https://arduinojson.org/>

					<i>КП.07140500.006.40.00.00 ПЗ</i>		
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	<div style="text-align: center;"> <i>Список использованных источников</i> </div>		
<i>Разраб.</i>							
<i>Провер.</i>							
<i>Н. Контр.</i>							
<i>Утв.</i>							
					<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
						32	1
					<i>ККАТ зр.4ВМ-12</i>		

Приложения

Приложение 1

ESP32-CAM

```
#include "esp_camera.h"
#include <WiFi.h>
#define TXD_PIN 1 // TX (U0T) на ESP32-CAM
#define CAMERA_MODEL_AI_THINKER
#include "camera_pins.h"
#include <HTTPClient.h>

const char *ssid = "SystemAutoDetected";
const char *password = "ESP8266ESP32CAM";
const char* serverUrl = "http://192.168.4.2/coordinates";
IPAddress staticIP(192, 168, 4, 1); // Статический IP-адрес для камеры
IPAddress gateway(192, 168, 4, 1); // Шлюз (адрес ESP8266 в режиме AP)
IPAddress subnet(255, 255, 255, 0); // Маска подсети

size_t positionX = 0;
size_t positionY = 0;
WiFiServer server(80);
#define TXD_PIN 1 // TX0 (U0T) на ESP32-CAM

void sendCoordinates(int x, int y) {
    Serial1.print("X:"); Serial1.print(x);
    Serial1.print(",Y:"); Serial1.println(y);
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        http.begin(serverUrl);
        http.addHeader("Content-Type", "application/json");
        String payload = "{\"x\":\"" + String(x) + "\",\"y\":\"" + String(y) + "\"}";
        int httpResponseCode = http.POST(payload);
        if (httpResponseCode > 0) {
            // Serial.printf("Response: %s\n", http.getString().c_str());
            // Serial.printf("Coordinates: X = %d | Y = %d", x, y);
        } else {
            // Serial.printf("Error on sending POST: %d\n", httpResponseCode);
        }
    }
}
```

					<i>КП.07140500.006.40.00.00 ПЗ</i>		
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	<i>Приложения</i>		
<i>Разраб.</i>							
<i>Провер.</i>							
<i>Н. Контр.</i>							
<i>Утв.</i>							
					<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
						33	11
					<i>ККАТ зр.4ВМ-12</i>		


```

        http.end();
    } else {
        // Serial.println("WiFi disconnected!");
    }
}

void setCameraExposure(int level) {
    sensor_t *s = esp_camera_sensor_get();
    if (s != nullptr) {
        s->set_exposure_ctrl(s, 1); // Включение автоматической экспозиции (1
- включено, 0 - выключено)
        s->set_aec2(s, 0); // Выключение усовершенствованного контроля
экспозиции
        s->set_aec_value(s, level); // Установка уровня экспозиции (-2 до +2
или 0-120 в зависимости от камеры)
    }
}

void setup() {
    Serial.begin(115200);
    Serial.println();
    // Конфигурация камеры
    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_sccb_sda = SIOD_GPIO_NUM;
    config.pin_sccb_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_RGB565;
    config.frame_size = FRAMESIZE_XGA;

```

					<i>КП.07140500.006.40.00.00 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		34

```

config.jpeg_quality = 30;
config.fb_count = 1;
if (psramFound()) {
    // Serial.println("PSRAM detected.");
    config.fb_location = CAMERA_FB_IN_PSRAM;
} else {
    // Serial.println("PSRAM not found. Switching to DRAM.");
    config.fb_location = CAMERA_FB_IN_DRAM;
}
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    // Serial.printf("Camera init failed with error 0x%x", err);
    return;
}
setCameraExposure(-1);

WiFi.config(staticIP, gateway, subnet);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
// Serial.println("\nWiFi connected.");
// Serial.print("Use 'http://");
// Serial.print(WiFi.localIP());
// Serial.println("' to connect.");
server.begin();
    Serial1.begin(115200, SERIAL_8N1, -1, TXD_PIN); // UART TX на GPIO1
}
// void debugPixelConversion(uint16_t color565) {
//     uint8_t r, g, b;
//     RGB565toRGB888(color565, r, g, b);
//     Serial.printf("RGB565: 0x%04X -> RGB888: (%d, %d, %d)\n", color565, r, g,
b);
// }
uint16_t toLittleEndian(uint16_t color) {
    return (color >> 8) | (color << 8); // Swap bytes if needed
}
void RGB565toRGB888(uint16_t color565, uint8_t &r, uint8_t &g, uint8_t &b) {
    color565 = toLittleEndian(color565); // Ensure little-endian representation
    r = ((color565 >> 11) & 0x1F) * 255 / 31; // Convert 5 bits to 8 bits
    g = ((color565 >> 5) & 0x3F) * 255 / 63; // Convert 6 bits to 8 bits
    b = (color565 & 0x1F) * 255 / 31; // Convert 5 bits to 8 bits
}

```

					КП.07140500.006.40.00.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		35

```

bool isColorInRange(uint8_t r, uint8_t g, uint8_t b,
                    uint8_t minR, uint8_t maxR,
                    uint8_t minG, uint8_t maxG,
                    uint8_t minB, uint8_t maxB) {
    return (r >= minR && r <= maxR &&
            g >= minG && g <= maxG &&
            b >= minB && b <= maxB);
}

void processFrame(camera_fb_t *fb,
                  uint8_t minR, uint8_t maxR,
                  uint8_t minG, uint8_t maxG,
                  uint8_t minB, uint8_t maxB) {
    if (!fb) {
        Serial.println("Invalid frame buffer!");
        return;
    }
    uint16_t *pixels = (uint16_t *)fb->buf; // RGB565 pixels
    size_t width = fb->width;
    size_t height = fb->height;
    bool detected = false;
    int laserX = 0, laserY = 0;
    for (size_t y = 0; y < height; y++) {
        for (size_t x = 0; x < width; x++) {
            size_t index = y * width + x;
            uint8_t r, g, b;
            RGB565toRGB888(pixels[index], r, g, b);
            if (isColorInRange(r, g, b, minR, maxR, minG, maxG, minB, maxB)) {
                Serial.printf("Laser detected at (%d, %d) with RGB (%d, %d, %d)\n", x,
y, r, g, b);
                detected = true;
                laserX = x;
                laserY = y;
                positionX = x;
                positionY = y;
                break;
            }
        }
        if (detected) break;
    }
    if (detected) {

        drawCircleOutline(pixels, width, height, laserX, laserY, 10, 0xFFFF); //
White outline
    } else {

```

					КП.07140500.006.40.00.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		36

```

        Serial.println("No laser detected.");
    }
}

void calculateSectorAverage(uint16_t *pixels, size_t width, size_t height,
size_t sectorX, size_t sectorY, size_t sectorWidth, size_t sectorHeight, uint8_t
&avgR, uint8_t &avgG, uint8_t &avgB) {
    uint32_t sumR = 0, sumG = 0, sumB = 0;
    size_t pixelCount = 0;
    for (size_t y = sectorY; y < sectorY + sectorHeight; y++) {
        for (size_t x = sectorX; x < sectorX + sectorWidth; x++) {
            if (x >= width || y >= height) continue; // За пределами кадра
            size_t index = y * width + x;
            uint8_t r, g, b;
            RGB565toRGB888(pixels[index], r, g, b);
            sumR += r;
            sumG += g;
            sumB += b;
            pixelCount++;
        }
    }
    if (pixelCount > 0) {
        avgR = sumR / pixelCount;
        avgG = sumG / pixelCount;
        avgB = sumB / pixelCount;
    } else {
        avgR = avgG = avgB = 0;
    }
}

void drawCircleOutline(uint16_t *pixels, size_t width, size_t height, int
centerX, int centerY, int radius, uint16_t color) {
    for (int y = -radius; y <= radius; y++) {
        for (int x = -radius; x <= radius; x++) {
            int distance = x * x + y * y;
            if (distance >= (radius - 1) * (radius - 1) && distance <= radius *
radius) {
                int posX = centerX + x;
                int posY = centerY + y;
                if (posX >= 0 && posX < width && posY >= 0 && posY < height) {
                    pixels[posY * width + posX] = color;
                }
            }
        }
    }
}

```

					КП.07140500.006.40.00.00 ПЗ	Лист
						37
Изм.	Лист	№ докум.	Подпись	Дата		

```

    }
}

void sendJpegFrame(WiFiClient &client, camera_fb_t *fb) {
    uint8_t *jpeg_buf = NULL;
    size_t jpeg_len = 0;
    if (!frame2jpg(fb, 80, &jpeg_buf, &jpeg_len)) {
        Serial.println("Failed to convert frame to JPEG");
        client.println("HTTP/1.1 500 Internal Server Error");
        client.println("Content-Type: text/plain");
        client.println("Connection: close");
        client.println();
        client.println("Failed to convert frame to JPEG");
        return;
    }
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: image/jpeg");
    client.println("Content-Length: " + String(jpeg_len));
    client.println("Connection: close");
    client.println();
    client.write(jpeg_buf, jpeg_len);
    free(jpeg_buf);
}

void handleFrameProcessing() {
    camera_fb_t* fb = esp_camera_fb_get();
    if (!fb) {
        Serial.println("Failed to capture frame");
        return;
    }
    // Определение диапазонов цвета
    uint8_t minR = 220, maxR = 255;
    uint8_t minG = 129, maxG = 195;
    uint8_t minB = 129, maxB = 195;
    // Обработка изображения
    processFrame(fb, minR, maxR, minG, maxG, minB, maxB);
    // Отправка координат
    sendCoordinates(positionX, positionY);
    // Освобождение памяти
    esp_camera_fb_return(fb);
    Serial.println("Frame processed.");
}

unsigned long lastTime = 0; // Время последней отправки

```

					КП.07140500.006.40.00.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		38

```

const unsigned long interval = 100; // Интервал 100 мс
void loop() {
    unsigned long currentTime = millis();
    // Проверка, прошло ли 100 мс
    if (currentTime - lastTime >= interval) {
        lastTime = currentTime; // Обновить время последней отправки
        handleFrameProcessing();
        // Обработка входящих HTTP-запросов
        WiFiClient client = server.available();
        if (!client) {
            return;
        }
        String request = client.readStringUntil('\r');
        Serial.println(request);
        client.flush();
        camera_fb_t* fb = esp_camera_fb_get();
        if (!fb) {
            Serial.println("Failed to capture frame");
            client.println("HTTP/1.1 500 Internal Server Error");
            client.println("Content-Type: text/plain");
            client.println("Connection: close");
            client.println();
            client.println("Failed to capture frame");
            client.stop();
            return;
        }
        // Отправка изображения и координат
        sendJpegFrame(client, fb);
        sendCoordinates(positionX, positionY);
        // Освобождение памяти
        esp_camera_fb_return(fb);
        client.stop();
        Serial.println("Image sent.");
    }
}

```

ESP8266 NODEMCU

```

#include <ESP8266WiFi.h>
#include <ESPAsyncWebServer.h>
#include <ArduinoJson.h>
#include <Servo.h>
#include <SoftwareSerial.h>
const char* apSSID = "SystemAutoDetected";
const char* apPassword = "ESP8266ESP32CAM";

```

					КП.07140500.006.40.00.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		39

```

IPAddress apIP(192, 168, 4, 2);          // Задаём статический IP для ESP8266
IPAddress apSubnet(255, 255, 255, 0); // Маска подсети

#define SERVO_X_PIN D5
#define SERVO_Y_PIN D6

Servo servoX;
Servo servoY;

SoftwareSerial mySerial(13, 15); // RX = GPIO13, TX = GPIO15 (TX не
используется)

AsyncWebServer server(80);
int currentX = 0;
int currentY = 0;
const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE html>
<html>
<head>
  <title>ESP8266 Web Server</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <style>
    body { font-family: Arial, sans-serif; text-align: center; margin-top:
50px; }
    h1 { color: #333; }
    p { font-size: 1.2em; }
  </style>
</head>
<body>
  <h1>ESP8266 Асинхронный Веб-Сервер</h1>
  <p>Сервер получает координаты для управления.</p>
  <p>Текущие координаты:</p>
  <p>X:<span id="coordX">0</span>,Y:<spanid="coordY">0</span></p>
  <script>
    function updateCoordinates() {
      fetch('/get-coordinates')
        .then(response => response.json())
        .then(data => {
          document.getElementById('coordX').textContent = data.x;
          document.getElementById('coordY').textContent = data.y;
        })
        .catch(error => console.error('Error:', error));
    }
    setInterval(updateCoordinates, 500);
  </script>
</body>
</html>
)rawliteral";

```

					КП.07140500.006.40.00.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		40

```

void smoothMoveServo(Servo &servo, int currentAngle, int targetAngle, int
stepDelay = 10, int stepSize = 1) {
    if (currentAngle < targetAngle) {
        for (int angle = currentAngle; angle <= targetAngle; angle += stepSize) {
            servo.write(angle);
            delay(stepDelay); // Небольшая задержка для плавного движения
        }
    } else {
        for (int angle = currentAngle; angle >= targetAngle; angle -= stepSize) {
            servo.write(angle);
            delay(stepDelay);
        }
    }
}

int lastX = -1; // Переменные для хранения предыдущих координат
int lastY = -1;

void moveToRegion(int x, int y) {
    const int imageWidth = 1024;           // Ширина изображения
    const int imageHeight = 768;           // Высота изображения
    const int SERVO_MIN_ANGLE = 0;         // Минимальный угол сервопривода
    const int SERVO_MAX_ANGLE = 180;       // Максимальный угол сервопривода
    const float CAMERA_VIEW_ANGLE_X = 25.0; // Угол обзора камеры по горизонтали
    const float CAMERA_VIEW_ANGLE_Y = 20.0; // Угол обзора камеры по вертикали

    const int centerX = imageWidth / 2;    // Центр по оси X
    const int centerY = imageHeight / 2;    // Центр по оси Y

    if (x == lastX && y == lastY) {
        // Если координаты не изменились, ничего не делаем
        // Serial.printf("Координаты не изменились: X = %d, Y = %d\n", x, y);
        return;
    }
    // Сохраняем новые координаты
    lastX = x;
    lastY = y;

    // Определение смещения от центра
    int deltaX = x - centerX;
    int deltaY = centerY - y; // Инверсия по Y, так как камера сверху

    // Инвертирование смещения по оси X, если необходимо
    deltaX = -deltaX;

```

					КП.07140500.006.40.00.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		41


```

        // Преобразование смещения в углы с уменьшением чувствительности
        float angleOffsetX = ((float)deltaX / centerX) * (CAMERA_VIEW_ANGLE_X / 2) *
0.5; // Уменьшен коэффициент
        float angleOffsetY = ((float)deltaY / centerY) * (CAMERA_VIEW_ANGLE_Y / 2) *
0.5; // Уменьшен коэффициент

        // Получение текущих углов сервоприводов
        int currentServoXAngle = servoX.read();
        int currentServoYAngle = servoY.read();

        // Вычисление целевых углов
        int targetServoXAngle = currentServoXAngle + angleOffsetX * (SERVO_MAX_ANGLE
/ CAMERA_VIEW_ANGLE_X);
        int targetServoYAngle = currentServoYAngle + angleOffsetY * (SERVO_MAX_ANGLE
/ CAMERA_VIEW_ANGLE_Y);

        // Ограничение углов в допустимом диапазоне
        targetServoXAngle = constrain(targetServoXAngle, SERVO_MIN_ANGLE,
SERVO_MAX_ANGLE);
        targetServoYAngle = constrain(targetServoYAngle, SERVO_MIN_ANGLE,
SERVO_MAX_ANGLE);

        // Добавлен порог для минимального отклонения
        const int MIN_MOVE_THRESHOLD = 5; // Минимальное изменение для движения

        // Если отклонение от центра превышает порог, двигаем сервоприводы
        if (abs(deltaX) > MIN_MOVE_THRESHOLD || abs(deltaY) > MIN_MOVE_THRESHOLD) {
            // Наведение сервоприводов
            smoothMoveServo(servoX, currentServoXAngle, targetServoXAngle);
            smoothMoveServo(servoY, currentServoYAngle, targetServoYAngle);

            // Лог для отладки
            // Serial.printf("Наведение завершено: ServoX = %d, ServoY = %d
(Координаты: X = %d, Y = %d)\n",
                targetServoXAngle, targetServoYAngle, x, y);
        } else {
            // Serial.printf("Лазер в центре, нет необходимости в движении: ServoX =
%d, ServoY = %d (Координаты: X = %d, Y = %d)\n",
                currentServoXAngle, currentServoYAngle, x, y);
        }
    }

    void setup() {

```

					КП.07140500.006.40.00.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		42

```

Serial.begin(115200);
mySerial.begin(115200);    // Виртуальный Serial для ESP32-CAM
servoX.attach(SERVO_X_PIN);
servoY.attach(SERVO_Y_PIN);
servoX.write(0);
servoY.write(0);
WiFi.softAPConfig(apIP, apIP, apSubnet);
WiFi.softAP(apSSID, apPassword);
Serial.println("Точка доступа создана!");
Serial.print("Название сети: ");
Serial.println(apSSID);
Serial.print("IP-адрес: ");
Serial.println(WiFi.softAPIP());
server.on("/", HTTP_GET, [] (AsyncWebServerRequest *request) {
    request->send_P(200, "text/html; charset=UTF-8", index_html);
});

server.on("/get-coordinates", HTTP_GET, [] (AsyncWebServerRequest *request) {
    DynamicJsonDocument doc(1024);
    doc["x"] = currentX;
    doc["y"] = currentY;
    String json;
    serializeJson(doc, json);
    request->send(200, "application/json", json);
});
server.begin();
Serial.println("Сервер запущен!");
}

void loop() {
    if (mySerial.available()) {
        String data = mySerial.readStringUntil('\n'); // Читаем строку до конца
        Serial.print("Принято по UART: ");
        Serial.println(data);
        mySerial.flush();
        int x, y;
        if (sscanf(data.c_str(), "X:%d,Y:%d", &x, &y) == 2) {
            currentX = x;
            currentY = y;
            moveToRegion(currentX, currentY);
        }
    }
}

```

					КП.07140500.006.40.00.00 ПЗ	Лист
						43
Изм.	Лист	№ докум.	Подпись	Дата		