

# Nine Fives Programmable SPDT Switch

---

Reference Manual

*Nine Fives*

*None*

## Table of contents

---

|                                   |    |
|-----------------------------------|----|
| 1. Introduction                   | 3  |
| 1.1 Device Variants               | 3  |
| 1.2 About This Manual             | 3  |
| 2. Getting Started                | 4  |
| 2.1 Physical Connections          | 4  |
| 2.2 Available Ports               | 4  |
| 2.3 Accessing the Web UI          | 4  |
| 2.4 Connecting via SCPI           | 4  |
| 2.5 Using the REST API            | 5  |
| 2.6 Default Network Configuration | 5  |
| 3. Web UI                         | 6  |
| 3.1 Web UI Overview               | 6  |
| 3.2 System Information            | 8  |
| 3.3 Network Configuration         | 10 |
| 3.4 Firmware Update               | 13 |
| 3.5 SPDT Control                  | 15 |
| 4. Touch UI                       | 17 |
| 4.1 Touch UI - SPDT               | 17 |
| 5. REST API                       | 21 |
| 5.1 REST API Overview             | 21 |
| 5.2 System Endpoints              | 22 |
| 5.3 Network Endpoints             | 24 |
| 5.4 Firmware Endpoints            | 27 |
| 5.5 SPDT Endpoints                | 30 |
| 6. SCPI API                       | 0  |
| 6.1 SCPI Overview                 | 0  |
| 6.2 Common Commands (IEEE 488.2)  | 0  |
| 6.3 System Commands               | 0  |
| 6.4 Network Commands              | 0  |
| 6.5 SPDT Switch Commands          | 0  |

# 1. Introduction

---

The Nine Fives Programmable SPDT Switch, Programmable SP4T 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 the following device variants:

| VARIANT                         | DESCRIPTION                                          |
|---------------------------------|------------------------------------------------------|
| <b>Programmable SPDT Switch</b> | RF switch with two throws (RF0 / RF1)                |
| <b>Programmable SP4T Switch</b> | RF switch with four throws (RF0-RF3)                 |
| <b>Programmable Attenuator</b>  | Variable RF attenuator (0-95.25 dB in 0.25 dB steps) |

See the [SPDT](#), [SP4T](#), or [attenuator](#) product specifications for RF performance, power limits, environmental ratings, and connector details.

All 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::::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

## System Information

## Network Configuration

## Firmware Update

HOSTNAME:  
POE-SWITCH-223

MODEL:  
POE-SWITCH

SERIAL:  
223

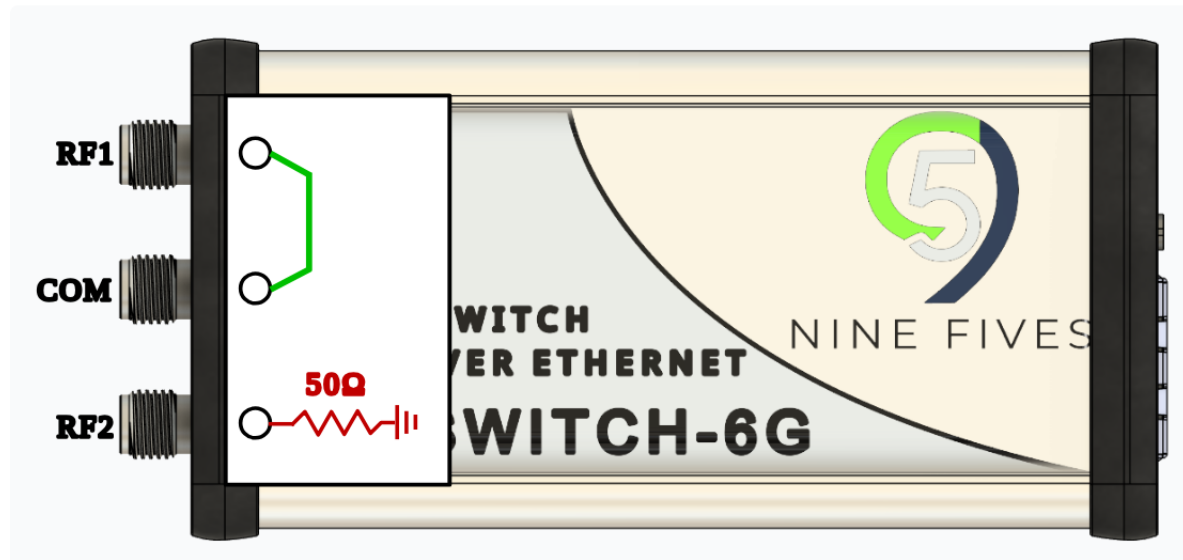
STATUS:  
Connected via usb

TEMP (C):  
47.2

## State Diagram

RF1

RF2



Nine Fives - POE-SWITCH-223 © 2025 | 4/15/2026, 10:06:43 PM

### 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.



Connected

System InformationNetwork ConfigurationFirmware Update

HOSTNAME:  
POE-SWITCH-223

MODEL:  
POE-SWITCH

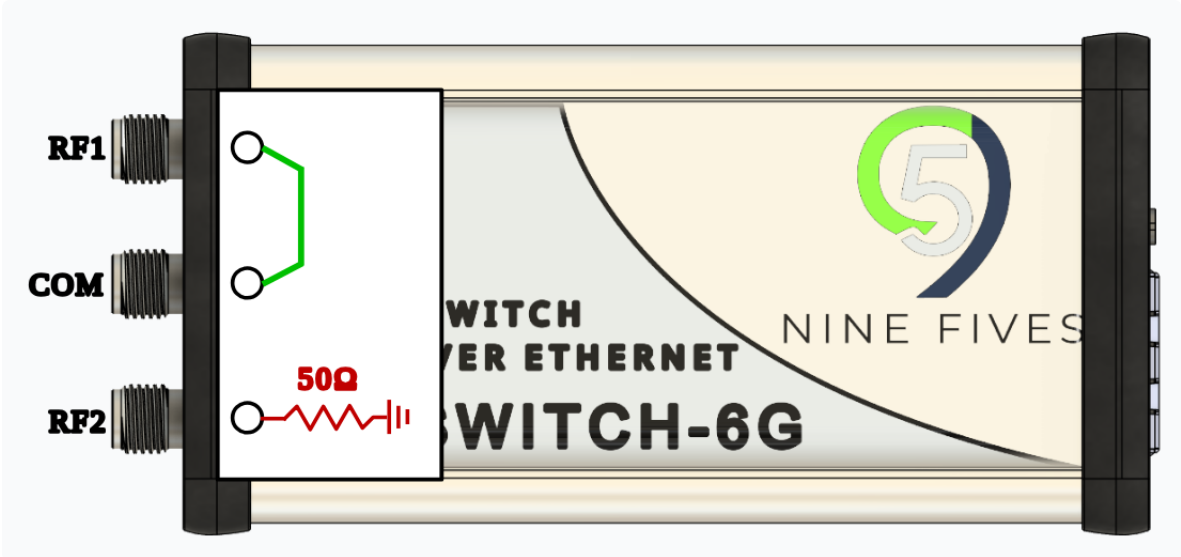
SERIAL:  
223

STATUS:  
Connected via usb

TEMP (C):  
47.2

State Diagram

RF1RF2



Nine Fives - POE-SWITCH-223 © 2025 | 4/15/2026, 10:06:43 PM

### 3.2.1 Displayed Fields

| FIELD           | DESCRIPTION                                                                                      |
|-----------------|--------------------------------------------------------------------------------------------------|
| <b>Hostname</b> | The device hostname, displayed as a clickable link to <code>http://&lt;hostname&gt;.local</code> |
| <b>Model</b>    | The device model identifier                                                                      |
| <b>Serial</b>   | The device serial number                                                                         |
| <b>Status</b>   | Connection type — "Connected via eth", "Connected via usb", or "Disconnected"                    |
| <b>Temp (C)</b> | 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

Ethernet

Mode:  
☒ DHCP Client(with static fallback)    ☐ DHCP Client    ☐ Static IP

Fallback IP Address:  
192.168.0.95

Netmask:  
255.255.0.0

Gateway:  
192.168.1.1

DHCP Timeout (seconds):  
30

MAC Address:  
00:15:5d:53:47:e6

Save Ethernet Config

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.                                                                 |

Ethernet

Mode:  
☐ DHCP Client(with static fallback)    ☐ DHCP Client    ☒ Static IP

IP Address:  
192.168.0.95

Netmask:  
255.255.0.0

Gateway:  
192.168.1.1

MAC Address:  
00:15:5d:53:47:e6

Save Ethernet Config

Configuration Fields

| FIELD                               | DESCRIPTION                                                                                                    | EDITABLE                         |
|-------------------------------------|----------------------------------------------------------------------------------------------------------------|----------------------------------|
| IP Address /<br>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

USB-C Ethernet Gadget

Mode:

☒ DHCP Host ☐ Static IP ☐ Disabled

IP Address:

192.168.42.95

Netmask:

255.255.255.0

MAC Address:

N/A

DHCP Range Start:

192.168.42.100

DHCP Range End:

192.168.42.200

Lease Time:

30s

Save USB-C Config

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., <input type="text" value="30s"/> , <input type="text" value="1h"/> , <input type="text" value="24h"/> ) |

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

3.3.3 Factory Reset

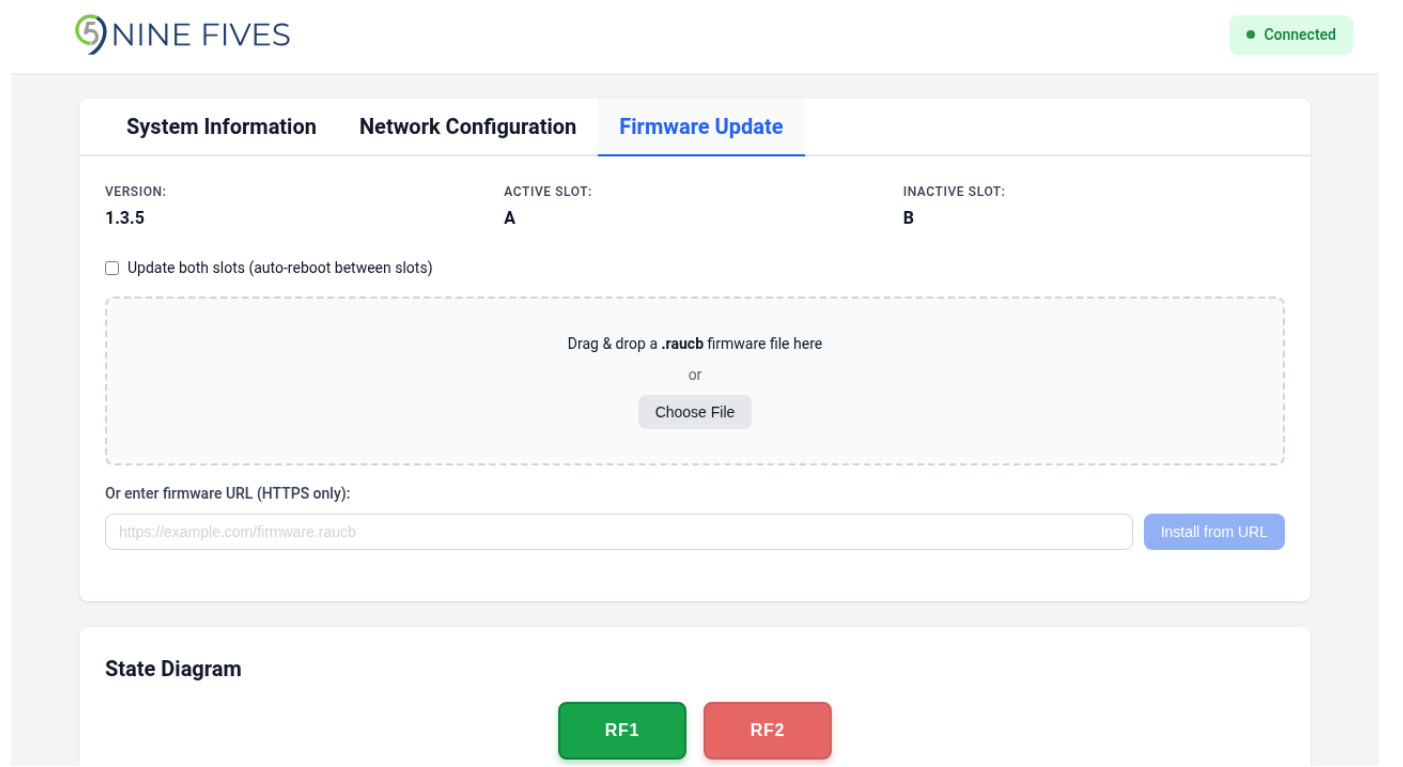
Click **Reset Networking Settings to Factory Defaults** at the bottom of the network configuration panel to restore all network settings to their factory defaults. A confirmation dialog appears before the reset is applied.

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                                                                                                                                                                                               |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Single slot</b><br>(default) | Installs firmware to the inactive slot only. After installation, click <b>Reboot to Activate New Firmware</b> to boot into the new firmware.                                                              |
| <b>Both slots</b>               | Check the <b>Update both slots</b> checkbox before uploading <i>or</i> installing from a URL. The device installs firmware to the inactive slot, reboots automatically, then installs to the second slot. |

### 3.4.4 Both-Slots Updates

The firmware bundle is too large to stage on the device's persistent storage, so it must be re-supplied for the second slot after the mid-cycle reboot. How that happens depends on how you started the update:

- **File upload:** keep the browser tab open. The page holds the bundle in memory and re-uploads it automatically once the device comes back up to finish the second slot.
- **URL install:** the device re-downloads the bundle from the URL on its own — no need to keep the tab open.

If a file-upload both-slots update is interrupted (for example the tab is closed or reloaded before the second slot finishes), the device is left running the new firmware on the first slot with the previous firmware still on the second slot — both remain bootable. When you next open the firmware page, a banner lets you **re-select the same firmware file to finish** the second slot, or **cancel** to leave it as-is.

### 3.4.5 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, and keeps polling across the automatic reboot during a both-slots update.

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 SPDT Control

The **State Diagram** card provides direct control over the RF switch position.



Connected

System InformationNetwork ConfigurationFirmware Update

HOSTNAME:  
POE-SWITCH-223

MODEL:  
POE-SWITCH

SERIAL:  
223

STATUS:  
Connected via usb

TEMP (C):  
47.2

State Diagram

RF1RF2



#### 3.5.1 Controls

Two buttons select the active switch position:

| BUTTON | DESCRIPTION                         |
|--------|-------------------------------------|
| RF0    | Routes the signal through RF port 0 |
| RF1    | Routes the signal through RF port 1 |

The active position button is highlighted. Both buttons are disabled when the device is disconnected.

#### 3.5.2 State Diagram

A diagram below the buttons shows the current signal routing path, updating in real time when the switch position changes.

**System Information****Network Configuration****Firmware Update**HOSTNAME:  
POE-SWITCH-223MODEL:  
**POE-SWITCH**SERIAL:  
**223**STATUS:  
**Connected via usb**TEMP (C):  
**47.2****State Diagram**

RF1

RF2



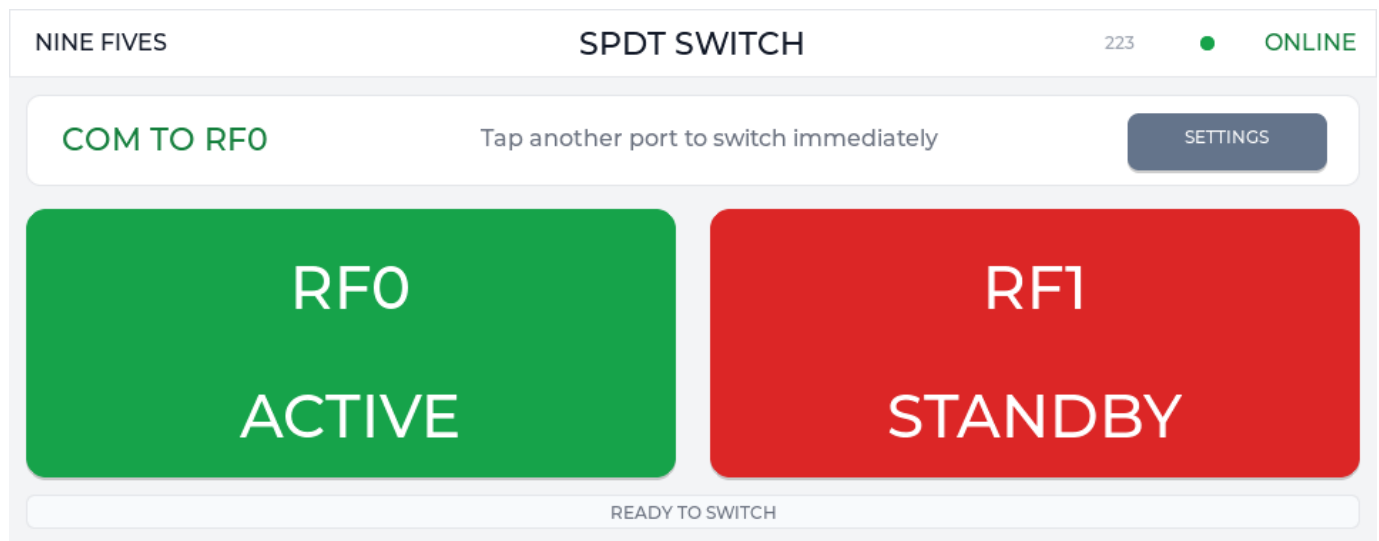
## 4. Touch UI

### 4.1 Touch UI - SPDT

The SPDT touch UI provides direct control of the RF path from the front-panel touchscreen. The main screen is built around two large port cards and a route summary so the active path is visible at a glance.

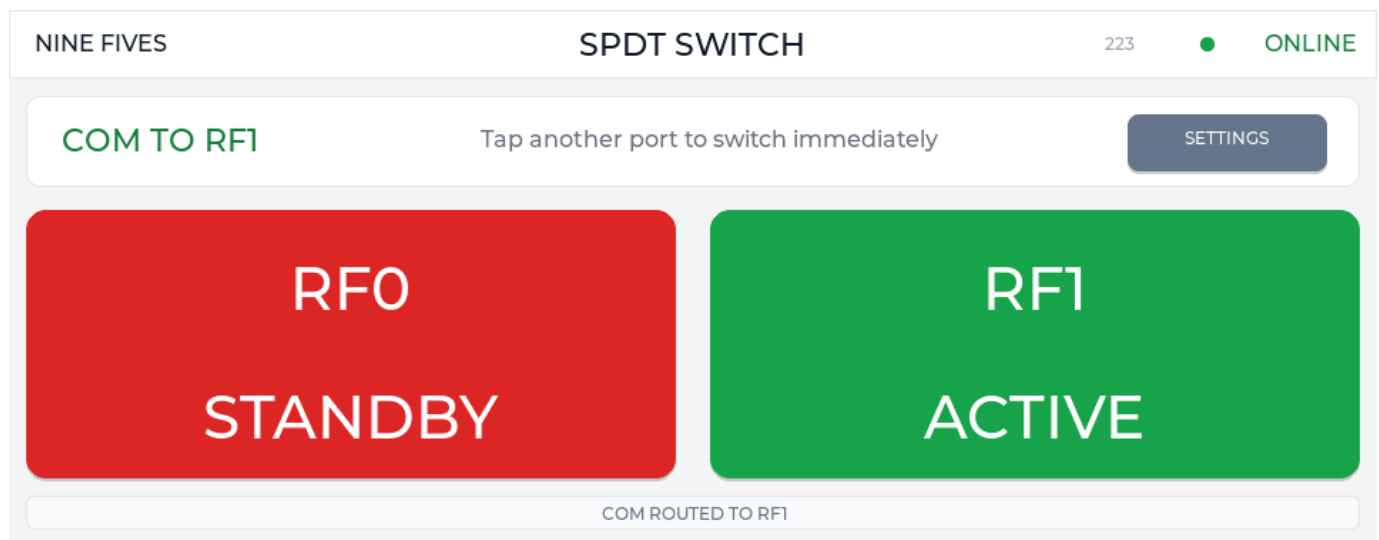
For the four-throw variant, see [Touch UI - SP4T](#).

#### 4.1.1 RF0 Active



When **RF0** is active, the RF0 card is green, the RF1 card is red, and the route summary reads **COM TO RF0**. Tapping **RF1** immediately requests a route change through the local runtime API.

#### 4.1.2 RF1 Active



When **RF1** is active, the RF1 card is green, the RF0 card is red, and the route summary reads **COM TO RF1**.

4.1.3 Network Settings

NINE FIVES

NETWORK

521

●

ONLINE

ETHERNET

10.0.0.1/24

static

USB-C

192.168.42.95/24

DHCP server

BACK

STATUS

READY TO ATTENUATE

RESET NETWORKING

The **Settings** control opens the shared network overview. The overview shows Ethernet and USB-C live addresses, the active mode for each interface, and the reset networking action.

Ethernet Settings

NINE FIVES

ETHERNET

521

●

ONLINE

DHCP + FALLBACK

DHCP CLIENT

STATIC

STATIC IP

10.0.0.1

EDIT

NETMASK

255.255.255.0

EDIT

GATEWAY

10.0.0.254

EDIT

DHCP FALLBACK TIMEOUT

30

OFF

LIVE ADDRESS

10.0.0.1/24

BACK

READY TO ATTENUATE

SAVE ETHERNET

The Ethernet screen selects DHCP fallback, DHCP client, or static mode. It also exposes the configured IP address, netmask, gateway, fallback timeout, live address, and save action.

## IP Editor

The screenshot shows the 'IP Editor' interface. At the top, it says 'NINE FIVES' on the left, 'EDIT ADDRESS' in the center, and '521' with a green dot and 'ONLINE' on the right. Below this, there are four octet editors. Each octet editor consists of a blue button with a '+' sign, a large number (10, 0, 0, 1 respectively), and a grey button with a '-' sign. At the bottom left is a grey 'BACK' button, and at the bottom right is a green 'APPLY VALUE' button.

Address fields open the four-octet editor. Each octet can be adjusted with the + and - controls before applying the value.

The screenshot shows the 'IP Editor' interface with a numeric keypad. On the left, there is a numeric keypad with buttons for digits 1-9, 0, and a 'CLR' button. To the right of the keypad are two octet editors. The first octet editor has a blue '+' button, the number '0', and a grey '-' button. The second octet editor has a blue '+' button, the number '1', and a grey '-' button. At the bottom left is a grey 'BACK' button, and at the bottom right is a green 'APPLY VALUE' button.

Tapping an octet opens the numeric keypad for direct entry. **Apply Value** writes the edited address back to the network settings screen.

USB-C Settings

NINE FIVES

USB-C

521

ONLINE

DHCP HOST

STATIC

DISABLED

DEVICE IP

192.168.42.95

EDIT

NETMASK

255.255.255.0

EDIT

LEASE TIME

30s

-

+

DHCP START

192.168.42.100

EDIT

DHCP END

192.168.42.200

EDIT

LIVE ADDRESS

192.168.42.95/24

BACK

READY TO ATTENUATE

SAVE USB-C

The USB-C screen selects DHCP host, static, or disabled mode. In DHCP host mode it also shows the DHCP lease time and address range.

4.1.4 Controls

| CONTROL    | DESCRIPTION                                           |
|------------|-------------------------------------------------------|
| RF0        | Routes COM to RF0.                                    |
| RF1        | Routes COM to RF1.                                    |
| Settings   | Opens the shared network configuration screens.       |
| Status bar | Shows the latest runtime status or switching message. |

## 5. REST API

---

### 5.1 REST API Overview

---

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

#### 5.1.1 Base URL

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

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

#### 5.1.2 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.

#### 5.1.3 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/spdt
```

## 5.2 System Endpoints

### 5.2.1 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                                                                                                             |
|-----------------------------|--------|-------------------------------------------------------------------------------------------------------------------------|
| <code>device</code>         | string | Device name                                                                                                             |
| <code>hostname</code>       | string | System hostname                                                                                                         |
| <code>model</code>          | string | Device model identifier                                                                                                 |
| <code>serial</code>         | string | Serial number                                                                                                           |
| <code>status</code>         | string | Connection status: <code>"Connected via eth"</code> , <code>"Connected via usb"</code> , or <code>"Disconnected"</code> |
| <code>temperature</code>    | number | Device temperature in °C                                                                                                |
| <code>connectedIface</code> | string | Network interface the client is connected through ( <code>eth0</code> or <code>usb0</code> )                            |

**Sub-Endpoints**

Individual fields are available at dedicated endpoints:

| ENDPOINT                                           | RETURNS                                  |
|----------------------------------------------------|------------------------------------------|
| <code>GET /api/system/status/device</code>         | <code>{"device": "..."} </code>          |
| <code>GET /api/system/status/hostname</code>       | <code>{"hostname": "..."} </code>        |
| <code>GET /api/system/status/model</code>          | <code>{"model": "..."} </code>           |
| <code>GET /api/system/status/serial</code>         | <code>{"serial": "..."} </code>          |
| <code>GET /api/system/status/status</code>         | <code>{"status": "..."} </code>          |
| <code>GET /api/system/status/temperature</code>    | <code>{"temperature": 45.2} </code>      |
| <code>GET /api/system/status/connectedIface</code> | <code>{"connectedIface": "eth0"} </code> |

### 5.2.2 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..."  
}
```

## 5.3 Network Endpoints

### 5.3.1 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.

### 5.3.2 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

**5.3.3 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).

## 5.4 Firmware Endpoints

### 5.4.1 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                                                             |
|-------------------------------|---------|-------------------------------------------------------------------------|
| <code>version</code>          | string  | Currently running firmware version                                      |
| <code>activeSlot</code>       | string  | Active firmware slot ( <code>A</code> or <code>B</code> )               |
| <code>inactiveSlot</code>     | string  | Inactive firmware slot                                                  |
| <code>inactiveVersion</code>  | string  | Firmware version on the inactive slot                                   |
| <code>updateInProgress</code> | boolean | Whether an update is currently running                                  |
| <code>updateProgress</code>   | number  | Progress percentage (0-100), present during updates                     |
| <code>updateMessage</code>    | string  | Human-readable status message, present during updates                   |
| <code>updateMode</code>       | string  | Empty for single-slot updates, <code>full</code> for both-slots updates |
| <code>updatePhase</code>      | string  | Current phase during a full update (see below)                          |
| <code>lastError</code>        | string  | Error message from the last failed update, if any                       |

#### Full Update Phases

| PHASE                          | DESCRIPTION                                          |
|--------------------------------|------------------------------------------------------|
| <code>installing first</code>  | Installing firmware to the inactive slot             |
| <code>rebooting</code>         | First slot done, device is about to reboot           |
| <code>installing second</code> | After reboot, installing firmware to the second slot |

### 5.4.2 POST /api/firmware/update

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

| PARAMETER                         | DESCRIPTION                                                       |
|-----------------------------------|-------------------------------------------------------------------|
| Form field: <code>firmware</code> | The <code>.raucb</code> file (max 256 MiB)                        |
| Query: <code>mode</code>          | Empty (default) for single slot, <code>full</code> 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. The bundle is **not** staged on `/data` — only a small marker is kept across the reboot. After the reboot the status reports `updatePhase: "awaiting second upload"`, and the caller must re-upload the same bundle to `POST /api/firmware/update/resume` to finish the second slot. (URL-sourced full updates skip this — see below.)

Errors:

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

5.4.3 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.

| PARAMETER                | DESCRIPTION                                                       |
|--------------------------|-------------------------------------------------------------------|
| Query: <code>mode</code> | Empty (default) for single slot, <code>full</code> for both slots |

With `mode=full`, the device re-downloads the bundle from the URL after the reboot and finishes the second slot autonomously — no further client action required.

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

5.4.4 POST /api/firmware/update/resume

Finish the second slot of an interrupted **upload-sourced** full update by re-uploading the bundle. Uses `multipart/form-data` with a `firmware` field, like `/api/firmware/update`. Valid only while `updatePhase` is `awaiting second upload`; the bundle version must match the first slot.

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

**Errors:**

- `400` — Missing file, invalid form data, or non-`.raucb` file
- `409` — Not awaiting a second-slot upload, or bundle version mismatch

### 5.4.5 POST /api/firmware/update/cancel

---

Abandon an interrupted full update, clearing the pending second-slot marker. The device stays on the new firmware; the second slot keeps its previous version. Both slots remain bootable.

```
curl -X POST http://<device-ip>/api/firmware/update/cancel
```

## 5.5 SPDT Endpoints

### RENAMED FROM /API/SWITCH

These endpoints were previously served at `/api/switch`. On SPDT hardware that path is kept as a deprecated alias and still works, but new automation should use `/api/spdt`. On SP4T hardware `/api/switch` aliases `/api/sp4t` instead, carrying the 0-3 range.

### 5.5.1 GET /api/spdt

Returns the currently routed throw.

#### Response:

```
{
  "state": 0
}
```

| STATE | DESCRIPTION |
|-------|-------------|
| 0     | RF0         |
| 1     | RF1         |

### 5.5.2 POST /api/spdt

Sets the routed throw.

#### Request:

```
{
  "state": 1
}
```

#### Response:

```
{
  "state": 1
}
```

#### Errors:

- 400 — Invalid state value (must be 0 or 1)
- 500 — Failed to set SPDT state

#### Example:

```
# Route COM to RF1
curl -X POST -H "Content-Type: application/json" \
-d '{"state": 1}' \
http://<device-ip>/api/spdt

# Route COM to RF0
curl -X POST -H "Content-Type: application/json" \
-d '{"state": 0}' \
http://<device-ip>/api/spdt

# Query the current throw
curl http://<device-ip>/api/spdt
```