### Инструкция по использованию программы мониторинга CAN-шины

#### **1. Настройка конфигурации**

Перед запуском программы необходимо настроить файл settings.ini. Основные параметры:

##### **Секция [CAN]**

* interface: Имя CAN-интерфейса (например, can0, can1, vcan0).
* bitrate: Скорость передачи данных (например, 500000 бит/с).
* polling\_period: Интервал опроса CAN-шины (в секундах).
* request\_id: ID для отправки запросов (в шестнадцатеричном формате, например, 0x7DF).
* response\_id: ID для ответов (через запятую, например, 0x7e7, 0x7e8, 0x7e9).
* can\_retry\_interval: Интервал между попытками переподключения (в секундах).
* max\_can\_retries: Максимальное количество попыток переподключения.

##### **Секция [LOGGING]**

* enabled: Включение/отключение логирования (true/false).
* data\_log\_file: Путь к файлу для хранения данных CAN (в формате JSON).
* error\_log\_file: Путь к файлу для хранения ошибок.

##### **Секция [NATS]**

* enabled: Включение/отключение отправки данных в NATS (true/false).
* server: Адрес NATS-сервера (например, nats://10.10.0.110:4222).
* topic: Топик для публикации данных.
* buffer\_size: Размер буфера сообщений при недоступности сервера.

#### **2. Запуск программы**

1. Убедитесь, что CAN-интерфейс активен (например, с помощью команды ip link set can0 up).
2. Запустите программу командой:

python3 main.py

#### **3. Используемые CAN ID**

Программа работает с OBD-II стандартом, используя следующие ID:

* **Запросы**: 0x7DF (стандартный широковещательный ID для OBD-II запросов).
* **Ответы**: Фильтрация по ID 0x7E7, 0x7E8, 0x7E9 (стандартные ID для ответов от ЭБУ).

#### **4. Поддерживаемые PID (Parameter ID)**

Программа запрашивает следующие PID для получения данных:

* 0x05: Температура охлаждающей жидкости.
* 0x0C: Обороты двигателя (RPM).
* 0x0D: Скорость автомобиля.
* 0x2F: Уровень топлива.
* Другие PID, указанные в словаре expected\_pids в коде.

#### **5. Логирование данных**

* Данные сохраняются в файл, указанный в data\_log\_file (формат JSON).
* Ошибки записываются в файл, указанный в error\_log\_file.

#### **6. Отправка данных в NATS**

* Если NATS включен, данные публикуются в указанный топик.
* При недоступности сервера данные буферизуются и отправляются при восстановлении соединения.

#### **7. Действия при ошибках**

* При частых ошибках CAN-подключения программа автоматически перезапускает соединение.
* Интерфейс CAN может быть принудительно поднят при обнаружении его отключения.

#### **8. Завершение работы**

Для остановки программы нажмите Ctrl+C в терминале.

### Примеры использования

1. **Мониторинг температуры и RPM**:
   * Программа будет запрашивать PID 0x05 и 0x0C, отображая температуру и обороты двигателя.
2. **Отправка данных в NATS**:
   * Если NATS сервер доступен, данные будут публиковаться в топик can.data.
3. **Анализ логов**:
   * Проверьте файлы data\_log.json и error\_log.txt для диагностики.

Программа гибко настраивается через settings.ini и поддерживает расширение списка PID для мониторинга.