

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук  
Кафедра информационных и управляющих систем  
Направление подготовки 09.04.04 – Программная инженерия  
Направленность (профиль) образовательной программы Управление разработкой программного обеспечения

ДОПУСТИТЬ К ЗАЩИТЕ  
Зав. кафедрой  
\_\_\_\_\_ А.В. Бушманов  
« \_\_\_ » \_\_\_\_\_ 2025 г.

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

на тему: Проектирование и реализация облачной платформы сбора экологических данных

Исполнитель  
студент группы 3105-ом2

Чжоу Гуаньюй

\_\_\_\_\_  
(подпись, дата)

Руководитель  
доцент, канд. физ.-мат.  
наук

В.В. Ерёмина

\_\_\_\_\_  
(подпись, дата)

Руководитель научного  
содержания программы  
магистратуры  
профессор, доктор техн. наук

И.Е. Еремин

\_\_\_\_\_  
(подпись, дата)

Нормоконтроль  
инженер кафедры

В.Н. Адаменко

\_\_\_\_\_  
(подпись, дата)

Рецензент  
директор, ЦЦТиТО

А.А. Тодосейчук

\_\_\_\_\_  
(подпись, дата)

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
**(ФГБОУ ВО «АмГУ»)**

Институт компьютерных и инженерных наук  
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ

Зав. кафедрой

\_\_\_\_\_ А.В. Бушманов

« \_\_\_\_ » \_\_\_\_\_ 2025 г.

**З А Д А Н И Е**

К магистерской диссертации студента группы 3105-ом2 Чжоу Гуаньюй

1. Тема магистерской диссертации: Проектирование и реализация облачной платформы сбора экологических данных

(Утверждено приказом от 06.03.2025 № 609-уч)

2. Срок сдачи студентом законченной работы (проекта): 10.06.2025

3. Исходные данные к магистерской диссертации: историческое описание, документация разработчиков, интернет ресурсы, учебная литература

4. Содержание магистерской диссертации (перечень подлежащих разработке вопросов): Общая характеристика системы сбора экологических данных для облачной платформы на базе STM32, решение алгоритма сбора данных и разработка программного и аппаратного обеспечения, решение и программная реализация алгоритма сбора данных среды облачной платформы

5. Перечень материалов приложения (наличие чертежей, таблиц, графиков, схем, программных продуктов, иллюстративного материала и т.п.): \_\_\_\_\_

6. Рецензент магистерской диссертации: А.А. Тодосейчук, директор, ЦЦТиТО

7. Дата выдачи задания: 15.01.2025

Руководитель выпускной квалификационной работы: \_\_\_\_\_

В.В. Еремина, доцент, канд. физ.-мат. наук

(фамилия, имя, отчество, должность, уч.степень, уч.звание)

Заявление принял к исполнению \_\_\_\_\_

## РЕФЕРАТ

Магистерская диссертация содержит 84 страница, 36 рисунок, 31 источник

### МИКРОКОНТРОЛЛЕР STM32, ОБЛАЧНАЯ ПЛАТФОРМА СБОРА ДАННЫХ, ЯЗЫК C

Используя STM32 в качестве основного контроллера и протокол MQTT, можно обеспечить эффективную, надежную передачу данных в реальном времени на платформе ONENET. Инструменты разработчика WeChat используются для разработки апплетов взаимодействия человека с компьютером, а для реализации связи между ними используется сетевая связь. сервер и клиент, что делает его Главным управляющий компьютер, серверная и клиентская стороны имеют функцию мониторинга данных об окружающей среде в режиме реального времени.

Сочетая микроконтроллер STM32 и облачную платформу ONENET, осуществляется сбор данных об окружающей среде и облачный анализ, что обеспечивает надежную поддержку будущих приложений Интернета вещей и реализует сбор, анализ и передачу данных об окружающей среде. Система обладает хорошей стабильностью и высокой масштабируемостью.

Структура работы. Данная исследовательская работа состоит из введения, четырех основных глав, заключения и списка использованных источников.

## СОДЕРЖАНИЕ

Введение	5
1 Общая характеристика системы сбора экологических данных для облачной платформы на базе STM32	7
1.1 Анализ экологических данных и процессов их оптимального сбора	7
1.2 Обзор существующих методов решения аналогичных задач	16
2 Решение алгоритма сбора данных и разработка программного и аппаратного обеспечения	22
2.1 Предлагаемый компьютерный алгоритм для решения процесса сбора данных	22
2.2 Обзор функциональной конструкции программного обеспечения для сбора данных об окружающей среде	34
2.3 Разработка характеристик оборудования, выбранного для системы сбора данных	43
3 Решение и программная реализация алгоритма сбора данных среды облачной платформы	49
3.1 Основные этапы практического развития систем сбора данных об окружающей среде	49
3.2 Пример тестирования программного обеспечения системы сбора данных об окружающей среде	66
Заключение	81
Библиографические ссылки	82

## ВВЕДЕНИЕ

С развитием и распространением науки и технологий все чаще используются интеллектуальные устройства, что поднимает такие проблемы, как точность экологических проблем. Раньше сбор экологических данных в основном осуществлялся вручную. В настоящее время ручной сбор данных имеет такие недостатки, как низкая эффективность и ненаучная система управления. В настоящее время тепличное разведение в основном основано на ручном управлении, но ручное управление имеет множество недостатков: во-первых, мониторинг традиционных теплиц затруднен, что не способствует работе менеджеров и контролю за тепличной средой в случае возникновения ошибок. эксплуатации, это повлияет на рост урожая и вызовет огромные последствия. Во-вторых, традиционные теплицы оказывают определенное влияние на эффективность производства крупномасштабных хозяйственных культур из-за сложной установки оборудования и электропроводки, а также высокой стоимости. Хотя многие теплицы и фабрики также оснащены соответствующим оборудованием для отопления, увлажнения и вентиляции, управление этими видами оборудования в основном осуществляется вручную. В этом случае, когда объем работ по проверке и техническому обслуживанию оборудования велик, возникнут проблемы с высокой эффективностью. Интенсивность труда возрастает. Кроме того, этот метод не может точно отслеживать окружающую среду в режиме реального времени и не может удовлетворить потребности современного быстро развивающегося общества. Таким образом, разработка простой системы сбора экологических данных, которая объединяет сбор, передачу и отображение экологических данных, значительно снизит трудоемкость операторов и повысит точность мониторинга окружающей среды. В настоящее время традиционная технология экологического мониторинга широко используется в различных отраслях промышленности и превратилась в полно-

ценную систему. Интеллектуальная технология сбора экологических данных позволяет проводить централизованный мониторинг различных факторов окружающей среды и также применяется в некоторых развитых странах. Однако его стоимость высока, а продвижение затруднено. В настоящее время технология проводного мониторинга окружающей среды в стране и за рубежом достаточно развита, но сфера ее применения очень ограничена, и ее ограничения также очевидны. MQTT стал предпочтительным протоколом облачной связи для отечественного промышленного Интернета вещей. Следовательно, необходимо разработать систему, которая может передавать эти данные по беспроводной сети на центральный компьютер для обработки и выполнять некоторые операции дистанционного управления. Эта система представляет собой устройство для беспроводного сбора данных об окружающей среде и неограниченного дистанционного управления на основе сбора и передачи. И отображать данные об окружающей среде. Среда сбора данных на основе беспроводной технологии имеет достаточное пространство для развития и широкие перспективы применения в производстве, промышленном и сельскохозяйственном развитии. Таким образом, разработка этой интеллектуальной системы сбора экологических данных может быть широко использована для удовлетворения потребностей развития рынка.

# 1 ОБЩАЯ ХАРАКТЕРИСТИКА СИСТЕМЫ СБОРА ЭКОЛОГИЧЕСКИХ ДАННЫХ ДЛЯ ОБЛАЧНОЙ ПЛАТФОРМЫ НА БАЗЕ STM32

## 1.1 Анализ экологических данных и процессов их оптимального сбора

На основе анализа системных требований система сбора данных об окружающей среде на основе STM32 в основном состоит из сенсорных модулей, основных модулей управления, модулей отображения и модулей беспроводной передачи и управления ESP8266. В качестве основного контроллера используется процессор STM32, а для передачи обнаруженных данных на облачную платформу ONENET используется беспроводной модуль ESP8266.

Используются инструменты разработчика WeChat для разработки приложения WeChat для взаимодействия человека с компьютером и используйте сетевую связь для реализации связи на стороне сервера, чтобы приложение WeChat имел функцию наблюдения за информацией об окружающей среде и выполнения дистанционного управления.

Структура системы показана на рисунке 1. Сенсорный модуль осуществляет сбор данных и передает их на главный модуль управления через последовательный порт. Главный модуль управления отображает данные на экране дисплея посредством анализа и обработки и передает данные на облачную платформу через беспроводной модуль WIFI ESP8266. Клиентский апплет WeChat получает данные на облачную платформу для отображения на облачной платформе. В то же время клиентская программа WeChat может отправлять данные на облачную платформу. Отправка данных окончательно реализует работу модуля управления.

Каждый модуль работает вместе для обеспечения мониторинга окружающей среды в режиме реального времени.

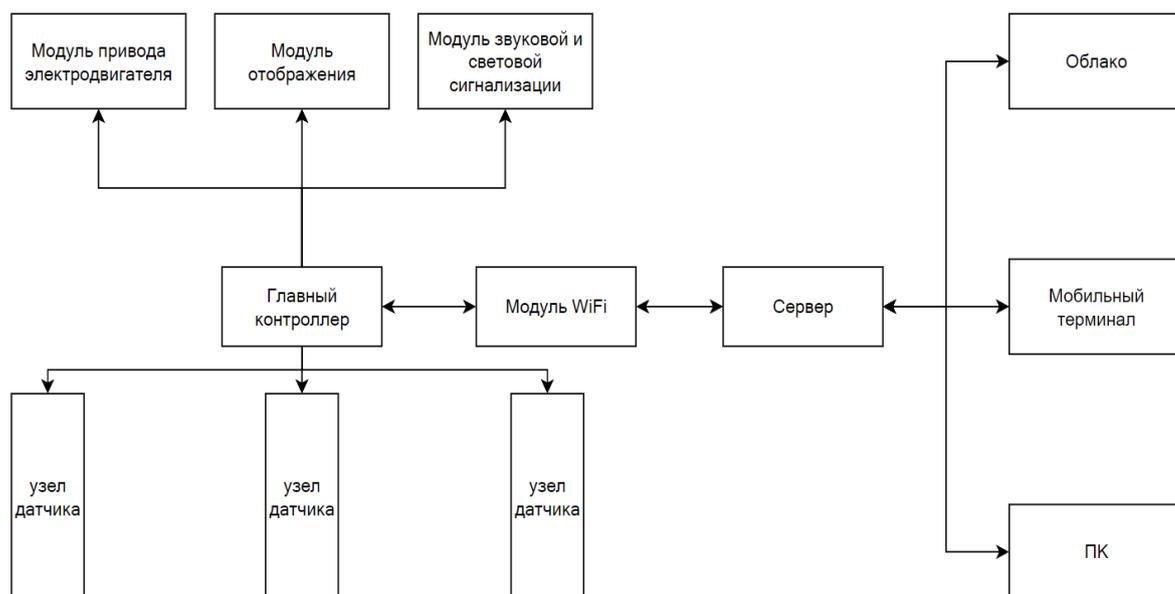


Рисунок 1 – Общая каркасная схема системы

Система предназначена для эффективного сбора и мониторинга данных об окружающей среде. Она объединяет датчики, хосты, облачные платформы и клиенты для достижения комплексного метода связи. Пользователи могут просматривать экологические данные в режиме реального времени через клиент, обеспечивая тем самым безопасность домашней среды и высокое качество производства в таких местах, как теплицы и фабрики. Система основана на технологиях беспроводной, последовательной и сетевой связи, и каждый модуль подключается друг к другу с помощью различных средств связи для достижения высококачественного взаимодействия человека с компьютером. На основе функций системы добавляются датчики температуры и влажности, датчики освещенности, датчики дыма и модули ESP8266 WIFI для осуществления мониторинга окружающей среды в реальном времени.

Процесс разработки системы можно разделить на следующие этапы:

- общее проектирование аппаратной среды: спланируйте и спроектируйте аппаратные компоненты системы;
- общая конструкция программной платформы: создание базовой программной среды системы;
- сбор данных об окружающей среде: используйте датчики для сбора

данных об окружающей среде;

- передача данных: предварительная передача данных осуществляется посредством последовательной связи;

- беспроводная передача и отображение данных: реализация беспроводной передачи и отображение данных в реальном времени;

- сетевое программирование: реализация передачи данных между клиентом и сервером;

- разработка и реализация интерфейса апплета WeChat: разработайте апплет WeChat, обеспечивающий удобный пользовательский интерфейс;

- проверка стабильности работы системы. Проверьте общую стабильность системы, чтобы убедиться в ее надежности и точности.

После анализа предмета исследования был применен комплексный подход к коммуникации с помощью сенсора, хоста, облачной платформы и клиента. В качестве главного контроллера было принято решение использовать STM32, благодаря его высокой производительности, энергоэффективности и широким возможностям периферийного подключения. Для проектирования интерактивных микропрограмм человеко-машинного взаимодействия использовались инструменты разработчика WeChat, обеспечивающие удобную реализацию пользовательского интерфейса и интеграцию с мобильными устройствами.

Связь между серверной и клиентской частями системы реализована посредством сетевых технологий, что позволило организовать эффективную передачу данных в режиме реального времени. Таким образом, главный контроллер, сервер и клиент находятся в постоянном взаимодействии и обладают возможностью непрерывного мониторинга параметров окружающей среды.

В систему также интегрированы механизмы обработки и хранения данных на облачной платформе, обеспечивающие масштабируемость, безопасность и доступность информации для последующего анализа. Для по-

вышения надежности были предусмотрены средства автоматической диагностики и оповещения при возникновении критических значений измеряемых параметров. Общий проект системы представлен на рисунке 2, где наглядно показаны основные компоненты архитектуры и их взаимосвязи.

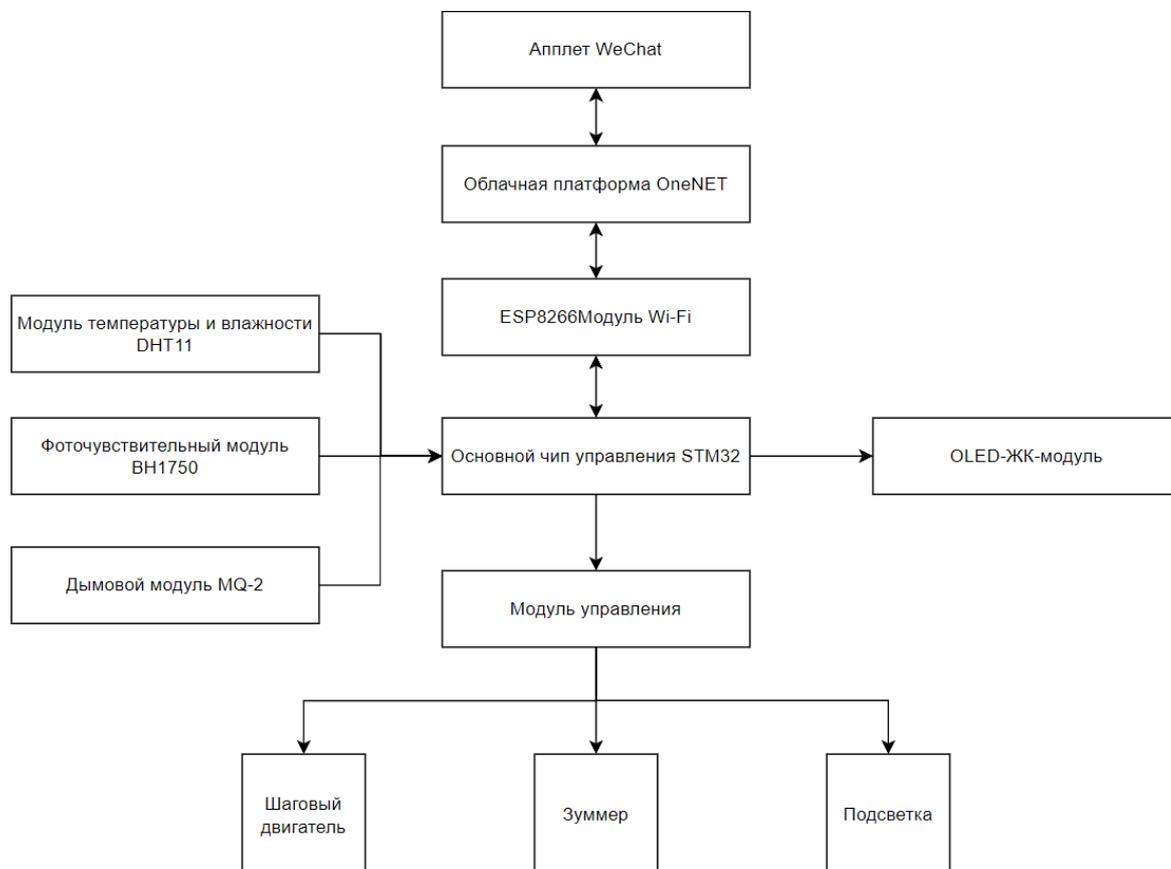


Рисунок 2 – Общий проектный чертеж системы

В основном система реализует следующие основные функции:

- сбор данных: используйте модуль датчика температуры и влажности, модуль датчика освещенности и модуль датчика дыма для сбора различных данных об окружающей среде в режиме реального времени для всестороннего мониторинга условий окружающей среды;

- беспроводная передача: данные отправляются и принимаются через модуль ESP8266 для реализации беспроводной связи между основным чипом управления и облачной платформой для обеспечения своевременности и надежности передачи данных;

- устройство сигнализации: оснащено звуковой и световой сигнализацией и устанавливает пороговые значения экологических данных. Как

только контролируемые данные об окружающей среде превысят заданный порог, система автоматически подаст сигнал тревоги, чтобы оперативно напомнить пользователям об аномальных условиях;

– интерфейс взаимодействия человека с компьютером: сервер облачной платформы получает данные об окружающей среде через модуль ESP8266, а клиент-апплет получает и отображает эти данные с сервера облачной платформы через сетевую связь, обеспечивая интуитивно понятный пользовательский интерфейс для обеспечения удобного мониторинга и управления окружающей средой.

Система структурно разделена на четыре сегмента, а именно на сегмент сбора данных, сегмент хоста, сегмент сервера и сегмент клиента.

Часть 1. Сбор данных: Сбор данных об окружающей среде осуществляется с помощью светочувствительного датчика, датчика дыма, датчика температуры и влажности и передачи данных через последовательный порт.

Часть 2. Хост-компьютер: STM32 в качестве главного чипа управления осуществляет прием данных через последовательный порт, с помощью беспроводного модуля ESP8266 входит в сеть Интернета через конфигурацию команды AT, через конфигурацию протокола MQTT, построенного по протоколу TCP/IP, входит в облачную платформу для передачи данных с сервером. Отображение данных через ЖК-экран и отправка полученных данных в модуль управления через последовательный порт.

Сервер MQTT, также известный как MQTT-брокер (Брокер), является основным компонентом, отвечающим за получение, обработку и распространение сообщений. Он выступает посредником между клиентами, обеспечивая плавную передачу сообщений от издателя подписчику. Некоторые основные функции и возможности серверов MQTT представлены ниже.

Режим публикации/подписки: клиенты могут публиковать сообщения в определенной теме (Теме), а другие клиенты могут подписываться на эти темы для получения сообщений. Сервер занимается доставкой всех этих сообщений.

Иерархия тем. Темы – это способ категоризации сообщений, позволяющий создать иерархическую структуру (например, дом/гостиная/температура) и помогающий организовывать и фильтровать сообщения.

Долговечность. постоянные сеансы: MQTT поддерживает постоянные сеансы. Даже если клиент отключается, сервер все равно может сохранять неполученные сообщения и доставлять их при повторном подключении клиента.

Хранение сообщений: вы можете настроить сервер на сохранение сообщений до тех пор, пока они не будут получены всеми подписчиками, или выбрать различные стратегии хранения в соответствии с потребностями.

Обеспечение качества (QoS):

– QoS 0 (не более одного раза): сообщение отправляется один раз, без повторных попыток и без гарантии прибытия сообщения;

– QoS 1 (хотя бы один раз): сообщение будет повторено для отправки, гарантируя, что сообщение будет получено хотя бы один раз, но могут быть дубликаты;

– QoS 2 (только один раз): сообщения гарантированно доставляются только один раз, что позволяет избежать дублирования за счет сложного процесса установления связи.

Аутентификация и авторизация. Сервер может потребовать от клиента предоставить имя пользователя и пароль для аутентификации. Существуют и другие способы контролировать, кто может публиковать сообщения и подписываться на них.

Передача данных осуществляется с помощью шифрования TLS/SSL для обеспечения безопасности данных во время передачи.

Управление подключением реализуется при помощи механизма Heartbeat: пакеты Heartbeat периодически передаются между клиентом и сервером, чтобы подтвердить, действительно ли соединение по-прежнему.

Автономные сообщения: поддержка сервера для хранения неотправленных сообщений, когда клиент отключен, гарантируя, что сообщения не

будут потеряны.

С помощью сервера MQTT осуществляется связь через модуль ESP8266WIFI, осуществляется сетевая связь с клиентом, отображаются данные окружающей среды.

Клиентская часть: Использование апплета WeChat в качестве клиента для реализации сетевой связи с сервером, получения и отображения данных об окружающей среде в режиме реального времени.

Ниже приводится анализ и изучение функций чипа ESP8266, включая режим его работы, протокол связи и способы его использования в системе дистанционного управления:

ESP8266 – это популярный модуль Wi-Fi, который в настоящее время представлен на рынке и широко используется в устройствах Интернета вещей (IoT) и умного дома. Он обладает мощными возможностями беспроводной связи и очень подходит для встраиваемых систем с точки зрения функциональности и гибкости.

ESP8266 может обмениваться данными с другими устройствами через последовательный интерфейс (UART). Это упрощает взаимодействие с микроконтроллерами, такими как Arduino, или другими платами разработки. ESP8266 поддерживает несколько режимов энергосбережения, которые снижают энергопотребление в периоды бездействия, тем самым продлевая срок службы батареи. Возможность подключения к беспроводной сети для связи с облачными серверами или другими сетевыми устройствами.

ESP8266 поддерживает несколько сетевых режимов, в том числе: Режим AP (точка доступа, точка доступа):

В этом режиме ESP8266 сам выступает в качестве точки доступа Wi-Fi, обеспечивая подключение к другим устройствам. Устройства могут подключаться к ESP8266 через Wi-Fi в качестве сервера для передачи данных или управления. Режим STA (станция, сайт): в этом режиме ESP8266 действует как клиент Wi-Fi и подключается к существующей беспроводной

сети (например, к домашнему маршрутизатору). Устройство может выходить в Интернет или взаимодействовать с другими сетевыми устройствами через маршрутизатор. Режим AP+STA: ESP8266 сочетает в себе функции режимов AP и STA, одновременно выступая в качестве точки доступа Wi-Fi и клиента. В этом режиме ESP8266 может напрямую взаимодействовать с компьютерами или другими устройствами (как точка доступа) или подключаться к другим сетям (как STA) для достижения более гибких сетевых функций.

ESP8266 поддерживает множество сетевых протоколов связи, в том числе: TCP/IP: ESP8266 поддерживает стандартный стек протоколов TCP/IP и может выполнять надежную передачу данных через TCP. Обычно используется для установления соединения с сервером для обмена данными. MQTT (транспорт телеметрии очереди сообщений):

MQTT – это облегченный протокол обмена сообщениями, особенно подходящий для использования с устройствами Интернета вещей.

Он основан на модели публикации/подписки, которая обеспечивает эффективную связь в сетевых средах с низкой пропускной способностью, высокой задержкой или ненадежными сетями. ESP8266 обеспечивает обмен сообщениями и удаленное управление между устройствами через MQTT. AT-команды (команды внимания) – распространенный способ управления ESP8266. Используйте AT-команды для настройки параметров сети, управления подключениями Wi-Fi, запроса статуса и многого другого. Например:

AT+CWJAP = «SSID», «ПАРОЛЬ»: подключитесь к указанной сети Wi-Fi.

AT+CIFSR: запрос текущего IP-адреса.

AT+CIPSTART=«TCP», «IP\_ADDRESS», PORT: установить TCP-соединение.

Используя режим AP+STA ESP8266, можно реализовать следующие функции:

– режим AP позволяет пользователям напрямую подключаться к

ESP8266 через компьютер или мобильный телефон для локального управления, например включения и выключения двигателей и звуковых сигналов;

– режим STA позволяет ESP8266 подключаться к домашнему беспроводному маршрутизатору, позволяя удаленно управлять устройством или взаимодействовать с другими устройствами и службами через Интернет.

Такая конфигурация позволяет не только локальное управление, но и удаленное управление через Интернет, что делает систему более гибкой и удобной.

Модуль датчика освещенности. В системе используется цифровой датчик освещенности BH1750FVI в сочетании с внутренним аналого-цифровым преобразованием для определения значений освещенности. BH1750FVI – это цифровой датчик освещенности, который использует внутренний фотодиод и аналого-цифровой преобразователь (АЦП) для измерения интенсивности света. Фотодиод преобразует оптический сигнал в электрический сигнал, а АЦП преобразует аналоговый электрический сигнал в цифровой сигнал для получения цифрового значения освещенности. Его можно использовать для обнаружения окружающего освещения, автоматической регулировки яркости экрана, интеллектуальных систем освещения и т. д.

Модуль датчика температуры и влажности. Система использует цифровой датчик температуры и влажности DHT11 для сбора данных о температуре и влажности и передает их в соответствии с протоколом передачи с временным разделением каналов (TDM) для получения необходимых данных. DHT11 – широко используемый цифровой датчик температуры и влажности, работающий в режиме вывода цифрового сигнала. Он измеряет температуру и влажность окружающей среды с помощью внутренних сенсорных элементов и передает данные в главный блок управления по протоколу мультиплексирования с временным разделением (TDM). Подходит для домашних метеостанций, систем мониторинга окружающей среды, управления кондиционером и увлажнителем и т. д.

Модуль датчика дыма. Модуль датчика дыма MQ-2 используется для контроля концентрации дыма в системе. Он может определить, превышает ли концентрация дыма в окружающей среде стандарт. Модуль оснащен регулируемым потенциометром для регулировки чувствительности выхода TTL. В сочетании со значениями других модулей окружающей среды можно более точно определить, есть ли в окружающей среде скрытые опасности. MQ-2 – это широко используемый датчик газа, который может обнаруживать в воздухе различные газы, такие как дым, угарный газ, метан и т. д. Принцип заключается в определении концентрации газа путем нагревания газа нагревательным элементом и определения концентрации газа. Изменить проводимость. Широко используется в системах пожарной сигнализации, мониторинге качества воздуха, обнаружении утечек газа и т. д.

Каждый из этих сенсорных модулей имеет разные области применения и характеристики и подходит для мониторинга различных данных об окружающей среде. Датчик освещенности BH1750FVI в основном используется для измерения интенсивности света, DHT11 используется для сбора данных о температуре и влажности, а MQ-2 используется для обнаружения дыма и концентрации газа. При интеграции этих датчиков следует учитывать их методы вывода, совместимость интерфейсов и методы обработки данных, чтобы гарантировать, что система может точно получать и обрабатывать данные об окружающей среде.

## **1.2 Обзор существующих методов решения аналогичных задач**

Ниже приводится анализ трех существующих систем экологического мониторинга в Китае:

– пекинская система мониторинга качества воздуха. Установите станции мониторинга качества воздуха по всему городу для мониторинга PM2.5, PM10, CO и других показателей. Для обеспечения передачи данных в реальном времени используется сочетание проводной и беспроводной связи. Данные анализируются через облачную платформу, отчеты о мониторинге ге-

нерируются в режиме реального времени и публикуются для общественности. Горожане могут проверить индекс качества воздуха (AQI) и концентрацию различных загрязняющих веществ через официальный сайт или мобильное приложение. Схема структуры системы показана на рисунке 3. Система может комплексно отражать состояние качества воздуха в различных районах. Благодаря мониторингу в реальном времени и анализу данных можно быстро выявить изменения качества воздуха и своевременно выдать ранние предупреждения. Открытие данных для общественности через официальные веб-сайты и мобильные приложения повысило внимание общественности и повысило ее участие в вопросах качества воздуха. Однако затраты на приобретение и обслуживание использования высокоточных датчиков и оборудования для мониторинга высоки. На качество воздуха сильно влияют метеорологические факторы (такие как скорость ветра, влажность и т. д.), и при интерпретации данных необходимо учитывать эти внешние факторы;

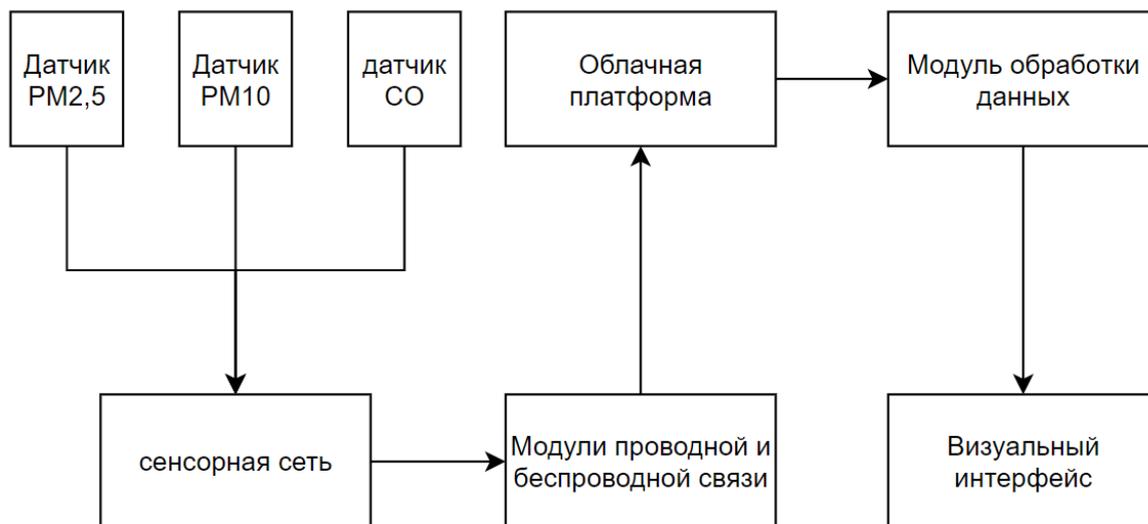


Рисунок 3 – Карта системы мониторинга качества воздуха в Пекине

– система мониторинга качества воды реки Янцзы. Вдоль реки Янцзы было установлено несколько станций мониторинга качества воды, оснащенных онлайн-датчиками качества воды для мониторинга температуры воды, pH, растворенного кислорода, мутности, содержания тяжелых металлов и

других показателей в режиме реального времени. Данные передаются в режиме реального времени в центральную базу данных посредством беспроводной связи. Отбор проб воды и лабораторный анализ проводятся регулярно для проверки точности данных онлайн-мониторинга. Схема структуры системы показана на рисунке 4. Эта система мониторинга может одновременно отслеживать несколько показателей качества воды и всесторонне оценивать состояние здоровья водных объектов. Объедините онлайн-мониторинг с лабораторным анализом, чтобы повысить точность и надежность данных. Однако существуют также проблемы с высокоточным оборудованием для мониторинга качества воды и высокими затратами на техническое обслуживание, особенно при его использовании в отдаленных районах. А следить за качеством воды он может только на берегу, что имеет ограничения;

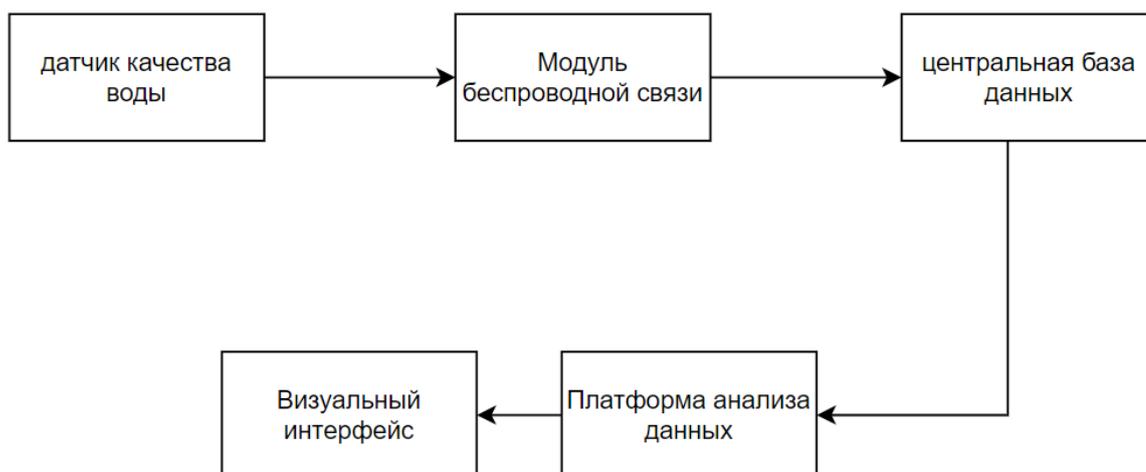


Рисунок 4 – Схема системы мониторинга качества воды реки Янцзы

– система мониторинга загрязнения почвы. В ключевых районах (таких как промышленные районы и строительные площадки) было развернуто несколько станций мониторинга почвы, которые используют электрохимические датчики для обнаружения тяжелых металлов и органических загрязнителей. Мониторинг в режиме реального времени и анализ исторических данных проб почвы выполняются через интегрированную базу данных. Схема структуры системы показана на рисунке 5. Эта система может кон-

тролировать ключевые области и эффективно выявлять источники загрязнения. Однако мониторинг почвы может проводиться не так часто, как мониторинг качества воздуха и воды, а обновление данных относительно отстает. На сбор проб почвы могут влиять региональные факторы, а некоторые промышленные зоны могут быть недоступны вручную, что приводит к увеличению затрат на отбор проб.

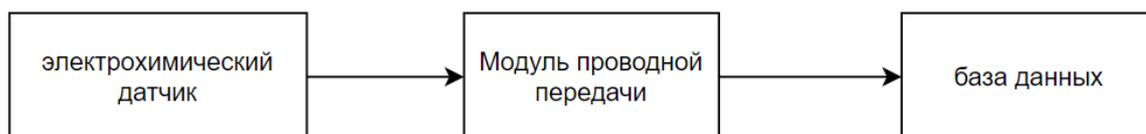


Рисунок 5 – Схема системы мониторинга загрязнения почвы

Данная система использует платформу беспроводной передачи данных с различными сенсорными модулями, объединяет последовательную связь и сетевую связь TCP и образует интеллектуальную систему беспроводного мониторинга окружающей среды. По сравнению с традиционным режимом мониторинга на месте она имеет следующие преимущества:

Система имеет функцию мониторинга экологической информации в режиме реального времени и может проводить комплексный мониторинг в соответствии с различными требованиями к параметрам, чтобы гарантировать, что данные мониторинга не будут пропущены или повторены. Благодаря функции беспроводной передачи система обеспечивает бесперебойную связь между клиентом и сервером. Хост может напрямую получать данные датчиков через последовательный порт, что позволяет работникам удаленно контролировать одну или несколько аналогичных сред, тем самым обеспечивая удобное и эффективное управление.

В системе используется модуль WIFI ESP8266, который устанавливается на хосте для реализации беспроводной связи между хостом и сервером облачной платформы. Сетевая связь основана на протоколе MQTT, который построен на основе протокола TCP/IP и обладает эффективными и надеж-

ными возможностями передачи данных в реальном времени, особенно подходящими для платформы ONENET. Чтобы обеспечить высокую стабильность и высокую производительность системы, основным процессором является STM32F103C8T6. Чип не только обладает превосходной стабильностью и производительностью, но также имеет большой объем памяти, подходящий для обработки сложных операций с данными.

В части беспроводной передачи системы используется чип ESP8266, который имеет характеристики последовательного порта и низкое энергопотребление. Он подключает интеллектуальные устройства пользователей к беспроводным сетям. Используйте сетевой протокол TCP/MQTT для беспроводной связи для реализации сети беспроводных терминалов. Параметры сетевой системы передаются с помощью AT-команды AT+<COMMAND>=<VALUE> и поддерживают функцию множественного соединения TCP. Сеть разделена на три режима: AP, STA и AP+STA. Эта конструкция использует режим AP+STA. Этот метод не только позволяет напрямую подключиться к системе через компьютер, но и позволяет удаленно управлять различными операциями через беспроводной маршрутизатор, без необходимости повторения ручных операций на месте, что удобно и быстро.

Все датчики системы представляют собой недорогие миниатюрные датчики с низким энергопотреблением, что значительно снижает затраты. Система разработана для обеспечения долгосрочной стабильной работы, непрерывной и эффективной работы в различных условиях окружающей среды, включая закрытые помещения или особые экстремальные условия. Это решает проблему несбалансированных точек отбора проб в традиционных системах мониторинга.

Замена существующего ручного мониторинга на местах для обеспечения доступа пользователей к экологическим данным в реальном масштабе времени через сеть. Таким образом, сотрудники могут контролировать и управлять окружающей средой в режиме реального времени с помощью

клиентских апплетов, что позволяет одному человеку контролировать несколько экологических условий одновременно, что значительно повышает эффективность работы и значительно снижает трудоемкость операторов и повышает точность сбора данных об окружающей среде. Он решает утомительные рабочие этапы традиционного сбора данных и устраняет опасность сбора данных в чрезвычайно суровых условиях. В то же время это снижает стоимость сбора и упрощает промышленное и сельскохозяйственное производство. Используя STM32 в качестве основного контроллера и протокол MQTT, можно обеспечить эффективную, надежную передачу данных в реальном времени на платформе ONENET. Инструменты разработчика WeChat используются для разработки апплетов взаимодействия человека с компьютером, а для реализации связи между ними используется сетевая связь сервера и клиента, что делает его Главным управляющим компьютером, серверная и клиентская стороны имеют функцию мониторинга данных об окружающей среде в режиме реального времени. Сочетая микроконтроллер STM32 и облачную платформу ONENET, осуществляется сбор данных об окружающей среде и облачный анализ, что обеспечивает надежную поддержку будущих приложений Интернета вещей и реализует сбор, анализ и передачу данных об окружающей среде. Система обладает хорошей стабильностью и высокой масштабируемостью.

## 2 РЕШЕНИЕ АЛГОРИТМА СБОРА ДАННЫХ И РАЗРАБОТКА ПРОГРАММНОГО И АППАРАТНОГО ОБЕСПЕЧЕНИЯ

### 2.1 Предлагаемый компьютерный алгоритм для решения процесса сбора данных

Алгоритм работы программного модуля мониторинга экологических данных включает следующие основные шаги:

- сбор данных. Модуль подключается к источникам информации, таким как датчики, базы данных или API внешних служб. Получение данных об окружающей среде (температура, влажность, качество воздуха, уровень загрязнения и др.) происходит регулярно или по запросу;

- обработка данных. Входящие данные проходят предварительное преобразование и фильтрацию – удаляются выбросы и ошибочные значения. Затем осуществляется анализ информации: вычисление средних значений, формирование статистических показателей, прогнозирование на основе накопленных данных и т. д.;

- хранение данных. Обработанная информация сохраняется в базе данных или в файлах (например, в формате JSON или в реляционной СУБД) для последующего использования и анализа. Это обеспечивает устойчивое хранение данных, возможность их централизованного доступа, а также интеграцию с другими модулями системы. Хранимые данные могут использоваться для построения отчетов, обучения моделей машинного обучения, проведения ретроспективного анализа, а также для выявления долгосрочных экологических тенденций и прогнозирования изменений в окружающей среде;

- представление данных. Результаты анализа и собранные показатели визуализируются в графическом интерфейсе пользователя. Это обеспечивает наглядное и удобное отображение текущего состояния окружающей среды;

– визуализация алгоритма. Для отображения последовательности действий используется блок-схема, как показано на рисунке 6.



Рисунок 6 – Блок-схема сбора данных

Для организации данных потребуется создать структуру, которая будет эффективно поддерживать как сбор, так и обработку данных. Рассмотрим следующие ключевые компоненты:

ER-диаграмма. Для структуры данных можно использовать ER-диаграмму для базы данных. Как показано ниже Рисунок 7. Основные сущности могут быть следующими:

Sensors (Датчики): информация о датчике (ID, тип, местоположение, дата установки).

Data (Данные): собранные значения (ID, sensor\_id, timestamp, значение показателя).

Alerts (Уведомления): события, когда показатели превышают допустимые нормы (ID, timestamp, sensor\_id, превышение нормы).

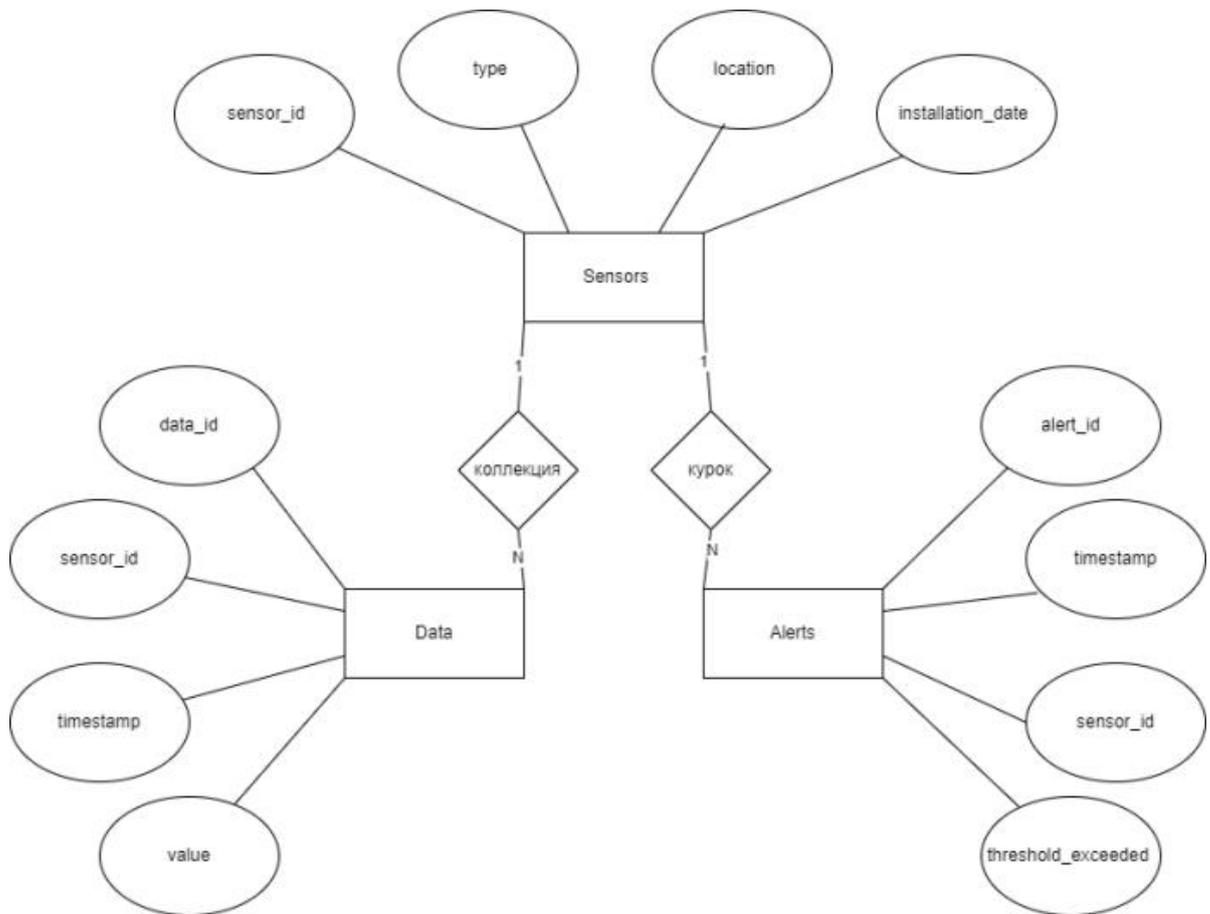


Рисунок 7 – Диаграмма базы данных ER

Обоснование структуры данных:

– Sensors (Датчики): каждый датчик имеет уникальный идентификатор, тип (например, датчик температуры, датчик загрязнения), местоположение и дату установки. Это необходимо для отслеживания информации о каждом источнике данных;

– Data (Данные): каждое значение данных привязано к конкретному датчику и времени сбора. Это нужно для того, чтобы хранить точную информацию о показаниях в динамике и использовать её для анализа;

– Alerts (Уведомления): генерируются, когда данные превышают заданные пороги или когда есть критические изменения, которые требуют внимания. Это важный элемент для мониторинга состояния экологии в реальном времени.

Общая структура программы разделена на следующие модули. Каждый модуль имеет независимые функции и работает вместе для сбора и управления данными.

Модуль сбора данных (реализован на языке C): модуль отвечает за считывание данных с датчиков окружающей среды. Обычно устройство сбора данных подключается к чипу STM32 и взаимодействует с компьютером через интерфейсы связи, такие как последовательный порт, I2C или SPI.

Используйте язык C для взаимодействия с оборудованием на нижнем уровне, чтобы обеспечить сбор данных в реальном времени. Программа на языке C может получить измеренное значение датчика, читая интерфейс датчика, и выполнить необходимое преобразование единиц измерения и очистку данных.

Модуль хранения и управления данными (реализация облачной платформы): облачная платформа хранит данные в базе данных. Модуль отвечает за получение загруженных данных из модуля сбора данных, их хранение и предоставление функций запроса и поиска данных.

Облачная платформа предоставит API для вызовов клиентских апплетов WeChat для запроса данных в реальном времени, получения исторических данных и т. д.

На облачной платформе могут выполняться такие задачи, как анализ данных, моделирование и оповещение, а клиенту может быть отправлена обратная связь.

Модуль пользовательского интерфейса (реализованный апплетом WeChat): пользователи взаимодействуют с системой через интерфейс апплета WeChat. Апплет получает данные, отправляя запросы к API облачной платформы, и представляет данные пользователю.

Этот модуль в основном обрабатывает отображение и представление данных и реагирует на действия пользователя, вызывая интерфейс, предоставляемый облачной платформой.

Мини-программа также может отправлять пользователям информацию о тревогах через механизм push, чтобы напомнить пользователям об аномальных параметрах окружающей среды.

Модуль передачи данных и синхронизации:

Этот модуль отвечает за обеспечение синхронизации данных между клиентом и облачной платформой. При передаче данных используется протокол HTTPS для обеспечения безопасности и целостности данных.

Загрузка и скачивание данных передаются в формате JSON, что делает обмен данными более кратким и простым для анализа. Диаграмма деятельности выглядит следующим образом Рисунок 8.

В данной системе используются различные типы сенсорных модулей для сбора информации об окружающей среде, включая параметры такие как температура, влажность, концентрация дыма и другие. Поскольку каждый датчик имеет свои особенности работы и формат выходных данных, для каждого из них требуется индивидуальная настройка программного обеспечения, обеспечивающая корректную интерпретацию и обработку получаемой информации.

Программное обеспечение разработано в среде Keil uVision5 с использованием языка программирования C и ориентировано на микроконтроллер STM32. Реализованные модули обеспечивают выполнение основных функций: инициализация датчиков, периодический сбор данных, фильтрация и предварительная обработка, объединение информации от различных источников, а также передача результатов на сервер или клиентское устройство через предусмотренные интерфейсы связи.

Ниже представлено описание программной реализации взаимодействия с каждым из подключённых датчиков.

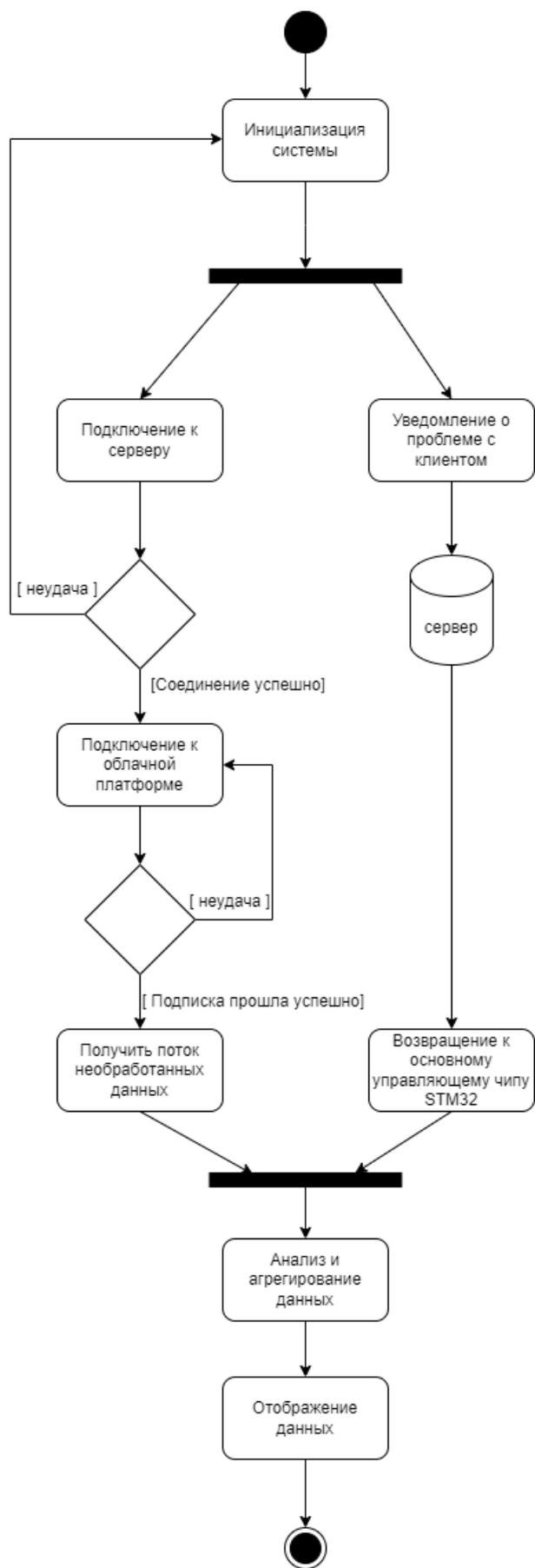


Рисунок 8 – Диаграмма деятельности

Данная система управляется путем подключения модуля датчика температуры и влажности DHT11 через контакт 8 GPIOA. Конкретный процесс сбора данных о температуре и влажности выглядит следующим образом: после того, как процессор посылает сигнал запуска, модуль температуры и влажности начинает работу, посылает ответный сигнал и отправляет пакет данных, представляющий температуру и влажность. Получив эту часть данных, процессор может использовать ее в соответствии с требованиями пользователя и преобразовывать связанные данные, такие как данные о температуре и влажности с десятичными знаками и данные о температуре и влажности без десятичных знаков. Конкретный процесс сбора данных о температуре и влажности показан на рисунке 9.

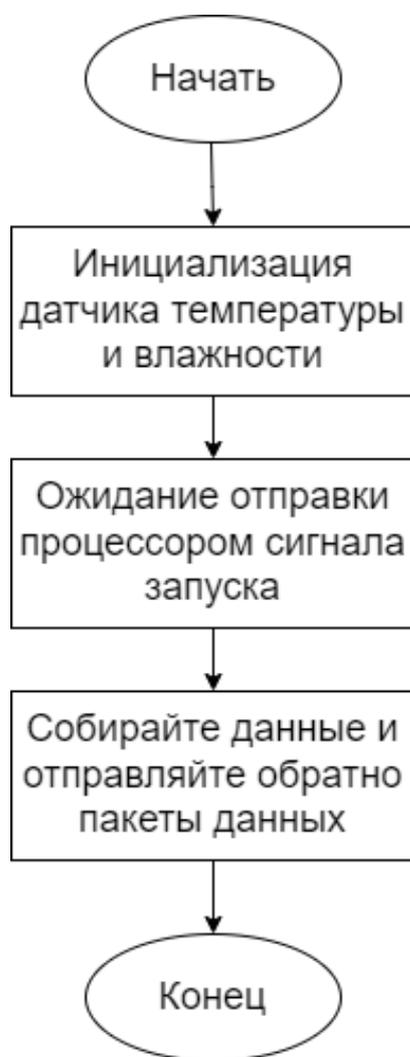


Рисунок 9 – Схема сбора данных датчика DHT11

Система легко получает 16-битные цифровые данные об освещении

через интерфейс шины I<sup>2</sup>C. Конкретный процесс внедрения показан на рисунке 10.



Рисунок 10 – Блок-схема сбора данных модуля фотодатчика BH1750

Основные процедуры, связанные со сбором данных об интенсивности света, следующие:

```
float LIght_Intensity(пусто)
{
return (float)(BH1750_Read_Measure()/1.1f*Resolurtion);
}
```

Сбор данных и рабочий процесс датчика дыма MQ-2 показаны на рисунке 11.



Рисунок 11 – Схема сбора данных модуля датчика дыма

Конкретная реализация кода модуля MQ-2 выглядит следующим образом:

```

if(timeCount % 40 == 0){
Ppm =(float) ADC_ConvertedValue*1000/4096*3.3;
UsartPrintf(USART_DEBUG,"Текущая концентрация: %.1f ppm\r\n",
pp m);
}
if(alarm_is_free == 10)
{
if((humidityH < 90) && (temperatureH < 32) && (Light < 10000) &&
(Ppm<2000))alarmFlag = 0;
else alarmFlag = 1;
}
  
```

```
if(alarm_is_free < 10)alarm_is_free++;
```

После включения системы необходимо настроить модуль ESP8266 для работы в сети. Если есть информация о беспроводной сети, устройство подключится к текущей беспроводной сети и установит сервер соединения TCP/IP. Конкретный процесс внедрения показан на рисунке 12.

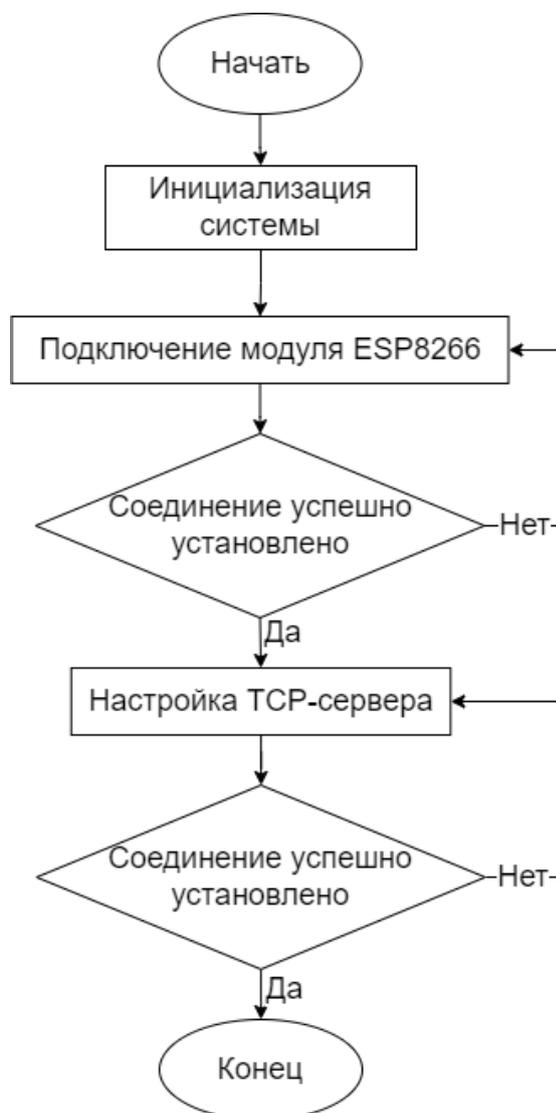


Рисунок 12 – Схема работы данных модуля ESP8266

Основная программа разработана и реализована. Сначала система инициализируется. После входа в режим AP+STA модуль ESP8266 настраивается и затем подключается к серверу. Каждый датчик собирает данные и отображает их на дисплее. В то же время собранные данные сравниваются. Если значение данных превышает установленный порог, будет сгенерирована инструкция по эксплуатации и отправлена в исполнительный модуль,

а данные будут переданы на сервер через модуль ESP8266. Данные можно получить с сервера через апплет WeChat и отобразить на странице клиента. Кроме того, можно вручную выдавать инструкции для реализации таких операций, как включение и выключение света и включение сигнализации. Основной процесс разработки программы показан на рисунке 13.

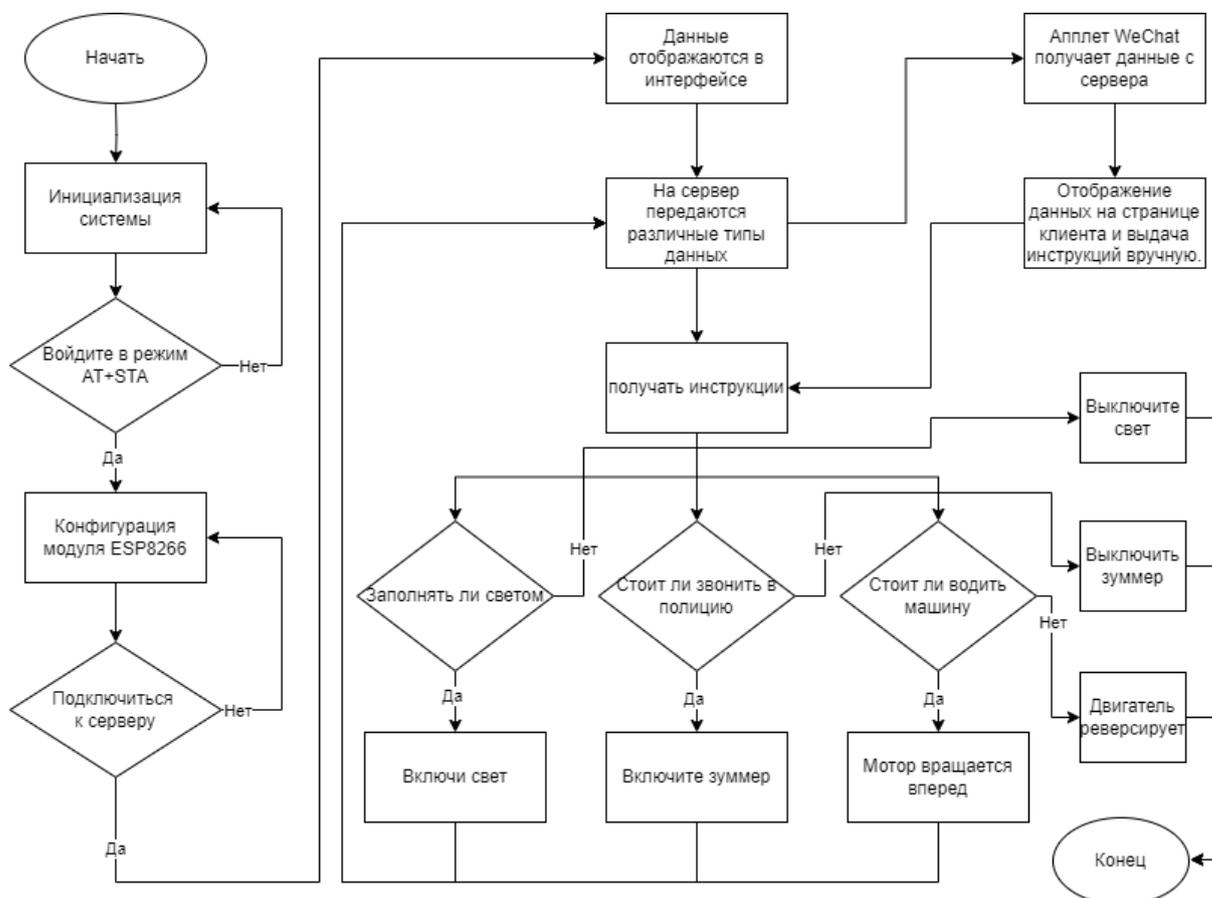


Рисунок 13 – Основная схема программы

Проектирование и реализация клиентской разработки. Клиент использует инструменты разработчика WeChat для разработки апплета WeChat, который отображает собранные данные и управляет модулем звуковой и световой сигнализации. После подключения к серверу MQTT подпишитесь на тему данных восходящего канала облачной платформы, чтобы получать сообщение (шестнадцатеричный поток байтов буфера), а затем выполните синтаксический анализ JSON для отображения соответствующей информации о данных. Клиент отправляет тему команды на облачную платформу, которая затем возвращается на главный управляющий чип STM32 для

управления соответствующим модулем. Процесс показан на рисунке 14.

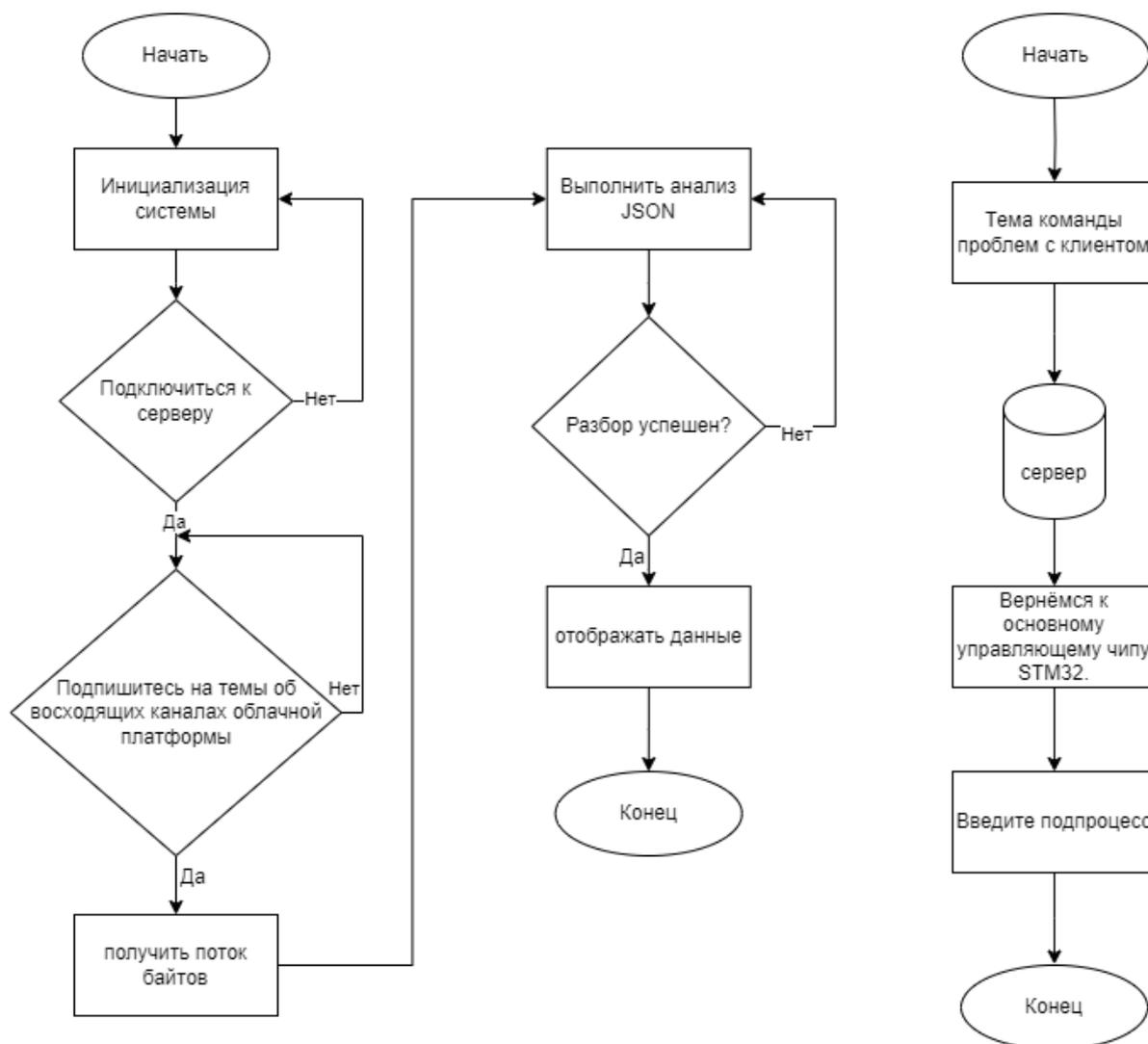


Рисунок 14 – Схема программы WeChat Mini

Экран данных отображает информацию от датчиков в реальном времени. На экране можно видеть температуру, влажность, уровень освещённости, наличие движения и другие параметры, фиксируемые системой мониторинга. Данные обновляются автоматически с заданной периодичностью, что позволяет пользователю получать актуальную информацию без задержек.

Экран отчета показывает текущее состояние среды на основе собранных и обработанных данных. Здесь представлены усреднённые значения за определённые промежутки времени, зафиксированные отклонения от

нормы и возможные рекомендации по регулированию условий. Этот экран помогает анализировать информацию и принимать обоснованные решения.

Командный переключатель используется для управления исполнительными устройствами. С его помощью можно включать и выключать освещение, а также активировать или деактивировать тревожную кнопку. Интерфейс управления выполнен в виде понятных элементов – тумблеров и кнопок с визуальной индикацией состояния.

Интерфейс пользовательского апплета представлен на рисунке 15. Он выполнен в лаконичном и понятном стиле, обеспечивающем быструю навигацию между экранами и лёгкость взаимодействия с системой. Все элементы расположены удобно, чтобы пользователь мог управлять системой без специальных навыков.



Рисунок 15 – Интерфейс пользовательского апплета

Откройте приложение WeChat, чтобы просмотреть собранные данные об окружающей среде в режиме реального времени, и используйте кнопки ниже, чтобы включать и выключать свет и звуковую сигнализацию.

## 2.2 Обзор функциональной конструкции программного обеспечения для сбора данных об окружающей среде

Программное обеспечение для сбора экологических данных предназначено для сбора и анализа данных об окружающей среде в режиме реального времени с помощью множества датчиков и оборудования для мониторинга, таких как качество воздуха, температура и влажность, интенсивность освещения и т. д. Для обеспечения эффективности и надежности системы была выбрана многопользовательская архитектура, то есть модель клиент-сервер, обеспечивающая одновременный доступ и операции с данными несколькими пользователями. В соответствии с потребностями программного обеспечения для сбора экологических данных была выбрана архитектура клиент-сервер, которая представляет собой типичную многопользовательскую архитектуру. В клиент-серверной архитектуре система разделяет функции на клиентскую и серверную части через сеть:

- клиент. Пользовательский операционный интерфейс, отвечающий за запрос данных с сервера и отображение результатов, а также предоставляющий пользователям интуитивно понятный операционный интерфейс для просмотра и анализа данных об окружающей среде;

- сервер. Облачный сервер используется для хранения, управления и обработки данных, а также обеспечивает функции хранения данных, вычислений и реагирования.

Эта архитектура имеет следующие преимущества:

- распределённые вычисления. Сервер может централизованно управлять данными, снижать нагрузку на клиента и повышать эффективность вычислений;

- многопользовательская поддержка. Несколько клиентов могут одновременно подключаться к серверу для запроса данных, анализа и управления;

- высокая масштабируемость. Ресурсы сервера можно расширять в соответствии с потребностями системы, например, путём увеличения объёма

хранилища или вычислительной мощности;

– надёжность. Система способна стабильно функционировать при одновременной работе большого количества клиентов, предотвращая потерю или повреждение данных;

– высокая доступность. Стабильность работы обеспечивается за счёт использования механизмов балансировки нагрузки и восстановления после сбоев;

– масштабируемость. Система поддерживает гибкое расширение функциональности, включая подключение новых сенсорных модулей и компонентов пользовательского интерфейса;

– безопасность. Передаваемые данные защищены с помощью механизмов шифрования и аутентификации, что гарантирует конфиденциальность и целостность пользовательской информации.

Согласно анализу спроса на программное обеспечение для сбора экологических данных, как показано в таблице 1 ниже, функции системы можно разделить на несколько модулей, каждый из которых выполняет различные задачи.

Таблицы 1 – Функциональные требования к сбору экологических данных

Управление пользователями	Регистрация пользователя и вход в систему. Поддержка регистрации и входа пользователя.
1	2
Сбор данных	Интерфейс датчика: поддерживает доступ и сбор данных от нескольких датчиков (температуры и влажности, качества воздуха, освещения и т. д.). Сбор данных в реальном времени: он может собирать данные об окружающей среде в режиме реального времени и поддерживает пакетный сбор данных.
Сбор данных	Проверка данных: проверьте достоверность и точность собранных данных.

Хранение данных	Управление базой данных: хранение собранных данных в базе данных.
Визуализация данных	Панель мониторинга: пользователи могут настроить панель мониторинга для мониторинга ключевых показателей в режиме реального времени.
Уведомления и оповещения	Настройка порога: пользователи могут устанавливать пороги мониторинга данных. Механизм сигнализации: как только данные превышают установленный порог, система автоматически отправляет уведомление.
Управление системой	Настройки системы: администраторы могут настраивать параметры системы, такие как интервалы сбора данных, пороговые значения сигналов тревоги и т. д.

Ниже представлено описание основных функциональных модулей системы мониторинга окружающей среды. Каждый модуль выполняет конкретные задачи в рамках общей архитектуры, обеспечивая стабильную и эффективную работу системы.

Модуль сбора данных. Отвечает за сбор информации об окружающей среде в режиме реального времени с помощью подключённых датчиков. Поддерживаются параметры, такие как температура, влажность, качество воздуха, уровень освещённости и другие. Модуль управляет оборудованием, периодически извлекает данные с датчиков и передаёт их на сервер для дальнейшей обработки.

Модуль хранения данных. Обеспечивает приём, долговременное хранение и резервное копирование собранных данных на сервере. Поддерживает функции работы с историческими данными, включая их сжатие, дубликацию и предоставление по запросу.

Модуль обработки и анализа данных. Выполняет анализ полученной информации в режиме реального времени. Поддерживает функции прогнозирования изменений параметров окружающей среды, обнаружения аномалий, формирования статистических отчётов и генерации предупреждений.

Модуль отображения данных. Отвечает за визуализацию информации на стороне клиента. Обеспечивает отображение данных в виде графиков и диаграмм, как в реальном времени, так и в ретроспективе. Реализована возможность просмотра отчётов и анализа динамики показателей.

Модуль тревожных сигналов и уведомлений. Автоматически формирует оповещения в случае превышения установленных пороговых значений параметров окружающей среды. Уведомления могут быть доставлены ответственному персоналу через различные каналы – например, в виде SMS-сообщений или электронных писем.

Модуль управления конфигурацией. Позволяет настраивать параметры системы, включая список используемого сенсорного оборудования, правила хранения и обработки данных, пороговые значения для оповещений и другие системные настройки.

Модуль обновления системы. Поддерживает управление обновлениями программного обеспечения, включая установку новых версий, исправлений и обновлений безопасности. Обеспечивает актуальность и защищённость всей системы.

Диаграмма вариантов использования, отражающая структуру и взаимодействие указанных модулей, представлена на рисунке 16.



Рисунок 16 – Диаграмма вариантов использования системы

По характеристикам системы сбора экологических данных в качестве модели жизненного цикла была выбрана инкрементальная модель (Incremental Model).

Инкрементная модель делит всю систему на несколько модулей (инкрементов), которые постепенно разрабатываются и поставляются:

Каждое приращение содержит часть функциональности, которую можно запускать независимо и выполнять свою конкретную функциональность до того, как программное обеспечение будет запущено в эксплуатацию.

В процессе разработки программного обеспечения можно постепенно получать готовые функциональные модули и обеспечивать обратную связь, чтобы обеспечить связь между разработкой системы и потребностями пользователя.

Этот метод подходит для разработки программного обеспечения для сбора данных, поскольку программное обеспечение можно внедрять поэтапно. Основные функции сбора и отображения могут быть реализованы на начальном этапе, а более сложные модули анализа данных и сигнализации могут быть добавлены позже.

Инкрементная модель разработки обладает рядом преимуществ, среди которых особенно выделяются высокая гибкость и диверсификация рисков. Благодаря поэтапному подходу к созданию системы, разработчики могут своевременно вносить изменения в план работ в соответствии с изменяющимися требованиями и отзывами пользователей, что делает процесс адаптивным и более ориентированным на конечный результат. Кроме того, инкрементальная реализация функциональности позволяет значительно снизить риск полного провала проекта: даже в случае возникновения проблем на одном из этапов остальная часть системы продолжает функционировать, обеспечивая устойчивость и стабильность процесса разработки.

Пользователь запускает программу сбора данных и попадает в основной интерфейс. Пользователь дает команду через интерфейс, чтобы начать сбор данных. Программное обеспечение для сбора данных отправляет запрос на устройство сбора данных для получения текущих данных о среде, а затем загружает данные на облачную платформу для анализа и хранения в базе данных. Облачная платформа отправляет данные на устройство сбора данных и возвращает собранные данные об окружающей среде. Интерфейс программного обеспечения для сбора данных отображает информацию о данных.

В то же время программное обеспечение определяет, превышают ли данные установленный порог, отправляет команду тревоги на облачную платформу, запускает модуль сигнализации и возвращает сообщение с напоминанием об успехе. отображается в интерфейсе программного обеспечения. Временная диаграмма программного обеспечения представлена ниже Рисунок 17.

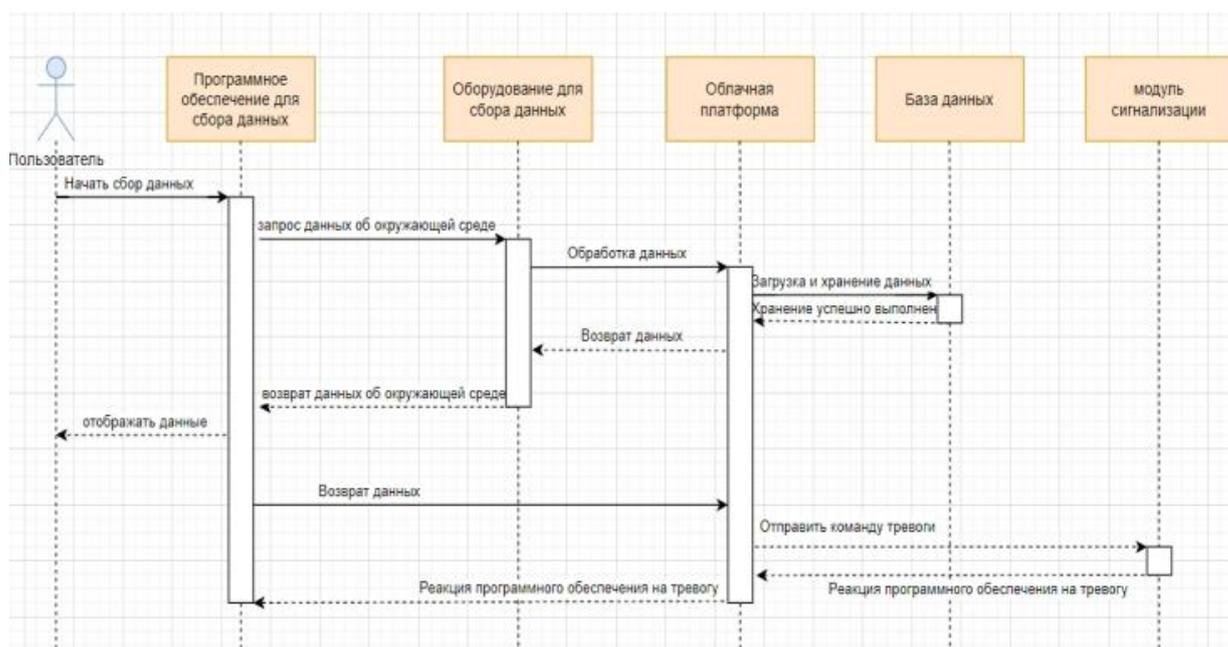


Рисунок 17 – Временная диаграмма программного обеспечения

Статус программного обеспечения в процессе сбора данных включает несколько этапов, каждый из которых выполняет определённую роль в обеспечении эффективной работы системы.

На начальной стадии, в состоянии ожидания, программное обеспечение находится в режиме по умолчанию после запуска. На этом этапе пользователи могут запустить сбор данных или просматривать историческую информацию. Важным действием на этом этапе является проверка состояния соединения и подготовка к запуску процесса сбора.

Далее система переходит в состояние сбора данных, в котором она начинает собирать данные с датчиков в режиме реального времени, включая параметры, такие как температура, влажность, качество воздуха и другие. В это время происходит регулярное обновление кэша данных в реальном времени, а также передача собранных данных в модуль обработки через определённые промежутки времени, например, каждую секунду. Важно отметить, что в любое время система может перейти в состояние исключения, если возникнут непредвиденные обстоятельства.

После завершения сбора данных система переходит в состояние хранения данных, где собранная информация записывается в базу данных или файловую систему. Здесь происходит проверка целостности данных, а также их запись в пакетном режиме для повышения эффективности хранения. Важно зафиксировать время хранения и статус данных.

Следующим этапом является обработка данных, когда система анализирует сохранённые данные и создаёт статистические отчёты. На этом этапе выполняется стирание недействительных данных, что способствует поддержанию качества информации.

После выполнения всех задач обработки и хранения данных система переходит в состояние окончания, где все процессы завершены, и система готова вернуться в режим ожидания. На этом этапе обновляется пользовательский интерфейс, чтобы отобразить актуальную информацию.

В случае возникновения ошибок во время сбора данных, таких как сбой датчика или потеря данных, система переходит в состояние обработки исключений. В этом состоянии система предлагает пользователю проверить устройство, а также автоматически повторно инициирует сбор данных или

сохраняет действительные на данный момент данные. Диаграмма состояния сбора данных представлена на рисунке 18.

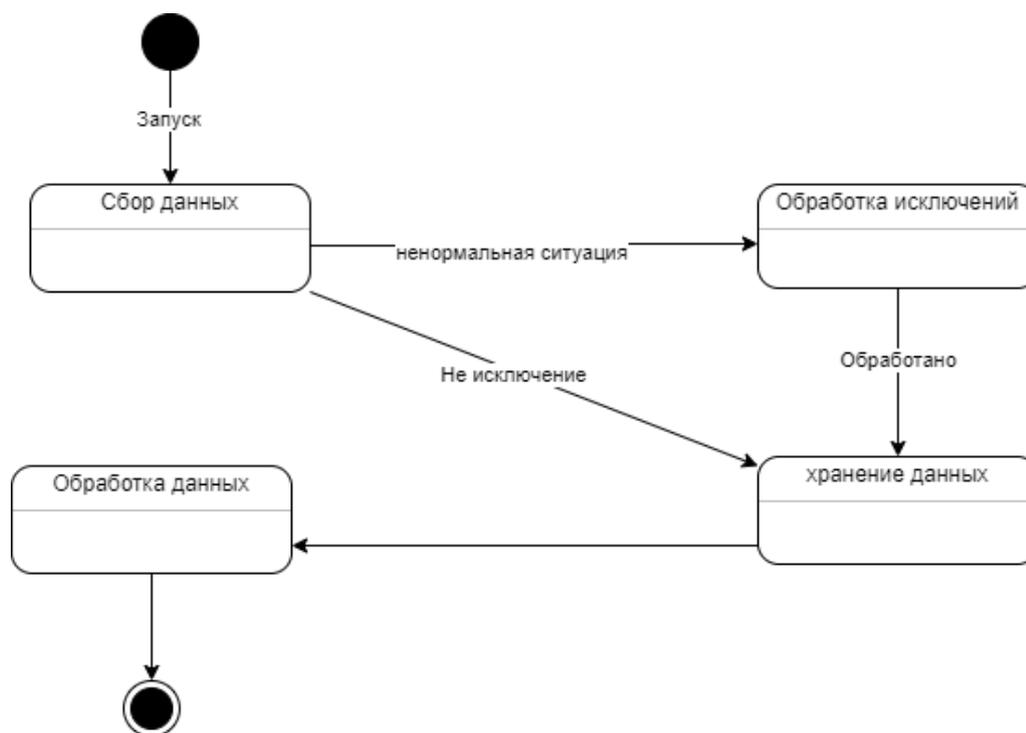


Рисунок 18 – Диаграмма состояния сбора данных

Переход состояний в системе происходит следующим образом. Из режима ожидания в режим сбора данных система переходит, когда активируется событие начала сбора данных. В этот момент программное обеспечение и оборудование инициализируются, и начинается процесс сбора информации с датчиков. Когда сбор данных завершается успешно, система переходит от сбора данных к их хранению. На этом этапе собранные данные сохраняются в режиме реального времени. После завершения хранения данных начинается следующий этап – обработка данных. На этом этапе выполняется анализ собранной информации.

По завершении обработки данных система переходит к состоянию завершения, что сигнализирует о завершении всех этапов и готовности вернуться в режим ожидания. Если в процессе работы системы возникает ошибка, независимо от текущего состояния, система переключается в режим обработки исключений. В этом режиме происходит обработка ошибок и попытка восстановить нормальное функционирование системы.

### 2.3 Разработка характеристик оборудования, выбранного для системы сбора данных

Система объединяет сбор, отображение и передачу данных с высокой степенью интеллекта, что снижает трудоемкость операторов и повышает точность мониторинга окружающей среды. В качестве центрального процессора регистратора данных используется микроконтроллер STM32F103C8T6. Структура проектирования оборудования показана на рисунке 19. Являясь представителем семейства микроконтроллеров средней плотности STM32- F103xx, STM32F103C8T6 объединяет высокопроизводительное 32-битное RISC-ядро Cortex-M3 на базе ARM, работающее на частоте 72 МГц, высоко- скоростную встроенную память (64 Кбайт флэш-памяти и 20 Кбайт SRAM), внутренний RC-генератор с заводской настройкой 8 МГц и широкий спектр усовершенствованных входов/выходов и периферийных устройств, подключенных к двум шинам APB. Кроме того, STM32F103C8T6 имеет 48 контактов, два 12-разрядных АЦП, 7-канальный контроллер прямого доступа к памяти, семь таймеров и девять интерфейсов связи (включая два I<sup>2</sup>C и два SPI, три USART, один USB и один CAN).



Рисунок 19 – Блок-схема конструкции оборудования

BOOT0 (контакт 44) и BOOT1 (контакт 20) заземлены, что означает, что в обычном режиме загрузки STM32 загружает свою встроенную операционную систему из Flash-памяти. D1 – светодиодный индикатор STM32, а D2 – светодиодный индикатор питания. Схема фильтра питания состоит из C3, C4, ..., C9. Y1 – пассивный кварцевый генератор с двумя конденсаторами емкостью 20 пФ, а X1 – активный кварцевый генератор с частотой 8 МГц.

Чип в основном отвечает за нормальную координацию работы сенсорных модулей нижнего компьютера. STM32 взаимодействует с датчиком BH1750 через шину I<sup>2</sup>C для получения данных об интенсивности света в реальном времени, а также взаимодействует с датчиком DHT11 через одну шину для получения данных о температуре и влажности в реальном времени. Собранные датчиком данные просто обрабатываются и анализируются, а затем собранные данные записываются во Flash-память STM32. Используя STM32 в качестве ядра управления сенсорным узлом, собранный аналоговый сигнал преобразуется в цифровой сигнал посредством аналого-цифрового преобразования для восприятия региональных данных об окружающей среде, таких как температура, влажность, интенсивность света и концентрация дыма.

В системе используется жидкокристаллический OLED-дисплей. Размер ЖК-модуля – 0,96 дюйма. Разрешение составляет 128\*64, используется чип драйвера SSD1306 и метод связи IIC. Модуль, отображающий различные типы данных в системе. Схема этого модуля показана на рисунке 20.

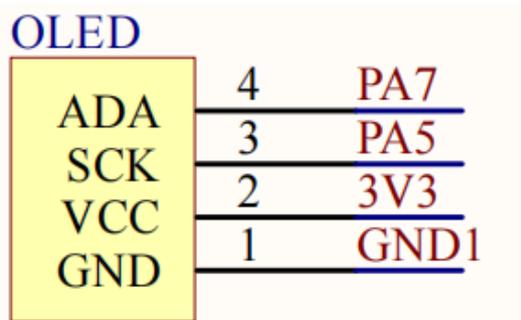


Рисунок 20 – Схема цепи ЖК-дисплея

Исполнительные устройства включают в себя модуль привода двигателя L298N, зуммер и светодиодную подсветку. Когда данные датчика достигают порогового значения, запускаются модуль звуковой и световой сигнализации и двигатель. Одновременно получать инструкции от клиента и выполнять соответствующие операции.

В данной системе используется датчик дыма MQ-2 для определения концентрации дыма в окружающей среде. Датчик обладает высокой чувствительностью к газу, длительным сроком службы и надежной стабильностью. Принцип его работы заключается в том, что при наличии дыма в окружающей среде проводимость датчика увеличивается с ростом концентрации дыма в воздухе, тем самым определяя концентрацию дыма в текущей окружающей среде. Простая схема используется для вывода высоких и низких электрических уровней и установки порогового значения, превышающего стандарт. Если выходной сигнал выше, это означает, что концентрация превышает норму. Схема этого модуля показана на рисунке 21.

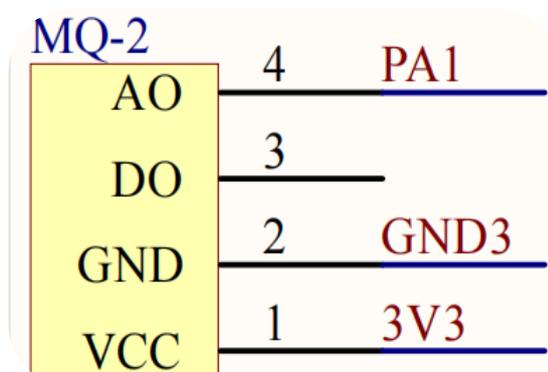


Рисунок 21 – Схема модуля датчика дыма MQ-2

Датчик дыма MQ-2 в системе отвечает за определение того, превышает ли концентрация дыма в окружающей среде норму.

Большинство сенсорных модулей выдают относительно слабые сигналы, и датчики дыма не являются исключением. Поэтому выходной сигнал должен обрабатываться набором схем для удовлетворения требований процессора по обнаружению входного сигнала. В системе используется неинвертирующая пропорциональная схема усилителя, что позволяет сократить

расширение оборудования и достичь наивысшей производительности по стоимости.

Цифровой датчик температуры и влажности DHT11 представляет собой комбинированный датчик температуры и влажности с калиброванным цифровым выходным сигналом. Он использует конструкцию с единым форматом данных шины и выполняет передачу данных каждые 4 миллисекунды. Разделяет строковые данные на десятичную часть и целую часть. Конкретный формат данных, следующий: первые 8 цифр – целая часть влажности, а последние 8 цифр – десятичная часть влажности. Следующие восемь цифр представляют целую часть температуры, следующие восемь цифр представляют десятичную часть температуры, а последние восемь цифр – данные калибровки. Система точна только до целой части температуры и влажности. Он обладает такими преимуществами, как низкая стоимость, высокая стабильность, быстрый отклик, высокая помехоустойчивость, высокая точность измерений (температура  $\pm 0,2$  °C, влажность  $\pm 2$  % относительной влажности) и т. д. Он подходит для мониторинга окружающей среды. Схема сенсорного модуля показана на рисунке 22.

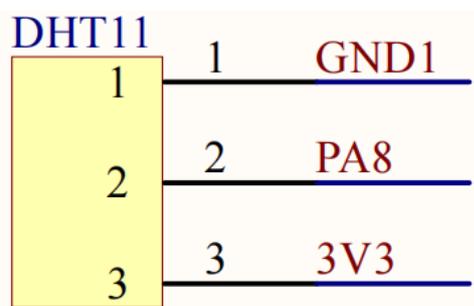


Рисунок 22 – Схема модуля датчика температуры и влажности

Сбор данных об окружающей среде требует сбора данных о температуре и влажности окружающей среды. Эта система использует цифровой датчик температуры и влажности DHT11 для сбора данных о температуре и влажности и получает необходимые данные в соответствии с протоколом передачи посредством временного мультиплексирования (TDM).

Модуль включает в себя резистивный датчик влажности и элемент измерения температуры NTC, подключенные к высокопроизводительному 8-битному микроконтроллеру. Чувствительный слой резистивного датчика влажности может быть поврежден химическими парами, а диффузия химикатов в чувствительном слое может привести к дрейфу измерений и снижению чувствительности. Фактическое оборудование показано на рисунке 23.

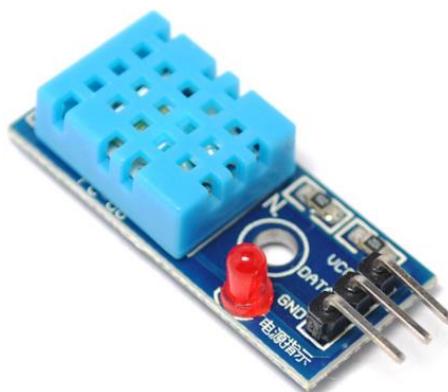


Рисунок 23 – Физическое изображение датчика DHT11

Как показано на рисунке 24, для проектирования схемы сбора данных об интенсивности света используется цифровой фотодатчик BH1750FVI. Благодаря таким преимуществам BH1750FVI, как высокое разрешение, широкий диапазон обнаружения (1 - 65535 люкс), спектр, близкий к человеческому глазу, малая зависимость от источника света и малое влияние инфракрасного излучения, 16-битные цифровые данные об освещенности можно легко получить через интерфейс шины I<sup>2</sup>C.

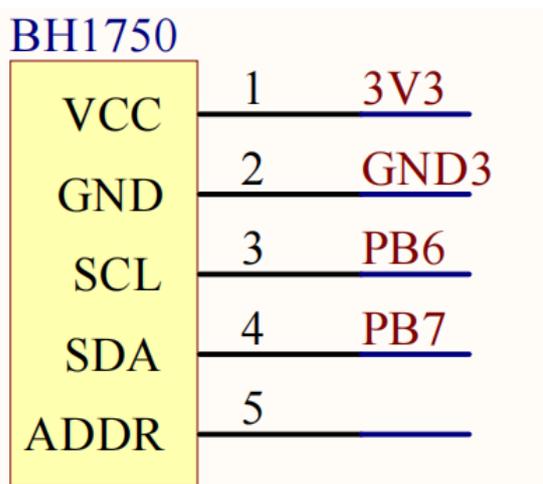


Рисунок 24 – Схема фоточувствительного датчика

При использовании данного датчика следует обратить внимание на его помехоустойчивую конструкцию, обеспечивающую подавление шума промышленной частоты 50 Гц и 60 Гц, а также снижение оптических помех от источников света, таких как люминесцентные и светодиодные лампы. Это позволяет обеспечить стабильную и точную работу в условиях, близких к реальной промышленной эксплуатации.

Для повышения энергоэффективности рекомендуется оптимизировать энергопотребление датчика. В частности, следует активировать спящий режим, позволяющий снизить потребляемый ток до уровня порядка 0,1 мкА в режиме ожидания. Это особенно важно при использовании датчика в автономных системах, питающихся от батарей или других источников с ограниченной ёмкостью.

Также важно оптимизировать диапазон измерений в зависимости от конкретных условий окружающей среды. Это можно сделать путём настройки апертуры входа света, что позволит контролировать уровень поступающего излучения, или установкой оптических фильтров, ограничивающих спектральный диапазон и уменьшающих влияние фонового освещения.

Дополнительно рекомендуется:

- учитывать температурные и влажностные параметры окружающей среды при калибровке устройства, так как они могут влиять на точность измерений;
- использовать экранирование или надёжное заземление, если датчик применяется в условиях высокой электромагнитной активности;
- производить периодическую проверку и перекалибровку датчика, чтобы сохранить точность измерений в течение всего срока эксплуатации;
- эти меры позволят обеспечить надёжную и долговечную работу устройства в составе измерительных и управляющих систем.

### 3 РЕШЕНИЕ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА СБОРА ДАННЫХ СРЕДЫ ОБЛАЧНОЙ ПЛАТФОРМЫ

#### **3.1. Основные этапы практического развития систем сбора данных об окружающей среде**

В этой главе будет подробно представлена реализация программного обеспечения для сбора экологических данных. Система в основном состоит из трех частей: клиентский интерфейс (разработанный с использованием апплета WeChat), программа сбора данных (разработанная с использованием языка C) и модуль хранения и обработки данных облачной платформы.

Чтобы обеспечить эффективность, масштабируемость и совместимость с существующей технической архитектурой программного обеспечения для сбора экологических данных, в процессе разработки были выбраны следующие технологии и инструменты.

**Язык C:** часть сбора данных разработана с использованием языка C. Язык C обладает эффективной производительностью и особенно подходит для обработки аппаратного интерфейса и задач сбора данных. Его возможности работы на низком уровне позволяют напрямую взаимодействовать с аппаратными устройствами (такими как датчики, контроллеры) для обеспечения сбора данных в режиме реального времени и точности.

**Апплет WeChat:** в качестве клиентского интерфейса он разработан с использованием апплета WeChat. Мини-программы WeChat могут обеспечить удобный пользовательский интерфейс, широкий охват пользователей и простоту использования. Благодаря встроенным кроссплатформенным функциям пользователи могут использовать систему на различных мобильных устройствах, а разработка и обслуживание относительно просты.

**Облачная платформа:** часть хранения и управления данными обрабатывается с использованием облачной платформы. Облачная платформа обеспечивает эффективное хранение и вычислительные ресурсы, поддержи-

вая хранение больших объемов данных, обновление данных в реальном времени и доступ к внешним API. Широко используемые платформы облачных сервисов (такие как Tencent Cloud, Alibaba Cloud и т. д.) могут предоставлять эластичные вычисления и системы хранения с высокой доступностью для обеспечения безопасности и надежности сбора данных.

Язык C обеспечивает эффективную базовую поддержку сбора данных; апплет WeChat предоставляет пользователям удобный доступ; облачная платформа гарантирует, что система хранения и управления данными может поддерживать крупномасштабную, высокопараллельную обработку данных об окружающей среде.

Общая структура программы разделена на следующие модули. Каждый модуль имеет независимые функции и работает вместе для сбора и управления данными:

Модуль сбора данных (реализован на языке C) отвечает за считывание данных с датчиков окружающей среды. Обычно устройство сбора данных подключается к чипу STM32 и взаимодействует с компьютером через интерфейсы связи, такие как последовательный порт, I2C или SPI.

Используйте язык C для взаимодействия с оборудованием на нижнем уровне, чтобы обеспечить сбор данных в реальном времени. Программа на языке C может получить измеренное значение датчика, читая интерфейс датчика, и выполнить необходимое преобразование единиц измерения и очистку данных.

Модуль хранения и управления данными (реализация облачной платформы). Облачная платформа хранит данные в базе данных. Этот модуль отвечает за получение загруженных данных из модуля сбора данных, их хранение и предоставление функций запроса и поиска данных.

Облачная платформа предоставит API для вызовов клиентских апплетов WeChat для запроса данных в реальном времени, получения исторических данных и т.д.

На облачной платформе могут выполняться такие задачи, как анализ данных, моделирование и оповещение, а клиенту может быть отправлена обратная связь.

Модуль пользовательского интерфейса (реализованный апплетом WeChat). Пользователи взаимодействуют с системой через интерфейс апплета WeChat. Апплет получает данные, отправляя запросы к API облачной платформы, и представляет данные пользователю.

Этот модуль в основном обрабатывает отображение и представление данных и реагирует на действия пользователя, вызывая интерфейс, предоставляемый облачной платформой.

Мини-программа также может отправлять пользователям информацию о тревогах через механизм push, чтобы напомнить пользователям об аномальных параметрах окружающей среды.

Модуль передачи данных и синхронизации отвечает за обеспечение синхронизации данных между клиентом и облачной платформой. При передаче данных используется протокол HTTPS для обеспечения безопасности и целостности данных.

Загрузка и скачивание данных передаются в формате JSON, что делает обмен данными более кратким и простым для анализа.

С точки зрения общего потока системы, этот проект принимает идею модульного программирования, то есть каждый функциональный модуль проектируется и пишется отдельно, и каждая функциональная подфункция вызывается основной функцией. С точки зрения проектирования программного обеспечения, есть подпрограмма сбора дыма, подпрограмма управления WIFI, подпрограмма сбора света, подпрограмма сбора температуры и влажности, подпрограмма управления светом, подпрограмма управления кнопками, подпрограмма OLED-дисплея, подпрограмма управления шаговым двигателем и т. д. После завершения вышеуказанных функций выполняется отладка последовательного порта на каждом функциональном модуле.

Во время отладки ESP8266, после включения всей системы, программа инициализирует блок, и модуль автоматически подключится к серверу MQTT. Когда помощник по отладке последовательного порта покажет, что соединение успешно установлено, система будет отправлять данные на сервер MQTT каждые 5 секунд.

При подключении ESP8266 к сети ему необходимо выполнить ряд задач по настройке, а именно:

- настройте последовательный порт: ESP8266 и микроконтроллер обмениваются данными с помощью USART2, TXD->PA2RXD->PA3. Ниже приведена процедура настройки;

- инициализация ESP8266: Модуль ESP8266 имеет несколько общих AT-команд для получения необходимой информации. Чтобы общаться с TCP, вам нужно сначала отправить AT-команды. Вам нужно сделать следующее для ESP8266.

Разработчик начинает процесс с отправки AT-команды и должен убедиться, что команда корректна. Затем выполняется мягкий запуск модуля с использованием команды AT+RST, которая возвращает ОК, если все прошло успешно. После этого необходимо настроить модуль в режим AT+CWMODE=1, что переводит его в режим станции, позволяя подключиться к точке доступа WIFI в текущей среде. Далее отправляется команда AT+CWDHCP=1 для активации режима DHCP. Когда это сделано, подключитесь к маршрутизатору, настроив учетные данные для доступа, такие как имя пользователя и пароль. После настройки подключений, с помощью команды AT+CIPSTART устанавливается подключение к TCP-соединению, в результате чего система получает номер порта и IP-адрес. Этот IP-адрес можно преобразовать в зарегистрированное доменное имя, либо непосредственно ввести зарегистрированное доменное имя для подключения.

Когда модуль ESP8266 подключен к Интернету, он может передавать данные с различных датчиков в облако, а апплет хост-компьютера может

извлекать необходимые данные с облачного сервера для получения информации об окружающей среде.

Отправка данных в ESP8266: В процессе передачи данных от микроконтроллера к ESP8266 необходимо отправлять команды в ESP8266 и проверять возвратные инструкции.

Система отлаживается с помощью модуля CP2102. TX и RX подключены к последовательному порту микроконтроллера. Данные, отправленные микроконтроллером, можно увидеть на основе информации, распечатанной отладочным программным обеспечением. Как показано на рисунке 25.

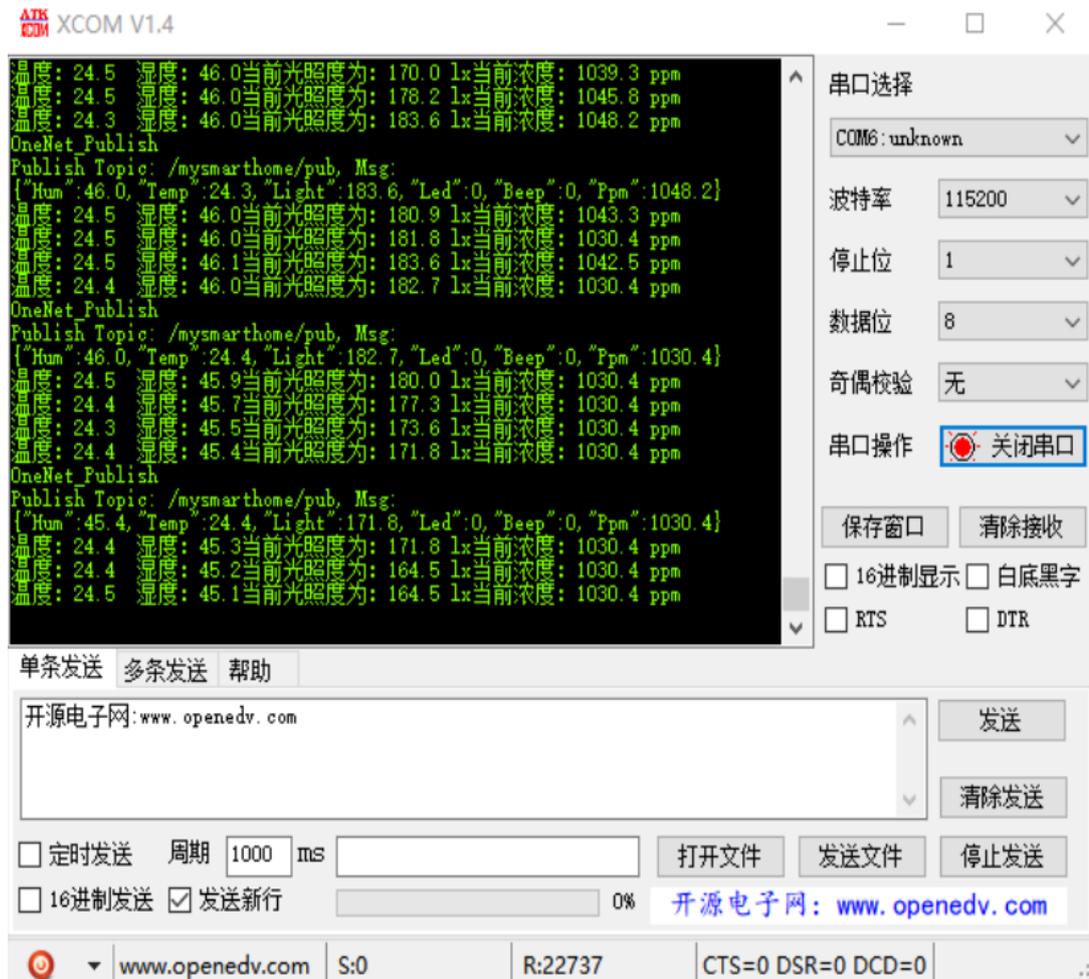


Рисунок 25 – Выход последовательного порта данных датчика

Когда различные датчики работают стабильно, вам необходимо использовать инструмент отладки последовательного порта для получения рабочих данных различных датчиков после обработки микроконтроллером.

Поэтому вам необходимо убедиться, что эти данные нормальные и отправляются каждые 5 секунд перед подключением к Интернету. Вы можете увидеть в инструменте отладки помощника, что данные, загруженные последовательным портом микроконтроллера, полны и необходимы, что означает, что данные различных типов локальных датчиков принимаются нормально. Таким образом, вы можете загрузить эти данные, чтобы убедиться, что загруженные данные являются верными и действительными.

После завершения отладки вам необходимо загрузить данные, использовать последовательный порт STM32 для отправки данных на ESP8266, а затем использовать ESP8266 для отправки данных на сервер MQTT. Когда ESP8266 подключен к серверу, он отправляет сообщение с данными каждые 5 секунд. Если апплет WeChat подписался на тему, то при отправке данных устройством сервер MQTT отправит собранную и обработанную устройством информацию апплету WeChat.

Самая важная часть удаленного беспроводного управления и мониторинга – это загрузка локальных данных на сервер MQTT, чтобы можно было удаленно контролировать и управлять системой. Поэтому беспроводная передача является важной частью всей системы для реализации удаленного управления устройствами.

В процессе исследования установление соединения между апплетом WeChat и STM32 является ключом к тому, можно ли удаленно управлять всей системой. После того, как данные, полученные терминалом апплета WeChat, обрабатываются ESP8266 и сервером MQTT, передаваемые данные используют ESP8266 для подключения к серверу MQTT. Espressif официально предоставила SDK, чтобы разработчики могли подключаться к серверу MQTT. Им нужно только изменить его в соответствии с соответствующими требуемыми функциями. На рисунке 26 показаны примерные отношения передачи данных во всем процессе коммуникации.



Рисунок 26 – Схема структуры принципа связи

Консоль сообщений EMQX IoT MQTT – это станция передачи удаленного управления. В протоколе агент публикует и подписывается на сообщения. Она легкая и имеет простую конструкцию. Характеристики этого протокола:

Протокол TCP/IP для передачи данных друг другу. Стоимость передачи данных низкая, а потребление сетевого трафика невелико. Он может подключаться к удаленным устройствам с минимальным объемом кода и пропускной способностью, что делает его пригодным для встраиваемых систем. С точки зрения сетевых коммуникаций продукта, использование услуг, предоставляемых EMQX, сделает соединение и связь между устройствами и приложениями более стабильными и эффективными. Он может подключать массивные устройства, обрабатывать большие объемы данных и легко реализовывать обработку данных.

В настоящее время двусторонняя передача может быть достигнута между устройствами и программами управления через него. Мало того, двусторонняя связь может быть достигнута также между устройствами. Рабочее состояние устройства может быть получено через апплет WeChat, а апплет WeChat также может отправлять инструкции устройству, чтобы устройство могло реагировать соответствующим образом.

В настоящее время на рынке присутствует множество интернет-ком-

паний, которые могут предоставлять облачные сервисы для создания платформ передачи данных, например: Tencent Cloud Service, предоставляемый Tencent, Alibaba Cloud Service, предоставляемый Alibaba, и т. д. Другие интернет-компании предоставляют аналогичные услуги. Каждая платформа имеет свои преимущества и недостатки. Система использует серверы Alibaba Cloud для создания платформы передачи данных. Шаги, следующие:

– скачать и установить версию EMOX с открытым исходным кодом. Вы можете напрямую использовать команду для ее загрузки, установки и запуска. Необходимо убедиться, что служба EMQX запущена корректно. Затем вы можете перейти в веб-интерфейс EMQX по адресу `http://localhost:18083` для выполнения дальнейшей настройки и мониторинга. По умолчанию логин и пароль администратора: `admin / public`. Также рекомендуется внести изменения в конфигурационные файлы, если требуется настроить сеть, аутентификацию или интеграцию с другими системами. Подробные инструкции доступны в официальной документации EMQX. Шаг показан на рисунке 27;

#### 安装包类型

Ubuntu 24.04 amd64 / deb

- 1 下载 `emqx-5.8.5-ubuntu24.04-amd64.deb` SHA256  

```
wget https://www.emqx.com/zh/downloads/broker/5.8.5/emqx-5.8.5-ubuntu24.04-amd64.deb
```
- 2 安装  

```
sudo apt install ./emqx-5.8.5-ubuntu24.04-amd64.deb
```
- 3 启动  

```
sudo systemctl start emqx
```

Рисунок 27 – Инструкции по загрузке, установке и запуску EMOX

– создание подключение, войти в консоль EMQX и подписаться на темы. Операция показана на рисунке 28;

New Subscription

\* Topic ⓘ

songkejin/pub

\* QoS Color ⌵

0 At most once ⌵ #1AA5F9 ↻

Alias ⓘ

Subscription Identifier

No Local  true  false

Retain as Published  true  false

Retain Handling Select ⌵

Cancel Confirm

Рисунок 28 – Операция подписки на тему

Система EMQX cloud предоставляет множество API, которые разработчики могут использовать, чтобы пользователи могли создавать свои собственные системы управления. В отличие от существующих облачных платформ на рынке, EMQX предоставляет больше интерфейсов обслуживания. Пользователи могут использовать эти интерфейсы данных и интеграцию данных для создания собственных облачных платформ и реализации функций мониторинга и управления продуктами, аналогичных функциям других

компаний. EMQX предоставляет множество функций управления для устройств.

Система отображает информацию о подключенных устройствах, контролирует их рабочее состояние и информирует пользователей о текущем статусе их подключения к сети. Она также поддерживает услугу подписки на темы для устройств, подключенных к интернету. Для нормальной двусторонней связи между прикладным ПО и устройствами необходимо, чтобы они подписались на соответствующие темы. В случае подключения нелегального устройства система может запретить его доступ и исключить из сети.

Для эффективной связи между устройствами и приложениями система предоставляет возможность выполнять статистический анализ и отслеживать темы. В режиме реального времени можно мониторить количество входящих и исходящих сообщений, включая отклоненные сообщения, через платформу.

Когда требуется отладить связь между устройствами и приложениями, существуют два основных способа. Один из них – использование онлайн-инструмента Websocket, предоставляемого EMQX, для отладки связи, а другой – использование MQTT.fx, инструмента для отладки MQTT-сообщений. Оба метода имеют схожие принципы работы: они обеспечивают двухстороннюю связь между устройствами и приложениями, подписываясь на определённые темы и отправляя сообщения по этим темам.

Для настройки связи через WebSocket необходимо указать адрес хоста и порт, а также убедиться, что доменное имя зарегистрировано более трёх месяцев назад. После этого можно проверить соединение с сервером с помощью тестового инструмента. Это позволит удостовериться в правильности взаимодействия устройства с сервером и подготовить приложение WeChat для дальнейшего подключения.

ESP8266 и микроконтроллер взаимодействуют через последовательный порт, а рабочий режим ESP8266 можно установить с помощью команд

последовательного порта. После этого ESP8266 отправит собранную информацию о данных различных датчиков на сервер MQTT, используя сервер MQTT в качестве станции передачи информации. Апплет может получать или выдавать инструкции для управления аппаратным устройством, подключившись к серверу MQTT. Однако перед подключением к серверу апплет WeChat должен знать, являются ли данные, загруженные ESP8266, нормальными, и использовать онлайн-инструмент Websocket или помощник по отладке MQTT.fx для успешного получения загруженных данных и обратного управления устройством с помощью инструкций в формате JSON.

В процессе используются эти два метода, и их общие принципы одинаковы, как указано ниже: Используя помощник по отладке MQTT.fx, вам необходимо ввести адрес сервера, к которому необходимо подключиться. Это разрешенное доменное имя. IP-адрес был разрешен под этим доменным именем. Требуется следующая конфигурация, как показано на рисунке 29.

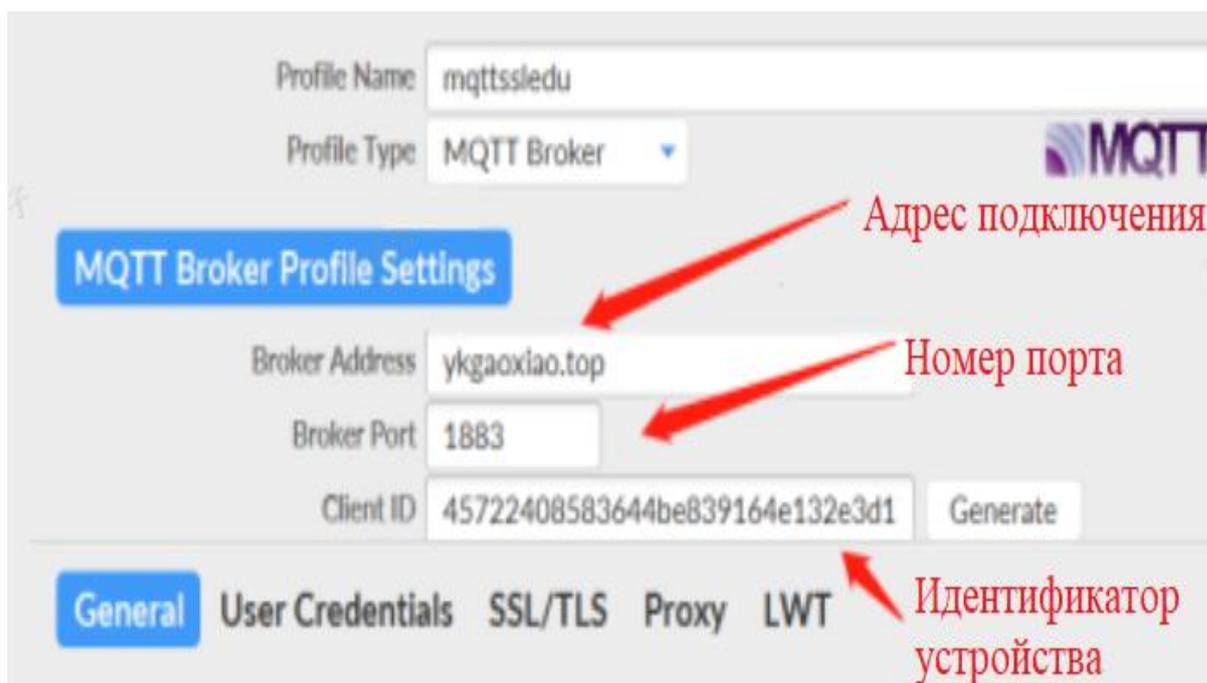


Рисунок 29 – Подключение к серверу

После подключения к серверу инструмент покажет, что соединение успешно установлено. В это время заполните тему подписки в инструменте, и будут отображены данные, загруженные устройством. Данные отправля-

ются каждые 5 секунд и сопровождаются временной меткой. Вы четко увидите информацию о работе устройства, отображаемую в формате JSON, как показано на рисунке 30.

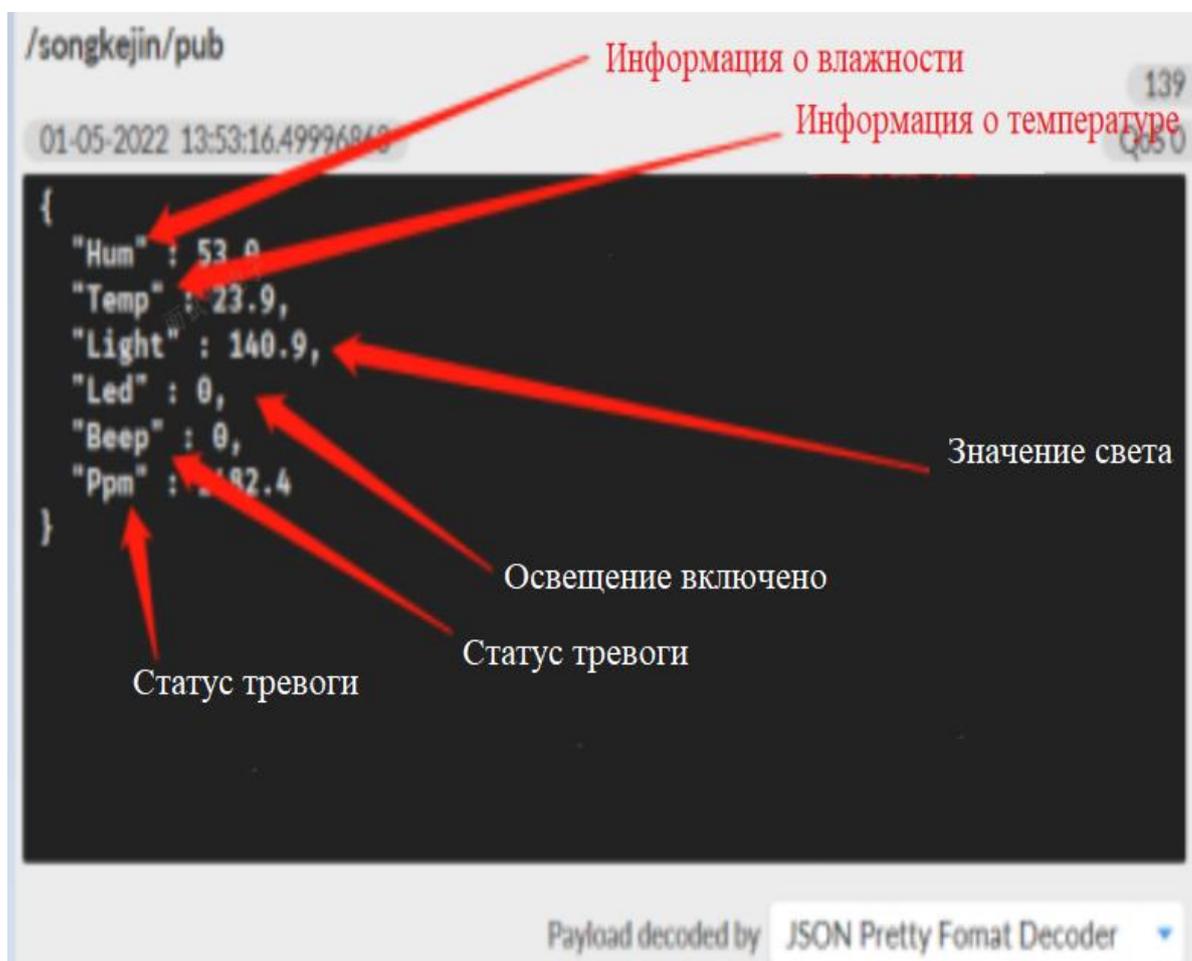


Рисунок 30 – Интерфейс помощника по отладке MQTT.fx

После успешного подключения ESP8266 к серверу MQTT данные могут быть отправлены на сервер MQTT. Сервер MQTT будет служить станцией передачи информации для предоставления необходимой информации для последующих апплетов WeChat.

С помощью вышеуказанных тестов связи было успешно продемонстрировано, что данные связи между устройством и сервером являются нормальными, а также их функции. Создав свой собственный сервер MQTT через Alibaba Cloud и успешно создав EMQX, вы можете получить сетевой доступ к устройству. В EMQX вы можете просматривать информацию об устройстве и напрямую отправлять команды управления для управления

устройством. Это закладывает основу для нормального взаимодействия между апплетом WeChat и сервером.

Сегодня WeChat является очень известным социальным программным обеспечением в Китае. С момента создания WeChat компанией Tencent количество ежедневных активных пользователей WeChat увеличивается из года в год. На данный момент количество пользователей WeChat привлекло почти 1 миллиард человек. Апплет WeChat на основе программного обеспечения WeChat появился естественным образом. Апплет WeChat отличается от обычного приложения для мобильного телефона. Обычное приложение для мобильного телефона требует от пользователей загрузки и установки, но апплет WeChat не требует этого сложного шага. Кроме того, апплет WeChat предоставляет разработчикам большое количество интерфейсов приложений для вызова, что значительно сокращает время разработки разработчиков. В соответствии с официальными документами, опубликованными самим WeChat, разработчики могут больше сосредоточиться на изучении того, что им нужно для своего бизнеса.

Апплет WeChat может подключаться к серверу MOTT, получать данные с сервера и отображать их на терминале апплета WeChat после обработки. В то же время апплет WeChat может также управлять нижестоящими устройствами и может вручную управлять устройствами в соответствии с фактическими условиями. В то же время он также может получать информацию, загружаемую устройством в режиме реального времени. Поскольку он имеет те же функциональные характеристики, что и приложение для мобильного телефона, эта система вводит апплет WeChat как интеллектуальный терминал в систему. Необходимо понимать конкретный статус сообщения устройства, а также может управлять устройством. Апплет WeChat сравнивается с другими программными решениями, которые также могут управлять устройством. Конкретные преимущества и различия показаны в таблице 2.

Таблицы 2 – Сравнение прикладного программного обеспечения

Платформа	Веб-версия для ПК	Приложение	Мини-программа wechat
1	2	3	4
Скачать	Загрузка не требуется, войдите через веб-страницу	Требуется загрузка	Загрузка не требуется
Установить	Ненужный	Необходимо установить	Установка не требуется
Сложность развития	Средняя сложность	Более высокая сложность	Средняя сложность
Удобство	Неудобно	Удобный	Удобный

Подводя итог, веб-страница должна сохранять исполняемый код на удаленном сервере, в то время как приложение должно быть загружено локально для использования и требует некоторого объема памяти. Программа WeChat mini сохраняет программу на официальном облачном сервере WeChat и даже может использовать официальную облачную базу данных WeChat для более сложного дизайна и разработки. Видно, что выбор разработки программы WeChat mini является разумным выбором.

В процессе разработки вы можете использовать официальный инструмент разработки WeChat под названием WeChat Development Tool, который имеет мощные функции отладки и компиляции. При проектировании вам необходимо разработать макет интерфейса в файле wxml. Файл wxss используется для проектирования стиля интерфейса, чтобы он выглядел более красиво, а файл js используется разработчиками для написания логики разработки и обработки различных типов данных.

При проектировании программного обеспечения вам необходимо сначала ввести библиотеку MQTT.JS. Это также должно облегчить установление соединения данных между апплетом WeChat и сервером MQTT. Функции библиотеки должны вызываться во время использования. Подобно STM32, апплет WeChat также должен подписываться и публиковать темы

для достижения взаимодействия данных в реальном времени. В процессе проектирования сложность разработки заключается в основном в написании логики разработки. Необходимо установить коммуникационное соединение между апплетом WeChat и сервером MQTT, как показано на рисунке 31.

```
const mqttHost = "ykgaoxiao.top";  
const mqttPort = 8084;  
const deviceSubTopic = "/songkejin/sub";  
const devicePubTopic = "/songkejin/pub";
```

Рисунок 31 – Подключение к серверу MQTT

В процессе подключения необходимо заполнить доменное имя сервера и номер порта. Аналогично необходимо подписаться на темы и опубликовать их. Основная функция – подключиться к устройству с наименьшими затратами, получать данные, отправленные с сервера, и отправлять команды на устройство. Апплет поддерживает только номер порта 8084.

После подключения к серверу MQTT апплет WeChat распечатает информацию о состоянии соединения, чтобы помочь разработчикам проверить, подключен ли сервер. После успешного подключения апплет WeChat немедленно подпишется на тему, а станция печати отобразит успешную подписку на данные устройства upstream Topic. После подписки на тему полученные данные представляют собой просто строку данных, которые не имеют формата JSON. Апплет WeChat не может напрямую использовать этот тип данных для отображения на странице интерфейса, поэтому требуется анализ JSON. При разработке апплета WeChat преобразование в формат JSON требует использования языка разработки апплета WeChat. `JSON.parse` преобразует данные в данные, которые могут быть отображены на странице

интерфейса. Область печати инструмента апплета WeChat распечатает проанализированные данные, как показано на рисунке 32.

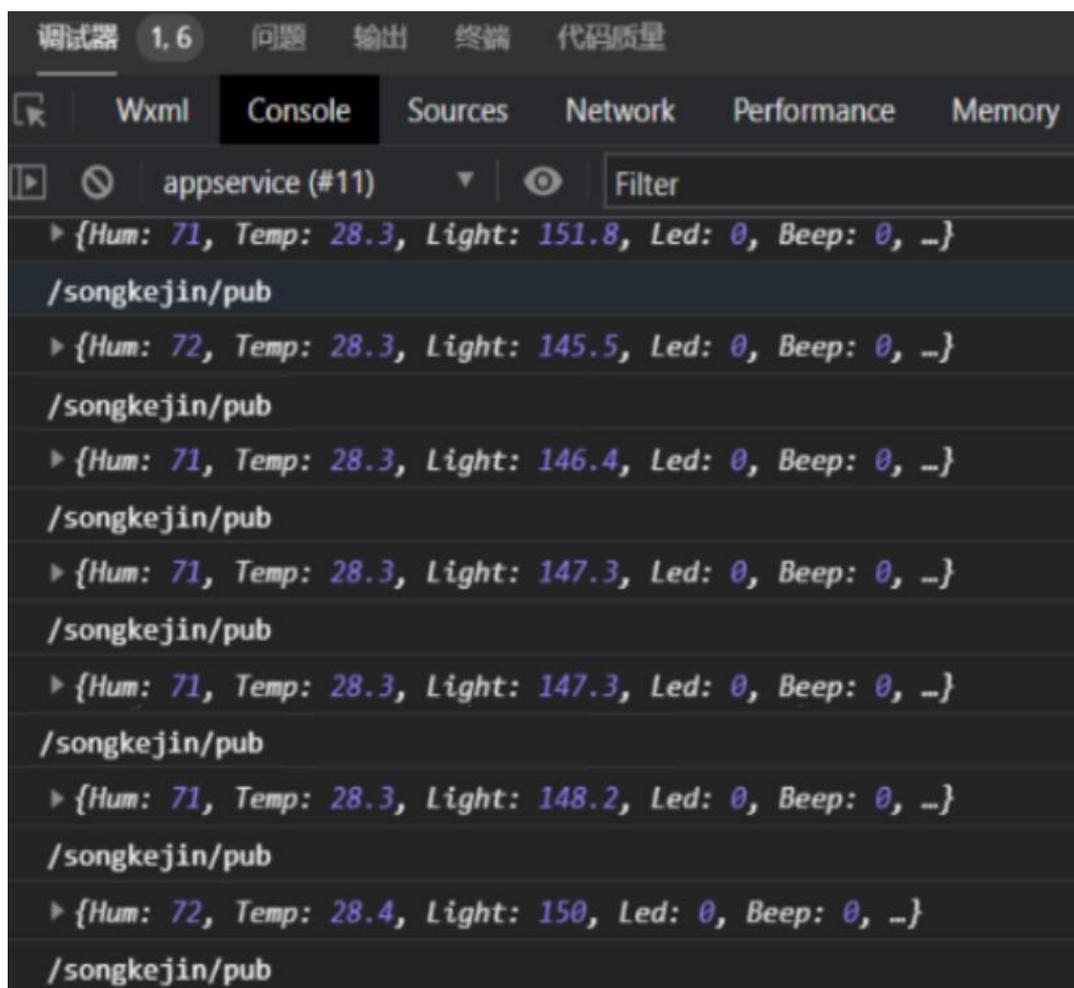


Рисунок 32 – Вывод области печати апплета WeChat

После того, как все вышеперечисленные работы были выполнены, в ходе совместных экспериментов по отладке было обнаружено, что информация, отправленная устройством, может быть получена апплетом WeChat, и данные могут быть успешно проанализированы и отображены на странице интерфейса. Затем устройство было отправлено для включения заполняющего света и проверки сигнализации, и было обнаружено, что они могут успешно отреагировать.

Устройство получает инструкции по включению заполняющего света и будильника от апплета WeChat, и устройство может выполнить соответствующие инструкции вовремя, указывая на то, что апплет WeChat успешно получил информацию об устройстве, отправленную устройством, и имеет

функцию управления устройством, указывая на то, что эксперимент прошел успешно. Фактическая картина показана на рисунке 33.

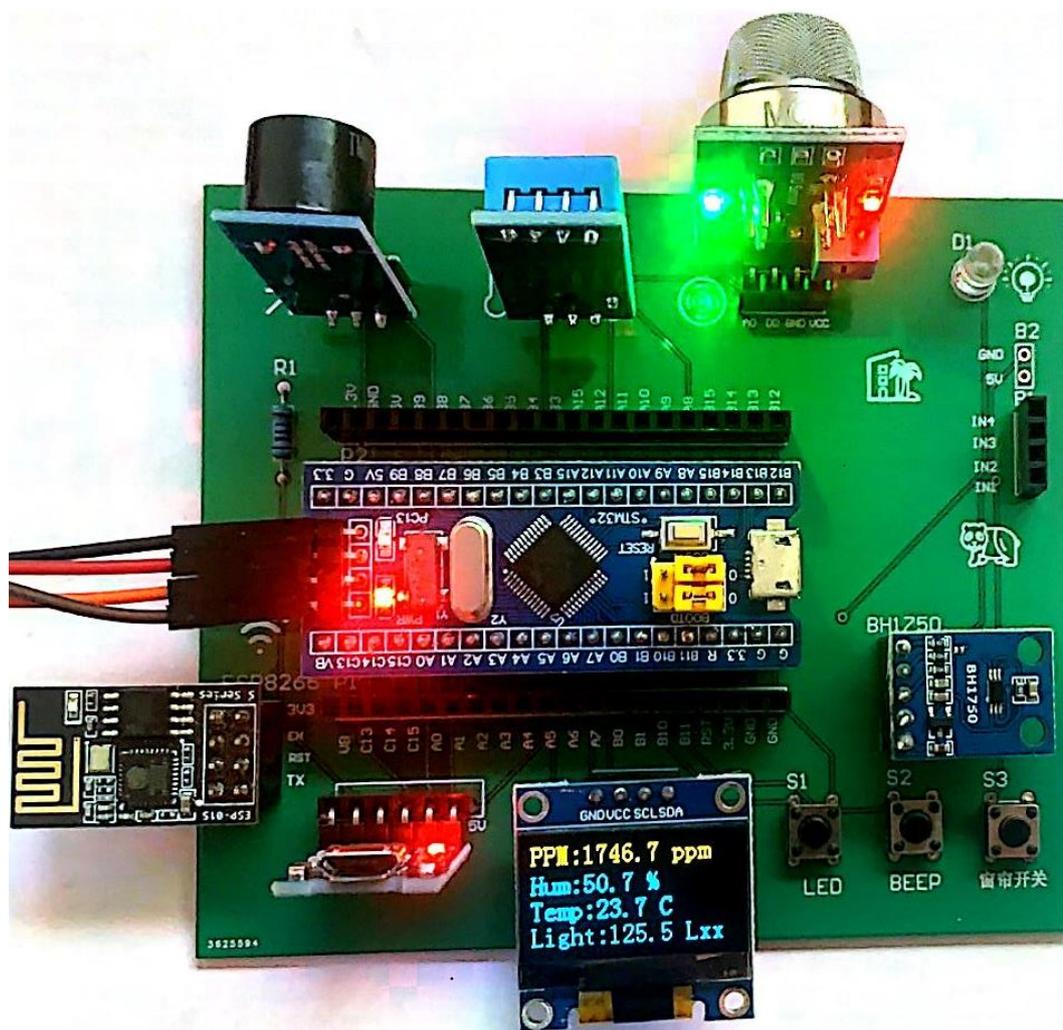


Рисунок 33 – Физическая схема конструкции оборудования

В процессе разработки и внедрения, после получения данных датчиков и загрузки их на сервер МОТТ, мы готовились к разработке апплета WeChat. Мы уже знали о существующем на рынке прикладном программном обеспечении и всесторонне сравнили преимущества и недостатки различного прикладного программного обеспечения. Наконец, мы выбрали апплет WeChat, который очень дружелюбен к разработчикам и имеет тот же рабочий эффект, что и APP, и не требует загрузки. Удобство неоспоримо. Используя инструмент разработки апплета WeChat, вы можете быстро раз-

работать необходимое вам оборудование для управления. В будущем применение апплета WeChat в промышленном управлении имеет большой потенциал и также очень расширяемо.

### **3.2. Пример тестирования программного обеспечения системы сбора данных об окружающей среде**

Целью тестирования этого программного обеспечения является проверка его функциональности, производительности и стабильности, чтобы гарантировать, что программное обеспечение может удовлетворить потребности пользователей в различных средах.

Стратегия тестирования включает модульное тестирование, интеграционное тестирование, системное тестирование и приемочное тестирование, что позволяет обеспечить высокое качество каждого функционального модуля и всей системы в целом.

Ниже представлено последовательное тестирование функциональных модулей системы. Одним из таких модулей является функция, которая обрабатывает данные о температуре и на основе входных данных вычисляет среднюю, максимальную и минимальную температуру. Функция критична для мониторинга окружающей среды и позволяет оперативно отслеживать изменения температурных показателей. Алгоритм температурного модуля, который реализует эту функциональность, показан на рисунке 34.

Для проверки работы функции `process\_Temperature\_data` необходимо подготовить несколько наборов входных данных. Каждый набор должен содержать различные значения температуры, чтобы протестировать корректность вычислений для разных сценариев. Например, можно использовать наборы данных с температурой в пределах нормальных значений, экстремальными температурами, а также данные с ошибками или выбросами, чтобы проверить устойчивость функции к неожиданным или некорректным данным.

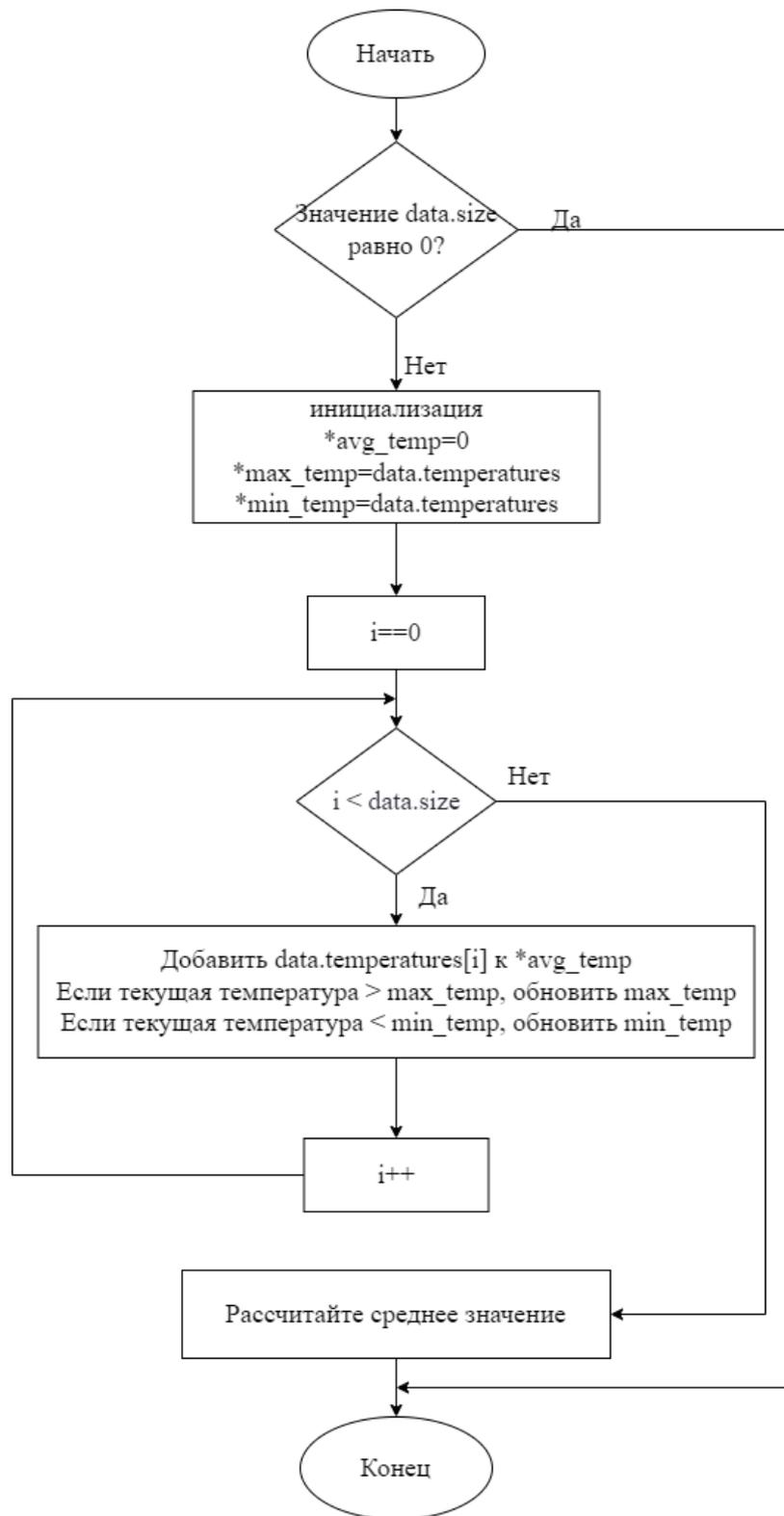


Рисунок 34 – Диаграмма алгоритма температуры

Для тестирования функции обработки температурных данных подготовлены следующие тестовые случаи, описанные ниже.

Тест 1: Нормальные данные с положительными температурами. Входные данные: {22.5, 25.0, 23.0, 24.5, 26.0}. Вычисления:

- средняя температура:  $(22.5 + 25.0 + 23.0 + 24.5 + 26.0) / 5 = 24.02$ ;
- максимальная температура: 26.0;
- минимальная температура: 22.5.

Тест 2: Набор данных с отрицательными температурами. Входные данные:  $\{-5.0, -3.0, -8.0, -2.5, -4.0\}$ . Вычисления:

- средняя температура:  $(-5.0 + -3.0 + -8.0 + -2.5 + -4.0) / 5 = -4.7$ ;
- максимальная температура: -2.5;
- минимальная температура: -8.0.

Тест 3: Набор данных с нулевыми температурами. Входные данные:  $\{0.0, 0.0, 0.0, 0.0, 0.0\}$ . Вычисления:

- средняя температура:  $(0.0 + 0.0 + 0.0 + 0.0 + 0.0) / 5 = 0.0$ ;
- максимальная температура: 0.0;
- минимальная температура: 0.0.

Тест 4: Пустой массив. Входные данные:  $\{\}$ .

Для пустого массива система должна либо вернуть значение по умолчанию (например, ноль или сообщение об ошибке), либо корректно обработать ситуацию, не вызывая сбоев в программе.

Тест 5: Набор данных с очень большими температурами. Входные данные:  $\{1000.0, 2000.0, 3000.0, 4000.0, 5000.0\}$ . Вычисления:

- средняя температура:  $(1000.0 + 2000.0 + 3000.0 + 4000.0 + 5000.0) / 5 = 3000.0$ ;
- максимальная температура: 5000.0;
- минимальная температура: 1000.0.

Тест 6: Набор данных с малыми температурными колебаниями. Входные данные:  $\{22.0, 22.1, 22.2, 22.0, 22.1\}$ . Вычисления:

- средняя температура:  $(22.0 + 22.1 + 22.2 + 22.0 + 22.1) / 5 = 22.08$ ;
- максимальная температура: 22.2;
- минимальная температура: 22.0.

Данные тесты помогут удостовериться, что функция обработки тем-

пературных данных правильно работает с различными типами входных данных, включая как стандартные, так и экстремальные значения.

Поскольку данные пустые, функция не должна изменять значения, и результат не должен быть вычислен.

Загрузка в систему VPL осуществляется после завершения всех этапов тестирования, включая формирование входных данных, выполнение расчётов и получение результатов. Как только функциональные тесты будут завершены, а корректность работы алгоритмов подтверждена, необходимо подготовить итоговый файл с реализованной функцией и результатами тестов. Этот файл должен соответствовать требованиям, установленным для системы VPL (Virtual Programming Lab).

Файл загружается в интерфейс VPL, где осуществляется автоматическое тестирование программного кода на соответствие критериям задания. Система VPL выполняет проверку заранее подготовленных тест-кейсов, оценивает корректность вывода, устойчивость программы к ошибкам и соблюдение логики работы. Результаты автоматической проверки отображаются в отчёте VPL и служат основанием для последующей оценки.

Успешное прохождение всех тестов в VPL подтверждает корректность реализации алгоритма и завершает этап функционального тестирования программного модуля.

Пример вывода результатов тестирования представлен следующим образом:

- test 1: avg: 24.02, max: 26.00, min: 22.50;
- test 2: avg: -4.70, max: -2.50, min: -8.00;
- test 3: avg: 0.00, max: 0.00, min: 0.00;
- test 4: no data available.

Обоснование выбора тестовых данных заключается в необходимости охвата как стандартных, так и граничных сценариев работы функции. В первом тесте проверяется корректность вычислений при наличии нормального

набора температурных значений с положительными числами. Это типичный случай для систем мониторинга окружающей среды, работающих в стандартных климатических условиях.

Второй тест предназначен для оценки корректности функционирования алгоритма при наличии отрицательных температур. Это особенно важно для обеспечения надежности работы системы в регионах с холодным климатом, где температуры могут опускаться ниже нуля.

Третий тест демонстрирует поведение функции при нулевых значениях температуры, что может быть полезно при эксплуатации системы в условиях постоянного холода, например, в морозильных складах или в условиях сурового климата.

Четвёртый тест моделирует ситуацию с отсутствием данных. Это крайний случай, при котором массив температурных значений пуст. Проверка данного сценария позволяет убедиться, что функция корректно обрабатывает отсутствие входных данных, не вызывая исключений или неправильных результатов. В частности, при отсутствии проверки может возникнуть попытка деления на ноль или обращения к несуществующим элементам массива. Для предотвращения подобных ошибок в функцию «process\_Temperature\_data» была добавлена логика обработки пустых массивов, обеспечивающая безопасное завершение выполнения и возврат корректного диагностического сообщения.

Функция обработки температурных данных имеет логическую структуру, аналогичную функции анализа показателей влажности. Модуль влажности также вычисляет среднее, максимальное и минимальное значения по входному массиву. Он применяется для отслеживания и контроля условий влажности в окружающей среде, что особенно важно в помещениях с особыми требованиями к микроклимату, таких как склады, теплицы или серверные помещения. Логика работы модуля влажности представлена на рисунке 35.

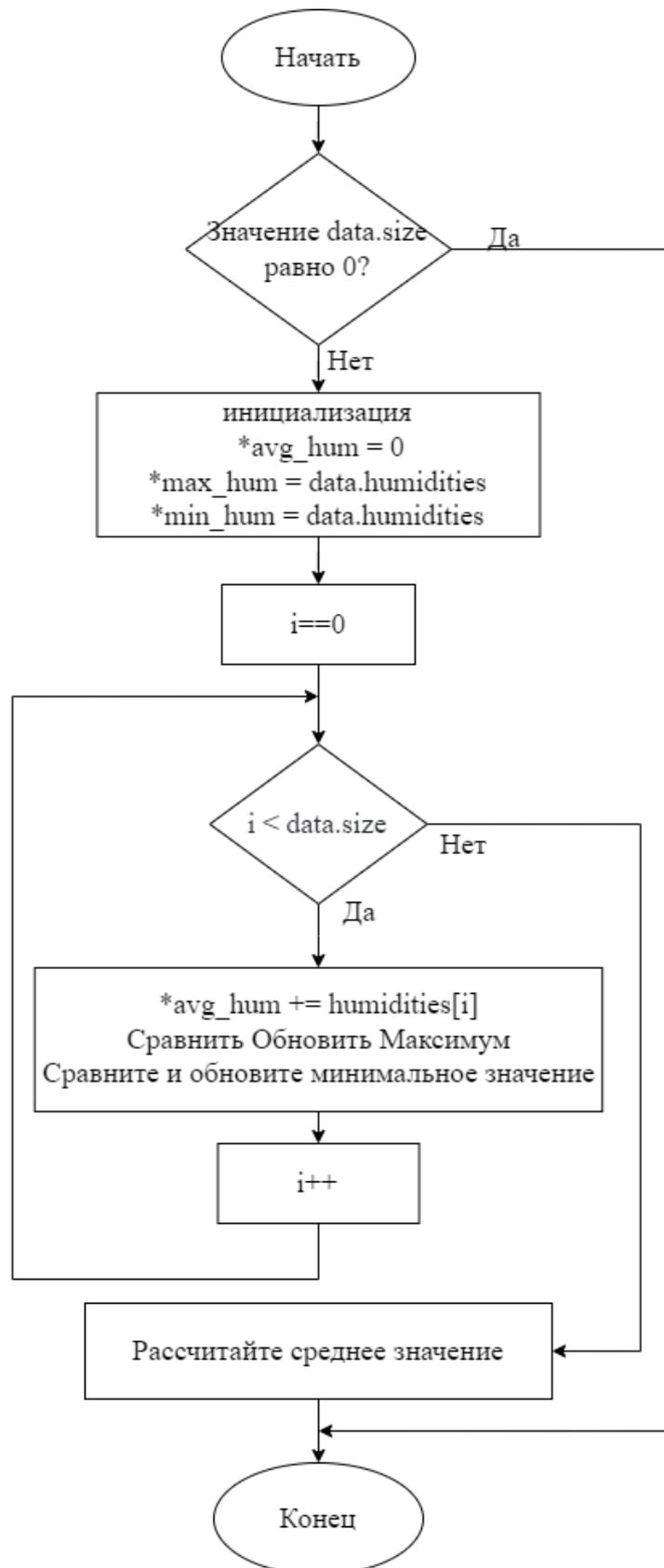


Рисунок 35 – Схема алгоритма влажности

Тестовые данные, включая нормальные значения, граничные условия и аномалии.

Таблицы 3 – Сравнение прикладного программного обеспечения

Тест	Описание	Входные данные	Ожидаемые результаты
1	2	3	4
Тест 1	Нормальные значения (в пределах 0–100 %)	{50.0, 60.0, 55.0, 58.0, 52.0}	Avg: 55.00, Max: 60.00, Min: 50.00
Тест 2	Аномальные значения (отрицательные и >100 %)	{-10.0, 20.0, 130.0, -5.0, 15.0}	Avg: 30.00, Max: 130.00, Min: -10.00
Тест 3	Все нулевые значения	{0.0, 0.0, 0.0, 0.0, 0.0}	Avg: 0.00, Max: 0.00, Min: 0.00
Тест 4	Пустой массив	{}	Нет данных (size == 0)

Вычисление выходных данных, Пример вывода функции для каждого теста:

- test 1: avg: 55.00, max: 60.00, min: 50.00;
- test 2: avg: 30.00, max: 130.00, min: -10.00;
- test 3: avg: 0.00, max: 0.00, min: 0.00;
- test 4: no data available.

Функция корректно обрабатывает аномальные значения (например, отрицательные или >100 %), что позволяет выявлять ошибки датчиков.

Пустой массив не изменяет указатели avg, max, min – требуется явная проверка size в вызывающем коде.

Рекомендации по тестированию:

Тест с одним элементом (например, {75.0} → все значения равны 75.00).

Проверить граничные значения (0 % и 100 %):

```
float humidities_5[] = {0.0, 100.0, 50.0}; // Avg: 50.00, Max: 100.00, Min: 0.00
```

Тест со смешанными данными (валидные + невалидные):

```
float humidities_6[] = {45.0, -20.0, 110.0}; // Avg: 45.00, Max: 110.00, Min: -20.00
```

Внедрить проверку диапазона влажности (0–100 %) и генерацию предупреждений при аномальных значениях.

Далее мы тестируем модули, связанные с освещением и дымом. Подготовьте данные для теста интенсивности света, как показано в таблице 4 ниже.

Таблица 4 – Данные испытаний интенсивности света

Тестовые случаи	Описывать	Входные данные	Ожидаемые результаты
1	2	3	4
Тест 1	Нормальное внутреннее освещение (100–500 Люкс)	{200.0, 350.0, 450.0, 300.0}	Среднее: 325,00, Максимум: 450,00, Минимум: 200,00
Тест 2	Экстремальное освещение (0 Люкс и 10000 Люкс)	{0.0, 10000.0, 500.0}	Среднее: 3500.00, Максимум: 10000.00, Минимум: 0.00
Тест 3	Отрицательное значение (ошибка датчика)	{-50.0, 200.0, 300.0}	Среднее: 150,00, Максимум: 300,00, Минимум: -50,00
Тест 4	Пустой массив	{}	Нет данных (size == 0)

Таблица 5 – Данные испытаний плотности дыма

Тестовые случаи	Описывать	Входные данные	Ожидаемые результаты
1	2	3	4
Тест 1	Диапазон безопасности (0–5 % или 0–500 ppm)	{0.5, 1.2, 3.0, 4.5}	Среднее: 2,30, Максимум: 4,50, Минимум: 0,50
Тест 2	Опасная концентрация (превышающая порог)	{8.0, 12.5, 20.0}	Среднее: 13.50, Максимальное: 20.00, Минимальное: 8.00
Тест 3	Отрицательное значение (неверный ввод)	{-2.0, 3.0, 5.0}	Среднее: 2.00, Максимум: 5.00, Минимум: -2.00
Тест 4	Одноэлементный массив	{15.0}	Среднее/максимальное/минимальное значение – все 15,00

Тест интенсивности света:

- test 1: avg: 325.00 lux, max: 450.00 lux, min: 200.00 lux;
- test 2: avg: 3500.00 lux, max: 10000.00 lux, min: 0.00 lux;
- test 3: avg: 150.00 lux, max: 300.00 lux, min: -50.00 lux;
- test 4: no light data available.

Тест на плотность дыма:

- test 1: avg: 2.30 %, max: 4.50 %, min: 0.50 %;
- test 2: avg: 13.50 %, max: 20.00 %, min: 8.00 %;
- test 3: avg: 2.00 %, max: 5.00 %, min: -2.00 %;
- test 4: avg: 15.00 %, max: 15.00 %, min: 15.00 %.

После тестирования были внесены некоторые улучшения: улучшена проверка входных данных, light: отрицательные значения должны отмечаться как сбой датчика (например, `if (light < 0) log_error()`). Дым: активация тревоги, когда концентрация отрицательная или превышает максимальный диапазон системы (например, 100 %); тест расширения граничных значений, свет: добавление сверхвысокой яркости (например, 100000 люкс, имитирующей прямой солнечный свет). Дым: проверка критических порогов безопасности (например, 5,0 % или 500 ppm); исправление логики функции, ошибка инициализации света: если функция возвращается напрямую, когда `data.size == 0`, вызывающей стороне необходимо проверить, был ли изменен указатель (рекомендуется возвращать код ошибки); дополнение к промышленному варианту использования, дым: имитация кратковременных пиков (например, {1,0, 2,0, 50,0, 2,0} для проверки временных опасностей). Освещение: добавлены данные о периодических изменениях (например, колебаниях освещения склада при его включении и выключении).

Тестовые случаи выдачи команд апплета WeChat и облачной платформы OneNET охватывают весь процесс выдачи команд включения/выключения света, включая нормальные, ненормальные, безопасные и производительные сценарии, а также проверяют надежность, режим реального

времени и безопасность апплета WeChat, выдающего команды включения/выключения света устройствам IoT через API облачной платформы OneNET.

Таблица 6 – Базовые функциональные тесты

ID теста	Описание сценария	Шаги тестирования	Ожидаемый результат
1	2	3	4
ТС-001	Успешная отправка команды «Включить свет»	1. Пользователь авторизуется в Mini Program и выбирает устройство. 2. Нажимает кнопку «Включить свет». 3. Проверяет журнал команд OneNET и статус устройства.	1. Mini Program отображает «Команда успешно отправлена». 2. OneNET фиксирует успешную отправку. 3. Устройство включает свет, статус синхронизируется с платформой.
ТС-002	Успешная отправка команды «Выключить свет»	1. Устройство включено. 2. Нажимает кнопку «Выключить свет». 3. Замеряет время отклика устройства.	1. Устройство выключается в течение 3 секунд. 2. Статус в Mini Program обновляется на «Выключено».
ТС-003	Многократная отправка команд (нагрузочный тест)	1. 10 раз нажимает «Включить» и «Выключить» в течение 5 секунд.	1. OneNET выполняет только последнюю команду. 2. Конечный статус устройства соответствует последней команде.

Таблица 7 – Тесты исключительных сценариев

ID теста	Описание сценария	Шаги тестирования	Ожидаемый результат
1	2	3	4
ТС-004	Отправка команды при разрыве сети	1. Отключает интернет в Mini Program. 2. Пытается отправить команду «Включить свет».	1. Mini Program показывает «Нет подключения к сети». 2. Команда не отправляется на OneNET.
ТС-005	Отправка команды при отключенном устройстве	1. Вручную отключает устройство. 2. Пытается отправить команду «Включить свет».	1. OneNET возвращает ошибку «Устройство offline». 2. Mini Program показывает "Устройство не подключено".
ТС-006	Недействительный токен или API-ключ	1. Использует просроченный токен для вызова API OneNET. 2. Пытается отправить команду.	1. OneNET возвращает «401 Unauthorized». 2. Mini Program показывает «Ошибка авторизации».

Таблица 8 – Тесты безопасности и прав доступа

ID теста	Описание сценария	Шаги тестирования	Ожидаемый результат
1	2	3	4
ТС-007	Неавторизованный доступ	1. Пользователь не авторизован. 2. Пытается отправить команду через поддельный запрос.	1. OneNET возвращает «403 Forbidden». 2. Устройство не реагирует.
ТС-008	Попытка управления чужим устройством	1. Пользователь А меняет device_id в запросе на устройство пользователя Б.	1. OneNET проверяет владельца устройства и блокирует команду. 2. Mini Program показывает «Доступ запрещен».

Таблица 9 – Тесты производительности и совместимости

ID теста	Описание сценария	Шаги тестирования	Ожидаемый результат
1	2	3	4
ТС-009	Высокая нагрузка (100 пользователей)	1. Эмулирует 100 одновременных запросов «Включить свет».	1. Успешность обработки запросов $\geq 99\%$ . 2. Среднее время отклика $< 2$ сек.
ТС-010	Задержки при разных типах сети	1. Тестирует отправку команд через Wi-Fi, 4G и при слабом сигнале.	1. Задержка при Wi-Fi/4G $< 1$ сек. 2. При слабом сигнале задержка $< 3$ сек с возможностью повтора.
ТС-011	Совместимость с разными устройствами	1. Подключает устройства разных брендов к OneNET. 2. Отправляет команды.	1. Все устройства корректно выполняют команды. 2. Статусы синхронизируются без ошибок.

В ходе тестирования были обнаружены следующие проблемы: для ТС-005 добавлена кнопка «Повторить» при ошибке подключения. Для ТС-009 оптимизировать обработку очередей на сервере.

Позже он будет изменен и скорректирован для подтверждения выполнения команды и получения АСК от устройства. Реализуйте локальное кэширование команд, чтобы их можно было повторно отправить при восстановлении сети. Расширенное ведение журнала для диагностики сбоев.

После отдельного тестирования вышеуказанной серии взаимосвязанных модулей проблемы были обнаружены и решены. После отладки системного оборудования и программного обеспечения был проведен совместный тест. Сбор, прием и передача различных данных были нормальными, и все данные могли нормально отображаться на странице. Все ожидаемые функции могли быть реализованы. Как показано на рисунке 36, данные последовательного порта, полученные данные облачной платформы и отображаемые данные страницы апплета, контролировались, а передача и отображение данных были правильными.

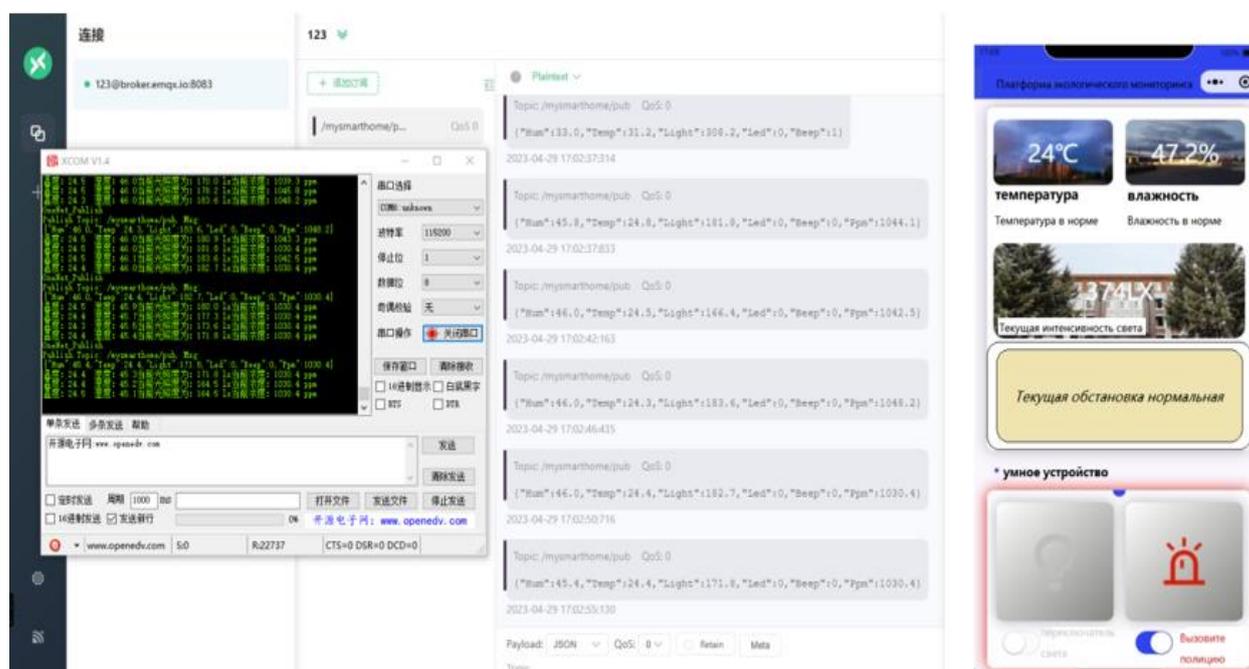


Рисунок 36 – Диаграмма отображения тестовых данных

Результаты испытаний системы показывают, что ожидания по проекту в основном выполнены, а функция сбора данных среды облачной платформы на базе STM32 успешно реализована.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения проекта была разработана и успешно протестирована система мониторинга окружающей среды, основанная на использовании датчиков температуры, влажности, освещённости и концентрации дыма. Основной управляющий блок на базе микроконтроллера STM32 осуществляет обработку и анализ данных, а модуль ESP8266 обеспечивает стабильную передачу информации на сервер MQTT. Пользовательское взаимодействие реализовано через мини-приложение, созданное в среде WeChat Developer Tools, что позволяет в реальном времени отслеживать параметры окружающей среды и управлять исполнительными устройствами.

Испытания подтвердили надёжность, экономичность и практичность системы. Её архитектура обеспечивает удобство эксплуатации и лёгкость масштабирования, что делает решение перспективным для внедрения в разнообразных условиях. Несмотря на потенциальные ограничения, связанные с использованием бесплатных общедоступных серверов MQTT, надёжность передачи данных может быть повышена путём перехода на частные серверные решения.

Проект обладает значительным потенциалом для дальнейшего развития в рамках концепции интеллектуальной среды обитания (Smart+). В будущем возможна адаптация системы под более широкий спектр задач, в том числе за счёт настройки чувствительности и интеграции дополнительных модулей мониторинга, что обеспечит её применимость в различных сценариях и повысит ценность решения для пользователей.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Алексеевский, П. И. Обучение студентов программированию с использованием отладочных комплектов STM32 Discovery / П. И. Алексеевский // Педагогическое образование в России. – 2018. – № 5. – С. 12–17.

2 Бурлов, В. Г. Мониторинг и экспертиза безопасности: учебное пособие / В. Г. Бурлов // Санкт-Петербургский политехнический университет Петра Великого, Высшая школа техносферной безопасности. – Санкт-Петербург. – 2018. – 13 с.

3 Выбираем и устанавливаем сигнализаторы загазованности / сайт «Газификация России» [Электронный ресурс]. – Режим доступа : <https://www.gazprommap.ru/articles/gas-check-devices/>. – 01.12.2024.

4 Горюноква, А. А. Современное состояние и подходы к разработке систем мониторинга загрязнения атмосферы / А. А. Горюноква // Известия ТулГУ. Технические науки. – 2013. – № 11. – С. 40-42.

5 Дулепов, В. И. Экосистемный анализ / В. И. Дулепов, О. А. Лескова. – Владивосток : ВГУЭС. – 2006. – 248 с.

6 Егоров, А. Ф. Методы идентификации мгновенных аварийных источников загрязнения атмосферного воздуха / А. Ф. Егоров, Т. В. Савицкая, С. П. Дударов // Химическая технология. – 2002. – №10. – С. 41-46.

7 Игнатенко, Г. К. Первичная обработка данных экологического мониторинга / Г. К. Игнатенко. – Обнинск : ИАТЭ. – 2006. – 76 с.

8 Инсайдерское руководство по STM32 [Электронный ресурс]. – Режим доступа : <http://redblot.ru/archives/966>. – 01.12.2024.

9 Кропотов, Ю. А. Марковские модели в автоматизированной системе мониторинга и прогнозирования экологического состояния промышленной зоны / Ю. А. Кропотов, Г. П. Суворова // Известия ОрелГТУ. Серия «Информационные системы и технологии». – 2008. – № 1–3/269(544). – С. 113-118.

10 Крыщенко, В. С. Проблемы почвенного мониторинга агроландшафтов: структура и модель данных / В. С. Крыщенко, О. М. Голозубов // Агрехимический вестник. – 2010. – № 5. – С. 11-18.

11 Мирзадинов, Р. А. Иерархия мониторинга окружающей среды / Р. А. Мирзадинов // Инновации в науке. – 2012. – № 10-2. – С. 20-26.

12 Мотузова, Г. В. Химическое загрязнение биосферы и его экологические последствия / Г. В. Мотузова, Е. А. Карпова // М. : Изд-во МГУ. – 2013.

13 Павлов, Н. И. Глобальные системы мониторинга окружающей среды и фоновый мониторинг / Н. И. Павлов, М. М. Кулеш // Вестник ТГЭУ. – 2000. – № 4. – С. 55-56.

14 Пути повышения производительности труда в сельском хозяйстве [Электронный ресурс]. – Режим доступа : <https://studfile.net/preview/>. – 01.12.2024.

15 Суворова, Г. П. Автоматизированная система экологического контроля предприятия / Г. П. Суворова // Методы и устройства передачи и обработки информации: межвуз. сб. научн. тр. Вып. 4 / под ред. В. В. Ромашиова, В. В. Булкина. – СПб. : Гидрометеиздат. – 2004. – С. 347.

16 Темичев, А. А. Интеллектуализация процесса оценки производительности систем мониторинга / А. А. Темичев, Р. А. Файзрахманов // Актуальные вопросы образования и науки: сборник научных трудов по материалам Международной научно-практической конференции: в 11 частях. – 2014. – С. 136–138.

17 Темичев, А. А. Проблемы современных распределённых систем мониторинга / А. А. Темичев, Р. А. Файзрахманов // Современные тенденции в образовании и науке: сборник научных трудов по материалам Международной научно-практической конференции: в 26 частях. – 2013. – С. 132–133.

18 Третьякова, Н. А. Основы экологии: учеб. пособие для вузов / Н. А. Третьякова; под науч. ред. М. Г. Шишова // Издательство Юрайт. – 2018. – 111 с.

19 Файзрахманов, Р. А. Технические аспекты создания комплексной автоматизированной информационной системы интеллектуального мониторинга окружающей среды / Р. А. Файзрахманов, Ю. А. Слаутин, В. Д. Володин, Р. Р. Бикметов, А. А. Шаронов // Устойчивое развитие. – 2013. – № 10. – С. 67–72.

20 Chue, H. A Data Acquisition System for A Hand-Held Electronic Nose (H2EN) / H. Chue, J. V. Hartfield, P. D. Wareham, P. A. Payne // Obermeier, E. (eds). Transducers '01 Eurosensors XV. – Berlin, Heidelberg : Springer. – 2001.

– P. 1684–1687.

21 DHT11 Humidity & Temperature Sensor [Электронный ресурс]. – Режим доступа : <https://files.scienceforum.ru/pdf/2022/61c1ceb99820f.pdf>. – 01.12.2024.

22 Dickens, C. Evaluating the global state of ecosystems and natural resources: Within and beyond the SDGs / C. Dickens, M. McCartney, D. Tickner, I. J. Harrison, P. Pacheco, B. Ndhlovu // Sustainability (Switzerland). – 2020. – Vol. 12. – № 18.

23 Erechtkhoukova, M. G. Investigation of monitoring designs for water quality assessment / M. G. Erechtkhoukova, P. A. Khaiteer // Proceedings of the 18th World IMACS Congress and MODSIM 2009 – International Congress on Modelling and Simulation: Interfacing Modelling and Simulation with Mathematical and Computational Sciences. – 2009. – P. 3612–3618.

24 ESP8266EX Datasheet [Электронный ресурс]. – Режим доступа : <https://cdn-shop.adafruit.com/product-files/2471/0AESP8266%20Datasheet.pdf>. – 01.12.2024.

25 Fauziah, F. Internal Audit Procedure on Environmental Management to Minimize Impact of Environmental Damage (Case Study at PT. Wonojati Wijoyo Kediri) / F. Fauziah // Journal of Economics, Finance and Management Studies. – 2023. – Vol. 6. – № 6. – P. 2629–2633.

26 I2C-bus specification and user manual [Электронный ресурс]. – Режим доступа : [http://www.nxp.com/documents/user\\_manual/UM10204.pdf](http://www.nxp.com/documents/user_manual/UM10204.pdf). – 22.11.2024.

27 Kumar, A. Environmental Monitoring Systems: A Review / A. Kumar, H. Kim, G. P. Hancke // IEEE Sensors Journal. – 2013. – Vol. 13, № 4. – P. 1329–1339.

28 Shamsadini, K. Environmental Audit / K. Shamsadini, M. A. Shahamabad, S. Bahonar // Bulletin of Science and Practice. – 2022. – P. 423–427.

29 STM32 – микроконтроллер для начинающих после Arduino [Электронный ресурс]. – Режим доступа : <https://arduinomaster.ru/stm32/stm32-mikrokontroller-dlya-nachinayushhih-posle-arduino>. – 01.11.2024.

30 WeChat Mini Program Development Guide [Электронный ресурс]. – Режим доступа : <https://developers.weixin.qq.com/miniprogram/>. – 01.12.2024.