

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ  
Заведующий кафедрой ЭВМ  
\_\_\_\_\_ Д.В. Топольский  
« \_\_\_ » \_\_\_\_\_ 2023 г.

Программно-аппаратный комплекс для контроля качества воздуха

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ  
ЮУРГУ-090301.2023.172 ПЗ ВКР

Руководитель работы,  
к.п.н., доцент каф. ЭВМ  
\_\_\_\_\_ Ю.Г. Плаксина  
« \_\_\_ » \_\_\_\_\_ 2023 г.

Автор работы,  
студент группы КЭ-406  
\_\_\_\_\_ Д.И. Арцер  
« \_\_\_ » \_\_\_\_\_ 2023 г.

Нормоконтролёр,  
ст. преп. каф. ЭВМ  
\_\_\_\_\_ С.В. Сяськов  
« \_\_\_ » \_\_\_\_\_ 2023 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ  
Заведующий кафедрой ЭВМ  
\_\_\_\_\_ Д.В.Топольский  
«\_\_» \_\_\_\_\_ 2023 г.

**ЗАДАНИЕ**

**на выпускную квалификационную работу бакалавра**  
студенту группы КЭ-406  
Арцер Джульетте Ильиничне  
обучающемуся по направлению  
09.03.01 «Информатика и вычислительная техника»

1. **Тема работы:** «Программно-аппаратный комплекс для контроля качества воздуха» утверждена приказом по университету от «25» апреля 2023 г. № 753-13/12
2. **Срок сдачи студентом законченной работы:** 01 июня 2023 г.
3. **Исходные данные к работе:** модель прибора для контроля качества воздуха, состоящий из:
  - 3.1 микроконтроллер;
  - 3.2 датчик углекислого газа;
  - 3.3. датчик микрочастиц пыли;
  - 3.4 датчик температуры и влажности.
4. **Перечень подлежащих разработке вопросов:**
  1. Аналитический обзор научно-технической, нормативной и методической литературы по тематике работы.

2. Разработка структуры программно-аппаратного комплекса.
3. Проектирование аппаратной части комплекса.
4. Разработка программной части комплекса.
5. Реализация и тестирование программно-аппаратного комплекса.

**5. Дата выдачи задания:** 2 декабря 2022 г.

Руководитель работы \_\_\_\_\_ /Ю.Г. Плаксина/

Студент \_\_\_\_\_ / Д.И. Арцер/

## КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	01.03.2023	
Разработка структуры программно-аппаратного комплекса	01.04.2023	
Проектирование аппаратной части комплекса.	15.04.2023	
Разработка программной части комплекса.	01.05.2023	
Реализация и тестирование программно-аппаратного комплекса.	15.05.2023	
Компоновка текста работы и сдача на нормоконтроль	24.05.2023	
Подготовка презентации и доклада	30.05.2023	

Руководитель работы \_\_\_\_\_/Ю.Г. Плаксина/

Студент \_\_\_\_\_/Д.И. Арцер/

## АННОТАЦИЯ

Д.И.Арцер. Программно-аппаратный комплекс для контроля качества воздуха. - Челябинск:ФГАОУ ВО «ЮУрГУ (НИУ)», ВШ ЭКН; 2023, 65 с., библиогр. список – 13 наим.

Проведен аналитический обзор современной научно-технической, нормативной, методической литературы, затрагивающей научно-техническую проблему.

В ходе выполнения практической работы классифицированы критерии для подобных приборов и собраны актуальные данные по техническим компонентам прибора. Приведены примеры оборудования для контроля за качеством воздуха от зарубежных и отечественных производителей. Указан ряд стандартов на которых основано большинство комплексов и систем контроля качества воздуха.

После анализа произведен подбор компонентов, синтезирована принципиальная электрическая схема устройства, изготовлен прототип устройства, создано программное обеспечение для реализации необходимого функционала, проведено итоговое тестирование прибора на соответствие требуемым характеристикам.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	7
1. АНАЛИТИЧЕСКИЙ ОБЗОР НАУЧНО-ТЕХНИЧЕСКОЙ, НОРМАТИВНОЙ И МЕТОДИЧЕСКОЙ ЛИТЕРАТУРЫ ПО ТЕМАТИКЕ РАБОТЫ.....	9
1.1. Аналитический обзор современных мониторинговых систем качества воздуха.....	9
1.2. Вывод по обзору на аналогичные продукты.....	16
1.3. Обзор способов оповещения.....	17
2. РАЗРАБОТКА СТРУКТУРЫ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА.....	20
2.1. Постановка задачи.....	20
2.2. Технические требования.....	20
2.3. Выбор датчиков.....	21
2.4. Выбор микроконтроллера.....	27
2.5. Функциональные требования.....	29
3. ПРОЕКТИРОВАНИЕ АППАРАТНОЙ ЧАСТИ КОМПЛЕКСА.....	33
4. РАЗРАБОТКА ПРОГРАММНОЙ ЧАСТИ КОМПЛЕКСА.....	37
5. РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА.....	47
ЗАКЛЮЧЕНИЕ.....	53
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	54
ПРИЛОЖЕНИЕ А. Схема электрическая принципиальная и перечень элементов всех модулей аппаратной части системы.....	56
ПРИЛОЖЕНИЕ Б. Фрагменты исходного кода аппаратной части системы.....	58

## ВВЕДЕНИЕ

Программно-аппаратный комплекс для контроля качества воздуха - это устройство, позволяющее анализировать важные параметры воздуха и сигнализировать об их ухудшении. Для обеспечения продуктивной работы сотрудников офиса требуется контроль условий, в которых офисные работники проводят большую часть своего времени. На продуктивность большое влияние оказывает здоровье сотрудников и одним из залогов хорошего здоровья является качество воздуха.

Создание системы, контролирующей качество воздуха - основная цель данного проекта. На качество воздуха влияют многие параметры, а именно:

- уровень содержания углекислого газа в воздухе (наиболее важный критерий);
- влажность;
- температура;
- содержание микрочастиц.

CO<sub>2</sub> — углекислый газ или диоксид углерода — бесцветный газ, который не имеет запаха при малых концентрациях. Углекислый газ выделяется людьми, животными и растениями, например, организм человека способен выделить около 1 кг углекислого газа в сутки. Повышенная концентрация углекислого газа в помещении может являться опасным для здоровья людей [1]. Внешнее это проявляется появлением ряда субъективных симптомов:

- головная боль;
- чувство разбитости;
- головокружение;
- раздражительность;
- нарушение сна.

В помещении важно поддерживать минимально возможную концентрацию уровня CO<sub>2</sub>, так как углекислый газ оказывает негативное влияние на организм человека. А основная причина повышенного содержания CO<sub>2</sub> в воздухе — это

люди, одновременно находящиеся в помещении, и их деятельность. Чем больше людей, тем активнее вырабатывается углекислый газ. Особо остро эта проблема стоит в офисах.

Также важным критерием качества воздуха является содержание микрочастиц пыли. Данные частицы образуются в результате загрязнения воздуха кусочками сажи, асфальта, частиц минеральных солей, соединения тяжелых металлов и частицах сажи. Также к вредным микрочастицам относятся некоторые аллергены и выбросы азота и серы. Все эти частицы накапливаются в организме и со временем вызывают серьезные проблемы с дыхательной системой человека и нарушением работы кровеносных сосудов. Это может быть причиной для увеличения нагрузки на сердце и риска возникновения тромбов [2].

Чтобы поддерживать оптимальные параметры воздуха в помещении, требуется использовать регулируемую вентиляцию или своевременно осуществлять проветривание. Но в процессе работы, эта задача уходит на второстепенный план и забывается. Так как в условиях заказчика не предполагается встраивать и налаживать вентиляцию, целью данного проекта будет создать систему, своевременно напоминающую работникам о недопустимом уровне всех вышеперечисленных показателей и о надлежащем скорейшем проветривании.



# 1 АНАЛИТИЧЕСКИЙ ОБЗОР НАУЧНО-ТЕХНИЧЕСКОЙ, НОРМАТИВНОЙ И МЕТОДИЧЕСКОЙ ЛИТЕРАТУРЫ ПО ТЕМАТИКЕ РАБОТЫ

## 1.1 Аналитический обзор современных мониторинговых систем качества воздуха

Многие компании, ведущие разработку инновационных приборов, работают над созданием домашних метеостанций для мониторинга качества воздуха в помещениях. Это важно как для домашнего использования, так и для производства. Ниже приведен список и краткий обзор наиболее успешных проектов.

### 1. Монитор качества воздуха от Даджет

Данный прибор анализирует содержание углекислого газа  $\text{CO}_2$  в воздухе и сигнализирует о чрезмерной его концентрации. Изображение прибора представлено на рисунке 1.1. Об уровне качества воздуха говорят светодиоды. Интуитивный интерфейс : зеленый, желтый и красный цвета загораются соответственно при  $<800$  ppm;  $800\dots1200$  ppm;  $>1200$  ppm.



Рисунок 1.1 - Монитор качества воздуха от Даджет

При превышении концентрации  $\text{CO}_2$  более 1200 ppm — прибор начинает издавать негромкие короткие сигналы [3]. На лицевой поверхности располагается кнопка MUTE, для отключения данной опции.

Дисплей имеет белую подсветку, которая не отключается. Она средней яркости и может мешать при домашнем использовании чувствительным людям спать. Питание прибора осуществляется от сети.

Монитор оснащен только датчиком  $\text{CO}$ , что не решает проблему нашего заказчика. Из плюсов можно выделить разнообразные способы уведомления пользователя об ухудшении качества воздуха: звуковой сигнал, светодиоды и дисплей. Однако прибор не оснащен подключением к сторонним девайсам и удаленным контролем.

## 2. Монитор воздуха Qingping ClearGrass

Детектор оснащен датчиками, измеряющими:  $\text{CO}_2$ , VOC,  $\text{PM}_{2.5}$ , что удовлетворяет требованиям заказчика. Изображение прибора представлено на рисунке 1.2. Также у прибора встроен сенсорный дисплей с автоматической регулировкой яркости [4]. На дисплее отражаются все измеряемые показатели.



Рисунок 1.2 - Монитор воздуха Qingping ClearGrass

Прибор не оснащен звуковой сигнализацией об ухудшении качества воздуха, что усложняет его использование: необходим постоянный зрительный контроль за дисплеем. Также монитору не хватает синхронизации с приложением или мессенджером для оперативного оповещения пользователя. Питание прибора осуществляется от аккумулятора, что также можно считать недостатком.

### 3. Датчик качества воздуха life control

Датчик анализирует и сообщает о содержании в воздухе углекислого газа  $\text{CO}_2$  и летучих органических веществ (ЛОВ), а также о температуре и влажности воздуха [5]. Изображение прибора представлено на рисунке 1.3. Работает устройство от батареи. Доступно дистанционное управление через мобильное приложение или сайт. Приложение адаптировано под Android/iOS.

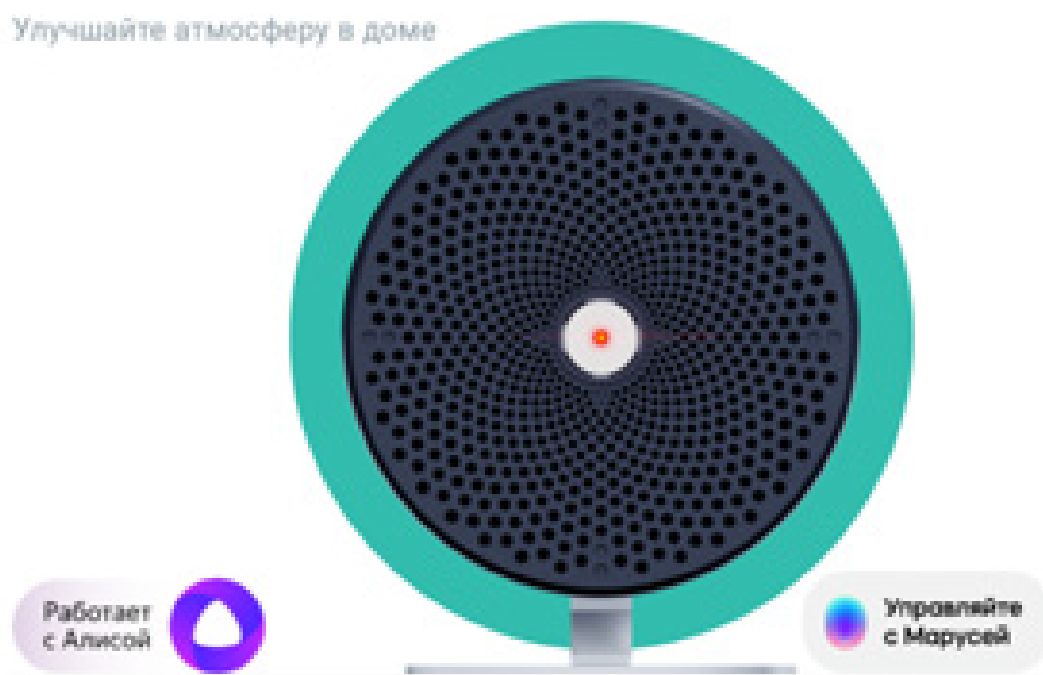


Рисунок 1.3 - Датчик качества воздуха life control

У устройства отсутствует звуковое оповещение и нет встроенного дисплея для контроля за качеством воздуха в режиме реального времени. Также прибор не оснащен датчиком мелкодисперсных частиц  $\text{PM}_{2.5}$ .

### 4. Датчик качества воздуха от IKEA Vindstyrka

ИКЕА представила компактный датчик качества воздуха с оригинальным названием Vindstyrka [6]. Он способен отслеживать четыре важных параметра и

предупреждать пользователя о вредном для здоровья уровне загрязнения. Изображение прибора представлено на рисунке 1.4. Устройство работает от сети и оснащается LCD-дисплеем, на котором отображаются влажность и температура воздуха, а также количество летучих органических соединений (tVOC) и мелкодисперсных частиц PM2,5 (менее 2,5 микрона в диаметре). Устройство не издает звуковое оповещение.



Рисунок 1.4 - Датчик качества воздуха от IKEA Vindstyrka

Устройство удовлетворяет почти все потребности заказчика, однако имеет самый важный недостаток: на данный момент товары данной компании недоступны в РФ. Датчик будет доступен для продаж только в апреле, тогда будет известно взаимодействие монитора с другими системами для удаленного оповещения.

## 5. Анализатор качества воздуха Noyafa JMS

Современное устройство от Noyafa (рисунок 1.5). Отображает качество воздуха в реальном времени по шкале от плохого к хорошему, автоматически включает звуковое оповещение при плохом качестве воздуха. Значение аккумулятора в правом верхнем углу показывает уровень заряда. Можно настроить яркость дисплея посредством нажатия кнопки "Яркость". С помощью кнопки "Звук" вкл./выкл. звуковое оповещение. Наличие звукового оповещения о качестве воздуха отображается в левом верхнем углу. Прошел испытания на соответствие RoHS, FCC, CE, Noyafa прошла систему сертификации качества ISO 9001 [6]. Производство в Китае, с таким же качеством, цена продукта более конкурентоспособна, чем у аналогичных конкурентов.

## 6. Монитор качества воздуха Qingping

Xiaomi Qingping Air Monitor Lite CGDN1 – вторая версия высокоточного датчика воздуха, которая имеет более компактный корпус, дисплей и отличный от первой модели набор датчиков [7].



Рисунок 1.5 - Анализатор качества воздуха Noyafa JMS

Монитор необходим, чтобы понимать параметры качества воздуха: концентрацию  $CO_2$ , количество твердых частиц, температуру, влажность и другие. Только имея необходимый объем данных с помощью таких бытовых приборов, как очиститель, увлажнитель, мойка воздуха или бризер, можно эффективно работать над их улучшением. Но, даже без наличия дополнительных бытовых приборов или возможности автоматизации подобных процессов, полезно знать, чем мы дышим и иметь возможность оперативно принять меры. Например, концентрация диоксида углерода (углекислый газ) в доме, квартире или офисе влияет на наше самочувствие, может вызывать головную боль и чувство усталости. Если в помещении без притока свежего воздуха находится несколько человек, то содержание углекислого газа в нем увеличивается в алгебраической прогрессии. Прибор изображен на рисунке 1.6.

#### 7. Анализатор воздуха VAKIO Atmosphere

Монитор качества воздуха Vakiо ATMOSPHERE - это климатическая станция для измерения температуры, влажности воздуха и уровня углекислого газа [9].



Рисунок 1.6 - Монитор качества воздуха Qingping

Монитор передает эти данные другим устройствам, отвечающим за состояние воздуха в помещении (например, VAKIO Window Smart, Base Smart,

OpenAir) и тем самым поддерживает чистоту и температуру воздуха в помещении без участия человека. Есть возможность самостоятельно установить границы, в которых нужно держать показатели воздуха и менять их со своего смартфона с помощью приложения Vakiо SMART Control или Яндекс станции Алиса. Прибор изображен на рисунке 1.7.



Рисунок 1.7 - Анализатор воздуха VAKIO Atmosphere

В таблице 1 представлен анализ вышеописанных приборов по основным критериям - это цена прибора, тип питания, наличие датчиков, тип оповещения пользователя и наличие дисплея.

Таблица 1 - Сравнение существующих аналогов

Название прибора	Цена руб.	Питание	Измеряемые параметры				Виды оповещения			Наличие дисплея
			CO <sub>2</sub>	t / влажн.	PM2.5	светодиод	в приложении	звук		
Монитор качества воздуха от Даджет.	7900	От сети	Да	Да	Нет	Да	Нет	Да	Да	
Монитор воздуха Qingring ClearGrass	7043	Аккумулятор	Да	Да	Да	Да	Нет	Нет	Да	
Датчик качества воздуха life control	1228	Батареи	Да	Да	Нет	Нет	Да	Нет	Нет	
Датчик от IKEA Vindstyrka	-	От сети	Нет	Да	Да	Нет	Нет	Нет	Да	
Анализатор Noyafa JMS	4675	Аккумулятор	Да	Да	Нет	Да	Нет	Да	Да	
Монитор качества воздуха Qingring	4346	Аккумулятор	Да	Да	Да	Нет	Да	Нет	Да	
Многофункциональный монитор качества воздуха [8]	11900	От сети	Да	Да	Нет	Да	Да	Нет	Да	



## 1.2 Вывод по обзору на аналогичные продукты

Устройства были проанализированы по наиболее важным критериям. Были выявлены их недостатки и достоинства. Большинство из представленных приборов доступны и просты в эксплуатации, они имеют минималистический дизайн и интуитивно понятное оповещение.

Многие из представленных моделей имеют тип питания от аккумулятора, что позволяет разместить прибор вдали от розеток. Одним из минусов приборов с типом питания от сети является неточное показание температуры. Однако при использовании аккумулятора, энергоёмкие датчики не смогут мониторить качество воздуха достаточно долго, так как они весьма энергозатратны, а значит батарея нужна очень мощная либо всё же следует выбрать тип зарядки от сети.

Лишь некоторые модели имели возможность оповещения пользователя посредством дополнительных сервисов, таких как приложение или сайт. Преимуществом отличаются приборы с упрощенным способом настройки оповещений и отсутствием сложностей при регистрации.

Также немногие из аналогов оснащены звуковым оповещением, что является неотъемлемой функцией для подобного прибора, так как при эксплуатации в условиях заказчика (производственное помещение) не всегда удобно заходить в приложение или постоянно смотреть на дисплей для определения уровня качества воздуха. В данном случае звуковой сигнал поможет вовремя среагировать и проветрить помещение.

Проанализировав подобные приборы можно сделать вывод, что на рынке очень мало моделей, оснащенных датчиком PM2.5, а этот датчик будет включен в проектируемый прибор.

Так как постоянно смотреть на прибор в ожидании светового сигнала, не является возможным, а звуковое оповещение не всегда будет заметно, наиболее оптимальным способом оповещения, исходя из вышеприведенного анализа, является оповещения на телефон из внешних источников. Приложение могло бы быть отличным вариантом для данного вида оповещения, однако в таком случае придётся адаптировать его под различные операционные системы смартфонов или

компьютеров, что вызывает определенные трудности для внедрения в компанию. Можно было бы использовать смс-рассылку, но это слишком затратно. В условиях поставленной задачи наиболее приемлемым вариантом будет отправка сообщений посредством мессенджера. Данный вариант не зависит от модели устройства и его операционной системы, а также отправка сообщений будет производиться бесплатно и безопасно.

### 1.3 Обзор способов оповещения

#### 1.3.1 Визуальное оповещение

##### 1.3.1.1 Оповещение цветным индикатором

Данный вид оповещения является удобным для быстрой реакции на изменения качества воздуха. При удобном расположении прибора, световой индикатор всегда будет на виду пользователя и даст знать, когда следует проветрить помещение.

Однако этот вид оповещения может быть проигнорирован, пользователю требуется постоянно смотреть на прибор, чтобы не пропустить изменения показателей, что не всегда может быть удобно. Также недостатком данного оповещения является неполнота передаваемой информации. С помощью светодиода можно оповестить пользователя только о необходимости проветрить помещение, но не получится передать те показатели, которые превысили норму.

##### 1.3.1.2 Оповещение на дисплее устройства

Оповещение пользователя с помощью дисплея интуитивно понятный способ, при этом на дисплее могут отображаться все необходимые параметры. Однако данный вариант имеет больше недостатков, в связи с отображением всех параметров, что не позволит пользователю быстро определить ухудшение параметров воздуха и своевременно среагировать.

### 1.3.2 Оповещение на цифровые устройства

#### 1.3.2.1 Оповещение в специализированном приложении

Данный вид оповещения может быть очень удобен в использовании. В приложении можно отобразить для пользователя различные показатели, которые измеряет прибор, и вывести статистику изменений.

Но с точки зрения разработки этот вид имеет некоторые недостатки. Во-первых, придется разрабатывать приложение сразу на нескольких ОС, чтобы пользователь с любым телефоном имел возможность установить приложение. Во-вторых, приложение придется постоянно поддерживать и обновлять.

#### 1.3.2.2 Вывод информации на сайт

Одним из способов оповещения является создание адаптивного сайта с возможностью авторизации и мониторинга качества воздуха. В этом случае не придется разрабатывать различные версии, как в варианте с приложением, и это определенно плюс. На сайте могут быть также отображены все интересные пользователю показатели, графики, статистика и так далее.

Но это бывает не всегда удобно для пользователя, придется постоянно заходить на сайт, чтобы проверить состояние качества воздуха. Настройка оповещений при закрытой веб-странице будет неудобна, а значит данный вид можно использовать лишь для мониторинга.

#### 1.3.2.3 Оповещения с помощью смс

Данный способ удобен с точки зрения разработчика. Нужно лишь знать номер телефона пользователя и настроить автоматическую отправку сообщения с необходимостью проветривания.

Однако у этого способа много недостатков. Во-первых, стоимость таких оповещений слишком высокая. Во-вторых, в сообщениях будет указана необходимость проветрить, но не будет возможности расширенного доступа к статистике.

#### 1.3.2.4 Оповещение в мессенджере

Этот способ наиболее удобен, как для пользователя, так и для разработчика. Мессенджер можно установить на любое устройство вне зависимости от ОС, будь

это телефон или компьютер. С помощью бота можно автоматизировать отправку сообщений пользователю, которые к тому же будут бесплатными. И наделить бота возможностью отвечать на вопросы пользователя о статистике. Добавить команды вывода графиков и информации об изменении показателей.

### 1.3.3 Звуковое оповещение

Данный вид оповещения очень удобен. Даже если нет любого другого оповещения, пользователь обязательно поймет, что необходимо осуществить проветривание. Звуковое оповещение происходит мгновенно. Это позволяет быстро реагировать на ситуацию и принимать соответствующие меры. Звуковое оповещение может быть эффективным способом оповещения, особенно если пользователь занят другой деятельностью, например, работает на компьютере или занимается физической работой.

#### Вывод по разделу один

В результате проведенного анализа выявлены основные цели по проектированию устройства. Проанализированы различные подходы к созданию мониторинговой системы качества воздуха. И выявлены критерии, важные для потребителя. Проанализированы различные способы оповещения и выбраны наиболее подходящие:

- звуковое оповещение;
- оповещение посредством светодиодов;
- оповещение пользователя в мессенджере посредством чат-бота.

## 2 РАЗРАБОТКА СТРУКТУРЫ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА

### 2.1 Постановка задачи

Требуется разработать программно-аппаратный комплекс, измеряющий и анализирующий различные параметры качества воздуха для последующей передачи их пользователю. На основании проведенного аналитического обзора и требований заказчика можно выделить следующие общие требования к устройству:

- прибор должен измерять качество воздуха по показателям: температура, влажность, содержание углекислого газа ( $CO_2$ ) и содержание мелких частиц пыли (PM2.5);
- необходимо осуществить оповещение пользователя в случае выхода значений, полученных с датчиков, за пределы нормы, для своевременного проветривания помещений;
- следует организовать возможность пользователя получать уведомления от устройства посредством телеграмм-бота;
- прибор должен обеспечивать возможность пользователя получать собранные данные с датчиков в любой момент времени посредством Телеграмм-бота.

### 2.2 Технические требования

Исходя из проведенного анализа, были выявлены положительные характеристики некоторых аналогов и учтены недостатки. Из чего сформировываются требования к технической составляющей комплекса, устройство должно обеспечиваться:

- наличием модуля Wi-Fi для передачи данных пользователю;
- питание устройство осуществляется через разъем типа Type-C;
- напряжение питания системы должно быть 5 Вольт для питания всех узлов системы;
- измерение температуры в пределах от +15 до + 40°C с точностью не хуже  $\pm 1^\circ\text{C}$ ;

- измерение относительной влажности в диапазоне 0 - 100 % и иметь погрешность измерений не более 5%;
- измерение концентрации углекислого газа в пределах до 3000 ppm и с точностью не хуже  $\pm 40$ ppm;
- измерение количества мелкодисперсных частиц PM2.5 от 1 до 2.5 мкм в воздухе.

К конструктиву прибора предъявляются следующие требования. Так как основная функция прибора - измерять различные показатели качества воздуха, для его корректной работы внутри корпуса требуется расположить датчики так, чтобы их работа не искажала эти показатели. Температурный датчик следует расположить на максимально возможном расстоянии от тепловыделяющих узлов системы, чтобы при нагревании других датчиков, истинное значение температуры не увеличивалось. Датчику пыли требуется обеспечить свободный приток воздуха, для корректной работы. Так как прибор подразумевает использование внутри жилых помещений, к степени защиты от влаги и пыли особых требований не предъявляется, и будет использоваться стандарт IP40. Питание прибора будет осуществляться от разъема USB от внешнего блока питания, разъем должен быть расположен на задней части устройства.

### 2.3 Выбор датчиков





Исходя из выявленных требований необходимо выбрать соответствующие компоненты для решения поставленной задачи. Датчики являются основными компонентами системы, так как осуществляют сбор данных. Выбор датчиков осуществляется в соответствии с изначальными требованиями.

#### 2.3.1 Датчик PM2.5

Мелкодисперсные частицы PM2.5 - это воздушный загрязнитель, в состав которого входят как твердые микрочастицы, так и мельчайшие капельки жидкостей. И те, и другие размером примерно от 10 нм до 2,5 мкм. Другие обозначения и названия частиц PM2.5: FSP (fine suspended particles), fine particles, fine particulate matter, мелкодисперсные взвешенные частицы, тонкодисперсная пыль [2].

Датчик PM2.5 способен отслеживать количество таких частиц в воздухе. Ниже представлена таблица с обзором на различные датчики такого типа (таблица 2).

Таблица 2 - Обзор на датчики PM2.5

Датчики	Цена руб.	Размер мм	Фото	Разрешение мкг / м3 мкм	Рабочее напряжение (В)
SDS011	1 980	70 x 70 x 22		0.3	5
PMSA003	1 012	35×38×12		0.3	5
PMS7003 / G7	1 675	48×37×12		0.3	5
DSM501A	460	45 x 59 x 17		1 - 2.5	5

Исходя из вышеприведенного анализа можно сделать вывод, что размер датчиков примерно одинаковый. В нашем приборе наилучшим вариантом будет использование самого маленького устройства. Самым большим датчиком является SDS011, поэтому лучше рассмотреть другие датчики. Рабочее напряжение у всех датчиков также одинаковое (5 В). Ценовой диапазон тоже небольшой. Разрешение приборов неудовлетворительно только у датчика DSM501A. Наилучшим датчиком по всем показателям выберем PMSA003.

### 2.3.2 Датчики температуры и влажности

Датчиков, измеряющих температуру и влажность, достаточно много на рынке, и отличаются они точностью, диапазоном измерений и своим размером. В наших интересах найти наиболее точный датчик небольшого размера, температурный диапазон может быть небольшим, так как измерения будут происходить в помещении. Ниже представлена сравнительная таблица подобных датчиков (таблица 3).

Датчиков температуры на рынке огромное количество и область их применения очень большая, в связи с этим варьируется диапазон измеряемых показателей и точность измерений, как и размер датчиков. В условиях нашего заказчика не требуются большие диапазоны температур, важна точность и компактность. Цена датчиков варьируется от 170 до 18 600 рублей. В будущем для широкого производства данного прибора требуется использование экономически выгодных компонентов. Наиболее оптимальным датчиком по всем параметрам является ВМЕ280. У этого устройства невысокая цена, компактный размер, наименьшее напряжение питания и наивысшая точность.

### 2.3.3 Датчики CO<sub>2</sub>

Датчики CO<sub>2</sub> – приборы, обеспечивающие качественный мониторинг окружающего воздуха и определение концентрации двуокиси углерода. Этот параметр играет наиболее важную роль в анализе качества воздуха, поэтому от выбора этого датчика зависит точность всего прибора.

Исходя из данных, приведенных в таблице 4, можно сделать вывод, что почти все датчики примерно одинакового размера и имеют усредненное напряжение питания 5 В. Диапазон измерений всех устройств удовлетворяет требованиям заказчика. Минимальное содержание углекислого газа в воздухе на природе составляет 350 ppm, а в закрытом помещении средним качеством воздуха считается 400-600 ppm углекислого газа. Выше 1000 ppm ощущается дискомфорт, а более 1400 может приводить к проблемам с самочувствием, т.к. выходит за пределы нормы [9]. Поэтому даже минимальный диапазон датчика S8 0053 будет приемлемым. Наивысшей точностью обладает датчик SCD30, но он также








является самым дорогим. А наименьшее потребление тока у датчика S8 0053, он же соответствует всем требованиям данного проекта, поэтому в разработке следует использовать его.

Таблица 3 - Обзор на датчики температуры и влажности

Датчик	Цена руб.	Размер мм	Фото	Напряжение питания В	Диапазон измерения температуры °C	Точность температуры °C	Диапазон измерения влажности	Точность влажности
AM2302	750	15 x 12 x 6		3.3 - 5	-40... +80	± 1	0 .. 100 %	± 2%
BME280	620	15 x 12		1.8 ... 5	-40 ... + 85	± 0.5	0 ... 100%	± 3%
DHT22	275	15x25x7.7		3 .. 6	-40 ..+80	± 0.5	0 .. 100%	± 2%
DHT11	170	16 x 12 x 6		3 .. 5	0 .. 50	± 2%	20 .. 90%	± 5%

Таблица 4 - Обзор на датчики  $CO_2$

Датчик	Цена (руб.)	Фото	Размер (мм)	Напряжение питания (В)	Среднее потребление тока (мА)	Диапазон измерений (ppm)	Точность измерений
MH-Z19B	2200		29,5 x 19,5 x 8,5	4.5 - 5	<60	0 - 5000	±50 ppm
MH-Z14A	2 500		57 x 34 x 16	4.5 - 5.5	<85	0 - 5000	± 50 ppm
SGP30	1 190		10 x 20	1.8 - 5	<70	400 - 60000	± 50 ppm
S8 0053	1 957		8.5 x 33.5 x 20	5	<18	400 - 2000	± 40 ppm
SCD30	6 683		35 x 23 x 7	5	<19	400 - 10.000	± 30 ppm

## 2.4 Выбор микроконтроллера

Для разработки устройства, требуется выбрать соответствующую плату для обработки сигналов. Сейчас наиболее популярны платы семейства Arduino.

Они обладают многими преимуществами. Такими, как: простой синтаксис программирования на C++, удобная среда разработки Arduino IDE, множество готовых библиотек и скетчей, а также низкое энергопотребление. Однако у Arduino есть и недостатки, такие как низкая скорость работы и ограниченное количество памяти. Для более мощных проектов на рынке появились платформы на основе микроконтроллера ESP32.

ESP32 WROOM-32 - это встраиваемый модуль на основе микроконтроллера ESP32 от компании Espressif Systems. Плата имеет встроенный Wi-Fi и Bluetooth, а также множество интерфейсов, таких как SPI, I2C, UART, ADC и другие. ESP32 WROOM-32 имеет два ядра процессора с тактовой частотой до 240 МГц, 520 Кбайт оперативное запоминающее устройство (ОЗУ) и 4 Мбайт флэш-памяти. Она также имеет поддержку различных операционных систем, включая FreeRTOS, что является преимуществом, так как это позволяет создавать задачи на основе данной операционной системы и использовать многозадачность.

Данный модуль можно сравнить с популярной Arduino UNO R3 (Сравнение представлено на таблице 5). Arduino UNO R3 - это микроконтроллерная плата, которая основана на микроконтроллере ATmega328P от компании Microchip. Плата имеет множество интерфейсов, таких как SPI, I2C, UART, ADC и другие. Arduino UNO R3 имеет одно ядро процессора с тактовой частотой до 16 МГц, 2 Кбайт ОЗУ и 32 Кбайт флэш-памяти. Она также имеет свою собственную среду разработки и библиотеки, которые позволяют разработчикам легко создавать проекты на Arduino.

Таблица 5 - Сравнение микроконтроллерных плат

Микроконтроллер	Цена (руб.)	Фото	Кол-во ядер	Частота (МГц)	Wi-Fi, Bluetooth	Flash	РАМ (КБ)	Интерфейсы подключения
ESP32	498		2	160	Да	16 МБ	16	SPI / I <sup>2</sup> C / UART / I <sup>2</sup> S / CAN
ARDUINO UNO R3	921		1	16	Нет	32 КБ	32	SPI / I <sup>2</sup> C / UART

Таким образом, разница между ESP32 WROOM-32 и Arduino UNO R3 заключается в их характеристиках и возможностях. ESP32 WROOM-32 имеет более мощный процессор, больше памяти и встроенный Wi-Fi и Bluetooth, что делает ее более подходящей для создания данного проекта. Arduino UNO R3, с другой стороны, является более простой платой, которая хорошо подходит для начинающих и для создания простых проектов. Из-за чего было принято решение выбрать модуль ESP32 WROOM-32.

## 2.5 Функциональные требования

Для разрабатываемой структуры можно выделить следующие функциональные требования:

- прибору необходимо осуществлять считывание показаний всех датчиков в период заданный пользователем;
- прибор должен иметь возможность обновления встроенного ПО;
- настройка и калибровка прибора должна осуществляться автоматически;
- измеренные данные должны храниться в памяти устройства на период не менее 10 дней;
- прибор должен иметь возможность оповещения пользователя, передачи измеренных показаний и отображение статистики измерений;

Взаимодействие прибора с пользователем происходит через интерфейс Телеграмм-бота. На рисунке 2.1 представлена UML диаграмма, иллюстрирующая взаимодействия пользователя с программно-аппаратным комплексом посредством web-сервиса Телеграмм-бота.

Исходя из пунктов 2.1 и 2.2, можно выделить следующие модули для реализации функциональных требований:

- система управления
- модуль датчиков
- модуль обмена данными с пользователем

Модуль датчиков - это аппаратная часть комплекса, включающая в себя периферийные устройства:

- датчик температуры и влажности;

- датчик углекислого газа CO<sub>2</sub>;
- датчик частиц пыли PM<sub>2.5</sub>.



Рисунок 2.1 - UML диаграмма взаимодействия пользователя с программно-аппаратным комплексом

Сбор данных с датчиков и обработку будет производить микроконтроллер (модуль системы управления). Обработанную полученную информацию микроконтроллер будет передавать в телеграмм-бот (модуль обмена данными с пользователем).

В качестве датчиков по результатам анализа в пункте 2.3 и 2.4 были выбраны: PMSA003 (датчик PM<sub>2.5</sub>), BME280 (датчик температуры и влажности) и S80053 (датчик углекислого газа CO<sub>2</sub>). А в качестве микроконтроллера - ESP32.

Структурная схема системы контроля качества воздуха приведена на рисунке 2.2. Здесь изображены все модули системы и связи между ними. Благодаря слаженной работе каждого модуля выполняются все функциональные требования программно-аппаратного комплекса.

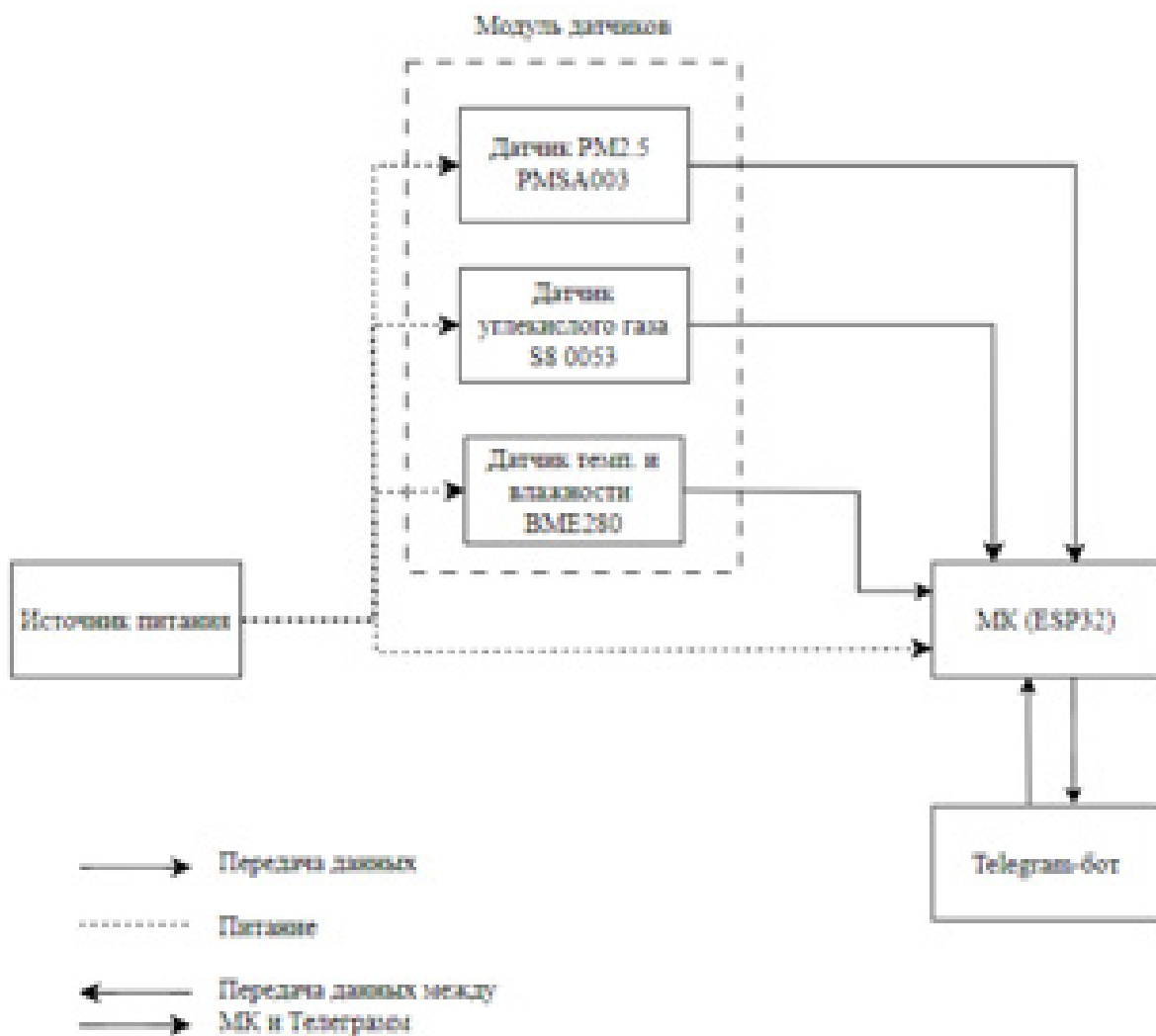


Рисунок 2.2 - Структурная схема системы контроля качества воздуха

Telegram-бот выполняет задачу связи устройства с пользователем для возможности получения данных с датчиков, а также отправки запросов на устройство для вывода данных. Передача данных между устройством и Telegram-ботом представлена на рисунке 2.3.



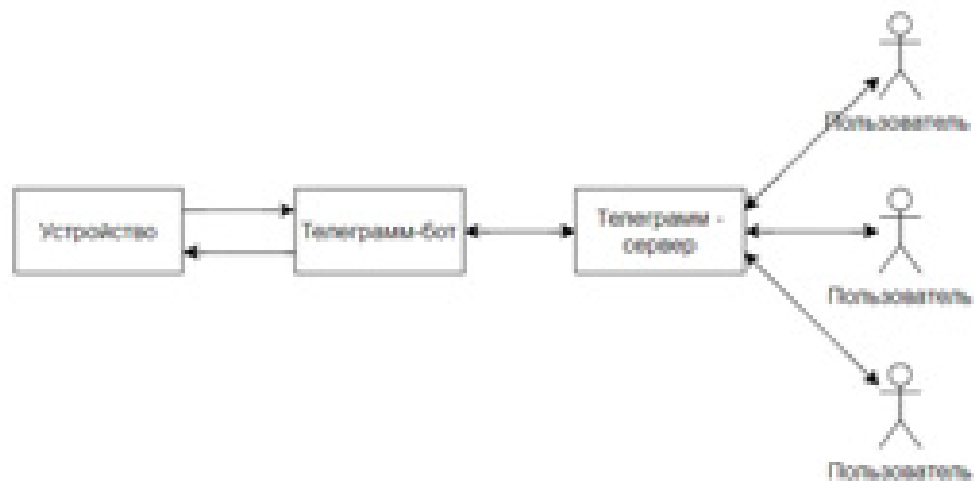


Рисунок 2.3 - Структурная схема работы Телеграмм-бота  
Вывод по разделу два

В данной главе был составлен ряд технических и функциональных требований к разрабатываемому комплексу. Также были отобраны наиболее подходящие компоненты для разрабатываемого прибора. ESP32 WROOM-32 в качестве платы и соответствующие датчики:

- датчик PMSA003 для измерения PM2.5;
- датчик BME280 для измерения температуры и влажности;
- датчик S8 0053 для измерения CO2.

Кроме того были разработаны структурные схемы взаимодействия всех модулей в устройстве:

- Структурная схема системы контроля качества воздуха
- Структурная схема работы Телеграмм-бота

### 3 ПРОЕКТИРОВАНИЕ АППАРАТНОЙ ЧАСТИ КОМПЛЕКСА

На основе структурной схемы устройства, разработанной во 2 главе, следует составить структурную схему связи аппаратных частей комплекса.

В качестве микроконтроллера был выбран ESP32. В качестве датчиков: PMSA003 (датчик PM2.5), BME280 (датчик температуры и влажности) и S80053 (датчик углекислого газа CO<sub>2</sub>). Структурная схема подключения микроконтроллера к датчикам представлена на рисунке 3.1.

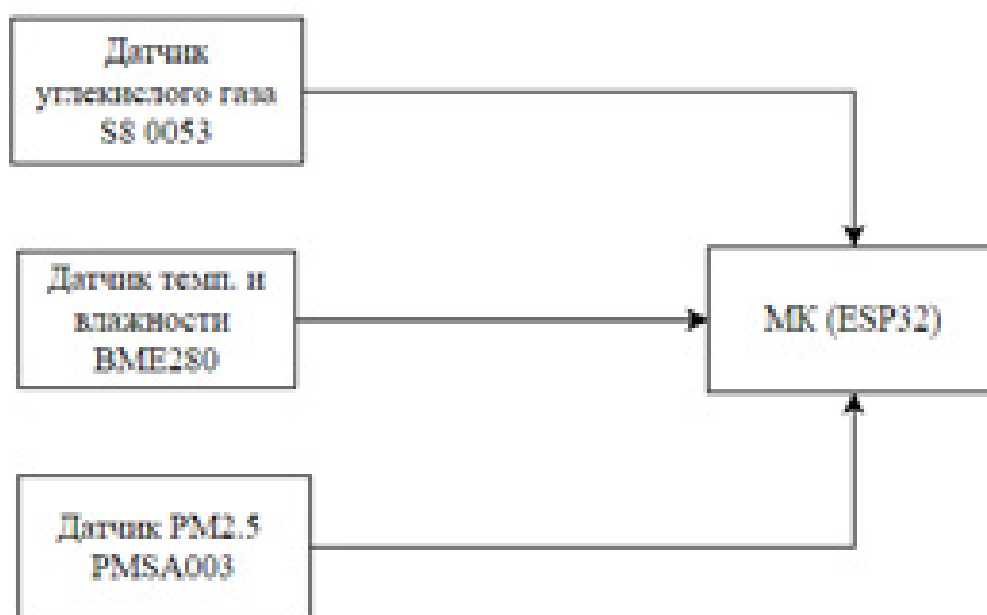


Рисунок 3.1 - Структурная электрическая схема подключения

На основании структурной схемы производится синтез функциональной схемы. Рассмотрим подключение всех датчиков к центральному микроконтроллеру. Датчик температуры и влажности подключается посредством интерфейса I2C. Питание модуля осуществляется по контактам VCC и GND. Протокол обмена данными I2C использует два провода для обмена информацией. Один используется для тактового сигнала (SCL), а другой – для отправки и получения данных (SDA). Схема подключения датчика к модулю представлена на рисунке 3.2.

Интерфейс I2C (или по другому IC) — это достаточно широко распространённый сетевой последовательный интерфейс, придуманный фирмой Philips и завоевавший популярность относительно высокой скоростью

передачи данных (обычно до 100 кбит/с, в современных микросхемах до 400 кбит/с), дешевой и простотой реализации.

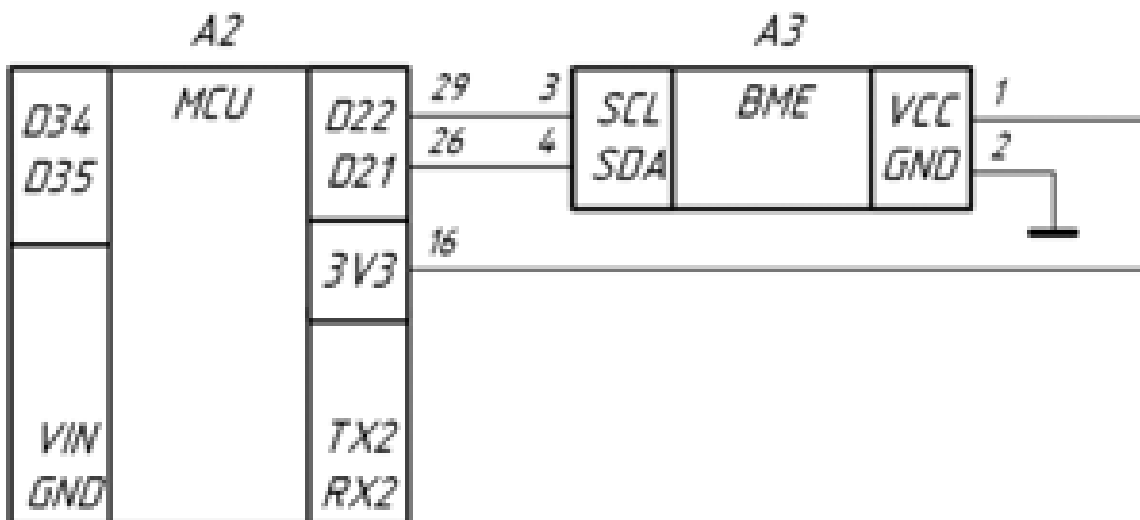


Рисунок 3.2 - Фрагмент схемы электрической принципиальной подключения датчика температуры

Датчик PMSA003 и S80053 подключены к устройству через интерфейс UART. Схема подключения датчика углекислого газа представлена на рисунке 3.3.

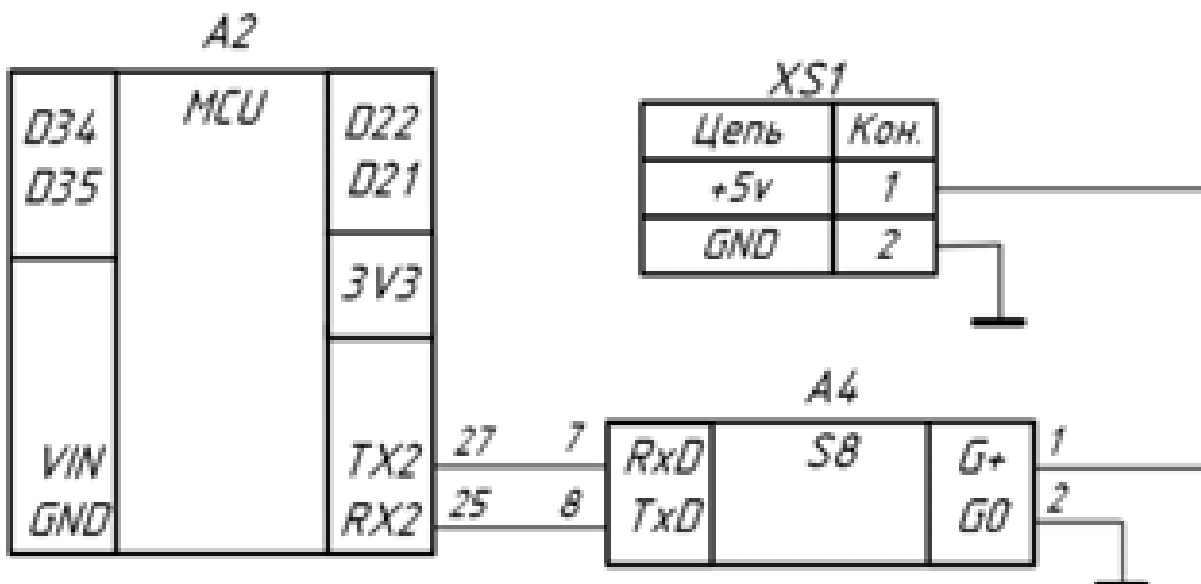


Рисунок 3.3 - Фрагмент схемы электрической принципиальной подключения датчика углекислого газа

UART означает универсальный асинхронный приёмопередатчик и представляет собой простой двухпроводной протокол для обмена последовательными данными. Асинхронность означает отсутствие общего тактового сигнала, поэтому для работы UART необходимо настроить одинаковую скорость передачи данных или битовую скорость на обеих сторонах соединения. Стартовые и стоповые биты используются, чтобы указать, где начинаются и заканчиваются пользовательские данные, или для «кадрирования» данных. Необязательный бит четности может использоваться для обнаружения однобитовых ошибок. UART по-прежнему является широко используемым протоколом последовательной передачи данных, но в последние годы был заменен в некоторых областях применения такими технологиями, как SPI, I2C, USB и Ethernet.

Схема подключения датчика микрочастиц пыли представлена на рисунке 3.4.

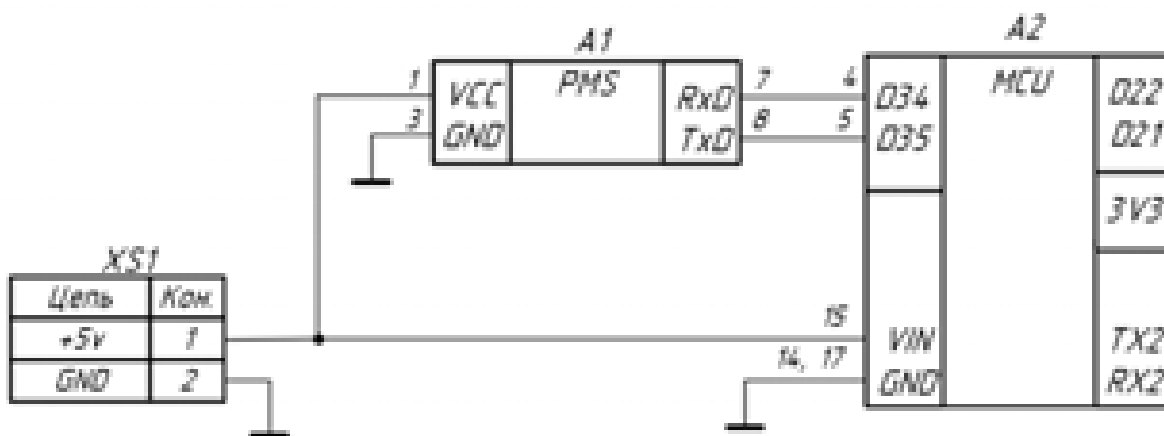


Рисунок 3.4 - Фрагмент схемы электрической принципиальной подключения датчика микрочастиц пыли

На основании фрагментов принципиальной схемы была синтезирована принципиальная электрическая схема. Полная принципиальная схема соединения всех модулей представлена в приложении А, чертеж А.2. Перечень элементов к схеме электрической принципиальной приведен в приложении А,

чертеж А.1. Фотография смонтированного прототипа устройства приведена на рисунке 3.5.

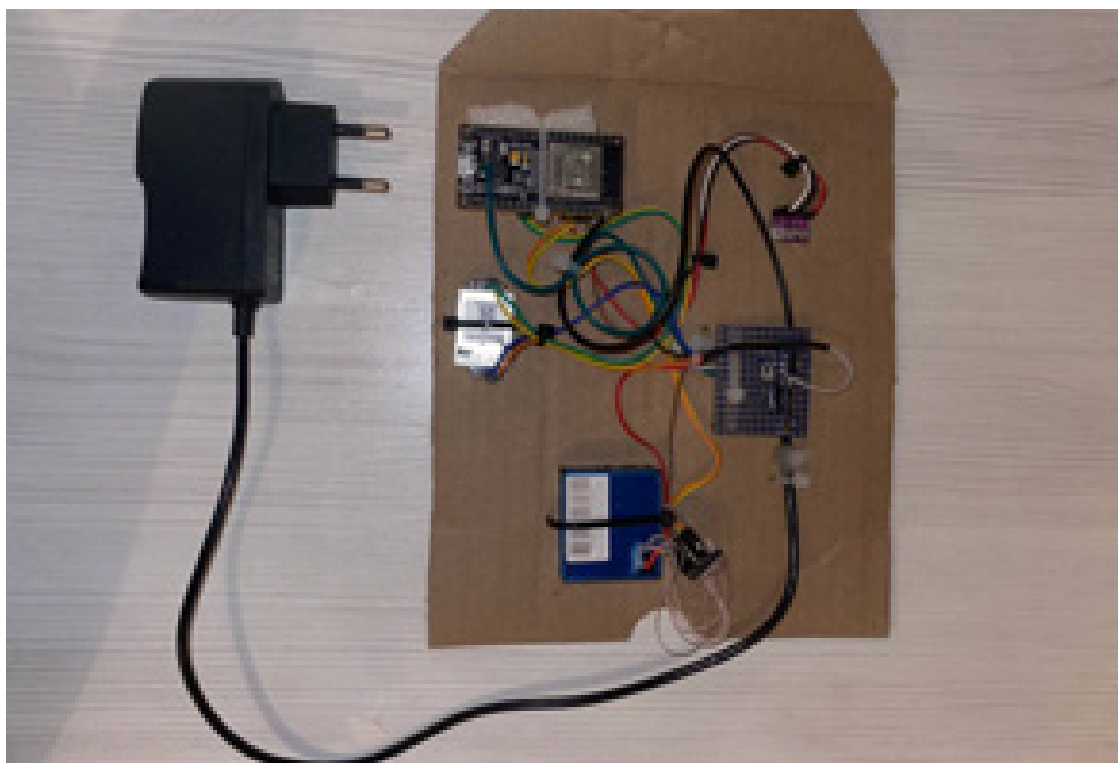


Рисунок 3.5 - Прототип устройств

Вывод по разделу три

В данной главе были составлены структурные электрические схемы и на основе которых была разработана принципиальная схема подключения датчиков к микроконтроллерному модулю. Также по результатам главы был собран прототип устройства.

#### 4 РАЗРАБОТКА ПРОГРАММНОЙ ЧАСТИ КОМПЛЕКСА

Программная часть комплекса будет отвечать за реализацию считывания показаний датчиков и подключения устройства к доступной сети Wi-Fi и работу телеграмм-бота. Программный код будет реализовываться на языке программирования C++. Также следует выбрать наиболее подходящий фреймворк для разработки между Espressif и Arduino.

Фреймворк Espressif - это набор средств разработки программного обеспечения (SDK) для микроконтроллеров ESP32 и ESP8266. Фреймворк Espressif сложнее, чем фреймворк Arduino, и предназначен для более опытных пользователей, которым требуется больший контроль над своими проектами. Он предоставляет более полный набор инструментов и библиотек, включая поддержку Wi-Fi, Bluetooth и других коммуникационных протоколов. Фреймворк Espressif также включает операционную систему реального времени (RTOS), которая позволяет более эффективно выполнять многозадачность и планировать задачи [12].

В свою очередь, Arduino - это популярная платформа с открытым исходным кодом, используемая для создания электронных проектов. Она представляет собой простую в использовании программную и аппаратную платформу для создания интерактивных объектов и прототипов. Она основана на упрощенной версии языка программирования C++ и предоставляет набор библиотек, облегчающих работу с различными датчиками, исполнительными механизмами и другими аппаратными компонентами. Одним из главных преимуществ фреймворка Arduino является его простота и легкость в использовании, что делает наиболее подходящей для данного проекта.

В целом, основное различие между фреймворком Arduino и фреймворком Espressif заключается в уровне их сложности и целевой аудитории, на которую они рассчитаны. Фреймворк Arduino проще и легче в использовании, в то время как фреймворк Espressif более сложен и предоставляет более продвинутые функции для опытных разработчиков, так

как возможностей Arduino для реализации программного комплекса будет достаточно, следует выбрать именно эту платформу для разработки.

Первым этапом знакомства пользователя с устройством будет подключение его к Wi-Fi. Для этого можно использовать автоматическое подключение к Wi-Fi с помощью библиотеки WiFiManager. Чтобы подключить устройство к сети, пользователю нужно будет с любого удобного устройства подключиться к устройству по Wi-Fi в режиме точки доступа (имя точки доступа будет ESP), а затем пользователь автоматически направится на страницу для подключения. С помощью веб-интерфейса пользователь сможет выбрать подходящую сеть из списка, ввести пароль и привязать устройства к данной сети. Изображение интерфейса выбора Wi-Fi сети показано на рисунке 4.1.

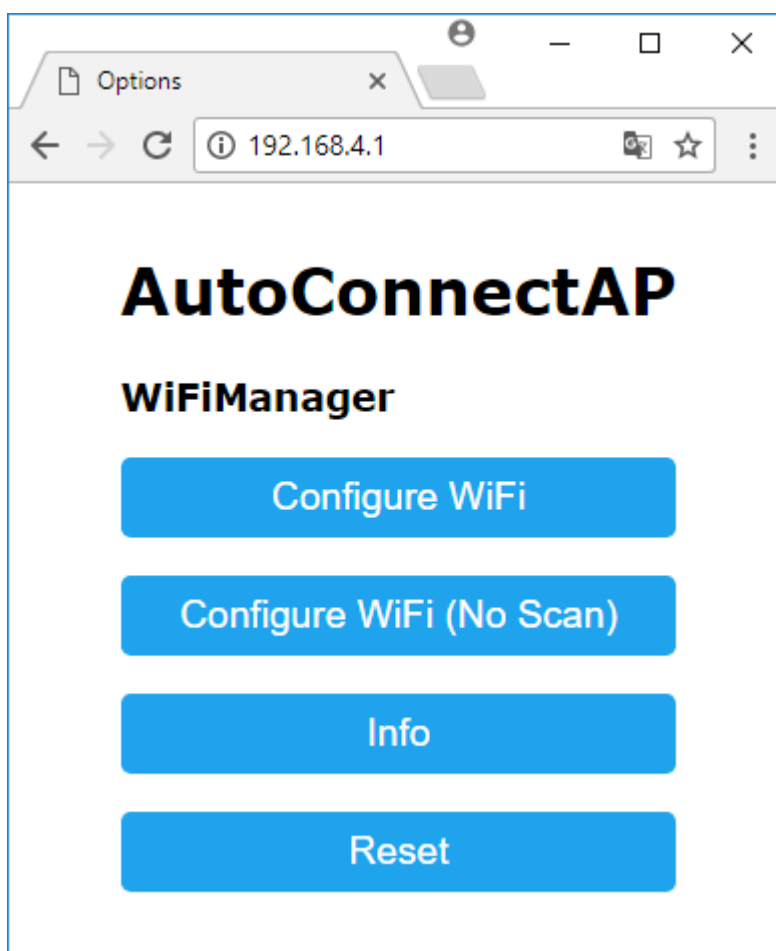


Рисунок 4.1 - Интерфейса выбора Wi-Fi сети

Второй этап - это взаимодействие пользователя с телеграмм-ботом. Функционал бота был уже ранее представлен во 2 главе. В его задачи включены следующие требования:

- приветствие пользователя и ознакомление его с функционалом;
- своевременная сигнализация об ухудшении измеряемых параметров;
- возможность детализировать данные за последний период.

Так как программирование будет происходить на базе микроконтроллера ESP32, следует воспользоваться преимуществами данного решения и использовать операционную систему FreeRTOS данного микропроцессора для реализации многозадачности в программе [11]. FreeRTOS — многозадачная операционная система реального времени (ОСРВ) для встраиваемых систем [10]. Благодаря данной операционной системе будет возможность создавать различные задачи, передача данных в которых будет происходить с помощью очередей. Схема передачи данных между задачами представлена на рисунке 4.2.

В программе потребуется реализовать несколько задач:

- 1) задача для считывания показаний температуры и относительной влажности с датчика BME280;
- 2) задача для считывания показаний углекислого газа с датчика SenseAir S8;
- 3) задача для считывания показаний количества микрочастиц пыли PM2.5 с датчика PMSA003;
- 4) обработка данных на выход за пределы нормы;
- 5) сохранение истории;
- 6) генерация графиков;
- 7) задача взаимодействия с телеграмм-ботом.



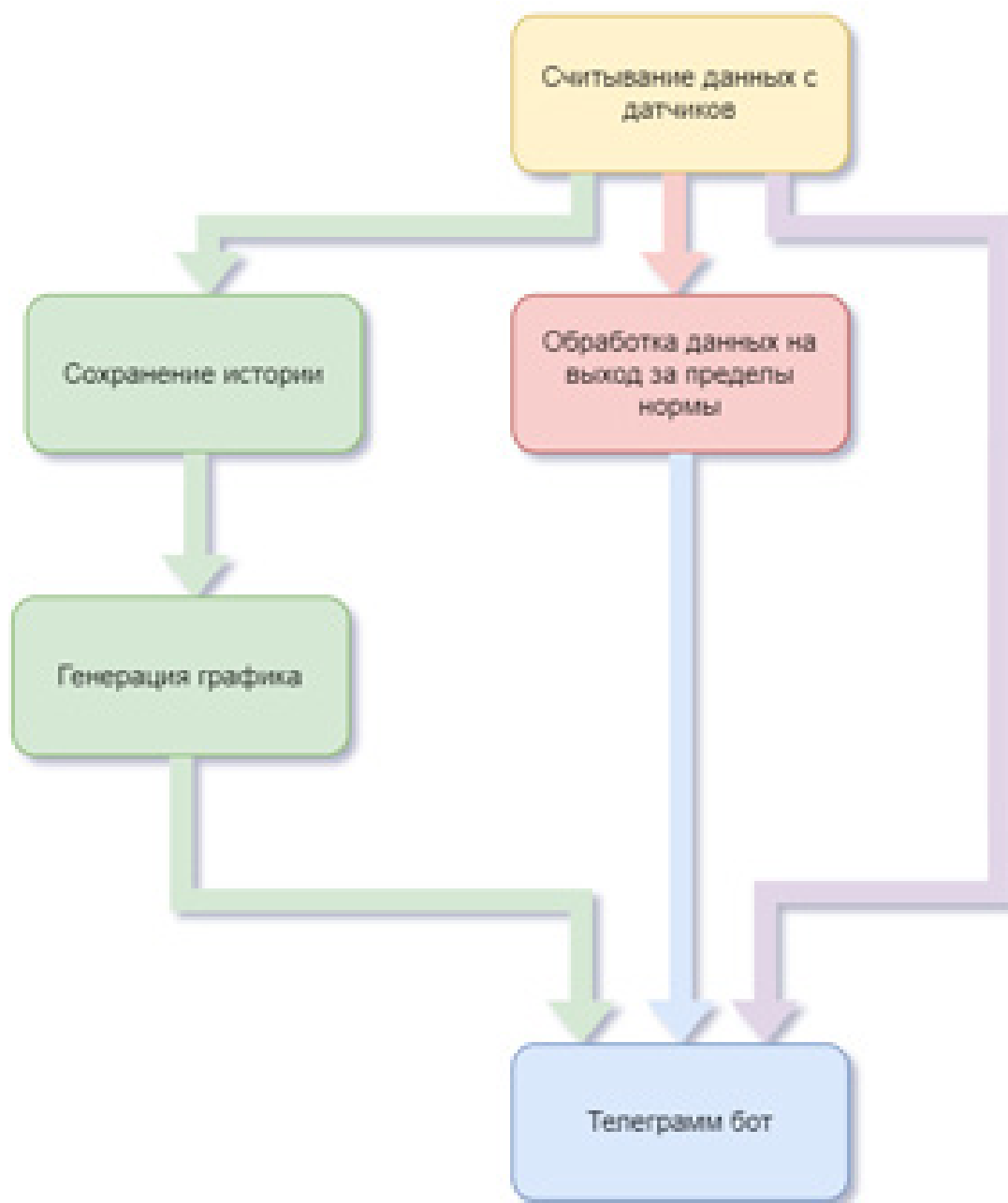


Рисунок 4.2 - Схема передачи данных между задачами

Первые три задачи будут иметь однотипный алгоритм. Поэтому на общей схеме они представлены в виде одного блока. В каждом датчике следует считать данные и отправить в соответствующую очередь. Алгоритмы этих задач представлены на рисунке 4.3. Данные, считанные с датчиков, будут передаваться в очередь для дальнейшей их обработки.



Рисунок 4.3 - Схема алгоритма считывания показаний с датчиков

Четвертая задача должна считывать данные из очереди и проверять не выходят ли показания за пределы. Для параметра температуры пределом будет являться 25°C. Для относительной влажности пределом является 60%. Для количества содержания углекислого газа - 1000 ppm. Для количества микрочастиц пыли PM2.5 - 10 мкг/м<sup>3</sup>. Алгоритм работы данной задачи представлен на рисунке 4.4.

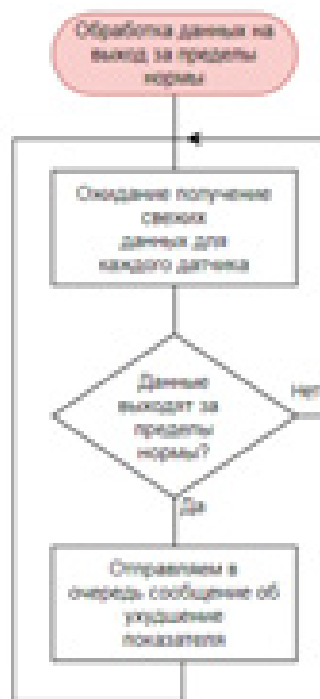


Рисунок 4.4 - Схема алгоритма обработки данных на выход за пределы нормы.

В задаче сохранения истории следует реализовать сбор данных со всех датчиков и последующее сохранение их в памяти устройства. Задача должна ожидать поступление данных посредством очередей. После получения следует записать данные от каждого датчика в соответствующий массив. Алгоритм задачи реализующей сохранение истории представлен на рисунке 4.5.



Рисунок 4.5 - Схема алгоритма сохранения истории

Для защиты от неверно интерпретируемых данных будут использованы мьютексы. Мьютекс в FreeRTOS - это средство синхронизации доступа к общему ресурсу между несколькими задачами в системе [13].

Мьютекс может находиться в двух состояниях: свободном и занятом. Задача может захватить мьютекс и заблокировать его для других задач, чтобы сделать доступным общий ресурс. Когда задача закончила использование ресурса, она освобождает мьютекс, чтобы другие задачи могли использовать общий ресурс.

Мьютексы в FreeRTOS будут использованы для синхронизации доступа задач сохранения данных и генерации графиков к общей области памяти.

Задача генерации графиков представляет собой алгоритм, в котором происходит ожидание данных от задачи, обрабатывающей сообщения от пользователя в телеграмм-боте. В случае если пользователь запросит вывести график, в задачу будет добавлено сообщение о необходимости сгенерировать ссылку на график. Для генерации графика будет использоваться доступный сервис Google Charts. Запрос будет отправляться с данными, взятыми из мьютекса. После генерации графика, алгоритм отправит в очередь ссылку на график, а задача по обработке сообщений в телеграмм отправит эту ссылку пользователю. Алгоритм задачи представлен на рисунке 4.6.

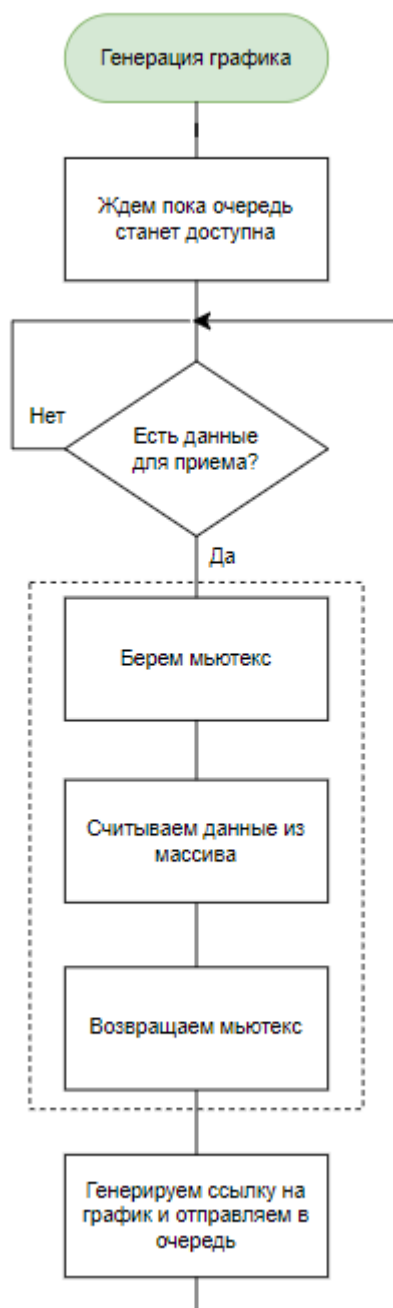


Рисунок 4.6 - Схема алгоритма генерации графика

В заключительной задаче следует отслеживать поступление новых сообщений от пользователя в телеграмм-боте. В случае, если пользователь требует отобразить текущие показания датчиков, задаче необходимо считать данные, поступившие в очереди от датчиков и отправить пользователю свежие показания в телеграмм-бот. Также задача будет отправлять пользователю график в случае запроса и сохранять id пользователя для будущей связи с ним.

Схема алгоритма задачи приема сообщений от пользователя в телеграмм-боте представлена на рисунке 4.7.

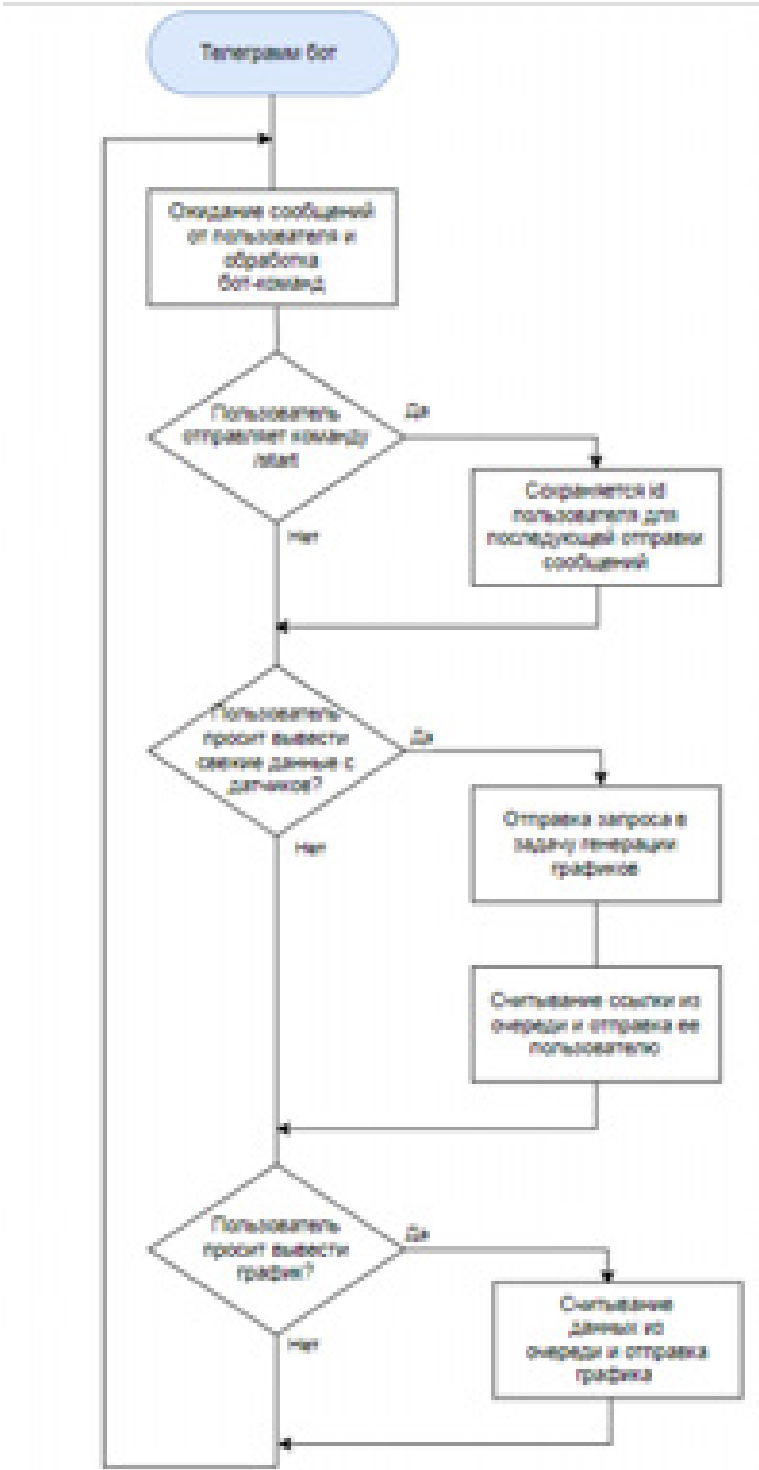


Рисунок 4.8 - Схема алгоритма задачи приема сообщений от пользователя в Телеграмм-Боте

Вывод по разделу четыре

В данной главе спроектирован алгоритм программы. Он состоит из 8 параллельных задач, каждая из которых реализует свой программный алгоритм. Спроектированы блок-схемы алгоритмов и схема передачи данных между задачами.

## 5 РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА

По результатам 4 главы был написан программный код и реализованы алгоритмы. Фрагменты листинга кода представлены в приложении Б. В ходе работы программы были проведены тестирования. Снимки экрана приведены на рисунках 5.1 - 5.4. На рисунке 5.1 изображена работа телеграмм-бота при выполнении стартовой команды. Происходит идентификация пользователя и открытие меню.

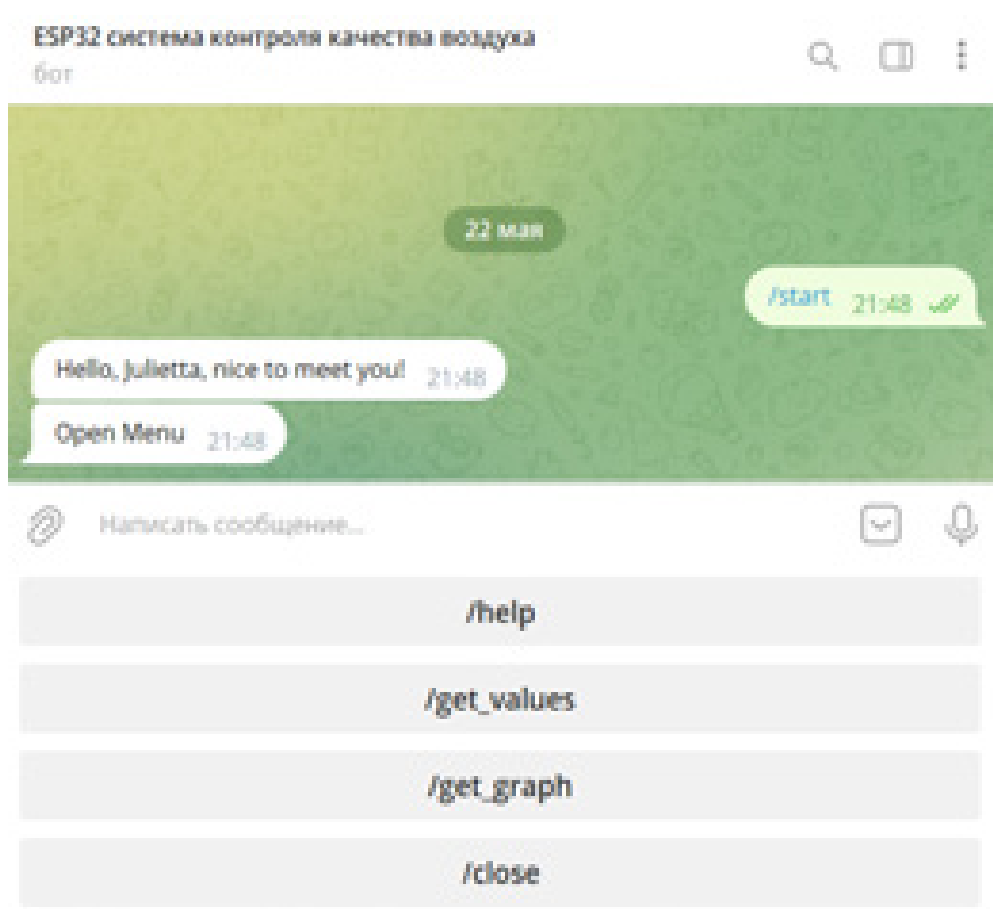


Рисунок 5.1 - Снимок экрана выполнение команды start

На рисунке 5.2 приведен снимок экрана при вызове команды help. Бот в ответ на команду отправляет список команд и их описание.



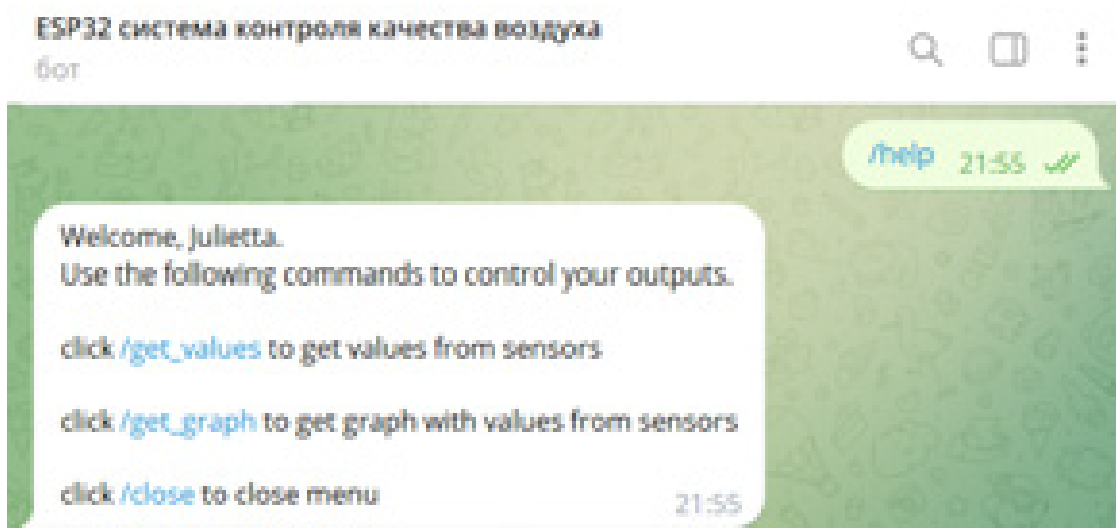


Рисунок 5.2 - Снимок экрана выполнения команды help

На рисунке 5.3 представлен снимок экрана при вызове команды `get_values`, которая присылает пользователю актуальные показания датчиков.

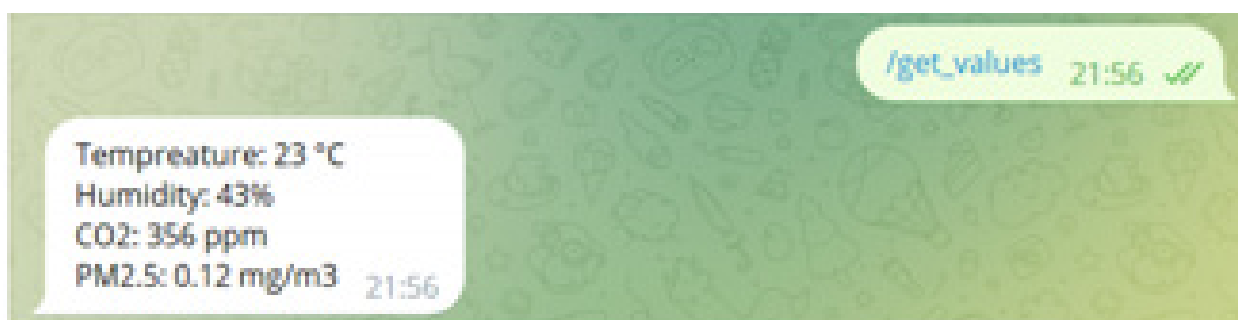


Рисунок 5.3 - Снимок экрана при вызове команды `get_values`

На рисунке 5.4 представлен график параметров при запросе пользователем графика. В ответе на команду `get_graph` бот присылает ссылку на график изменения последних актуальных показаний датчиков прибора.



Рисунок 5.4 - Изображение графика

В качестве тестирования были намеренно изменены параметры воздуха для проверки исправности работы задачи проверяющей параметры на выход за пределы нормы. Результаты тестирования приведены в таблице 6.

Таблица 6 - Тестирование прототипа устройства

Проверяемый параметр	Замеренное значение датчиком	Наличие сообщения о выходе параметра за пределы нормы
Температура	31	Да
Влажность	89%	Да
Углекислый газ	715 ppm	Да
Содержание микрочастиц пыли	512	Да

По результатам таблицы все тесты прошли успешно. Снимки экрана при прохождении тестирования представлены на рисунках 5.5 - 5.7

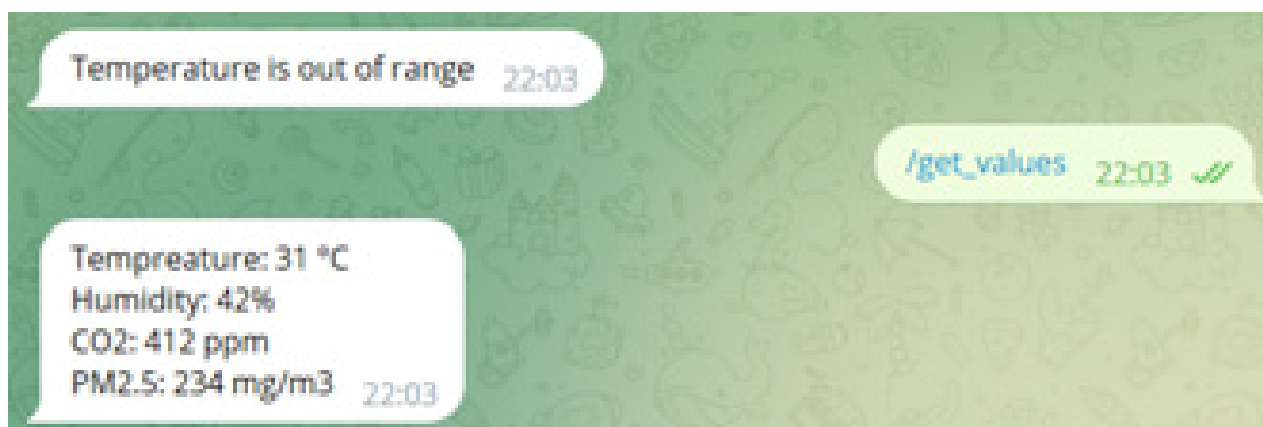


Рисунок 5.5 - Снимок экрана тест первый

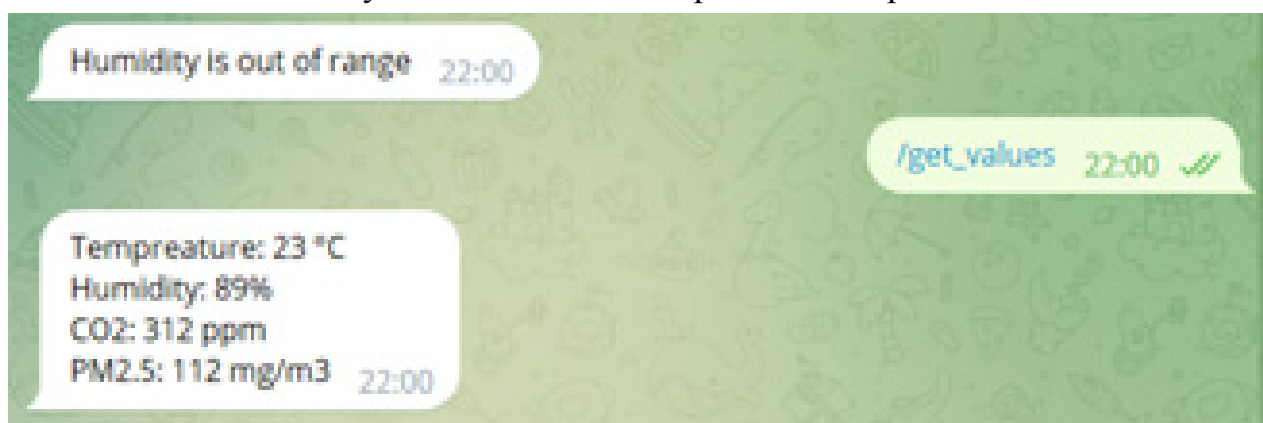


Рисунок 5.6 - Снимок экрана тест второй

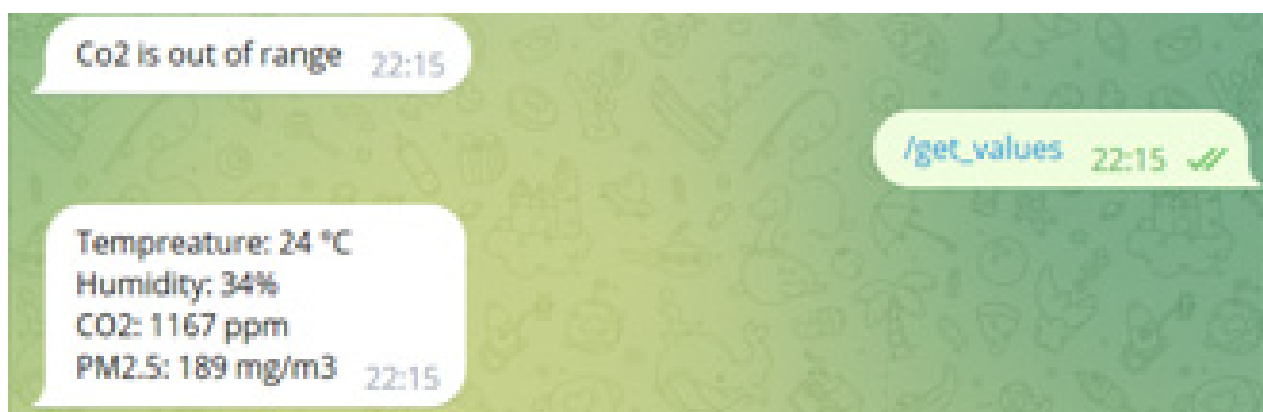


Рисунок 5.6 - Снимок экрана тест третий

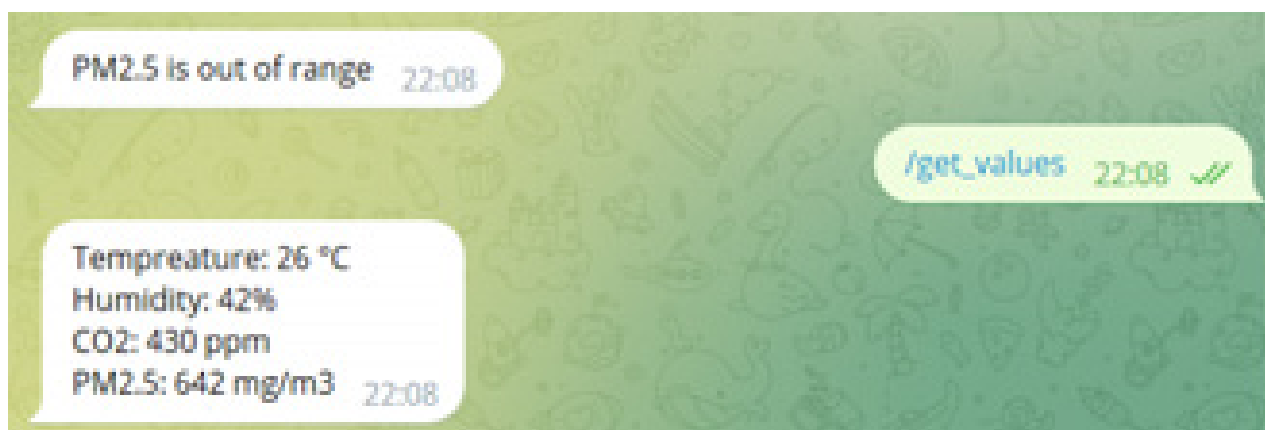


Рисунок 5.7 - Снимок экрана тест четвертый

Во второй главе при постановке задачи были сформулированы определенные требования к разрабатываемому прибору. Соблюдение всех требований можно увидеть в таблице 7.

Таблица 7 - Таблица анализа выполнения поставленных требований

Требования	Реализация
Прибор должен измерять качество воздуха по показателям: температура, влажность, содержание углекислого газа ( $CO_2$ ) и содержание мелких частиц пыли (PM2.5).	К устройству подключены датчик температуры и влажности, датчик углекислого газа и датчик, измеряющий содержание частиц пыли. В программной части реализован алгоритм считывания данных для всех датчиков. При тестировании устройство корректно работает.
Необходимо осуществить оповещение пользователя в случае выхода значений, полученных с датчиков, за пределы нормы, для своевременного проветривания помещений.	В программной части реализован алгоритм, осуществляющий проверку данных, считанных с датчиков на выход за предел. При тестировании бот успешно отправлял сообщения в телеграмм для оповещения о выходе параметров за пределы.
Следует организовать возможность пользователя получать уведомления от устройства посредством телеграмм-бота.	Алгоритм релизован посредством телеграмм-бота. При тестировании прототип успешно работает.

## Продолжение таблицы 7

Прибор должен обеспечивать возможность пользователя получать собранные данные с датчиков в любой момент времени посредством телеграмм-бота.	В телеграмм-боте пользователь может отправлять запросы на вывод данных от пользователя и получать ответ в виде актуальных параметров либо в виде графика. В ходе тестирования алгоритм успешно работал.
---------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

По результатам таблицы можно видеть, что все тесты успешно прошли проверку и каждое требование, поставленное изначально при планировании устройства, было выполнено.

Вывод по разделу пять

В результате загрузки программного обеспечения на прототип устройства все тесты были пройдены успешно. Все начальные требования выполнены. Прототип готов к использованию и передан заказчику.

## ЗАКЛЮЧЕНИЕ

В результате работы было разработано устройство для контроля качества воздуха. Прототип прошел все тестирования и готов к эксплуатации. Для решение поставленной задачи, были проведены следующие исследования:

1. Проведен аналитический обзор литературы и проанализированы существующие методы решения актуальной задачи. Также проведен обзор существующих аналогов на рынке.
2. В рамках работы разработана структура программно-аппаратного комплекса и выбраны необходимые компоненты.
3. Спроектирована архитектура аппаратной части комплекса, разработаны структурные и принципиальные схемы, на основе которых был сформирован прототип устройства.
4. Осуществлена разработка алгоритмов для решения поставленных задач.
5. Реализована программная часть комплекса и проведено тестирование прототипа. В результате тестирования выяснилось, что все поставленные задачи были выполнены. Прототип работает корректно и готов к использованию.

Таким образом, в данной работе были выполнены все поставленные требования и реализованы все задачи. Данное исследование может быть использовано для дальнейшей научной разработке в рамках данной предметной области.

В дальнейшем для данного прототипа планируется разработка корпуса и улучшение функционала.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 ГОСТ 30494-2011 Здания жилые и общественные. Параметры микроклимата в помещениях / Межгосударственной научно-технической комиссией по стандартизации, техническому нормированию и оценке соответствия в строительстве. - М.: Изд-во стандартов, 2013. - 15 с.
- 2 Частицы PM2.5: что это, откуда и почему об этом все говорят [Электронный ресурс]: URL: <https://www.airkaz.org/pm25.php> (Дата обращения 12.02.2023).
- 3 Руководство пользователя / Торговая марка «ДАДЖЕТ». Артикул KIT MT 8060. - 4 с.
- 4 Монитор качества воздуха Clear Grass / Qingping Air Detector - CGS1. [Электронный ресурс]: URL: <https://mi-house.ru/testeri/monitor-kachestva-vozduha-clear-grass-qingping-air-detector-cgs1-white.html> (Дата обращения: 05.02.2023).
- 5 Официальный сайт компании life control [Электронный ресурс] URL: <https://lifecontrol.ru/devices/air-sensor/> (Дата обращения: 05.02.2023).
- 6 Интернет площадка, информирующая о новинка в сфере информационных технологий 4PDA. [Электронный ресурс] URL: [https://4pda.to/2023/02/15/409824/ikea\\_vypustila\\_vysokotochnyj\\_datchik\\_kachestva\\_vozdukha/](https://4pda.to/2023/02/15/409824/ikea_vypustila_vysokotochnyj_datchik_kachestva_vozdukha/) (Дата обращения: 29.02.2023).
- 7 Анализатор Noyafa JMS12C [Электронный ресурс] URL: <https://www.noyafa.com/ru/products/air-quality-tester-jms-12.html> (Дата обращения: 05.02.2023).
- 8 Руководство пользователя Qingping Technology CGDN1 Air Monitor Lite Detector - 5 с.
- 9 Руководство по эксплуатации. Анализатор воздуха VAKIO Atmosphere. - 3 с.
- 11 Справочное руководство / Reference Manual for FreeRTOS version 10.0.0 issue 1. 2017 - 400 с.

12 Белова, А. В. Создаем устройства на микроконтроллерах // А. В. Белова. – Санкт Петербург: Наука и Техника, 2007. – 295 с.;

13 Официальный сайт FreeRTOS URL: <https://freertos.org> (Дата обращения: 12.04.2023).



## ПРИЛОЖЕНИЕ А



## ПРИЛОЖЕНИЕ Б

### Фрагменты исходного кода программной части системы

#### Листинг Б.1 - Исходный код для функции setup()

```
#include <stdio.h>

#include "FreeRTOS.h" // lib for queue

String chat_id;

// creating tasks's handles
TaskHandle_t task_read_temp_and_humidity = NULL; //all
TaskHandle_t task_read_co2 = NULL;
TaskHandle_t task_read_pm25 = NULL;
TaskHandle_t task_logger = NULL;
TaskHandle_t task_check_out_of_border = NULL;
TaskHandle_t task_send_message = NULL;

// creating queues
QueueHandle_t queue_for_check_border = NULL;
QueueHandle_t queue_messages = NULL;
QueueHandle_t queue_for_logger = NULL;

// struct for sensor's values
struct struct_data {
    char name; // name of parameter
    int value;
};

void setup() {
    Serial.begin(115200);
    connectWiFi();
    char success = 'y';
    unsigned status;
    status = bme.begin();
    S8_serial.begin(S8_BAUDRATE);
    sensor_S8 = new S8_UART(S8_serial);
    pms.begin(pmsSerial);
    pms.setMode(PASSIVE);
    queue_for_check_border = xQueueCreate(32, sizeof(struct_data));
    if (queue_for_check_border == NULL) {
        Serial.println("Error creating the queue");
        success = 'n';
    }
}
```

## Продолжение листинга Б.1

```

} else Serial.println("SUCCESS1");

queue_messages = xQueueCreate(1, sizeof(char));
if (queue_messages == NULL) {
    Serial.println("Error creating the queue");
    success = 'n';
} else Serial.println("SUCCESS2");
queue_for_logger = xQueueCreate(32, sizeof(struct_data));
if (queue_for_logger == NULL) {
    Serial.println("Error creating the queue");
    success = 'n';
} else Serial.println("SUCCESS3");
// creating and checking tasks
xTaskCreate(read_temp_and_humidity, "task_read_temp_and_humidity", 10000,
NULL, 1, &task_read_temp_and_humidity); //
if (task_read_temp_and_humidity == NULL) {
    Serial.println("task_read_temp_and_humidity isn't ok");
    success = 'n';
} else Serial.println("SUCCESS4");
xTaskCreate(read_co2, "task_read_co2", 10000, NULL, 1, &task_read_co2);
if (task_read_co2 == NULL) {
    success = 'n';
    Serial.println("task_read_co2 isn't ok");
} else Serial.println("SUCCESS5");
    xTaskCreate(read_pm25, "task_read_pm25", 10000, NULL, 1,
&task_read_pm25);
if (task_read_pm25 == NULL) {
    Serial.println("task_read_pm25 isn't ok");
    success = 'n';
} else Serial.println("SUCCESS6");
xTaskCreate(logger, "task_logger", 30000, NULL, 1, &task_logger);
if (task_logger == NULL) {
    Serial.println("task_logger isn't ok");
    success = 'n';
} else Serial.println("SUCCESS7");
xTaskCreate(check_out_of_border, "task_check_out_of_border", 20000, NULL,
1, &task_check_out_of_border);
if (task_check_out_of_border == NULL) {
    Serial.println("task_check_out_of_border isn't ok");
    success = 'n';
} else Serial.println("SUCCESS8");
    xTaskCreate(send_message, "task_send_message", 30000, NULL, 1,
&task_send_message);

```

```

if (task_send_message == NULL) {

    Serial.println("task_send_message isn't ok");
    success = 'n';

```

### Продолжение листинга Б.1

```

} else Serial.println("SUCCESS9");
if (success == 'n') Serial.println("REBOOT");
else Serial.println("SUCCESS");
// connect to tg bot
bot.attach(newMsg);
}

```

### Листинг Б.2 Исходный код для задачи по считыванию данных с датчиков

```

#include <Wire.h>
#include <SPI.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
#include "s8_uart.h"
#include "BLINKER_PMSX003ST.h"

Adafruit_BME280 bme; // I2C
unsigned long delayTime;

#if defined(USE_SOFTWARE_SERIAL) || defined(ARDUINO_ARCH_RP2040)
    #define S8_RX_PIN 2
    #define S8_TX_PIN 2
#else
    #define S8_UART_PORT 1
#endif
#ifdef USE_SOFTWARE_SERIAL
    SoftwareSerial S8_serial(S8_RX_PIN, S8_TX_PIN);
#else
    #if defined(ARDUINO_ARCH_RP2040)
        REDIRECT_STDOUT_TO(Serial)
        UART S8_serial(S8_TX_PIN, S8_RX_PIN, NC, NC);
    #else
        HardwareSerial S8_serial(S8_UART_PORT);
    #endif
#endif

HardwareSerial pmsSerial(2);
BLINKER_PMSX003ST pms;

```

```
void read_temp_and_humidity(void * pvParameters){
```

Продолжение приложения Б

```
    int temp;
    struct_data time_struct;
    int humidity;
    struct_data humidity_struct;
```

## Продолжение листинга Б.2

```
for(;;){
    temp = bme.readTemperature();
    time_struct.name = 't';
    time_struct.value = temp ;
    if (!xQueueSend(queue_for_check_border, &time_struct, 500))
Serial.println("Fail to send");
    if (!xQueueSend(queue_for_logger, &time_struct, 500))
Serial.println("Fail to send");
    humidity = bme.readHumidity();
    humidity_struct.name = 'h';
    humidity_struct.value = humidity;

    if (!xQueueSend(queue_for_check_border, &humidity_struct, 500))
Serial.println("Fail to send");
    if (!xQueueSend(queue_for_logger, &humidity_struct, 500))
Serial.println("Fail to send");
    delay(delayTime);
}
}
```

```
void read_co2(void * pvParameters){
    int co2;
    struct_data co2_struct;
    for(;;){
        sensor.co2 = sensor_S8->get_co2();
        co2 = sensor.co2;
        co2_struct.name = 'c' ;
        co2_struct.value = co2;
        if (!xQueueSend(queue_for_check_border, &co2_struct, 1000))
Serial.println("Fail to send");
        if (!xQueueSend(queue_for_logger, &co2_struct, 500))
Serial.println("Fail to send");
        delay(delayTime);
    }
}
```

```
void read_pm25(void * pvParameters){
```

```

int pm25;

pms.request();
if(!pms.read()){
    return;
}
struct_data pm25_struct;

```

Продолжение приложения Б

### Продолжение листинга Б.2

```

for(;;){
    pm25 = pms.getPmAt(2.5);
    pm25_struct.name = 'p';
    pm25_struct.value = pm25;

if (!xQueueSend(queue_for_check_border, &pm25_struct, 500))
Serial.println("Fail to send");
        if (!xQueueSend(queue_for_logger, &pm25_struct, 500))
Serial.println("Fail to send");

        delay(delayTime);
    }
}

```

### Листинг Б.3 - Исходный код для задачи по сохранению истории данных

```

void logger( void * pvParameters ){
    int temp_arr[1000];
    int humidity_arr[1000];
    int co2_arr[1000];
    int pm25_arr[1000];
    struct_data struct_for_readings;
    int index_temp = 0;
    int index_humidity = 0;
    int index_co2 = 0;
    int index_pm25 = 0;

    for(;;){
        if (xQueueReceive(queue_for_logger, &struct_for_readings, 1000)) {
            if (struct_for_readings.name == 't'){
                temp_arr[index_temp] = struct_for_readings.value;
                index_temp = (index_temp++)%10;
                Serial.println("Temp " + String(struct_for_readings.value));
            }
        }
    }
}

```

```

if (struct_for_readings.name == 'h'){
    humidity_arr[index_humidity] = struct_for_readings.value;
    index_humidity = (index_humidity++)%10;
    Serial.println("Hum " + String(struct_for_readings.value));
}

```

```

if (struct_for_readings.name == 'c'){
    co2_arr[index_co2] = struct_for_readings.value;

```

### Продолжение листинга Б.3

```

    index_co2 = (index_co2++)%10;
    Serial.println("CO2 " + String(struct_for_readings.value));
}

```

```

if (struct_for_readings.name == 'p'){
    temp_arr[index_pm25] = struct_for_readings.value;
    index_pm25 = (index_pm25++)%10;
    Serial.println("PM2.5 " + String(struct_for_readings.value));
}

```

```

}
}
}

```

### Листинг Б.4 - Исходный код для задачи по проверки данных на выход из массива

```

void check_out_of_border(void * pvParameters){
    struct_data recording;
    char message;
    int value;
    int co2_border = 1000;
    int pm25_border = 600;
    int temp_border = 28;
    int humidity_border = 60;
    for(;;){
        if (xQueueReceive(queue_for_check_border, & recording, 1000)) {
            value = recording.value;
            if (recording.name == 't'){
                if (value > temp_border){
                    message = '1';
                    if(!xQueueOverwrite(queue_sos_message, &message))
                        Serial.println("Failed to send item to queue_sos_message");
                }
            }
        }
    }
}

```



```

    }

    }
    if (recording.name == 'h') {
        if (value > humidity_border){
            message = '2';
                if(!xQueueOverwrite(queue_sos_message, &message))
Serial.println("Failed to send item to queue_sos_message");
        }
    }
    if (recording.name == 'c') {

```

#### Продолжение листинга Б.4

```

        if (value > co2_border){
            message = '3';
                if(!xQueueOverwrite(queue_sos_message, &message))
Serial.println("Failed to send item to queue_sos_message");
        }
    }
    if (recording.name == 'p') {
        if (value > pm25_border){
            message = '4';
                if(!xQueueOverwrite(queue_sos_message, &message))
Serial.println("Failed to send item to queue_sos_message");}}}
    }

```

#### Листинг Б.5 - Исходный код для функции wifi-manager

```

#include <WiFiManager.h>

void connectWiFi() {

    WiFiManager manager;

    bool success = manager.autoConnect("ESP", "password");

    if(!success) Serial.println("Failed to connect");

    else Serial.println("Connected");

}

```