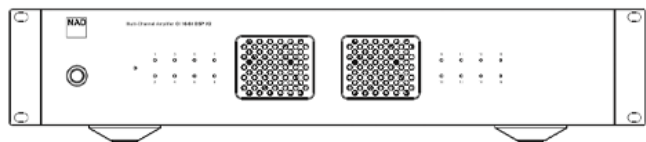


ИНСТРУКЦИЯ ПО ЭКСПЛУАТАЦИИ

Многоканальный усилитель NAD CI 16-60 DSP

NAD CI 16-60



x8



x1



x1



x4



x4



Класс: многоканальный *DSP*-усилитель для систем распределенного аудио и коммерческих инсталляций

ЦЕЛЕВОЕ НАЗНАЧЕНИЕ (ПРОФЕССИОНАЛЬНОЕ НЕБЫТОВОЕ ПРИМЕНЕНИЕ)

Многоканальный усилитель NAD CI 16-60 DSP **предназначен исключительно для профессионального небытового использования** в интеграционных проектах (CI – custom install). Разработано компанией NAD Electronics (Канада) для профессионального применения в коммерческих и пользовательских интеграционных проектах. Усилитель предназначен для использования в системах распределенного аудио — как в коммерческих помещениях (рестораны, отели, конференц-залы, магазины), так и в крупных жилых комплексах с многозонными аудиосистемами. Усилитель построен на модифицированной платформе Nurx UcD и оснащен полноценным DSP на каждый канал. Версия v2 предлагает улучшенный WebUI с расширенными возможностями мониторинга и управления.

Основные профессиональные сценарии применения:

Сфера применения	Тип задач
Коммерческие аудиосистемы	Фоновое озвучивание в ресторанах, барах, отелях, торговых центрах
Конференц-залы и переговорные комнаты	Распределенное озвучивание, многозонные системы
Образовательные учреждения	Озвучивание аудиторий, классных комнат, кампусов
Культовые сооружения	Церкви, синагоги, храмы — системы озвучивания
Студии звукозаписи (контрольные комнаты)	Распределенный мониторинг в комплексах студий
Домашние и профессиональные кинотеатры	Многоканальное усиление для зон ожидания и коридоров
Системы умного дома (Custom Integration)	Многозонное распределение аудио по помещениям
Трансляционные системы	Озвучивание аэропортов, вокзалов, бизнес-центров

Ключевые особенности для профессионального применения:

- 16 каналов усиления:** 16 каналов × 60 Вт (8 Ом), каждая пара каналов мостируется до 140 Вт для расширения мощности .
- Встроенный DSP на каждый канал:** Полноценная цифровая обработка сигнала с эквалайзерами, фильтрами, лимитерами, задержками и регулировкой фазы .
- Программируемость через Python и IP API:** Полный контроль через TCP/IP (порт 52000) и веб-интерфейс для автоматизации и интеграции с профессиональными системами управления .
- Виртуальная патч-панель (WebUI):** Маршрутизация любого входа на любой выход без физического переключения — "virtual patch bay" .

5. **Профессиональные аппаратные интерфейсы:** RS232, IR IN/OUT, 12V Trigger, петлевые выходы для каскадирования усилителей .
 6. **2U стоечное исполнение:** Компактный 19" корпус высотой 2U для экономии места в профессиональных стойках .
-

1. ВВЕДЕНИЕ И ОБЩИЕ СВЕДЕНИЯ

Настоящая инструкция предназначена для **квалифицированного персонала инсталляторов и системных интеграторов**, использующего многоканальный усилитель NAD CI 16-60 DSP в профессиональных целях. Усилитель построен на модифицированной платформе Nurx UcD (класс D) и оснащен сетевым управлением с веб-интерфейсом для полной конфигурации DSP и маршрутизации сигналов .

NAD CI 16-60 DSP версии v1 выпущен в 2021 году и регулярно получает обновления прошивки . Версия v2 (2025) доступна как обновленная модель .

Основные технические характеристики:

Параметр	Значение
Количество каналов	16
Выходная мощность (8 Ом)	60 Вт на канал (все каналы одновременно, 1 кГц, 0.05% THD)
Выходная мощность (4 Ом)	65 Вт на канал
Мощность в мостовом режиме (8 Ом)	140 Вт на канал (пары каналов)
THD (искажения)	0.05% (20 Гц - 3 кГц, 1-50 Вт)
Соотношение сигнал/шум	80 дБ (А-взвешенный)
Коэффициент демпфирования	110 (20 Гц - 1 кГц, 8 Ом)

Параметр	Значение
Частотный диапазон	20 Гц - 20 кГц (± 1 дБ)
Разделение каналов	60 дБ (1 кГц)
Пиковый выходной ток	15 А (1 Ом, 1 мс)
Потребляемая мощность (ожидание)	0.5 Вт, 3 Вт в сетевом ожидании

Подключения и разъемы:

Тип	Количество / Характеристики	Назначение
Аналоговые входы (RCA)	16 (по канално)	Сигнал от источника
Глобальные входы (RCA)	2 пары	Общий сигнал для нескольких зон
Петлевые выходы (RCA)	16 (по канално) + глобальные	Каскадирование на несколько усилителей
Акустические выходы	16 пар (Phoenix/Euroblock)	Подключение колонок
12V Trigger IN	2-контактный Phoenix	Дистанционное включение
IR IN / OUT	4-контактный Phoenix	Управление по ИК-кабелю
LAN (Ethernet)	RJ45	Сетевое управление (WebUI + API)
RS232	DB9	Серийное управление

2. ТРЕБОВАНИЯ БЕЗОПАСНОСТИ

Перед эксплуатацией системы ознакомьтесь со следующими требованиями:

- **Электропитание:** Используйте оба прилагаемых съемных кабеля питания. Подключайте к розетке с защитным заземлением. Напряжение питания: соответствует региону .
- **Вентиляция:** Обеспечьте свободный доступ воздуха. Рекомендуемый зазор: не менее 10 см слева, справа, сверху и сзади. Вентиляционные отверстия не должны блокироваться .
- **Температура:** Эксплуатируйте при температуре от 0 до 40°C. Усилитель предназначен для работы в умеренном климате .
- **Влажность:** Не допускайте попадания жидкости на корпус. Не эксплуатируйте вблизи воды. Прибор не должен подвергаться воздействию капель или брызг .
- **Перегрузка:** Усилитель оснащен защитой от перегрузки. Не перегружайте сетевые розетки и удлинители .
- **Чистка:** Отключите от сети перед чисткой. Используйте только сухую мягкую ткань. Не применяйте жидкие или аэрозольные чистящие средства .
- **Установка:** Используйте только стойки, подставки и крепления, рекомендованные производителем. При транспортировке используйте тележку с осторожностью .

⚠ ПРОФЕССИОНАЛЬНОЕ ПРИМЕЧАНИЕ: Усилитель предназначен для установки в профессиональные 19" стойки. Обязательно закрепите его винтами через монтажные отверстия на передней панели.

⚠ ПРОФЕССИОНАЛЬНОЕ ПРИМЕЧАНИЕ: При использовании в коммерческих инсталляциях с большим количеством каскадированных усилителей убедитесь, что питание всей системы рассчитано на общую нагрузку.

3. РАСПАКОВКА И ПРОВЕРКА КОМПЛЕКТАЦИИ

1. Извлеките компоненты из упаковки. **Сохраните оригинальную упаковку для транспортировки .**
2. Проверьте комплектацию:

Компонент	Описание
NAD CI 16-60 DSP	16-канальный многоканальный усилитель
Съемные кабели питания	2 шт.
8 × 4-контактных терминальных колодок	Для подключения колонок

Компонент	Описание
1 × 4-контактная терминальная колодка	Для IR IN/OUT
1 × 2-контактная терминальная колодка	Для +12V TRIGGER IN
4 ножки с монтажными винтами	Для настольной установки (опционально)
Руководства	Краткое руководство по установке
3. Осмотрите корпус на предмет повреждений.	

Примечание: Маркировка и табличка с рейтингом находятся на задней панели прибора .

4. УСТАНОВКА И РАЗМЕЩЕНИЕ (ПРОФЕССИОНАЛЬНАЯ КОНФИГУРАЦИЯ)

4.1. Габаритные размеры и вес

Параметр	Значение
Габариты (Ш × В × Г)	483 × 100 × 435 мм
Высота в стойке (RU)	2U
Вес (нетто)	10.3 кг (22.7 фунта)
Вес (брутто)	12.8 кг (28.2 фунта)

4.2. Размещение в профессиональной стойке

- **Крепление:** Установите усилитель в стандартную 19" стойку, закрепив винтами через монтажные отверстия на передней панели.
- **Вентиляция:** Оставьте не менее 10 см свободного пространства слева, справа, сверху и сзади для обеспечения циркуляции воздуха .
- **Вес:** Убедитесь, что стойка способна выдержать вес усилителя (10.3 кг).
- **Доступ:** Обеспечьте доступ к задней панели для подключения кабелей.

4.3. Подключение питания

1. Перед подключением убедитесь, что выключатель питания (на задней панели) находится в положении OFF.
2. Подключите **оба** прилагаемых кабеля питания к разъемам AC Mains Input на задней панели (усилитель может иметь два разъема питания для резервирования или повышенной мощности) .
3. Подключите кабели к розеткам сети переменного тока с заземлением.
4. Включите питание.

4.4. Подключение акустических систем

Усилитель использует 4-контактные терминальные колодки (Phoenix/Euroblock) для подключения колонок.

Процедура подключения:

1. Зачистите концы акустических кабелей на ~10 мм.
2. Вставьте провода в соответствующие отверстия терминальной колодки.
3. Затяните винты для фиксации проводов.
4. Вставьте терминальную колодку в соответствующий разъем на задней панели усилителя.

Схема подключения:

- Каждый канал имеет свою пару клемм.
- При мостовом режиме используйте специально отмеченные клеммы (L+ и R- или аналогичные).

4.5. Типовые схемы подключения

А. Многозонная система (16 зон стерео):

- Каждая зона (стерео) подключается к 2 последовательным каналам (1+2, 3+4, 5+6...)
- Каждый входной сигнал маршрутизируется через WebUI на соответствующие выходы

В. Мощные системы (мостовой режим):

- Объединение двух каналов в один мощный (140 Вт) для крупных зон

С. Каскадирование (Daisy Chain):

- Используйте петлевые выходы (RCA Thru/Loop Out) для передачи сигнала на дополнительные усилители

Д. Управляющие подключения:

- **12V Trigger IN:** Подключение к управляющему устройству (Crestron, Control4 и др.) для дистанционного включения
- **IR IN/OUT:** Подключение к системам ИК-управления
- **LAN (Ethernet):** Подключение к локальной сети для доступа к WebUI и IP API
- **RS232:** Подключение к последовательному порту управляющей системы

5. ПРОГРАММИРОВАНИЕ НА PYTHON (IP API)

5.1. Общие сведения

Усилитель NAD CI 16-60 DSP поддерживает управление через **TCP/IP протокол** с использованием шестнадцатеричных команд. API доступен через порт **52000**. Усилитель выступает в роли **TCP-сервера** .

Базовые параметры подключения:

- **Протокол:** TCP/IP (raw socket)
- **Порт:** 52000
- **IP-адрес:** IP-адрес усилителя в локальной сети
- **Формат команд:** Шестнадцатеричные последовательности
- **Кодировка:** Бинарная (HEX)

5.2. Определение IP-адреса устройства

IP-адрес усилителя можно определить несколькими способами:

1. **Через веб-интерфейс (WebUI)** — стандартный доступ по IP, назначенному DHCP
2. **Через утилиту обнаружения** (например, Advanced IP Scanner)
3. **Через DHCP-сервер** (список выданных адресов)

```
python
import socket
import time
from typing import Optional, Dict, List

def discover_ci_16_60_on_network() -> List[str]:
    """
    Обнаружение устройств NAD CI 16-60 DSP в локальной сети
    через широковещательный UDP запрос.

    Примечание: Для точного обнаружения может потребоваться
    использование специализированных утилит NAD.
    """
    # Альтернативный метод: сканирование порта 52000
    found_devices = []
    # Этот метод требует предварительного знания диапазона IP-адресов
    # Реализация через широковещательный запрос требует специфического протокола NAD
    return found_devices

def get_device_info(ip_address: str) -> Dict:
    """Получение информации об устройстве через WebUI (HTTP GET)"""
    try:
        import requests
        # WebUI доступен на порту 80
        response = requests.get(f"http://{ip_address}", timeout=5)
        # Примечание: WebUI возвращает HTML, парсинг не тривиален
        # Для получения статуса используйте API команды
        return {'ip': ip_address, 'webui_accessible': response.status_code == 200}
    except Exception as e:
        return {'ip': ip_address, 'error': str(e)}
```

5.3. Базовое подключение и отправка команд

API CI 16-60 DSP использует **шестнадцатеричные последовательности** для всех операций. Усилитель работает как TCP-сервер .

```
python
import socket
import struct
from typing import Optional, Dict, Any, List
```

```
class NADCI1660Controller:
```

```
    """
    Класс для управления NAD CI 16-60 DSP через IP API (TCP/IP).
    Усилитель является TCP-сервером на порту 52000 [citation:5].
    """
```

```
    def __init__(self, ip_address: str, port: int = 52000):
```

```
        """
        Инициализация контроллера
```

```
        Args:
```

```
            ip_address: IP-адрес усилителя в локальной сети
            port: TCP-порт (по умолчанию 52000) [citation:5]
```

```
        """
        self.ip = ip_address
        self.port = port
        self._socket = None
```

```
    def connect(self) -> bool:
```

```
        """
        Установка TCP-соединения с усилителем
```

```
        Returns:
```

```
            True при успешном подключении
```

```
        """
```

```
        try:
```

```
            self._socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            self._socket.settimeout(5)
            self._socket.connect((self.ip, self.port))
            return True
```

```
        except Exception as e:
```

```
            print(f"Ошибка подключения к {self.ip}:{self.port}: {e}")
            self._socket = None
            return False
```

```
    def disconnect(self):
```

```
        """Закрытие TCP-соединения"""
```

```
        if self._socket:
```

```
            self._socket.close()
```

```
self._socket = None
```

```
def _send_hex_command(self, hex_bytes: bytes) -> Optional[bytes]:
```

```
    """
```

```
    Отправка HEX-команды и получение ответа
```

```
    Args:
```

```
    hex_bytes: Байтовая последовательность команды
```

```
    Returns:
```

```
    Ответ в байтах или None при ошибке
```

```
    """
```

```
    if not self._socket:
```

```
        if not self.connect():
```

```
            return None
```

```
    try:
```

```
        self._socket.send(hex_bytes)
```

```
        response = self._socket.recv(1024)
```

```
        return response
```

```
    except Exception as e:
```

```
        print(f"Ошибка отправки команды: {e}")
```

```
        return None
```

```
# === Управление питанием [citation:5] ===
```

```
def power_on(self) -> bool:
```

```
    """
```

```
    Включение усилителя
```

```
    Команда: FF 55 01 01
```

```
    """
```

```
    cmd = bytes([0xFF, 0x55, 0x01, 0x01])
```

```
    response = self._send_hex_command(cmd)
```

```
    return response is not None
```

```
def power_off(self) -> bool:
```

```
    """
```

```
    Выключение усилителя
```

```
    Команда: FF 55 01 02
```

```
    """
```

```
    cmd = bytes([0xFF, 0x55, 0x01, 0x02])
```

```
    response = self._send_hex_command(cmd)
```

```

    return response is not None

def power_toggle(self) -> bool:
    """
    Переключение питания

    Команда: FF 55 01 03
    """
    cmd = bytes([0xFF, 0x55, 0x01, 0x03])
    response = self._send_hex_command(cmd)
    return response is not None

def get_power_status(self) -> Optional[str]:
    """
    Запрос статуса питания

    Команда: FF 55 01 70
    Возвращает: "Power Status On/Off" [citation:5]
    """
    cmd = bytes([0xFF, 0x55, 0x01, 0x70])
    response = self._send_hex_command(cmd)
    if response:
        # Парсинг ответа (формат: текстовый с ключами)
        return response.decode('ascii', errors='ignore').strip()
    return None

# === Управление методом питания [citation:5] ===

def set_power_method(self, method: int) -> bool:
    """
    Установка метода включения

    Args:
        method: 0 = Power Button, 1 = Always On, 2 = 12V trigger, 3 = Signal Sense [citation:5]

    Команда: FF 55 02 F8 [P1]
    """
    if method < 0 or method > 3:
        return False
    cmd = bytes([0xFF, 0x55, 0x02, 0xF8, method])
    response = self._send_hex_command(cmd)
    return response is not None

def set_eco_mode(self, enabled: bool) -> bool:

```

```
"""
```

Установка ECO режима

Args:

enabled: True - ECO mode on, False - ECO mode off [citation:5]

Команда: FF 55 02 F9 [P1] (P1 = 0 or 1)

```
"""
```

```
cmd = bytes([0xFF, 0x55, 0x02, 0xF9, 1 if enabled else 0])
response = self._send_hex_command(cmd)
return response is not None
```

=== Информация об устройстве [citation:5] ===

```
def get_device_name(self) -> Optional[str]:
    """Запрос имени устройства (FF 55 01 E0)"""
    cmd = bytes([0xFF, 0x55, 0x01, 0xE0])
    response = self._send_hex_command(cmd)
    if response:
        return response.decode('ascii', errors='ignore').strip()
    return None
```

```
def get_device_model(self) -> Optional[str]:
    """Запрос модели устройства (FF 55 01 E1)"""
    cmd = bytes([0xFF, 0x55, 0x01, 0xE1])
    response = self._send_hex_command(cmd)
    if response:
        return response.decode('ascii', errors='ignore').strip()
    return None
```

```
def get_firmware_version(self) -> Optional[str]:
    """Запрос версии прошивки (FF 55 01 E5)"""
    cmd = bytes([0xFF, 0x55, 0x01, 0xE5])
    response = self._send_hex_command(cmd)
    if response:
        return response.decode('ascii', errors='ignore').strip()
    return None
```

```
def get_serial_number(self) -> Optional[str]:
    """Запрос серийного номера (FF 55 01 E6)"""
    cmd = bytes([0xFF, 0x55, 0x01, 0xE6])
    response = self._send_hex_command(cmd)
    if response:
        return response.decode('ascii', errors='ignore').strip()
```

```

return None

# === Управление каналами [citation:5] ===

def set_channel_mute(self, channel: int, mute: bool) -> bool:
    """
    Включение/выключение отключения звука на канале

    Args:
        channel: Номер канала (1-16)
        mute: True - отключить звук, False - включить звук
    """
    # Формирование команды по спецификации API
    # Примечание: Полная спецификация команд для mute требует уточнения
    pass

def get_channel_status(self, channel: int) -> Optional[Dict]:
    """Получение статуса канала (включая DSP настройки)"""
    # Статус может быть получен через WebUI или специализированные API команды
    # Примечание: CI 16-60 V2 предоставляет улучшенный мониторинг статуса [citation:2]
    pass

# === Управление DSP настройками (через WebUI) ===

def configure_dsp_via_webui(self, config: Dict):
    """
    Настройка DSP параметров через WebUI.
    CI 16-60 DSP предоставляет веб-интерфейс для конфигурации [citation:2][citation:4].

    Доступные настройки:
    - Peak Limiters
    - RMS Limiters
    - Phase Control
    - Delay
    - Tilt EQ [citation:2]
    - Готовые DSP профили для PSB, DALI [citation:2]
    """
    # Конфигурация через HTTP POST запросы к WebUI
    pass

```

5.4. Веб-интерфейс (WebUI) управления

Усилитель CI 16-60 DSP предоставляет полноценный **веб-интерфейс** для конфигурации всех параметров. WebUI доступен через стандартный HTTP порт 80 по IP-адресу устройства .

Функции WebUI (после обновления прошивки 2024 года):

1. DashBoard (Панель мониторинга):

- Реальное время статуса усилителя
- Использование каналов
- Температура и состояние питания

2. Zone Configuration (Конфигурация зон):

- Создание и редактирование зон
- Управление отключением звука, режимом сна
- Глобальный контроль входов

3. DSP Controls (Управление DSP):

- Peak и RMS лимитеры
- Контроль фазы и задержки (Delay)
- Tilt EQ для быстрой коррекции
- Готовые DSP профили для популярных брендов акустики (DALI, PSB)

4. Input/Output Setup (Настройка входа/выхода):

- Виртуальная патч-панель (маршрутизация любого входа на любой выход)
- Настройка чувствительности входов

5. Remote Monitoring (Удаленный мониторинг):

- Интеграция с платформами, такими как Domotz
- Расширенная обратная связь для стороннего ПО

6. Security (Безопасность):

- Защита паролем для предотвращения несанкционированных изменений конфигурации

```
python
```

```
class NADCI1660WebUI:
```

```
    """
```

```
    Класс для управления NAD CI 16-60 DSP через веб-интерфейс (HTTP)
```

```
    """
```

```
    def __init__(self, ip_address: str):
```

```
        self.ip = ip_address
```

```
        self.base_url = f"http://{ip_address}"
```

```
        self.session = requests.Session()
```

```

def get_dashboard_status(self) -> Optional[Dict]:
    """
    Получение статуса с панели мониторинга [citation:2]
    """
    try:
        # Получение статуса (требуется аутентификация, если включена)
        response = self.session.get(f"{self.base_url}/api/status", timeout=10)
        if response.status_code == 200:
            return response.json()
        return None
    except Exception as e:
        print(f"Ошибка получения статуса: {e}")
        return None

def configure_zone(self, zone_id: int, config: Dict) -> bool:
    """
    Конфигурация зоны через Zone Configuration Page [citation:2]

    Args:
        zone_id: Идентификатор зоны
        config: Словарь с параметрами зоны (inputs, mute, sleep, global controls)
    """
    try:
        response = self.session.post(
            f"{self.base_url}/api/zones/{zone_id}",
            json=config,
            timeout=10
        )
        return response.status_code == 200
    except Exception as e:
        print(f"Ошибка конфигурации зоны: {e}")
        return False

def configure_dsp(self, channel: int, dsp_settings: Dict) -> bool:
    """
    Конфигурация DSP для канала [citation:2]

    Args:
        channel: Номер канала (1-16)
        dsp_settings: Словарь с настройками DSP
            (limiters, phase, delay, eq, tilt)
    """
    try:

```

```

response = self.session.post(
    f"{self.base_url}/api/dsp/channel/{channel}",
    json=dsp_settings,
    timeout=10
)
return response.status_code == 200
except Exception as e:
    print(f"Ошибка конфигурации DSP: {e}")
    return False

def load_speaker_profile(self, channel: int, brand: str, model: str) -> bool:
    """
    Загрузка готового DSP профиля для акустических систем [citation:2]

    Args:
        channel: Номер канала (1-16)
        brand: Бренд акустики (например, "DALI", "PSB")
        model: Модель акустической системы
    """
    try:
        response = self.session.post(
            f"{self.base_url}/api/dsp/profile/{channel}",
            json={"brand": brand, "model": model},
            timeout=10
        )
        return response.status_code == 200
    except Exception as e:
        print(f"Ошибка загрузки профиля: {e}")
        return False

def set_virtual_patch(self, input_channel: int, output_channel: int) -> bool:
    """
    Настройка виртуальной патч-панели [citation:4][citation:9]

    Args:
        input_channel: Входной канал (1-16)
        output_channel: Выходной канал (1-16)
    """
    try:
        response = self.session.post(
            f"{self.base_url}/api/routing",
            json={"input": input_channel, "output": output_channel},
            timeout=10
        )

```

```

return response.status_code == 200
except Exception as e:
    print(f'Ошибка настройки маршрутизации: {e}')
return False

```

5.5. Справочник команд IP API (TCP/IP порт 52000)

API для CI 16-60 DSP использует шестнадцатеричные команды. Ниже приведены основные команды на основе спецификации NAD .

Формат команд: FF 55 XX YY [P1] [P2]...

Управление питанием:

Команда (HEX)	Функция	Параметры
FF 55 01 01	Power ON	-
FF 55 01 02	Power OFF	-
FF 55 01 03	Power Toggle	-
FF 55 01 70	Запрос статуса питания	-
FF 55 02 F8 [P1]	Set Power Method	P1: 0=Button,1=Always On,2=12V Trigger,3=Signal Sense
FF 55 02 F9 [P1]	Set ECO Mode	P1: 0=OFF, 1=ON

Информация об устройстве:

Команда (HEX)	Функция	Возвращаемое значение
FF 55 01 E0	Запрос имени устройства	"Device Name"
FF 55 01 E1	Запрос модели устройства	"Device Model"

Команда (HEX)	Функция	Возвращаемое значение
FF 55 01 E5	Запрос версии прошивки	"Firmware Version"
FF 55 01 E6	Запрос серийного номера	"Serial Number"
FF 55 01 19	Запрос температуры	"Main Unit Temperature xC"
FF 55 01 1A	Запрос операционного статуса	"Amplifier Status"

Сброс:

Команда (HEX)	Функция
FF 55 01 F8	Factory Reset Unit

Управление каналами:

Команда (HEX)	Функция	Параметры
FF 55 03 F1 [P1] R	Запрос входного усиления	P1: канал (1-16)
FF 55 03 F2 [P1] P2	Установка входного усиления	P1: канал, P2: 0-4 (соответствует -6.6 дБ)
FF 55 03 F5 [P1] R	Запрос режима входа	P1: канал (1-16)
FF 55 03 F6 [P1] R	Запрос мостового режима	P1: канал (1-16)

Примечания:

- 16-канальные модели имеют **8 зон** (Zone A-H)
- Входные каналы автоматически спариваются при выборе зоны (1+2, 3+4, и т.д.)
- При обновлении прошивки с v2.26 до v2.37 могут измениться параметры API (требуется обновление драйверов управления)

6. УПРАВЛЕНИЕ DSP

6.1. Цифровая обработка сигнала (DSP)

Усилитель CI 16-60 DSP оснащен полноценным DSP на каждый канал .

Доступные DSP параметры:

Параметр	Описание
Peak Limiters	Ограничители пиков — защита динамиков от перегрузки
RMS Limiters	Ограничители по среднеквадратичному значению — защита от перегрева
Phase Control	Контроль фазы — компенсация фазовых сдвигов
Delay	Задержка сигнала — синхронизация звука в разных зонах
Tilt EQ	Наклонная эквалализация — быстрая коррекция баланса ВЧ/НЧ
Parametric EQ	Параметрический эквалайзер (доступен через WebUI)

6.2. Готовые DSP профили

В прошивке 2024 года добавлена поддержка готовых DSP профилей для популярных брендов акустических систем:

- **DALI** — оптимизированные профили
- **PSB Speakers** — профили от родственного бренда

Эти профили позволяют быстро оптимизировать работу усилителя с конкретными моделями акустики.

6.3. Виртуальная патч-панель (Virtual Patch Bay)

Усилитель позволяет маршрутизировать **любой входной сигнал на любой выход** (или несколько выходов) программно, без физического переключения кабелей. Это критически важно для:

- Многозонных систем с гибкой маршрутизацией
 - Быстрого переключения конфигураций
 - Удаленного управления без доступа к оборудованию
-

7. ПРОФЕССИОНАЛЬНЫЕ ФУНКЦИИ

7.1. Режим моста (Bridge Mode)

Любая последовательная пара каналов может быть объединена в мостовой режим:

Конфигурация	Мощность
Стандартный режим	16 × 60 Вт (8 Ом)
Мостовой режим (8 пар)	8 × 140 Вт (8 Ом)

Применение:

- Увеличение мощности для зон с низкочувствительными динамиками
- Драйвер пассивных сабвуферов
- Дальние зоны с длинными кабельными линиями

7.2. Профессиональные аппаратные интерфейсы

Интерфейс	Назначение
12V Trigger IN	Включение усилителя по сигналу от управляющей системы (3-30 Vdc)
IR IN / OUT	Передача ИК-сигналов управления через кабель
RS232	Серийное управление для интеграции с AMX, Crestron
Ethernet (RJ45)	Сетевое управление (WebUI + IP API порт 52000)
Глобальные входы/выходы RCA	Каскадирование нескольких усилителей

7.3. Методы включения (Power Methods)

Усилитель поддерживает несколько методов включения, настраиваемых через API:

Метод	Описание
Power Button	Включение кнопкой на передней панели
Always On	Постоянное включение
12V Trigger	Включение по сигналу на входе Trigger
Signal Sense	Включение при обнаружении сигнала на входе

7.4. ECO режим

Усилитель имеет ECO режим с энергопотреблением в режиме ожидания всего **0.5 Вт** (в обычном режиме 3 Вт) .

7.5. Профессиональные системные интеграции

Усилитель интегрируется с ведущими платформами управления (через API, RS232 или сторонние драйверы):

- **Control4**
- **Crestron**
- **RTI**
- **URC**
- **Elan**
- **Lutron**
- **Domotz** (Remote Monitoring Platform)
- **KNX**
- **PUSH**

7.6. Удаленный мониторинг (Remote Monitoring)

Улучшенная обратная связь для сторонних платформ позволяет:

- Отслеживать статус питания
- Контролировать температуру
- Мониторить использование каналов
- Диагностировать неисправности

8. ПОИСК И УСТРАНЕНИЕ НЕИСПРАВНОСТЕЙ

Проблема

Решение

Усилитель не включается

Проверьте оба кабеля питания. Убедитесь, что выключатель на задней панели в положении ON. Проверьте настройки Power Method (если используется 12V Trigger или Signal Sense)

Проблема	Решение
Нет звука на канале	Проверьте настройки виртуальной патч-панели (маршрутизация входов на выходы). Проверьте настройки mute для канала. Убедитесь, что мостовой режим настроен правильно
Усилитель не подключается к сети	Проверьте кабель Ethernet. Убедитесь, что DHCP включен (WebUI) или IP настроен статически. Проверьте доступность порта 52000
API команды не работают	Убедитесь, что используется порт 52000. Проверьте формат HEX команд. При обновлении с v2.26 до v2.37 может потребоваться обновление драйвера управления
Шум/фон на выходе	Проверьте заземление всей системы. Убедитесь, что акустические кабели не проходят рядом с силовыми. Проверьте настройки входного усиления
Перегрев	Обеспечьте вентиляционный зазор 10 см со всех сторон. Проверьте работу вентиляторов (запрос через API FF 55 02 1B [P1])
Не удается войти в WebUI	Проверьте, что усилитель имеет IP-адрес в сети. Убедитесь, что защита паролем не включена (или используйте пароль администратора)

9. ОБНОВЛЕНИЕ ПРОШИВКИ

9.1. Версии прошивки

CI 16-60 DSP имеет регулярные обновления прошивки :

Версия	Дата	Основные изменения
v2.37	Август 2025	Текущая версия
v2.08	2025-01-25	Fix: popping issue on muting channels
v2.06	2024-09-24	Fix: network dropping issue
v2.05	2021-12-03	Изменение диапазона громкости, остановка для регуляторов, команды статуса
v2.04	2021-05-14	Улучшение Load Preset, добавление Mute для каждого канала
v2.03	2021-04-07	Fix: проблема включения через 12V Trigger в ECO режиме
v1.99	2020-11-23	Fix: проблема полярности в суммарном режиме
v1.95	2020-10-12	Добавление DSP web-функции

9.2. Процедура обновления

Обновление прошивки выполняется **через веб-интерфейс (WebUI)** или через **сервисный порт** (по инструкции производителя) .

Примечание: При обновлении с v2.26 до v2.37 потребуется также обновить управляющий драйвер (если используется), так как изменились параметры API .

10. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

- **Чистка:** Отключите от сети перед чисткой. Используйте только сухую мягкую ткань. Не используйте жидкости или аэрозоли .
- **Вентиляция:** Регулярно проверяйте, чтобы вентиляционные отверстия не были заблокированы пылью.
- **Обновление ПО:** Выполняется через WebUI (OTA — Over The Air) .
- **Транспортировка:** Используйте оригинальную упаковку.

- **Сервис:** При повреждениях обращайтесь к авторизованному сервисному центру NAD .
-

ПРИЛОЖЕНИЕ: ПОЛНЫЕ ТЕХНИЧЕСКИЕ СПЕЦИФИКАЦИИ

Параметр	Значение
Модель	CI 16-60 DSP
Тип	Многоканальный DSP усилитель
Количество каналов	16
Мощность (8 Ом)	60 Вт (все каналы, 1 кГц, 0.05% THD)
Мощность (4 Ом)	65 Вт (все каналы, 1 кГц, 0.05% THD)
Мощность в мосту (8 Ом)	140 Вт (1 кГц, 0.05% THD)
THD+N	0.05% (20 Гц - 3 кГц, 1-50 Вт)
Соотношение сигнал/шум	80 дБ (А-взвешенный)
Коэффициент демпфирования	110
Частотный диапазон	20 Гц - 20 кГц (± 1 дБ)
Разделение каналов	60 дБ (1 кГц)
Входной импеданс	20 кОм//220 pF
Чувствительность	760 мВ (для 50 Вт в 8 Ом)

Параметр	Значение
Пиковый выходной ток	15 А (1 Ом, 1 мс)
Потребление (ожидание)	0.5 Вт
Потребление (сетевой standby)	3 Вт
Trigger IN уровень	3-30 Vdc
Аналоговый Audio Sense	3 ± 0.5 mBrms
Габариты (Ш × В × Г)	483 × 100 × 435 мм
Высота в стойке	2U
Вес нетто	10.3 кг (22.7 фунта)
Вес брутто	12.8 кг (28.2 фунта)
Артикул	CI1660DSPV2

NAD Electronics