PiTrezor — Quick Start & User Instructions

Buildroot image for a Trezor■like wallet on Raspberry Pi

What is PiTrezor?

PiTrezor turns a Raspberry Pi (Zero/3/4) with a touchscreen into a Trezor■style wallet. You build a minimal, locked■down image with Buildroot, flash it to an SD card, and boot the Pi. On first boot, you'll see a splash screen, perform touchscreen calibration, and then the wallet UI starts.

What You Need

- Ubuntu 22.04/24.04 (or Debian■based) build machine; x86_64 recommended for fastest cross■compile.
- Raspberry Pi Zero, 3/3B+, or 4 with power supply.
- Supported 3.5" (or similar) touchscreen (e.g., Waveshare).
- MicroSD card (8 GB or larger) and a card reader.
- USB cable to connect the Pi to your computer running Trezor Suite.

Step 1 — Install Build Prerequisites (on your PC)

```
sudo apt update
sudo apt install -y build-essential git curl unzip python3 rsync bc bison flex \
  libssl-dev libncurses5-dev libncursesw5-dev gawk wget
```

Step 2 — Get the Code (with submodules)

Clone the repository and include all submodules the first time:

```
git clone --recurse-submodules git@github.com:Octavepi/buildroot-pitrezor.git
cd buildroot-pitrezor
```

If you forgot the flag, bring in submodules later with:

```
git submodule update --init --recursive
```

Step 3 — Build the Image

The build script auto■detects paths (absolute BR2_EXTERNAL) and auto■generates br-ext/Config.in. You pass your Pi model, the LCD overlay name, and rotation (0/90/180/270).

```
chmod +x build.sh
./build.sh <pi-model> <overlay-name> <rotation>

# Examples
./build.sh rpi4 waveshare35a 180
./build.sh rpi3 rpi-display 0
```

Overlay names come from the LCD-show project (e.g., waveshare35a, waveshare35b, rpi-display). Pick the entry that matches your display model. If you're unsure, check your display's documentation.

At the start of the build, you'll see lines confirming the resolved paths, e.g.:

- ■ `Using BR2_EXTERNAL=/full/path/to/buildroot-pitrezor/br-ext`
- ■ `Buildroot directory=/full/path/to/buildroot-pitrezor/third_party/buildroot`

When the build completes, the SD card image appears at: output/images/sdcard.img

Step 4 — Flash the Image to an SD Card (No command line needed)
Use one of these GUI tools:

- Balena Etcher — etcher.balena.io
- Raspberry Pi Imager — www.raspberrypi.com/software/

Steps in either tool:

- Open the program.
- Select the image file: output/images/sdcard.img.
- Insert your SD card and choose it as the target.
- Click Flash and wait until it completes, then safely eject the card.

Step 5 — First Boot & Calibration

- Insert the SD card into the Pi, attach your LCD, and power on.
- A branded splash screen appears, followed by a one■time touchscreen calibration.
- After calibration, you'll see "Calibration complete. Starting wallet…" and the wallet UI starts.
- By design there is no SSH/console access; the device behaves like a dedicated hardware wallet.

Step 6 — Connect to Trezor Suite

- Use a USB cable to connect the Pi to your computer.
- Open Trezor Suite on the computer; the PiTrezor should appear as a USB HID device.
- Follow on■screen prompts in Suite to pair and use the wallet.

Linux USB Access Note (udev rules)
On Linux, Trezor Suite normally installs the required udev rules automatically. If Suite does not detect PiTrezor when connected, you may need to add them manually:

```
wget https://raw.githubusercontent.com/trezor/trezor-common/master/udev/51-trezor.rules
sudo mv 51-trezor.rules /etc/udev/rules.d/
sudo udevadm control --reload-rules
sudo udevadm trigger
```

Then unplug/replug your PiTrezor and restart Trezor Suite.

Updating or Changing Screens

- To change rotation or switch displays, rebuild with a different <overlay-name> and/or rotation and re■flash.
- Re■flashing re■triggers first■boot calibration.

Firmware Updates
Unlike a real Trezor hardware wallet, PiTrezor cannot be updated directly from Trezor Suite. Suite's firmware update mechanism expects a bootloader and flash memory layout that the Pi does not have. Instead, when Trezor firmware is updated upstream, you (or maintainers) must rebuild PiTrezor against the new firmware.

- Maintainers may publish new ready-to-flash images on GitHub Releases.
- Users update by reflashing the new image with Etcher or Raspberry Pi Imager.
- Your wallet seed remains valid; just restore it after flashing.

This approach keeps PiTrezor secure and avoids opening network or write-access paths inside the device.

If Something Doesn’t Look Right (Quick Checks)

- Display stays blank: try the alternate overlay for your model (e.g., waveshare35b vs waveshare35a) and rebuild.
- Suite doesn’t detect device: unplug/replug USB; try a different cable/port; ensure your OS allows HID access.
- Build errors about missing submodules: run git submodule update --init --recursive and rebuild.

Security Notes

- Treat PiTrezor like a real wallet: protect your seed and physical device.
- No network/SSH access is exposed by default; interaction is via USB HID to Trezor Suite.
- Back up your seed before rebuilding or re■flashing.

Acknowledgment

This guide and the PiTrezor project were created with the assistance of ChatGPT. Bring your ideas to life with https://chat.openai.com