Nine Fives Programmable Attenuator

Reference Manual

Nine Fives

None

Table of contents

1. Introduction

The Nine Fives Programmable SPDT Switch and Programmable Attenuator are network-controlled RF instruments. Each device provides three communication interfaces for monitoring and control:

- **Web UI** (Port 80) — Browser-based graphical interface for configuration and control
- **REST API** (Port 80) — JSON API for integration with custom applications and automation scripts
- **SCPI over TCP** (Port 5025) — Industry-standard text protocol for test and measurement applications
- **SCPI over HiSLIP** (Port 4880) — High-Speed LAN Instrument Protocol (IVI-6.1) for VISA-compatible instrument communication

All interfaces provide equivalent control over the device. Choose the interface that best fits your application.

1.1 Device Variants

Nine Fives produces two device variants:

VARIANT	DESCRIPTION
Programmable SPDT Switch	RF switch with two positions (RF0 / RF1)
Programmable Attenuator	Variable RF attenuator (0–62.5 dB in 0.25 dB steps)

Both variants share common system, network, and firmware management capabilities. The variant-specific commands and endpoints are documented in their respective sections.

1.2 About This Manual

This manual covers all available commands and endpoints for your device variant. Each section includes complete syntax, parameters, examples, and error codes.

2. Getting Started

2.1 Physical Connections

The device has two network interfaces:

INTERFACE	CONNECTOR	DEFAULT IP	DESCRIPTION
Ethernet	RJ-45	Assigned by DHCP (fallback: 192.168.0.95)	Primary network connection
USB-C	USB-C port	192.168.42.95	Direct connection to a host computer

Connect to the device using either interface. The USB-C connection is useful for initial setup or when no Ethernet network is available.

2.2 Available Ports

PORT	PROTOCOL	DESCRIPTION
80	HTTP	Web UI and REST API
5025	TCP	SCPI raw socket
4880	TCP	SCPI over HiSLIP (IVI-6.1)

2.3 Accessing the Web UI

Open a web browser and navigate to the device's IP address:

```
http://192.168.0.95
```

Or, if connected via USB-C:

```
http://192.168.42.95
```

The web UI provides a graphical interface for all device functions including system monitoring, network configuration, firmware updates, and device control.

2.4 Connecting via SCPI

2.4.1 Raw TCP

Use any TCP client to connect to port 5025:

```
telnet <device-ip> 5025
```

Send a command to verify the connection:

```
*IDN?
```

The device responds with its identification string.

2.4.2 HiSLIP (VISA)

Use a VISA library (such as PyVISA) with the following resource string:

```
TCPIP::<device-ip>::hislip0::INSTR
```

Example using Python:

```
import pyvisa
rm = pyvisa.ResourceManager()
inst = rm.open_resource("TCPIP::192.168.0.95::hislip0::INSTR")
print(inst.query("*IDN?"))
```

2.5 Using the REST API

The REST API accepts JSON requests on port 80. Example using `curl`:

```
curl http://192.168.0.95/api/system/status
```

See the REST API section for complete endpoint documentation.

2.6 Default Network Configuration

SETTING	ETHERNET	USB-C
Mode	DHCP Client with Static Fallback	DHCP Host
IP Address	192.168.0.95 (fallback)	192.168.42.95
Subnet Mask	255.255.0.0	255.255.255.0
DHCP Timeout	30 seconds	—

In the default Ethernet mode (`DHCP Client with Static Fallback`), the device first attempts to obtain an IP address from a DHCP server. If no DHCP server responds within 30 seconds, it falls back to the configured static IP address.

The USB-C interface runs a DHCP server by default, automatically assigning an IP address to the connected host computer.

3. Web UI

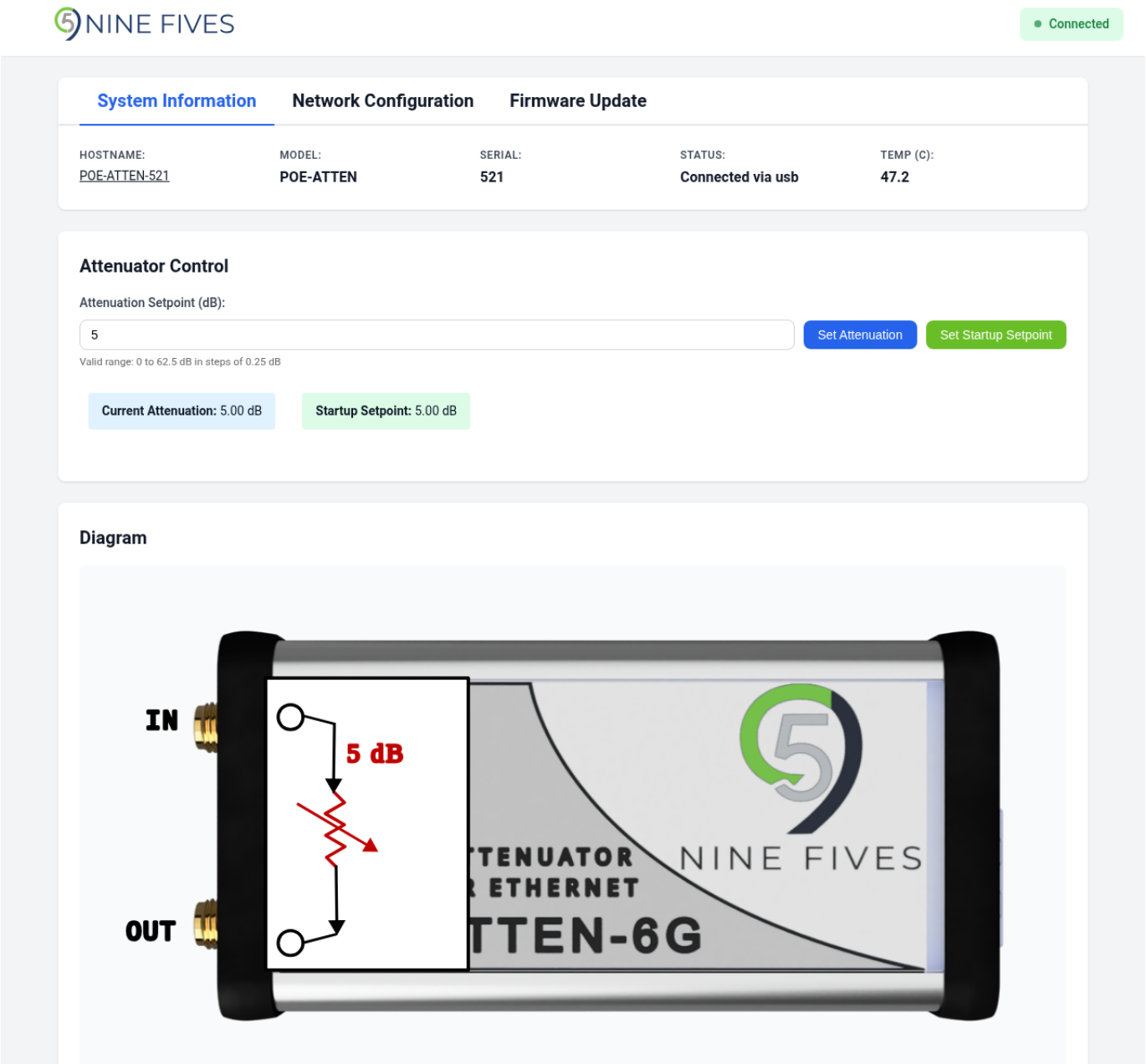
3.1 Web UI Overview

The Nine Fives web interface is accessible at `http://<device-ip>` on port 80. It provides a browser-based graphical interface for monitoring and controlling the device.

3.1.1 Layout

The interface consists of:

- **Header** — Nine Fives logo and connection status indicator
- **Tab bar** — Switches between System Information, Network Configuration, and Firmware Update views
- **Device control card** — Variant-specific controls (switch state or attenuation setpoint)
- **Notifications** — Success and error messages appear at the bottom of the page and auto-dismiss after a few seconds



3.1.2 Connection Status

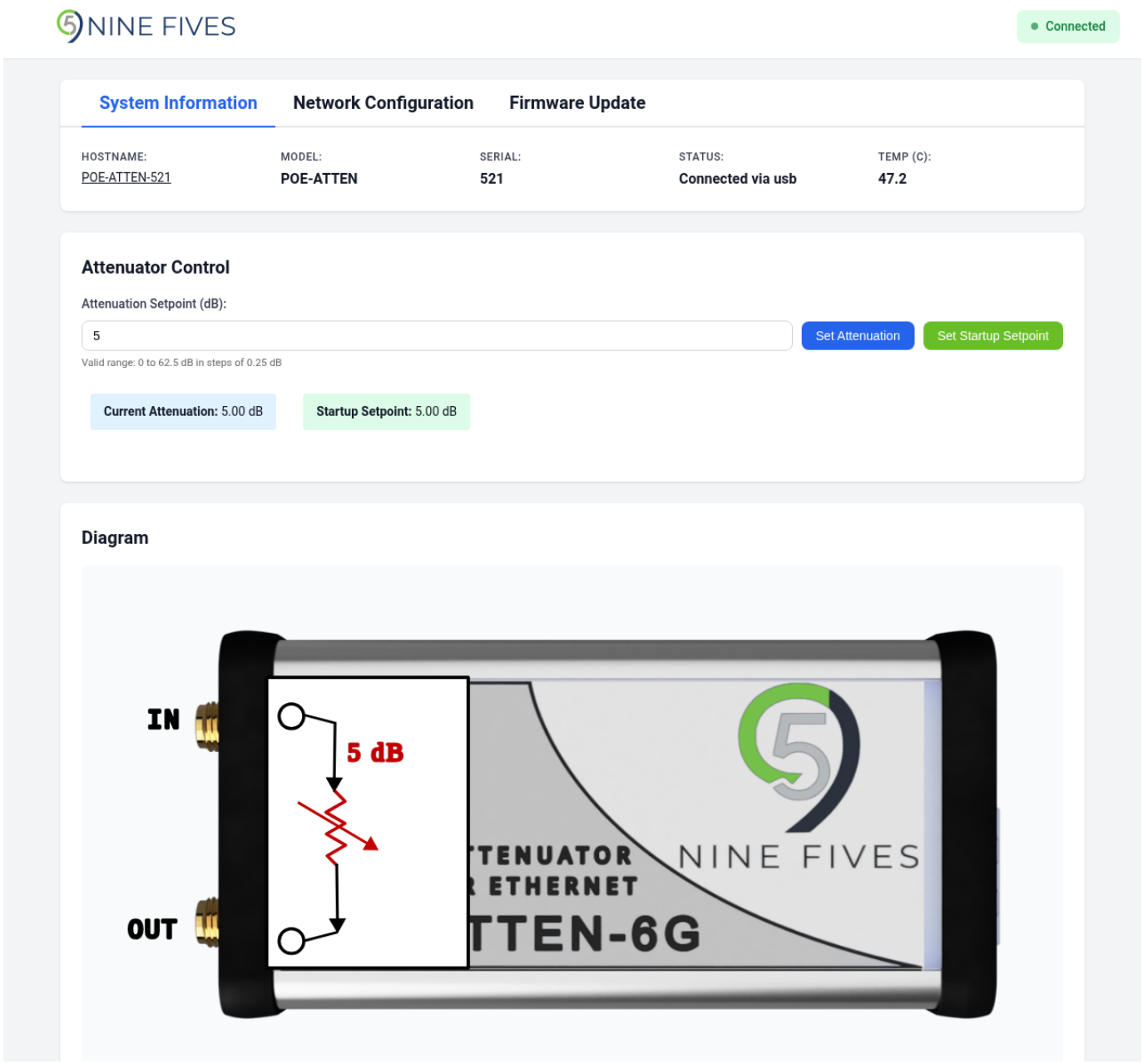
The header displays a connection status badge:

- **Connected** (green) — The browser is communicating with the device
- **Disconnected** (red) — Communication with the device has been lost

The interface polls the device every second to update the displayed state. If communication is lost, the status indicator turns red and controls are disabled until the connection is restored.

3.2 System Information

The **System Information** tab displays the current status of the device.



3.2.1 Displayed Fields

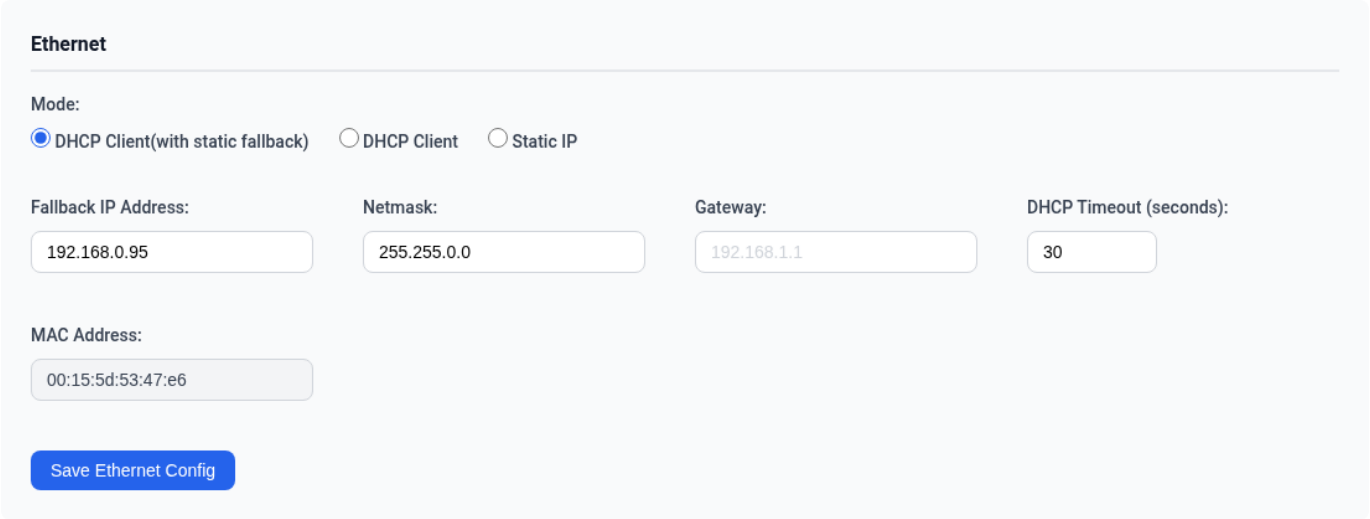
FIELD	DESCRIPTION
Hostname	The device hostname, displayed as a clickable link to `http://<hostname>.local`
Model	The device model identifier
Serial	The device serial number
Status	Connection type — "Connected via eth", "Connected via usb", or "Disconnected"
Temp (C)	The device temperature in degrees Celsius, updated in real time

The temperature reading updates automatically every second.

3.3 Network Configuration

The **Network Configuration** tab allows you to view and modify the network settings for both the Ethernet and USB-C interfaces.

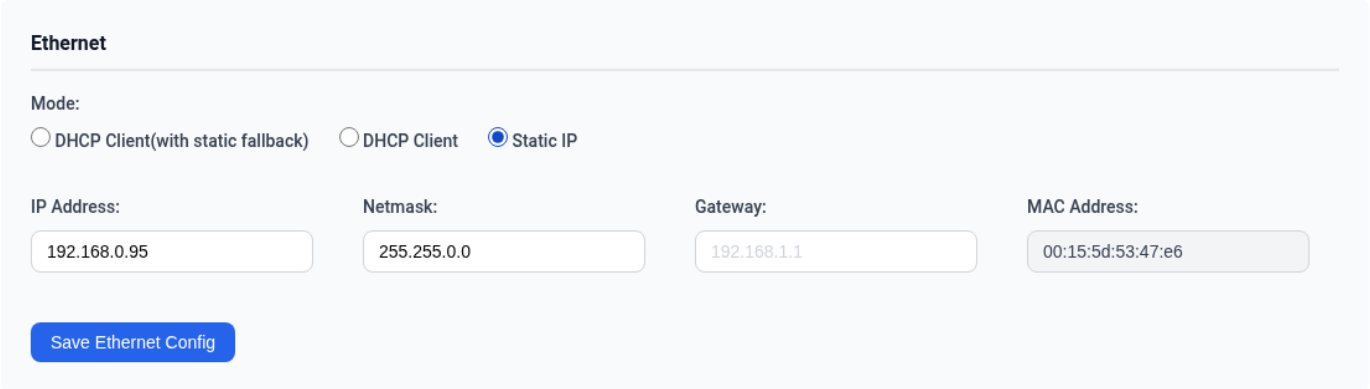
3.3.1 Ethernet Settings



Mode Selection

Select the Ethernet operating mode using the radio buttons:

MODE	DESCRIPTION
DHCP Client (with static fallback)	Attempts DHCP; falls back to the configured static IP address after a timeout. This is the factory default.
DHCP Client	Obtains an IP address from a DHCP server on the network. The IP address and netmask fields are disabled in this mode.
Static IP	Uses the configured IP address, netmask, and gateway.



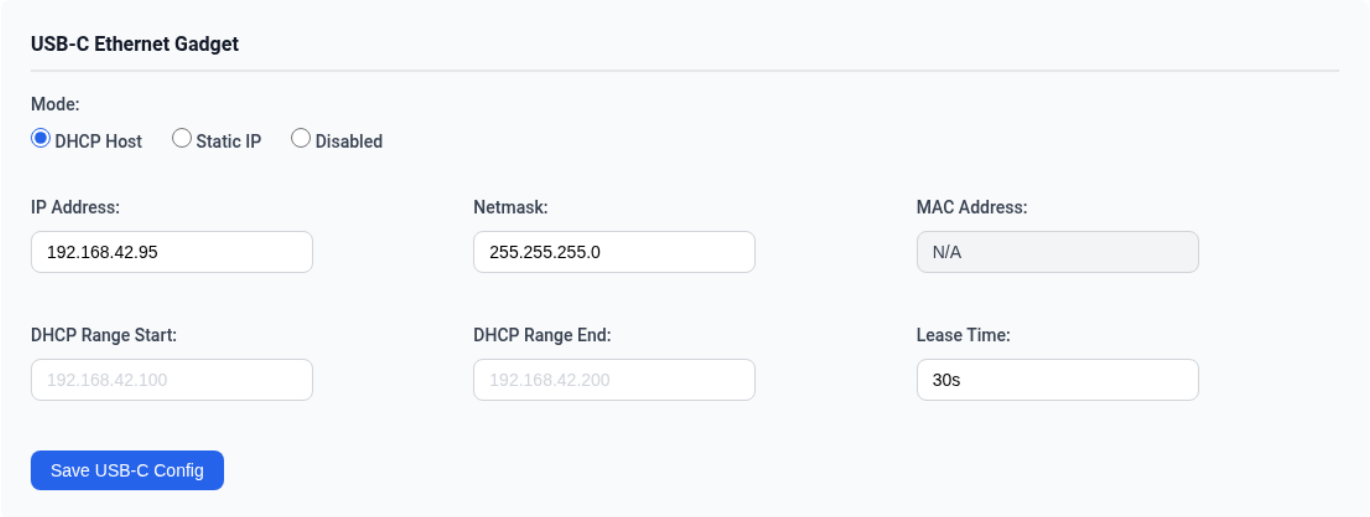
Configuration Fields

FIELD	DESCRIPTION	EDITABLE
IP Address / Fallback IP Address	The configured IP address. Label changes based on mode.	Yes (except in DHCP Client mode)
Netmask	The subnet mask	Yes (except in DHCP Client mode)
Gateway	The default gateway	Yes (except in DHCP Client mode)
DHCP Timeout	Seconds to wait for DHCP before falling back to static. Only shown in DHCP Client (with static fallback) mode.	Yes
MAC Address	The Ethernet interface MAC address	No (read-only)
Live IP	The currently active IP address on the interface	No (read-only)

Click **Save Ethernet Config** to apply changes. Settings take effect immediately.

> **WARNING**
>
> Changing network settings while connected via Ethernet may result in loss of connectivity. Ensure you have an alternative way to reach the device (e.g., via USB-C) before making changes.

3.3.2 USB-C Ethernet Gadget Settings



Mode Selection

MODE	DESCRIPTION
DHCP Host	Runs a DHCP server, automatically assigning an IP address to the connected computer. This is the factory default.
Static IP	Uses the configured IP address and netmask.
Disabled	Brings down the USB-C network interface. All fields are hidden.

Configuration Fields

FIELD	DESCRIPTION	EDITABLE
IP Address	The device IP address on the USB-C interface	Yes
Netmask	The subnet mask	Yes
MAC Address	The USB-C interface MAC address	No (read-only)
Live IP	The currently active IP address on the interface	No (read-only)

When **DHCP Host** mode is selected, additional fields appear:

FIELD	DESCRIPTION
DHCP Range Start	First IP address in the DHCP pool
DHCP Range End	Last IP address in the DHCP pool
Lease Time	DHCP lease duration (e.g., `30s`, `1h`, `24h`)

Click **Save USB-C Config** to apply changes.

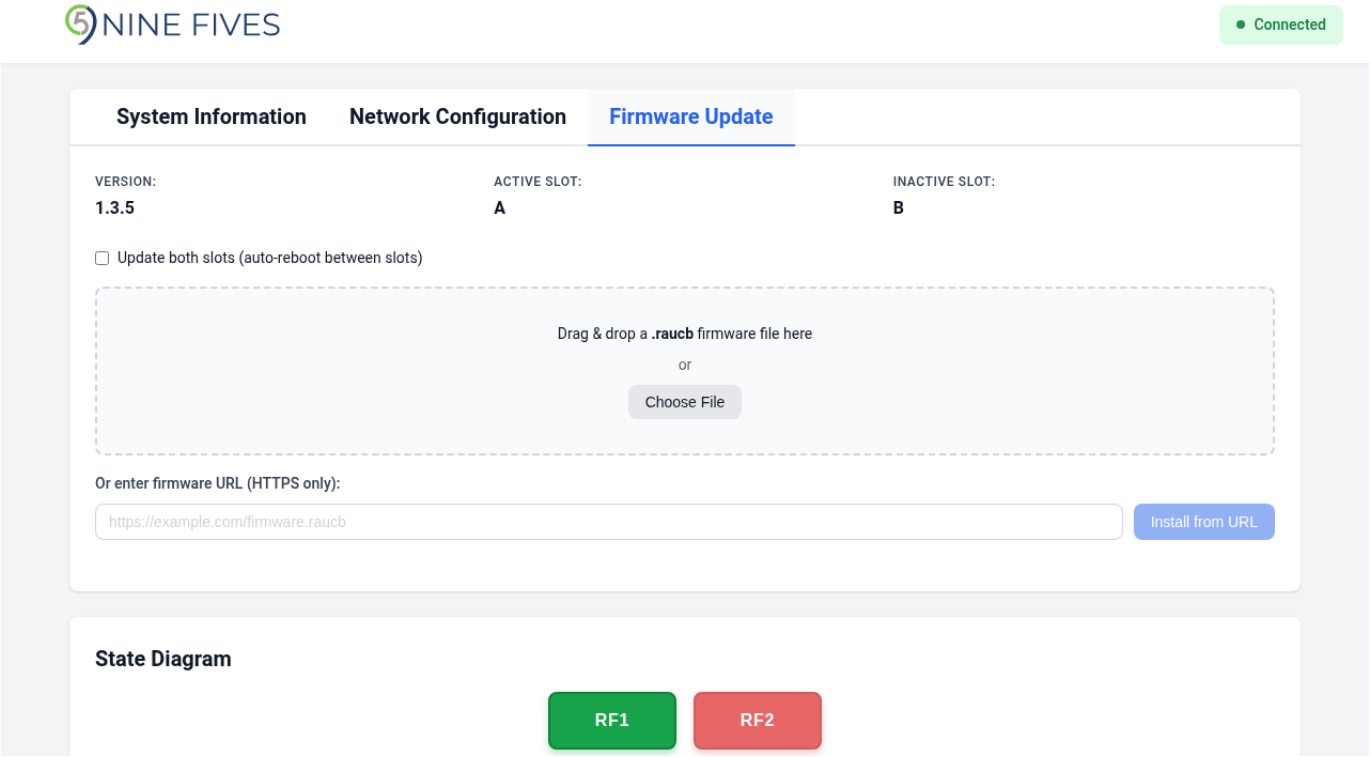
3.3.3 Factory Reset

Click **Reset Networking Settings to Factory Defaults** at the bottom of the network configuration panel to restore all network settings to their factory defaults. A confirmation dialog appears before the reset is applied.

> **WARNING**
>
> The device will briefly be unreachable while factory default settings are applied. The default configuration uses DHCP Client with Static Fallback mode on Ethernet (fallback IP: 192.168.0.95) and DHCP Host mode on USB-C (IP: 192.168.42.95).

3.4 Firmware Update

The **Firmware Update** tab displays the current firmware version and provides tools for updating the device firmware.



3.4.1 Firmware Information

FIELD	DESCRIPTION
Version	The currently running firmware version
Active Slot	The firmware slot currently booted (A or B)
Inactive Slot	The alternate firmware slot and its version

The device uses an A/B slot scheme. Firmware updates are installed to the inactive slot, so the currently running firmware is never modified during an update.

3.4.2 Update Methods

File Upload

Drag and drop a `.raucb` firmware bundle onto the upload area, or click **Choose File** to select one from your computer.

URL Install

Enter an HTTPS URL pointing to a `.raucb` firmware bundle and click **Install from URL**. The device downloads and installs the firmware directly.

3.4.3 Update Modes

MODE	DESCRIPTION
Single slot (default)	Installs firmware to the inactive slot only. After installation, click **Reboot to Activate New Firmware** to boot into the new firmware.
Both slots	Check the **Update both slots** checkbox before uploading. The device installs firmware to the inactive slot, reboots automatically, then installs to the second slot.

3.4.4 Update Progress

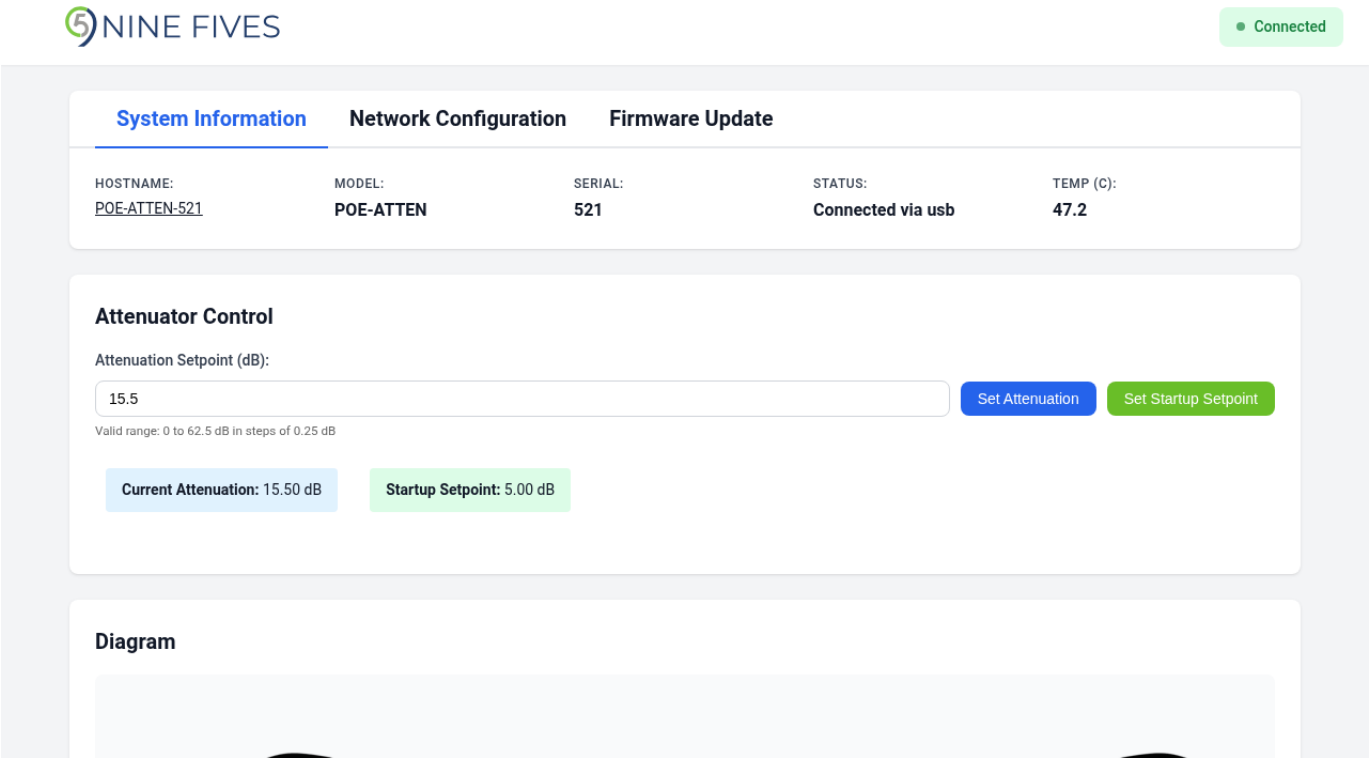
During an update, the upload area is replaced by a progress bar and status message. The interface polls for status every 2 seconds.

After a successful single-slot update, a **Reboot to Activate New Firmware** button appears. After a successful both-slots update, a confirmation message is displayed.

If an error occurs, the error message is displayed below the firmware information.

3.5 Attenuator Control

The **Attenuator Control** card provides direct control over the RF attenuation level.



3.5.1 Controls

CONTROL	DESCRIPTION
Attenuation Setpoint (dB)	Numeric input field. Valid range: 0 to 62.5 dB in steps of 0.25 dB.
Set Attenuation	Applies the entered value immediately.
Set Startup Setpoint	Saves the entered value as the power-on default. The device will apply this attenuation every time it boots.

3.5.2 Display

FIELD	DESCRIPTION
Current Attenuation	The attenuation value currently applied to the hardware, updated in real time.
Startup Setpoint	The saved power-on default attenuation value.

3.5.3 Diagram

A block diagram below the controls illustrates the signal path through the attenuator, displaying the current attenuation value.

4. REST API

4.1 REST API Overview

The device provides a JSON REST API on port 80 for programmatic control and monitoring.

4.1.1 Base URL

```
http://<device-ip>
```

All endpoints accept and return JSON (`Content-Type: application/json`) unless otherwise noted.

4.1.2 Error Handling

HTTP STATUS	DESCRIPTION
200	Success
202	Accepted (asynchronous operation started)
400	Bad Request — invalid input or parameters
405	Method Not Allowed — wrong HTTP method
409	Conflict — operation already in progress
500	Internal Server Error
503	Service Unavailable

Error responses are returned as plain text with a descriptive message.

4.1.3 Examples

The examples in this section use `curl`. Replace `<device-ip>` with your device's IP address.

```
# Query system status
curl http://<device-ip>/api/system/status

# POST with JSON body
curl -X POST -H "Content-Type: application/json" \
  -d '{"state": 1}' \
  http://<device-ip>/api/switch
```

4.2 System Endpoints

4.2.1 GET /api/system/status

Returns complete system status information.

Response:

```
{
  "device": "Nine Fives Programmable SPDT Switch",
  "hostname": "nf-device",
  "model": "NF-SW-01",
  "serial": "ABC123",
  "status": "Connected via eth",
  "temperature": 45.2,
  "connectedIface": "eth0"
}
```

FIELD	TYPE	DESCRIPTION
`device`	string	Device name
`hostname`	string	System hostname
`model`	string	Device model identifier
`serial`	string	Serial number
`status`	string	Connection status: `"Connected via eth"`, `"Connected via usb"`, or `"Disconnected"`
`temperature`	number	Device temperature in °C
`connectedIface`	string	Network interface the client is connected through (`eth0` or `usb0`)

Sub-Endpoints

Individual fields are available at dedicated endpoints:

ENDPOINT	RETURNS
`GET /api/system/status/device`	`{"device": "..."}`
`GET /api/system/status/hostname`	`{"hostname": "..."}`
`GET /api/system/status/model`	`{"model": "..."}`
`GET /api/system/status/serial`	`{"serial": "..."}`
`GET /api/system/status/status`	`{"status": "..."}`
`GET /api/system/status/temperature`	`{"temperature": 45.2}`
`GET /api/system/status/connectedIface`	`{"connectedIface": "eth0"}`

4.2.2 POST /api/system/reboot

Reboots the device. The device will be unreachable for approximately 30 seconds.

Request: No body required.

Response:

```
{
  "status": "ok",
```

```
  "message": "Rebooting in 3 seconds..."
}
```

4.3 Network Endpoints

4.3.1 GET /api/network/config

Returns network configuration for both the Ethernet and USB-C interfaces, including configured settings and live interface information.

Response:


```
{
  "ethernet": {
    "mode": "dhcp_client_fallback",
    "ip_address": "192.168.0.95",
    "netmask": "255.255.0.0",
    "gateway": "",
    "dhcp_fallback_timeout": 30,
    "dhcp_server": {
      "range_start": "192.168.0.100",
      "range_end": "192.168.0.199",
      "lease_time": "30s"
    },
    "live_ip": "192.168.0.95",
    "mac": "00:11:22:33:44:55"
  },
  "usb_gadget": {
    "mode": "dhcp_host",
    "ip_address": "192.168.42.95",
    "netmask": "255.255.255.0",
    "enabled": true,
    "dhcp_server": {
      "range_start": "192.168.42.100",
      "range_end": "192.168.42.200",
      "lease_time": "30s"
    },
    "live_ip": "192.168.42.95",
    "mac": "00:11:22:33:44:56"
  }
}
```


Ethernet Modes

MODE	DESCRIPTION
`dhcp client fallback`	Try DHCP; fall back to static IP after timeout (factory default)
`static`	Static IP address
`dhcp client`	Obtain IP from a network DHCP server

USB-C Modes

MODE	DESCRIPTION
`dhcp host`	Run DHCP server for the connected host computer (factory default)
`static`	Static IP address
`disabled`	Interface is brought down

Read-Only Fields

The `live ip` and `mac` fields are read-only and reflect the current state of the interface.

4.3.2 POST /api/network/config

Updates network configuration and applies it immediately. You can update Ethernet, USB-C, or both interfaces in a single request. Only include the fields you want to change.

> **WARNING**
>
> Changing network settings while connected may result in loss of connectivity. Settings take effect immediately.

Example — set Ethernet to static IP:

```
curl -X POST -H "Content-Type: application/json" -d '{
  "ethernet": {
    "mode": "static",
    "ip_address": "192.168.1.100",
    "netmask": "255.255.255.0",
    "gateway": "192.168.1.1"
  }
}' http://<device-ip>/api/network/config
```

Example — set Ethernet to DHCP client mode:

```
curl -X POST -H "Content-Type: application/json" -d '{
  "ethernet": {
    "mode": "dhcp_client"
  }
}' http://<device-ip>/api/network/config
```

Example — set Ethernet to DHCP client with static fallback:

```
curl -X POST -H "Content-Type: application/json" -d '{
  "ethernet": {
    "mode": "dhcp_client_fallback",
    "ip_address": "192.168.0.95",
    "netmask": "255.255.0.0",
    "dhcp_fallback_timeout": 30
  }
}' http://<device-ip>/api/network/config
```

Example — disable USB-C networking:

```
curl -X POST -H "Content-Type: application/json" -d '{
  "usb_gadget": {
    "mode": "disabled",
    "enabled": false
  }
}' http://<device-ip>/api/network/config
```

Response:

```
{
  "status": "ok"
}
```

Errors:

- `400` — Invalid request body, invalid or unknown mode, or invalid IP address
- `500` — Failed to save or apply configuration

4.3.3 POST /api/network/reset

Restores factory default network configuration and applies it immediately.

Request: No body required.

Response:

```
{
  "status": "ok"
}
```

> **WARNING**
>
> The device will briefly be unreachable while factory default settings are applied. The default configuration is DHCP Client with Static Fallback on Ethernet (fallback IP: 192.168.0.95) and DHCP Host on USB-C (IP: 192.168.42.95).

4.4 Firmware Endpoints

4.4.1 GET /api/firmware/status

Returns current firmware version, slot information, and update progress.

Response (idle):

```
{
  "version": "1.2.0-abc1234",
  "activeSlot": "A",
  "inactiveSlot": "B",
  "inactiveVersion": "1.1.0",
  "updateInProgress": false
}
```

FIELD	TYPE	DESCRIPTION
`version`	string	Currently running firmware version
`activeSlot`	string	Active firmware slot (`A` or `B`)
`inactiveSlot`	string	Inactive firmware slot
`inactiveVersion`	string	Firmware version on the inactive slot
`updateInProgress`	boolean	Whether an update is currently running
`updateProgress`	number	Progress percentage (0–100), present during updates
`updateMessage`	string	Human-readable status message, present during updates
`updateMode`	string	Empty for single-slot updates, `"full"` for both-slots updates
`updatePhase`	string	Current phase during a full update (see below)
`lastError`	string	Error message from the last failed update, if any

Full Update Phases

PHASE	DESCRIPTION
`installing_first`	Installing firmware to the inactive slot
`rebooting`	First slot done, device is about to reboot
`installing_second`	After reboot, installing firmware to the second slot

4.4.2 POST /api/firmware/update

Upload a `.raucb` firmware bundle for installation. Uses `multipart/form-data` encoding.

PARAMETER	DESCRIPTION
Form field: `firmware`	The `.raucb` file (max 256 MiB)
Query: `mode`	Empty (default) for single slot, `full` for both slots

Example — single slot:

```
curl -X POST -F "firmware=@firmware-1.2.0.raucb" \
  http://<device-ip>/api/firmware/update
```

Example — both slots:

```
curl -X POST -F "firmware=@firmware-1.2.0.raucb" \
  "http://<device-ip>/api/firmware/update?mode=full"
```

Response (202 Accepted):

```
{
  "status": "accepted",
  "message": "Installing firmware from firmware-1.2.0.raucb"
}
```

Installation runs asynchronously. Poll `GET /api/firmware/status` for progress.

In `mode=full`, the device reboots automatically after the first slot and installs the second slot on startup.

Errors:

- `400` — Missing file, invalid form data, or non-`.raucb` file
- `409` — Update already in progress
- `503` — Firmware manager not available

4.4.3 POST /api/firmware/update/url

Download and install a firmware bundle from a URL.

Request:

```
{
  "url": "https://example.com/firmware-1.2.0.raucb"
}
```

Only HTTPS URLs are accepted.

Response (202 Accepted):

```
{
  "status": "accepted",
  "message": "Downloading and installing firmware from URL"
}
```

Errors:

- `400` — Invalid URL, missing URL, or non-HTTPS URL
- `409` — Update already in progress

4.5 Attenuator Endpoints

4.5.1 GET /api/attenuator

Returns the current attenuation setpoint in dB.

Response:

```
{
  "setpoint": 15.5
}
```

4.5.2 POST /api/attenuator

Sets the attenuation setpoint in dB.

Request:

```
{
  "setpoint": 20.0
}
```

Valid range: 0 to 62.5 dB in steps of 0.25 dB.

Response:

```
{
  "setpoint": 20.0
}
```

Errors:

- `400` — Invalid request body
- `500` — Failed to set setpoint (value out of range or invalid step size)

4.5.3 GET /api/attenuator/startup

Returns the startup attenuation setpoint — the value applied when the device boots.

Response:

```
{
  "startup_setpoint": 62.5
}
```

4.5.4 POST /api/attenuator/startup

Sets the startup attenuation setpoint. This value is persisted and applied on every device boot.

Request:

```
{
  "startup_setpoint": 30.0
}
```

Valid range: 0 to 62.5 dB in steps of 0.25 dB.

Response:

```
{
  "startup_setpoint": 30.0
}
```

Errors:

- `400` — Invalid request body
- `500` — Failed to save startup setpoint

Examples:

```
# Set attenuation to 20 dB
curl -X POST -H "Content-Type: application/json" \
  -d '{"setpoint": 20.0}' \
  http://<device-ip>/api/attenuator

# Query current setpoint
curl http://<device-ip>/api/attenuator

# Save 30 dB as the startup default
curl -X POST -H "Content-Type: application/json" \
  -d '{"startup_setpoint": 30.0}' \
  http://<device-ip>/api/attenuator/startup
```

5. SCPI API

5.1 SCPI Overview

The device implements a SCPI (Standard Commands for Programmable Instruments) command interface over two transport protocols.

5.1.1 Transport Protocols

Raw TCP (Port 5025)

PROPERTY	VALUE
Protocol	TCP
Port	5025
Format	Text-based, newline-terminated

Example connections:

```
telnet <device-ip> 5025
nc <device-ip> 5025
```

HiSLIP (Port 4880)

HiSLIP (High-Speed LAN Instrument Protocol) is defined by IVI Foundation specification IVI-6.1. It provides improved performance over raw TCP and supports VISA resource strings.

PROPERTY	VALUE
Protocol	HiSLIP (IVI-6.1)
Port	4880
Vendor ID	NF

VISA resource string:

```
TCPIP::<device-ip>::hislip0::INSTR
```

Example (Python with PyVISA):

```
import pyvisa
rm = pyvisa.ResourceManager()
inst = rm.open_resource("TCPIP::192.168.0.95::hislip0::INSTR")
print(inst.query("*IDN?"))
```

All commands documented in this section are available on both transport protocols.

5.1.2 Command Syntax

Case Insensitivity

Commands are case-insensitive:

```
*IDN?
*idn?
*Idn?
```

Short Forms

SCPI supports abbreviated commands. Uppercase letters in the documentation indicate the minimum required characters:

FULL FORM	SHORT FORM
`:SYSTem:NETwork:ETHernet:MODE?`	`:SYST:NET:ETH:MODE?`
`:SYSTem:FIRMware:VERSion?`	`:SYST:FIRM:VERS?`
`SWITch:MAIN:STATe?`	`SWIT:MAIN:STAT?`

Multiple Commands

Multiple commands can be sent on one line, separated by semicolons:

```
*IDN?; :ATT?
```

5.1.3 Error Codes

CODE RANGE	DESCRIPTION
-100 to -199	Command errors (invalid syntax, missing or invalid parameters)
-200 to -299	Execution errors (hardware operation failures)

CODE	DESCRIPTION
-101	Invalid or missing parameter
-102	Invalid state value
-103	Invalid IP address
-104	Invalid or unsupported mode
-108	Invalid attenuation value
-200	Hardware or configuration operation failed

5.2 Common Commands (IEEE 488.2)

These commands are available on all device variants.

5.2.1 *IDN?

Query instrument identification.

Response:

```
Nine Fives,<controller_type>,<serial>,<version>
```

Where `<controller_type>` is `Switch_Controller` or `Attenuator_Controller`.

Example:

```
*IDN?
Nine Fives,Switch_Controller,0001,1.0.0
```

5.2.2 *RST

Reset instrument to default state.

- **Programmable SPDT Switch:** Sets switch to position A (state 0)
- **Programmable Attenuator:** Sets attenuation to 0 dB

Example:

```
*RST
```

5.2.3 *CLS

Clear status and error queue.

Example:

```
*CLS
```

5.3 System Commands

These commands are available on all device variants.

5.3.1 :SYSTem:FIRMware:VERSion?

Query the current firmware version.

Short form: `:SYST:FIRM:VERS?`

Response: Quoted string

Example:

```
:SYST:FIRM:VERS?
"1.2.0-abc1234"
```

5.3.2 :SYSTem:FIRMware:STATus?

Query firmware update status.

Short form: `:SYST:FIRM:STAT?`

Response: Status string in one of the following formats:

RESPONSE	DESCRIPTION
`"IDLE"`	No update in progress
`"UPDATING,<progress>,<message>"`	Update in progress with percentage and status
`"COMPLETE"`	Update completed, awaiting reboot
`"ERROR,<message>"`	Last update failed

Example:

```
:SYST:FIRM:STAT?
"IDLE"
```

5.3.3 :SYSTem:FIRMware:UPDate \<url>

Download a firmware bundle from a URL and install it. Only HTTPS URLs are accepted. The update runs asynchronously — poll `:SYSTem:FIRMware:STATus?` for progress.

Short form: `:SYST:FIRM:UPD`

Example:

```
:SYST:FIRM:UPD "https://example.com/firmware-1.2.0.raucb"
```

Errors:

- `-101` — Invalid or missing URL parameter
- `-200` — Update failed (already in progress, invalid URL scheme, etc.)

5.3.4 :SYSTem:REBoot

Reboot the device. The device will be unreachable for approximately 30 seconds.

Short form: `:SYST:REB`

Example:

```
:SYST:REB
```

5.4 Network Commands

These commands configure the Ethernet and USB-C network interfaces. Write commands take effect immediately.

5.4.1 Ethernet Interface

:SYSTem:NETwork:ETHernet:MODE

Query or set the Ethernet operating mode.

Short form: `:SYST:NET:ETH:MODE`

MODE	DESCRIPTION
`static`	Static IP address
`dhcp client`	Obtain IP from a DHCP server
`dhcp client fallback`	Try DHCP; fall back to static after timeout

Examples:

```
:SYST:NET:ETH:MODE?
dhcp_client_fallback

:SYST:NET:ETH:MODE static
```

:SYSTem:NETwork:ETHernet:IPADdress

Query or set the configured Ethernet IP address.

Short form: `:SYST:NET:ETH:IPAD`

Examples:

```
:SYST:NET:ETH:IPAD?
192.168.0.95

:SYST:NET:ETH:IPAD 192.168.1.100
```

:SYSTem:NETwork:ETHernet:SMASk

Query or set the Ethernet subnet mask.

Short form: `:SYST:NET:ETH:SMAS`

Examples:

```
:SYST:NET:ETH:SMAS?
255.255.0.0

:SYST:NET:ETH:SMAS 255.255.255.0
```

:SYSTem:NETwork:ETHernet:GATEway

Query or set the Ethernet default gateway.

Short form: `:SYST:NET:ETH:GATE`

Examples:

```
:SYST:NET:ETH:GATE?
192.168.1.1

:SYST:NET:ETH:GATE 10.0.0.1
```

:SYSTem:NETwork:ETHernet:DHCPTimeout

Query or set the DHCP fallback timeout in seconds. Used in `dhcp client fallback` mode.

Short form: `:SYST:NET:ETH:DHCPT`

Examples:

```
:SYST:NET:ETH:DHCPT?
30

:SYST:NET:ETH:DHCPT 60
```

:SYSTem:NETwork:ETHernet:DHCP:STARt

Query or set the DHCP server pool start address.

Short form: `:SYST:NET:ETH:DHCP:STAR`

Examples:

```
:SYST:NET:ETH:DHCP:STAR?
192.168.0.100

:SYST:NET:ETH:DHCP:STAR 192.168.0.50
```

:SYSTem:NETwork:ETHernet:DHCP:END

Query or set the DHCP server pool end address.

Short form: `:SYST:NET:ETH:DHCP:END`

Examples:

```
:SYST:NET:ETH:DHCP:END?
192.168.0.199

:SYST:NET:ETH:DHCP:END 192.168.0.250
```

:SYSTem:NETwork:ETHernet:DHCP:LEASetime

Query or set the DHCP lease time.

Short form: `:SYST:NET:ETH:DHCP:LEAS`

Examples:

```
:SYST:NET:ETH:DHCP:LEAS?
30s

:SYST:NET:ETH:DHCP:LEAS 1h
```

:SYSTem:NETwork:ETHernet:LIVeip?

Query the live IP address currently active on the Ethernet interface. Read-only.

Short form: `:SYST:NET:ETH:LIV?`

Example:

```
:SYST:NET:ETH:LIV?
192.168.0.95
```

:SYSTem:NETwork:ETHernet:MAC?

Query the Ethernet MAC address. Read-only.

Short form: `:SYST:NET:ETH:MAC?`

Example:

```
:SYST:NET:ETH:MAC?
00:11:22:33:44:55
```

5.4.2 USB-C Gadget Interface

:SYSTem:NETwork:USB:MODE

Query or set the USB-C networking mode.

Short form: `:SYST:NET:USB:MODE`

MODE	DESCRIPTION
`static`	Static IP address
`dhcp host`	Run DHCP server for connected host
`disabled`	Interface is brought down

Examples:

```
:SYST:NET:USB:MODE?
dhcp_host

:SYST:NET:USB:MODE static
```

:SYSTem:NETwork:USB:IPADdress

Query or set the configured USB-C IP address.

Short form: `:SYST:NET:USB:IPAD`

Examples:

```
:SYST:NET:USB:IPAD?
192.168.42.95

:SYST:NET:USB:IPAD 192.168.42.10
```

:SYSTem:NETwork:USB:SMASk

Query or set the USB-C subnet mask.

Short form: `:SYST:NET:USB:SMAS`

Examples:

```
:SYST:NET:USB:SMAS?
255.255.255.0

:SYST:NET:USB:SMAS 255.255.255.0
```

:SYSTem:NETwork:USB:ENABled

Query or set the USB-C interface enabled state.

Short form: `:SYST:NET:USB:ENAB`

Examples:

```
:SYST:NET:USB:ENAB?
1

:SYST:NET:USB:ENAB 0
```

:SYSTem:NETwork:USB:DHCP:STARt

Query or set the USB-C DHCP server pool start address.

Short form: `:SYST:NET:USB:DHCP:STAR`

Examples:

```
:SYST:NET:USB:DHCP:STAR?
192.168.42.100

:SYST:NET:USB:DHCP:STAR 192.168.42.50
```

:SYSTem:NETwork:USB:DHCP:END

Query or set the USB-C DHCP server pool end address.

Short form: `:SYST:NET:USB:DHCP:END`

Examples:

```
:SYST:NET:USB:DHCP:END?
192.168.42.200

:SYST:NET:USB:DHCP:END 192.168.42.250
```

:SYSTem:NETwork:USB:DHCP:LEASetime

Query or set the USB-C DHCP lease time.

Short form: `:SYST:NET:USB:DHCP:LEAS`

Examples:

```
:SYST:NET:USB:DHCP:LEAS?
30s

:SYST:NET:USB:DHCP:LEAS 24h
```

:SYSTem:NETwork:USB:LIVeip?

Query the live IP address currently active on the USB-C interface. Read-only.

Short form: `:SYST:NET:USB:LIV?`

Example:

```
:SYST:NET:USB:LIV?
192.168.42.95
```

:SYSTem:NETwork:USB:MAC?

Query the USB-C MAC address. Read-only.

Short form: `:SYST:NET:USB:MAC?`

Example:

```
:SYST:NET:USB:MAC?
00:11:22:33:44:56
```

5.4.3 Network Reset

:SYSTem:NETwork:RESet

Restore factory default network configuration and apply immediately.

Short form: `:SYST:NET:RES`

Example:

```
:SYST:NET:RES
```

5.4.4 Quick Reference

Ethernet

COMMAND	SHORT FORM	R/W	DESCRIPTION
`:SYSTem:NETwork:ETHernet:MODE`	`:SYST:NET:ETH:MODE`	R/W	Ethernet operating mode
`:SYSTem:NETwork:ETHernet:IPADdress`	`:SYST:NET:ETH:IPAD`	R/W	Configured IP address
`:SYSTem:NETwork:ETHernet:SMASk`	`:SYST:NET:ETH:SMAS`	R/W	Subnet mask
`:SYSTem:NETwork:ETHernet:GATEway`	`:SYST:NET:ETH:GATE`	R/W	Default gateway
`:SYSTem:NETwork:ETHernet:DHCPTimeout`	`:SYST:NET:ETH:DHCPT`	R/W	DHCP fallback timeout (seconds)
`:SYSTem:NETwork:ETHernet:DHCP:STARt`	`:SYST:NET:ETH:DHCP:STAR`	R/W	DHCP pool start
`:SYSTem:NETwork:ETHernet:DHCP:END`	`:SYST:NET:ETH:DHCP:END`	R/W	DHCP pool end
`:SYSTem:NETwork:ETHernet:DHCP:LEASetime`	`:SYST:NET:ETH:DHCP:LEAS`	R/W	DHCP lease time
`:SYSTem:NETwork:ETHernet:LIVeip?`	`:SYST:NET:ETH:LIV?`	R	Live IP address
`:SYSTem:NETwork:ETHernet:MAC?`	`:SYST:NET:ETH:MAC?`	R	MAC address

USB-C

COMMAND	SHORT FORM	R/W	DESCRIPTION
`:SYSTem:NETwork:USB:MODE`	`:SYST:NET:USB:MODE`	R/W	USB-C operating mode
`:SYSTem:NETwork:USB:IPADdress`	`:SYST:NET:USB:IPAD`	R/W	Configured IP address
`:SYSTem:NETwork:USB:SMASk`	`:SYST:NET:USB:SMAS`	R/W	Subnet mask
`:SYSTem:NETwork:USB:ENABled`	`:SYST:NET:USB:ENAB`	R/W	Enabled state (0/1)
`:SYSTem:NETwork:USB:DHCP:STARt`	`:SYST:NET:USB:DHCP:STAR`	R/W	DHCP pool start
`:SYSTem:NETwork:USB:DHCP:END`	`:SYST:NET:USB:DHCP:END`	R/W	DHCP pool end
`:SYSTem:NETwork:USB:DHCP:LEASetime`	`:SYST:NET:USB:DHCP:LEAS`	R/W	DHCP lease time
`:SYSTem:NETwork:USB:LIVeip?`	`:SYST:NET:USB:LIV?`	R	Live IP address
`:SYSTem:NETwork:USB:MAC?`	`:SYST:NET:USB:MAC?`	R	MAC address

5.5 Attenuator Commands

5.5.1 :ATT?

Query the current attenuation setpoint.

Response: Floating point value in dB

Example:

```
:ATT?
31.25
```

5.5.2 :SETATT \<value>

Set the attenuation setpoint.

Parameters:

- `<value>` — Attenuation in dB. Valid range: 0 to 62.5, in steps of 0.25.

Example:

```
:SETATT 45.5
```

Errors:

- `-108` — Invalid attenuation value (out of range or invalid step size)

5.5.3 :STARTUPATT:VALUE?

Query the startup attenuation value — the attenuation applied when the device boots.

Short form: `:STARTUPATT:VAL?`

Response: Floating point value in dB

Example:

```
:STARTUPATT:VALUE?
62.5
```

5.5.4 :STARTUPATT:VALUE \<value>

Set the startup attenuation value. This value is persisted and applied on every device boot.

Short form: `:STARTUPATT:VAL`

Parameters:

- `<value>` — Attenuation in dB. Valid range: 0 to 62.5, in steps of 0.25.

Example:

```
:STARTUPATT:VALUE 30.0
```

Errors:

- `-108` — Invalid attenuation value (out of range or invalid step size)

5.5.5 Quick Reference

COMMAND	DESCRIPTION
`*IDN?`	Query identification
`*RST`	Reset to 0 dB
`*CLS`	Clear error queue
`:ATT?`	Query current attenuation
`:SETATT <dB>`	Set attenuation (0–62.5 dB, 0.25 steps)
`:STARTUPATT:VALUE?`	Query startup attenuation
`:STARTUPATT:VALUE <dB>`	Set startup attenuation
`:SYSTem:FIRMware:VERSion?`	Query firmware version
`:SYSTem:FIRMware:STATus?`	Query update status
`:SYSTem:FIRMware:UPDate <url>`	Install firmware from URL
`:SYSTem:REBoot`	Reboot device