Государственное бюджетное общеобразовательное учреждение
города Москвы "Школа № 641 имени Сергея Есенина"

«Цифровой двойник устройства сбора данных с использованием дополненной реальности»

Выполнили:
Мутьева Александра Кирилловна,
Горбань Дарья Александровна,
Маргушин Роман Евгеньевич,
Саакян Грант Рштунович,
Исаков Тимур Алексеевич

Москва
2021

Оглавление

1. Введение 3
2. Оценка требований 4
   2.1. Анализ технологических требований 4
   2.2. Обоснование выбора программного обеспечения 6
   2.3. Обоснование выбора датчиков и сенсорных модулей 7
3. Сценарий разработки продукта 9
   3.1. Конечный выбор оборудования и обеспечения 9
   3.2. Поток работ 10
4. Конечный комплекс 11
   4.1. Принцип работы кода 12
5. Приложение 1 13
6. Приложение 2 14
7. Приложение 3 15
8. Приложение 4 16
9. Приложение 5 17

Введение

Современные технологии позволяют человеку всего за пару секунд узнать состояние окружающей среды в своем городе. Но, к сожалению, эти технологии работают не так хорошо в больших городах, где погода в разных его концах может очень отличаться. Это создает большие неудобства для различных групп людей. Например, некоторые хронические болезни не дают человеку нормально существовать в сухом воздухе, или же флористы, выращивающие экзотические растения, не могут допустить слишком низкой температуры для своих растений.

Как узнать температуру не в своем городе, не в своем районе, а непосредственно вокруг себя? Справиться с этой проблемой помогает установка, собирающая данные о состоянии окружающей среды.

Оценка требований

Анализ технологических требований

Исходя из применения данная установка, должна отвечать следующим требованиям:

- Исправно работать при температурном режиме воздуха от +5 до +30 градусов;
- Иметь подсистему сбора данных, представляющую собой программно-аппаратный комплекс, состоящий из микроконтроллера и датчиков;
- Быть автономным, что позволит системе работать бесперебойно и информировать пользователя независимо от наличия электричества;
- Устройство должно считывать показания с датчиков и, обрабатывая их, передавать на облачную платформу раз в 10 секунд посредством беспроводной связи и MQTT-запросов. Также должен быть предусмотрен вывод итоговых показаний локально, что позволит контролировать процесс при необходимости;
- Иметь программную подсистему сервера, реализующую функции передач, приема и обработки информации, поступающей от подсистемы сбора данных, а также предполагать под собой базу данных, хранящую информацию об устройствах сбора данных и показаний, им соответствующих;
- Программная подсистема должна содержать в себе пользовательский интерфейс для возможности просмотра базы данных, а также иметь возможность принимать, обрабатывать и передавать поступающие данные посредством MQTT-запросов;
- Иметь программную подсистему визуализации информации в формате дополненной реальности, представляющую собой мобильное приложение, отображающее показания данных в режиме реального

времени при наведении камеры устройства на конечную подсистему сбора данных.

Подсистема сбора данных должна содержать в себе датчики и сенсорные модули, позволяющие установке информировать владельца о таких показателях, как:

- Температура (представленная в градусах Цельсия);
- Влажность воздуха (в %);
- Уровень освещенность в помещении (в %);
- Уровень зашумленности в помещении (в %);
- Уровень содержания углекислого газа (в %);
- Факт наличия или отсутствия движения (в %).
- Требование к точности измерений и выводу данных в приложении дополненной реальности:
- Вывод показаний датчиков производится в формате вещественного числа (с точностью до 2 знака после запятой);
- Вывод данных сопровождается пиктограммами, соответствующими каждому датчику, позволяющими однозначно понять его предназначение.

Обоснование выбора программного обеспечения

Наиболее удобной средой программирования подсистемы сбора данных была выбрана среда Arduino IDE (на базе языка C/C++). Преимуществами среды являются:

- Быстрая загрузка кода в микроконтроллер напрямую из среды программирования. Это экономит время и облегчает процесс обновления кода;
- Подсветка синтаксиса, помогающая быстрее ориентироваться в программе и воспринимать ее;
- В основе среды лежит язык C/C++, что позволяет легко заменять встроенные библиотеки Arduino IDE на аналогичную библиотеку C/C++.

Для реализации подсистемы визуализации информации предпочтительной оказалась межплатформенная среда разработки Unity. Ее основными преимуществами, значительно облегчающими рабочий процесс является поддержка программирования под множество различных платформ без значительных изменений в коде, а также ее популярность. Чем популярнее среда, тем больше всевозможных библиотек и плагинов, значительно ускоряющими процесс разработки, для нее создается.

Формат дополненной реальности удобно использовать интеграцию Vuforia SDK с уже обозначенной выше Unity. Комплект средств разработки предлагает широкий спектр функций и целевых объектов, позволяющих сделать процесс разработки дополненной реальности более гибким.

Выбор микроконтроллера пал на ESP32, относящийся к серии микроконтроллеров с низким энергопотреблением. Главным преимуществом этого микроконтроллера является его небольшой размер. В отличие от других модулей поддерживающих связь с Интернетом, ESP32 подключается к сети без проводов, кроме того микроконтроллер имеет гораздо большее количество пинов, что позволит в случае необходимости дополнить подсистему без дополнительных плат расширения.

Обоснование выбора датчиков и сенсорных модулей

Влажность и температура воздуха

Для соответствия конечного комплекса выдвигаемым требованиям, погрешность в показаниях должна быть минимальной. Существует множество разнообразных датчиков и совмещенных модулей для считывания влажности и температуры окружающей среды, тем не менее предпочтительным является популярный датчик DHT11. Его преимущества заключаются в широком диапазоне температур (0-50 °C) и высокая точность измерений (погрешность менее 5%). Следовательно датчик сможет правильно функционировать в любом закрытом помещении и передавать достоверные показания.

Уровень освещенности

Практически все существующие на данный момент датчики освещенности являются производными фоторезистора. Тем не менее для более точной оценки освещенности помещения дополнительные действия должны быть минимизированы, ведь каждое из них несет в себе свою погрешность. Поэтому простой фоторезистор вкупе с резистором на 10кОм является лучшим вариантом.

Уровень зашумленности

Для измерения уровня зашумленности идеально подходит микрофон совмещенный с аудиоусилителями на базе чипа OPA134. Совмещенный модуль обладает высокой чувствительностью, обеспеченной несколькими усилителями. Кроме того, модуль фиксирует уровень громкости вокруг себя каждый промежуток времени, что ускоряет его работу. Также важным преимуществом модуля является вывод непосредственно уровня громкости, а не сторонних переменных, требующих дальнейшей обработки.

Содержание вредоносных газов

Решение задачи, поставленной перед нами, должно нести максимальную пользу. Поэтому после анализа существующих датчиков газа предпочтение было отдано датчику MQ-2, восприимчивому в большей степени к выхлопным

газам и углеводородной смеси. Находясь в помещении, человек постоянно изменяет количество углекислого газа в нем. Высокое содержание углекислого газа в воздухе приводит к сонливости, потере концентрации и головной боли, но эти проблемы не возникают при регулярном проветривании помещения. Большее практическое применение имеет отслеживание содержания в воздухе выхлопных газов. Как известно, в их состав входят токсичные соединения, таящие куда большую опасность, нежели углекислый газ. Особенно важен контроль качества воздуха в больших городах, где загрязнение атмосферы чрезмерно.

Наличие движения

Предлагаемые рынком датчики движения практически не отличаются в своих характеристиках. Наиболее же подходящим же под стоящие перед нами требования является датчик движения KS0052, имеющий ряд преимуществ:

- Его размер значительно меньше остальных датчиков движения;
- Имеет широкий диапазон рабочей температуры (от -20 до 85℃);
- Максимальная задержка вывода менее 3 секунд.

Кроме того важными преимуществами описанных датчиков является их низкое энергопотребление и небольшие размеры, что, во-первых, позволит установке работать без подзарядки длительное время и, во-вторых, даст возможность сделать подсистему сбора данных достаточно компактной. Также все датчики имеют подходящий под требования диапазон рабочей температуры, что также немаловажно

.

Сценарий разработки продукта

Конечный выбор оборудования и обеспечения

Исходя из проведенного анализа, подсистемы проекта было решено реализовывать следующим образом.

Подсистема сбора данных представляет собой систему из микроконтроллера ESP32 с интегрированным Wi-Fi контроллером и ряда сенсоров:

- Датчик влажности DHT11, собирающий информацию о влажности и температуре воздуха;
- Датчик освещенности представлен фоторезистором;
- Датчик движения KS0052, фиксирующий факт движения;
- Датчик звука представлен сенсорным модулем, состоящим из микрофона и аудиоусилителя на базе чипа OPA134. Данный модуль регистрирует механические волны и возвращает сигнал соответствующий уровню громкости;
- Датчик газа MQ-2, чувствительный к углеводородному газу.

Подсистема сервера представляет собой облачную платформу Интернета вещей, реализующую хранение данных, передачу их в подсистему визуализации, а также выводит данные при команде от пользователя.

Подсистема визуализации данных представляет собой мобильное приложение дополненной реальности на операционной системе Android. Формат дополненной реальности реализован путем интеграции платформ Unity и Vuforia.

Поток работ

Для сокращения времени, требуемого для реализации проекта, было решено внедрить методику AGILE, позволяющую не терять в эффективности на протяжении всей работы. Одним из её достоинств является постепенное составление документации, необходимой на каждом этапе.

На основе проделанного анализа, а также поиска оборудования и программного обеспечения для создания программно-аппаратного комплекса была составлена следующая система, координирующая очередность выполнения операций:

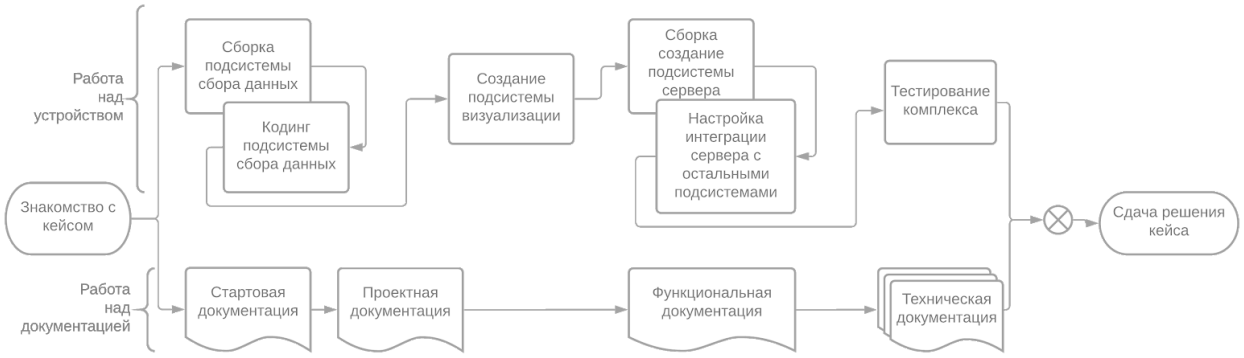

Рис. 1 - сокращенный workflow

С полной версией можно ознакомиться в Приложении 1 или по ссылке.

Конечный комплекс

Конечный программно-аппаратный комплекс представлен совокупностью трех подсистем. Первая занимается сбором данных и последующей отправкой путем беспроводной связи на сервер. Сервер является частью второй подсистемы, реализующей хранение поступающих данных за все время работы установки, вывод актуальных показаний данных для возможности просмотра их пользователем и передачу собранных данных в приложение дополненной реальности. Приложение предоставляет пользователю возможность увидеть актуальные показатели микроклимата помещения путем наведения камеры на подсистему сбора данных.

Наглядное представление работы комплекса можно увидеть в Приложении 2 (ER-модель), Приложении 2 (Функциональная схема) и Приложении 3 (Структурная схема).

Подсистема визуализации данных остается на стадии доработки (выделенный цветом этап), так как возникла проблема с интеграцией MQTT сервера и приложения дополненной реальности. Если проблему не удастся устранить, будет выбран другой сервер.

Рис. 2 - Нынешний этап реализации проекта

Фото- и видеоматериалы можно посмотреть по ссылке.

Принцип работы кода

Программный код, управляющий работой микроконтроллера подсистемы сбора данных выполняет четыре основные функции:

- Обеспечение стабильного подключения к беспроводной сети Wi-Fi, необходимой для бесперебойной передачи данных в хранилище. Эта функция реализована в самом начале программы в теле процедуры “EspMQTTClient client”;
- Считывание показателей с датчиков. Данная функция реализуется методом digitalRead()/analogRead();
- Вывод собранных показателей на монитор последовательного порта. Функция реализуется командами Serial.print()/Serial.println();
- Непосредственная отправка снятых показаний в базу данных. Пересылка осуществляется командой client.publish(“источник”, отправляемое значение).

Последние три функции, описанные выше применяются ко всем датчикам, и, по итогу, мы получаем на выходе пакет дискретных данных, выведенных на монитор последовательного порта (для необходимого контроля работы) и загруженных на MQTT сервер.

Полный код можно найти в Приложении 5.

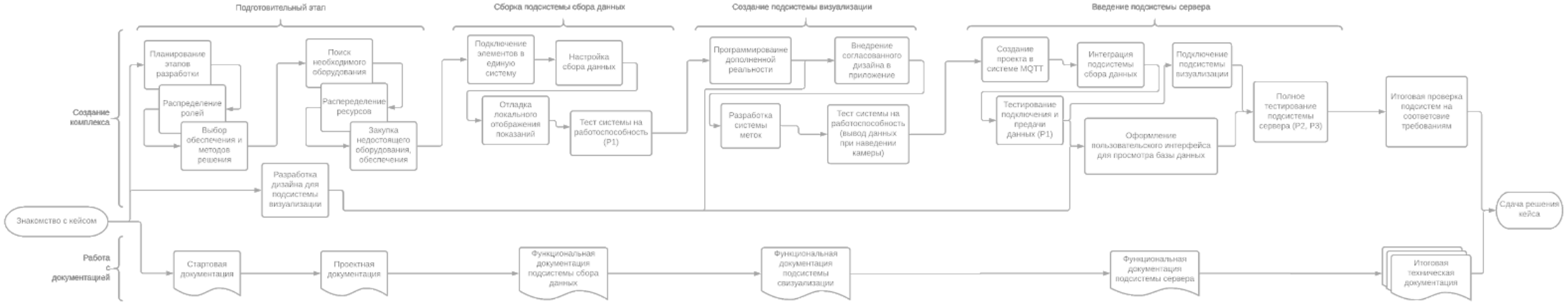

Рис. 3 - Полный workflow

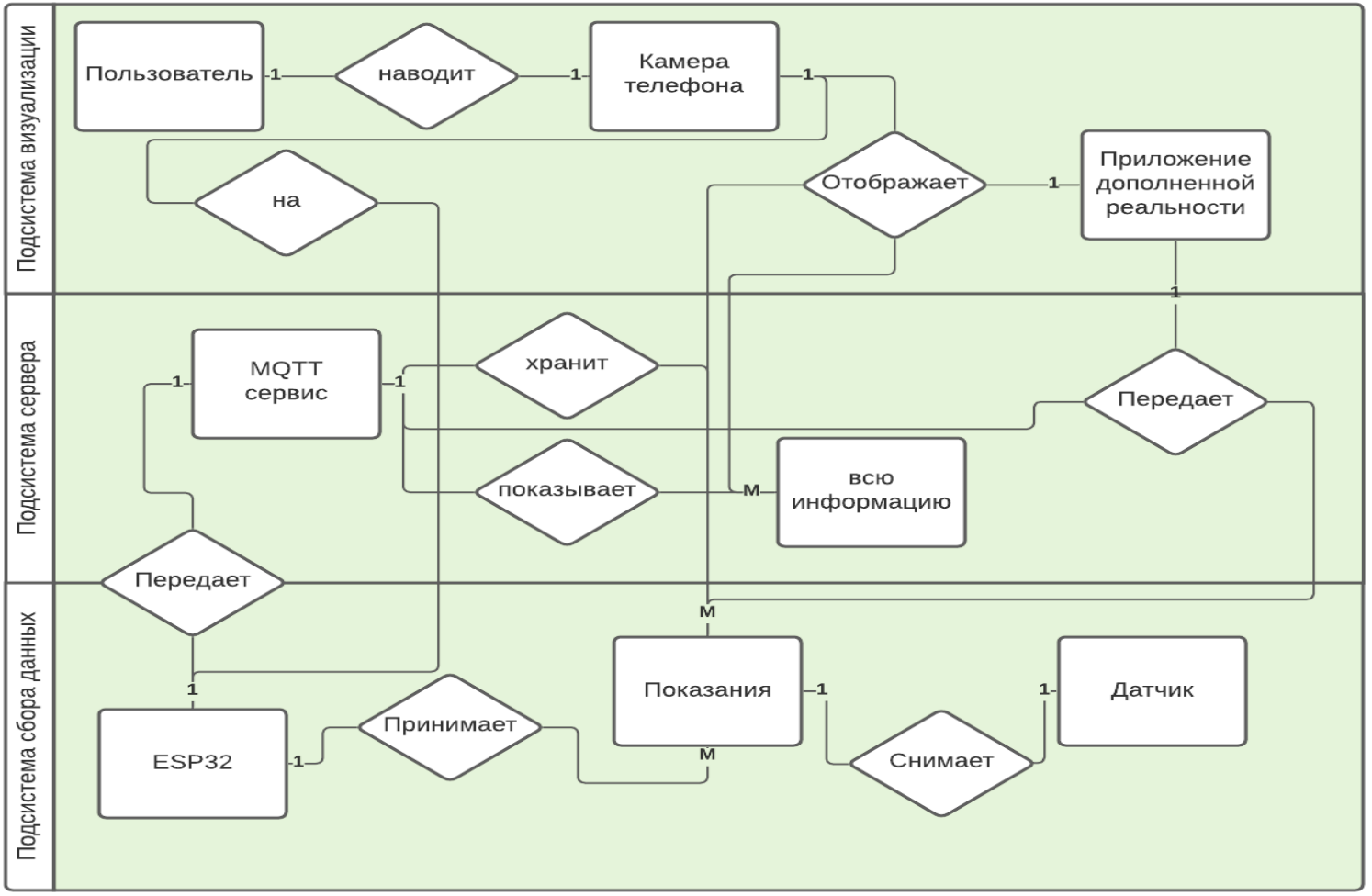

Рис. 4 - ER модель

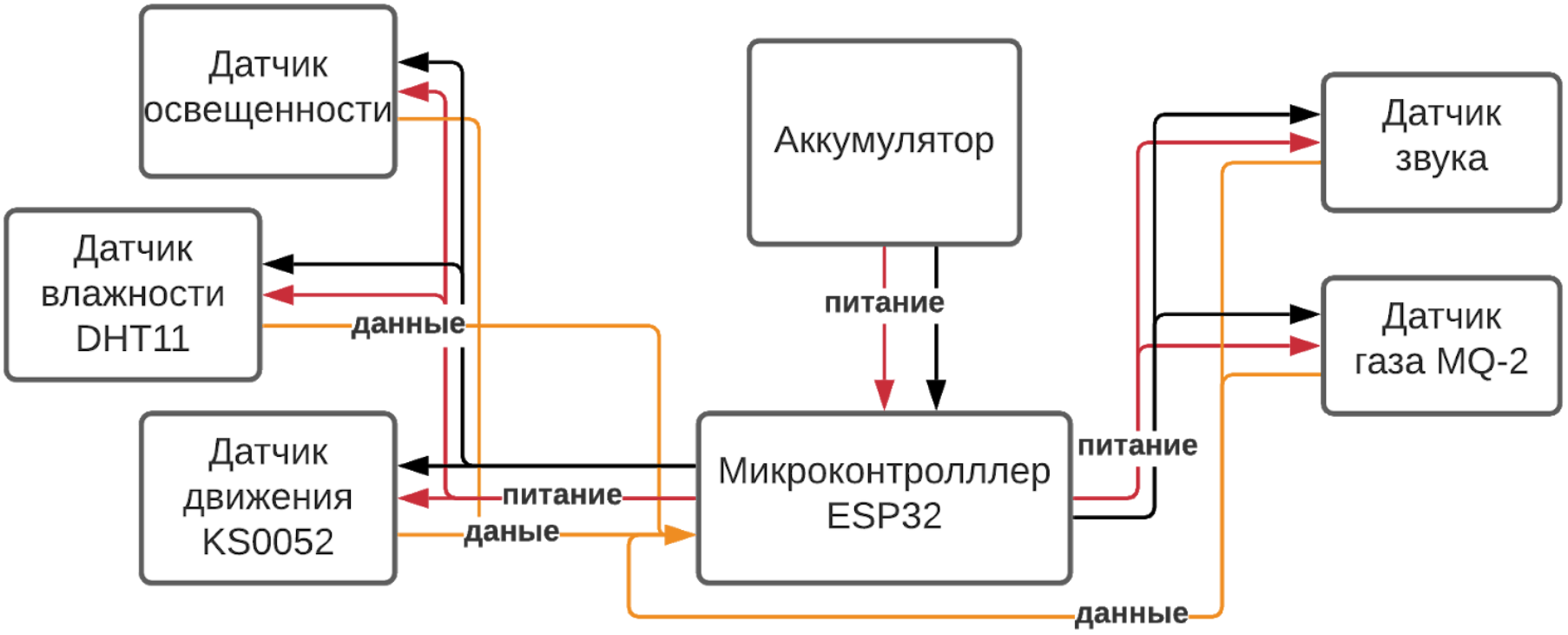

Рис 5 - Функциональная схема подсистемы сбора данных

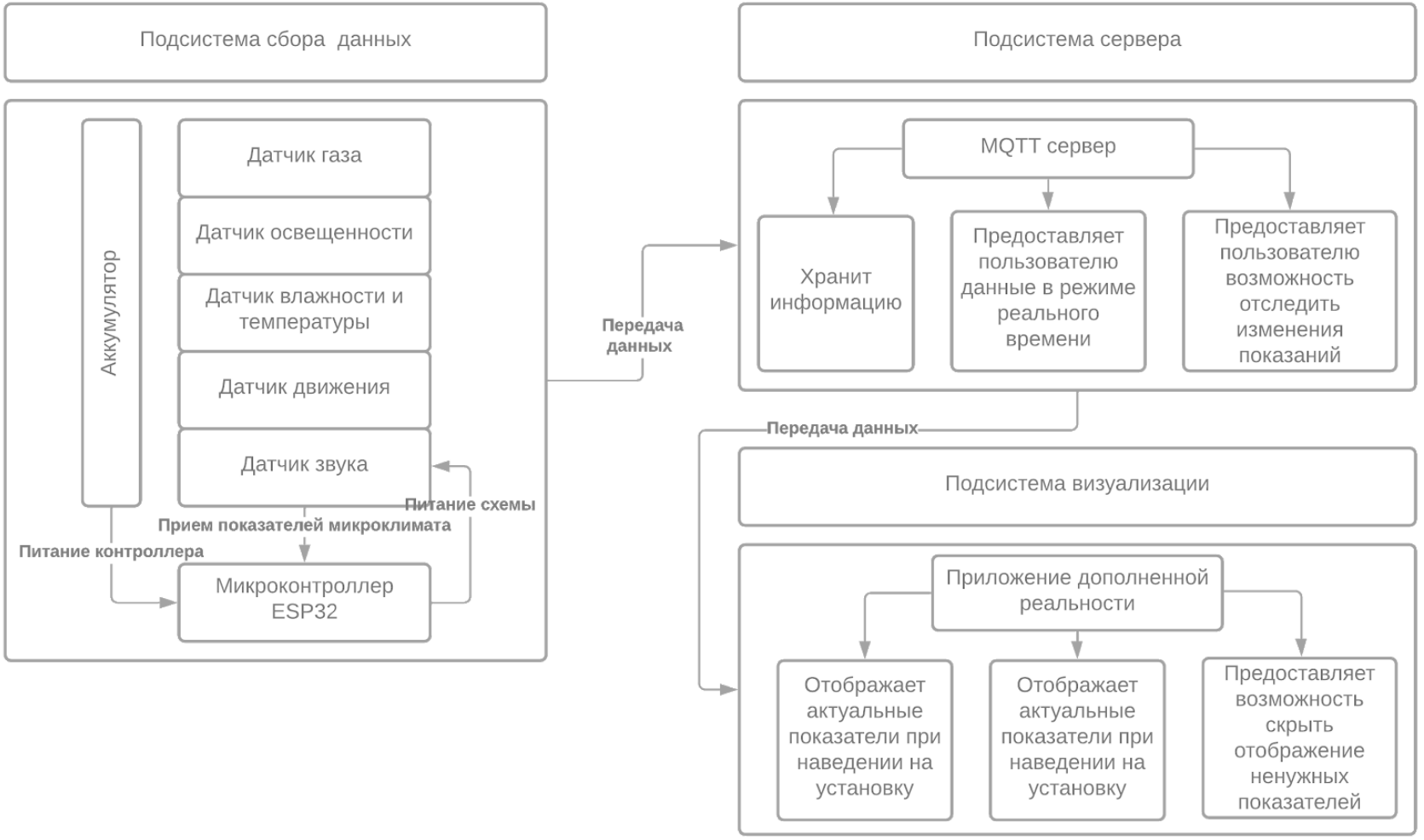

Рис. 6 - Структурная схема

```
#include "Arduino.h"
#include "EspMQTTClient.h"
#include "DHT.h"

#define PUB_DELAY (5 * 1000)

#define gasPin 2
byte sensorPin = 3;
#define DHTPIN 5
#define lightPin 14

DHT dht(DHTPIN, DHT11);

EspMQTTClient client(
  "Wi-Fi Name",
  "Wi-Fi password",
  "dev.rightech.io",
  "Server Name"
);

void setup() {
  Serial.begin(9600);
  dht.begin();
}

void onConnectionEstablished() {
  Serial.println("connected");
}

long last = 0;

void loop() {
  client.loop();
  long now = millis();
  if (client.isConnected() && (now - last > PUB_DELAY)) {

    float t = dht.readTemperature();
    Serial.print("Температура: ");
    Serial.println(t);
    client.publish("base/state/temperature", String(t));
```

```
    float h = dht.readHumidity();
    Serial.print("Влажность: ");
    Serial.println(h);
    client.publish("base/state/humidity", String(h));


    int light = analogRead(lightPin);
    int lightLevel = map(light, 0, 1023, 0, 100);
    Serial.print("Уровень освещенности: ");
    Serial.println(lightLevel);
    client.publish("base/state/light", String(lightLevel));


    int gas = analogRead(gasPin);
    int gasLevel = map(gas, 0, 450, 0, 100);
    Serial.print("Уровень газа: ");
    Serial.print(gasLevel);
    Serial.println("%");
    client.publish("base/state/gas", String(gasLevel));


    byte state = digitalRead(sensorPin);
    if(state == 1){Serial.println("Замечено движение!");
    delay(500);}
    else if(state == 0){Serial.println("Движение отсутсвует!");
    delay(1000);}
    client.publish("base/state/state", String(state));

    last = now;}
}
```