**Описание системы мониторинга датчиков на базе Raspberry Pi**

**Общее описание**

Эта система представляет собой комплексное решение для сбора, обработки и передачи данных с различных датчиков через интерфейс I2C. Основные компоненты включают:

1. **IMU датчик (ICM-20948)** - 9-осевой датчик (акселерометр, гироскоп, магнитометр)
2. **Датчик температуры/влажности (SHTC3)**
3. **Газовый датчик (SGM58031)** - 4-канальный аналоговый сенсор
4. **Систему передачи данных через NATS**
5. **Логирование данных в файлы**

**Архитектура системы**

**1. Docker-контейнер (docker-compose.yml)**

* Привилегированный режим для доступа к аппаратным интерфейсам
* Монтирование директорий для логов и конфигурации
* Доступ к шине I2C (/dev/i2c-1)
* Сетевой режим "host" для работы с NATS

**2. Основное приложение (main.py)**

Содержит следующие ключевые компоненты:

**Классы датчиков:**

* ICM20948 - работа с IMU датчиком (акселерометр, гироскоп, магнитометр)
* SHTC3 - работа с датчиком температуры и влажности
* SGM58031 - работа с газовым датчиком

**Основные функции:**

* Чтение и калибровка данных с датчиков
* Фильтрация и обработка данных (AHRS алгоритм для IMU)
* Логирование данных в JSON-формат
* Буферизация и отправка данных через NATS
* Обработка ошибок и восстановление соединений

**3. Конфигурация (settings.ini)**

Гибкая система настроек с разделами:

1. **Логирование**:
   * Пути к файлам логов
2. **NATS**:
   * Адрес сервера и параметры подключения
   * Настройки буферизации
3. **Датчики**:
   * Интервалы опроса
   * Включение/отключение отдельных датчиков
4. **Калибровка**:
   * Параметры калибровки для каждого датчика
   * Масштабные коэффициенты и смещения

**4. Зависимости (requirements.txt)**

* smbus - для работы с I2C
* lgpio - низкоуровневый доступ к GPIO
* nats-py - клиент для работы с NATS

**Особенности реализации**

1. **Асинхронная архитектура**:
   * Использование asyncio для эффективного управления циклами опроса
   * Неблокирующая отправка данных
2. **Устойчивость к ошибкам**:
   * Буферизация данных при недоступности сервера
   * Ограничение частоты повторения одинаковых ошибок
   * Автоматическое восстановление соединений
3. **Гибкая конфигурация**:
   * Все параметры вынесены в конфигурационный файл
   * Индивидуальные настройки для каждого датчика
   * Возможность включения/отключения датчиков
4. **Калибровка**:
   * Поддержка программной калибровки всех датчиков
   * Масштабирование и смещение для каждого канала
5. **Логирование**:
   * Два типа логов (JSON для данных, текстовый для ошибок)
   * Вращение логов для предотвращения переполнения

**Поток данных**

1. Инициализация датчиков с учетом конфигурации
2. Основной цикл:
   * Опрос включенных датчиков
   * Применение калибровочных коэффициентов
   * Обработка данных (фильтрация, преобразование)
   * Сохранение в локальный лог
   * Добавление в буфер для отправки
3. Периодическая отправка данных на NATS сервер
4. Обработка ошибок и восстановление соединений

**Сценарии использования**

1. **Мониторинг состояния оборудования**:
   * Контроль ориентации и вибраций
   * Мониторинг температуры и влажности
2. **Промышленные приложения**:
   * Контроль параметров окружающей среды
   * Обнаружение утечек газов
3. **Умный дом**:
   * Системы климат-контроля
   * Датчики безопасности
4. **Научные исследования**:
   * Сбор данных с датчиков
   * Анализ динамики изменений

Система особенно полезна в приложениях, где требуется надежный сбор и передача данных с различных датчиков с возможностью гибкой настройки и калибровки.