**Описание системы мониторинга UPS HAT (E) для Raspberry Pi**

**Общее описание**

Это комплексное решение для мониторинга и управления источником бесперебойного питания (UPS) HAT (E) для Raspberry Pi. Система предоставляет:

1. Детальный мониторинг параметров батареи
2. Защиту от глубокого разряда
3. Логирование данных
4. Удаленную передачу данных через NATS
5. Автоматическое безопасное отключение системы при критических состояниях

**Компоненты системы**

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

* Создает изолированное окружение для работы приложения
* Особенности конфигурации:
  + Привилегированный режим (privileged: true) для доступа к аппаратным ресурсам
  + Монтирование директорий для логов и конфигурации
  + Доступ к шине I2C для взаимодействия с UPS HAT
  + Режим сети "host" для максимальной производительности

**2. Основное приложение (main.py)**

Реализует весь функционал мониторинга и управления:

**Ключевые классы и функции:**

* UPSMonitor - основной класс для работы с UPS через I2C
  + Чтение данных батареи (напряжение, ток, уровень заряда)
  + Мониторинг состояния зарядки
  + Защитные механизмы
  + Калибровка измерений

**Функционал:**

* Непрерывный мониторинг параметров:
  + Напряжение и ток батареи
  + Напряжения отдельных ячеек
  + Состояние зарядки
  + Оставшаяся емкость
  + Расчет времени до разряда/заряда

**Защитные механизмы:**

* Автоматическое отключение при:
  + Критически низком напряжении (аварийное)
  + Продолжительном низком напряжении
* Настраиваемые пороги срабатывания

**Логирование:**

* Два типа логов:
  + error\_log.txt - ошибки и критические события
  + data\_log.json - структурированные данные мониторинга

**Интеграция:**

* Отправка данных на NATS сервер для удаленного мониторинга

**3. Конфигурация (settings.ini)**

Подробный конфигурационный файл с разделами:

1. **Основные настройки**:
   * Параметры подключения к NATS
   * Интервалы опроса и отправки данных
   * Пути для логов
2. **Параметры батареи**:
   * Тип и характеристики аккумулятора
   * Критические напряжения
3. **Калибровка**:
   * Точная настройка измерений тока
   * Формулы и инструкции для калибровки
4. **Защита**:
   * Пороги срабатывания защитных механизмов
   * Время реакции системы

**4. Зависимости (requirements.txt)**

* nats-py - для работы с NATS
* smbus - для работы с I2C
* configparser - для чтения конфигурации

**Особенности реализации**

1. **Асинхронная архитектура** - использование asyncio для эффективной работы
2. **Потокобезопасное логирование** - механизмы блокировки файлов
3. **Гибкая конфигурация** - все параметры вынесены в конфигурационный файл
4. **Защита данных** - буферизация и повторные попытки отправки
5. **Калибровка** - поддержка точной настройки измерений

**Сценарии использования**

1. **Мониторинг состояния батареи**:
   * Локальное хранение истории
   * Удаленная передача данных
2. **Защита оборудования**:
   * Предотвращение глубокого разряда
   * Безопасное отключение системы
3. **Анализ энергопотребления**:
   * Точные измерения тока и напряжения
   * Расчет оставшегося времени работы

Система предназначена для использования в ответственных применениях Raspberry Pi, где важно обеспечить стабильное питание и корректное завершение работы при отключении электроэнергии.