# SFP Tool User Manual


- SFP Tool User Manual
  - Purpose
  - Safety and Handling
  - System Overview
  - Before You Start
  - Powering On and Local Operation
  - Connecting the App
    - Bluetooth
    - USB
  - App Navigation
  - Status Page
  - Transceiver Page
  - EEPROM Page
  - Passwords Page
    - Stored
    - Change Password
    - Cracking
  - Scripts Page
  - Settings Page
    - Device Name
    - OLED
  - Special Settings
    - Hardware Revision
    - Display Type
  - Battery Indicator
  - Firmware Update
    - Normal Update from the App
    - Recovery Update from the Browser
  - Logs
  - Troubleshooting
    - The app cannot find the device
    - Display output looks wrong
    - No battery percentage is shown
    - Firmware update fails
    - EEPROM write did not behave as expected
  - Good Operating Practice


# SFP Tool User Manual

## Purpose

SFP Tool is a portable service device for reading, checking, and reprogramming SFP and QSFP transceivers. It can be operated directly on the hardware or remotely from the desktop and Android app over USB or Bluetooth Low Energy.

## Safety and Handling

- Use the device only with compatible SFP or QSFP modules.
- Insert and remove transceivers carefully to avoid connector damage.
- Do not interrupt power while writing EEPROM, changing passwords, or updating firmware.
- Verify the correct hardware revision and OLED type in special settings before diagnosing display or battery issues.
- Use password cracking and EEPROM rewriting only where you are authorized to do so.

## System Overview

The system has four major parts:

- The handheld hardware with ESP32-S3, display, buttons, battery support, and transceiver sockets.
- The firmware running on the ESP32-S3.
- The React and Tauri app for desktop and Android.
- The browser-based recovery updater for first flash or crash-loop recovery.

## Before You Start

You should have:

- A charged SFP Tool device.
- A supported 1.3 inch OLED installed and configured as `SSD1306` or `SH1106`.
- A compatible SFP or QSFP module if you plan to inspect or modify a transceiver.
- The desktop or Android app if you want remote control.
- A Chromium-based browser if you need to use the web recovery updater.

## Powering On and Local Operation

After power-up, the firmware initializes the display, transceiver interfaces, BLE services, battery monitoring, and optional USB HID transport. The on-device display is intended for direct field use without a permanently attached phone or PC.

Use the front buttons to navigate menus locally. The exact menu contents depend on

firmware version, but the app exposes the full operational feature set and is the preferred interface for advanced work.

---

# Connecting the App

## Bluetooth

1. Power on the device.
2. Open the app.
3. Wait for the BLE scan to list nearby tools.
4. Select the device from the scanner page.

The BLE device name can be changed in the app settings page. The default advertised prefix is `SFP-TOOL-`.

## USB

1. Connect the tool to the host using USB.
2. Open the app.
3. Select the USB entry from the scanner page.

USB is useful for faster transfers, especially for firmware updates and EEPROM operations.

---

# App Navigation

After connecting, the app exposes these pages from the main menu:

- `Status`
- `Transceiver`
- `EEPROM`
- `Passwords`
- `Scripts`
- `About`
- `Settings`
- `Disconnect`

Holding the `Settings` menu entry opens `Special Settings`.

---

# Status Page

The status page shows whether an SFP or QSFP module is currently detected.

Use this page to verify basic physical insertion before attempting reads or writes.

# Transceiver Page

The transceiver page displays module information and digital diagnostic monitoring data when available. Typical values include vendor-identifying data and measured optics or power telemetry exposed by the module.

Use this page for quick health checks before making EEPROM changes.

# EEPROM Page

The EEPROM section has two tabs:

- `Editor` for direct EEPROM access.
- `ESP Files` for files stored on the tool itself.

Use the editor when you need to inspect or modify EEPROM bytes. Confirm the target address range and module type before writing changes.

# Passwords Page

The passwords area has three tabs:

- `Stored`
- `Change Password`
- `Cracking`

## Stored

Use this tab to manage passwords stored on the device for later reuse.

## Change Password

To change a transceiver password:

1. Enter the current password as exactly 8 hexadecimal characters.
2. Enter the new password as exactly 8 hexadecimal characters.
3. Press `Change Password`.

The app rejects invalid lengths and prevents reusing the same value as the old password.

## Cracking

Use the cracking page only when you are permitted to recover a password. This operation can take time depending on the target module and transport.

# Scripts Page

Scripts provide repeatable EEPROM programming workflows. They are stored on the tool and can also be imported from or exported to files in desktop builds.

The built-in editor supports commands such as:

- `password 00112233`
- `name "VENDOR"`
- `model "MODEL-123"`
- `serial "SN000001"`
- `mfg_date "240101"`
- `bytes 0xA0 0x01 0x02`
- `checksum`
- `write`

Every programming script should end with `write`, otherwise validation will report an error.

# Settings Page

The settings page exposes normal operator settings.

### Device Name

Changing the device name updates both:

- The BLE advertised name.
- The USB HID product name.

The maximum name length is 24 characters. The device reboots to apply the new name.

### OLED

You can configure:

- Brightness from `0` to `255`.
- Idle timeout from `0` to `65535` seconds.

After the idle timeout expires, the display dims to 25 percent brightness. If brightness is set to `0`, the display can turn fully off until a button is pressed.

# Special Settings

Special settings are intended for service or build-time correction.

### Hardware Revision

- `1.0`: no battery gauge support.
- `1.1`: current default hardware with battery support.

### Display Type

- `SSD1306`
- `SH1106`

Set these values correctly if the display behaves incorrectly or the battery indicator is missing on supported hardware.

---

## Battery Indicator

When battery telemetry is available, the app bar shows:

- Battery percentage.
- Charging state.
- Active transport (`USB` or `BLE`).

If battery information is unavailable, verify the configured hardware revision.

---

## Firmware Update

The about page supports firmware upload to the device.

### Normal Update from the App

1. Open `About`.
2. Choose a firmware file or trigger automatic download if available.
3. Start the update.
4. Wait until the upload completes and the device reboots.

USB uses larger transfer chunks than BLE and is usually faster.

### Recovery Update from the Browser

Use the browser-based updater when:

- The device is new and has not been flashed yet.
- The firmware is broken.
- The device is stuck in a crash loop and the normal app path is unavailable.

Use a Chromium-based browser such as Chrome or Edge for this workflow.

---

## Logs

On the about page, tapping the build number repeatedly opens the logs viewer. This is

intended for diagnostics and service use.

---

# Troubleshooting

### The app cannot find the device

- Confirm the tool is powered on.
- Check whether the device appears over USB instead of BLE.
- Reconnect and reopen the scanner page.
- If you recently renamed the device, reconnect after the reboot.

### Display output looks wrong

- Check `Special Settings` and confirm the OLED type is correct.
- Confirm the installed display matches `SSD1306` or `SH1106`.

### No battery percentage is shown

- Verify the hardware revision is set correctly.
- Revision `1.0` does not provide battery gauge support.

### Firmware update fails

- Retry over USB if possible.
- Keep the device powered during the entire update.
- If the device no longer starts correctly, use the browser-based recovery updater.

### EEPROM write did not behave as expected

- Re-read the EEPROM before writing again.
- Confirm the correct memory window and module type.
- Recalculate checksums where required.

---

# Good Operating Practice

- Read first, then modify.
- Save EEPROM-related files before making destructive changes.
- Validate scripts before running them on live modules.
- Prefer USB for large transfers and service work.
- Keep firmware, app, and published documentation aligned.