**Расширенный список CAN ID для OBD-II и других протоколов**

Программа поддерживает фильтрацию по ID ответов. Вот основные диапазоны и конкретные ID, которые можно использовать:

**1. Стандартные OBD-II ID (ISO 15765-4)**

| **ID (Hex)** | **Описание** |
| --- | --- |
| **0x7E8** | Основной ответ ЭБУ двигателя (по умолчанию) |
| **0x7E9** | Ответ от трансмиссии (TCM) |
| **0x7EA** | Ответ от ABS/ESP |
| **0x7EB** | Ответ от климат-контроля |
| **0x7EC** | Ответ от кузовного модуля (BCM) |
| **0x7ED** | Ответ от модуля подушек безопасности |
| **0x7EE** | Ответ от модуля освещения |
| **0x7EF** | Ответ от модуля телематики |

**Пример для**settings.ini**:**

response\_id = 0x7E8, 0x7E9, 0x7EA, 0x7EB

**2. Расширенные ID для диагностики**

| **ID (Hex)** | **Описание** |
| --- | --- |
| **0x18DAF1xx** | Физический адрес ЭБУ (xx — номер модуля) |
| **0x18DB33xx** | Функциональный адрес (широковещательный) |
| **0x7DF** | Широковещательный запрос (используется для запросов) |

**Важно**: Для работы с этими ID может потребоваться поддержка расширенного формата (is\_extended\_id=True в python-can).

**3. ID для протокола J1939 (грузовики, спецтехника)**

| **ID (Hex)** | **Описание** |
| --- | --- |
| **0x18FEE500** | Данные двигателя (RPM, температура) |
| **0x18FEF100** | Давление в топливной системе |
| **0x18FFA000** | Скорость автомобиля |

**Настройка:**

* Требуется изменить bitrate на **250000** в settings.ini.
* В коде добавить фильтры:

can\_filters = [{"can\_id": 0x18FEE500, "can\_mask": 0x1FFFF000}]

**4. Произвольные ID для нестандартных систем**

Некоторые производители используют собственные ID:

* **Tesla**: 0x102, 0x103 (данные батареи).
* **BMW/Mercedes**: Диапазон 0x600-0x7FF.
* **Гибридные авто**: 0x7C0-0x7CF (данные высоковольтной системы).

**Как добавить:**

1. В settings.ini укажите нужные ID:

response\_id = 0x7E8, 0x102, 0x7C0

1. В коде добавьте декодер для новых PID в функцию decode\_pid().

**5. Фильтрация по маске**

Если нужно принимать диапазон ID (например, все ответы от 0x7E0 до 0x7EF):

can\_filters = [{"can\_id": 0x7E0, "can\_mask": 0x7F0}] *# Фильтр по первым 3 цифрам*

**Как определить подходящие ID для вашего авто?**

1. **Используйте OBD-сканер** (например, Torque Pro) для просмотра ID в реальном времени.
2. **Анализ дампа CAN-шины**:

candump can0 -l *# Запись трафика*

1. **Документация производителя** (например, для VAG-группы ID часто начинаются с 0x7E0).

**Ошибки при настройке ID**

1. **Нет ответа**:
   * Убедитесь, что request\_id (по умолчанию 0x7DF) корректный.
   * Проверьте, поддерживает ли авто запрашиваемые PID (0x05, 0x0C и т.д.).
2. **Некорректные данные**:
   * Возможно, ID используется для другого протокола (например, J1939). Проверьте bitrate.

**Пример для грузовиков (J1939)**

[CAN]

interface = can0

bitrate = 250000

request\_id = 0x18DB33F9 # Функциональный запрос

response\_id = 0x18FEE500, 0x18FEF100 # Данные двигателя и топлива

Для таких случаев потребуется доработать функцию decode\_pid() под специфику J1939.