

Complete Guide: Flashing Raspberry Pi OS from Mac and Windows

Required Materials

Hardware

- Raspberry Pi 4B (4GB or 8GB RAM recommended)
- MicroSD card (32GB or larger, Class 10 or better)
- MicroSD card reader/adaptor
- Ethernet cable (for initial setup)
- Power supply (USB-C, 5V 3A for Pi 4)

Software

- Raspberry Pi Imager (official tool)
 - Computer running macOS or Windows
-

Method 1: Using Raspberry Pi Imager (Recommended)

macOS Installation

Step 1: Download Raspberry Pi Imager

```
bash
```

```
# Option A: Download from website
```

```
# Visit: https://www.raspberrypi.com/software/
```

```
# Download "Raspberry Pi Imager for macOS"
```

```
# Option B: Install via Homebrew
```

```
brew install --cask raspberry-pi-imager
```

Step 2: Prepare SD Card

1. Insert microSD card into Mac's card reader
2. Open **Disk Utility** (Applications → Utilities)
3. Select your SD card from the sidebar
4. Click **Erase**
5. Format: **MS-DOS (FAT)**
6. Name: **PI_SETUP**

7. Click **Erase**

Step 3: Flash OS with Imager

1. Launch **Raspberry Pi Imager**
2. Click **CHOOSE OS**
3. Select **Raspberry Pi OS (64-bit) → Raspberry Pi OS Lite (64-bit)**
4. Click **CHOOSE STORAGE**
5. Select your SD card
6. **IMPORTANT:** Click the **gear icon (⚙️)** for advanced options

Step 4: Configure Advanced Settings

- ☒ Enable SSH
 - o Use password authentication
 - Username: `pi`
 - Password: `[create strong password]`
- ☒ Configure WiFi (if using wireless)
 - SSID: `[your network name]`
 - Password: `[your WiFi password]`
 - WiFi country: `US` (or your country)
- ☒ Set locale settings
 - Time zone: `[your timezone]`
 - Keyboard layout: `[your layout]`
- ☒ Skip first-run wizard

Step 5: Write Image

1. Click **SAVE** to save advanced settings
2. Click **WRITE**
3. Enter your Mac password when prompted
4. Wait for writing and verification (10-15 minutes)

Step 6: Post-Flash Setup (macOS)

```
bash
```

```
# After successful flash, SD card remounts as "bootfs"
```

```
cd /Volumes/bootfs
```

```
# Verify SSH is enabled (should see ssh file)
```

```
ls -la ssh
```

```
# If WiFi wasn't configured in imager, create wpa_supplicant.conf
```

```
sudo nano wpa_supplicant.conf
```

WiFi Configuration File (if needed):

```
country=US
```

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
```

```
update_config=1
```

```
network={
```

```
    ssid="YourNetworkName"
```

```
    psk="YourWiFiPassword"
```

```
    key_mgmt=WPA-PSK
```

```
}
```

Windows Installation

Step 1: Download Raspberry Pi Imager

1. Visit <https://www.raspberrypi.com/software/>
2. Download **Raspberry Pi Imager for Windows**
3. Run the installer (`imager_x.x.x.exe`)
4. Follow installation wizard
5. Launch **Raspberry Pi Imager**

Step 2: Prepare SD Card

1. Insert microSD card into Windows PC
2. Open **File Explorer**
3. Right-click on SD card drive
4. Select **Format...**
5. File system: **FAT32**
6. Allocation unit size: **Default**
7. Volume label: **PI_SETUP**

8. Check **Quick Format**

9. Click **Start**

Step 3: Flash OS with Imager

1. Open **Raspberry Pi Imager**

2. Click **CHOOSE OS**

3. Select **Raspberry Pi OS (64-bit) → Raspberry Pi OS Lite (64-bit)**

4. Click **CHOOSE STORAGE**

5. Select your SD card (verify size matches)

6. Click the **gear icon (⚙️)** for advanced options

Step 4: Configure Advanced Settings

☒ Enable SSH

- o Use password authentication

Username: pi

Password: [create strong password - write it down!]

☒ Configure WiFi

SSID: [your network name]

Password: [your WiFi password]

WiFi country: US

☒ Set locale settings

Time zone: America/New_York (or your timezone)

Keyboard layout: us

☒ Skip first-run wizard

[Check this box]

☒ Eject media when finished

[Check this box]

Step 5: Write Image

1. Click **SAVE**

2. Click **WRITE**

3. Click **YES** when Windows asks for permission

4. Wait for writing process (10-20 minutes)

5. Click **CONTINUE** when finished

Step 6: Post-Flash Verification (Windows)

```
cmd

# Open Command Prompt
# Navigate to SD card (usually D: or E:)
D:
dir

# Should see files like:
# bootcode.bin, config.txt, ssh, wpa_supplicant.conf
```

Method 2: Manual Flashing (Alternative)


macOS Manual Method

Step 1: Download OS Image

```
bash

# Download Raspberry Pi OS Lite
curl -L -o raspios-lite.zip https://downloads.raspberrypi.org/raspios_lite_arm64/images/raspios

# Extract the image
unxz 2024-07-04-raspios-bookworm-arm64-lite.img.xz
```



Step 2: Identify SD Card

```
bash

# List all disks
diskutil list

# Look for your SD card (usually /dev/disk2 or /dev/disk3)
# Example output:
# /dev/disk2 (external, physical):
#   0: FDisk_partition_scheme      *32.0 GB   disk2
#   1: Windows_FAT_32              32.0 GB   disk2s1
```

Step 3: Flash Image

```
bash
```

```
# Unmount the SD card (replace disk2 with your disk)
```

```
diskutil unmountDisk /dev/disk2
```

```
# Flash the image (be very careful with disk number!)
```

```
sudo dd bs=1m if=2024-07-04-raspbian-bookworm-arm64-lite.img of=/dev/rdisk2
```

```
# This takes 10-15 minutes, no progress indicator
```

```
# Press Ctrl+T to see progress
```

Step 4: Enable SSH and WiFi

```
bash
```

```
# After dd completes, remount
```

```
diskutil mountDisk /dev/disk2
```

```
# Enable SSH
```

```
touch /Volumes/bootfs/ssh
```

```
# Configure WiFi
```

```
nano /Volumes/bootfs/wpa_supplicant.conf
```

Windows Manual Method

Step 1: Download Tools

1. Download **Win32 Disk Imager** or **Rufus**
2. Download **Raspberry Pi OS Lite** from official website
3. Extract the `.img` file using **7-Zip**

Step 2: Flash with Win32 Disk Imager

1. Launch **Win32 Disk Imager** as Administrator
2. **Image File:** Browse to your `.img` file
3. **Device:** Select SD card drive letter
4. Click **Write**
5. Confirm overwrite warning
6. Wait for completion

Step 3: Enable SSH (Windows)

cmd

```
# After flashing, SD card appears as boot drive
# Open Command Prompt as Administrator
cd /d E:\
```

```
# Create empty ssh file (no extension)
echo. > ssh
```

```
# Create WiFi config
notepad wpa_supplicant.conf
```

First Boot and SSH Connection

Find Your Pi's IP Address

Method 1: Router Admin Page

1. Open browser to `http://192.168.1.1` (or your router IP)
2. Login with admin credentials
3. Look for **Connected Devices** or **DHCP Client List**
4. Find device named "raspberrypi" or with MAC starting with "DC:A6:32"

Method 2: Network Scan (macOS/Linux)

```
bash

# Install nmap if needed
brew install nmap # macOS
# sudo apt install nmap # Linux

# Scan local network
nmap -sn 192.168.1.0/24 | grep -B2 -A2 "Raspberry"

# Alternative: ping broadcast
ping raspberrypi.local
```

Method 3: Network Scan (Windows)

```
cmd
```

```
# Use built-in arp command  
arp -a | findstr "dc-a6-32"
```

```
# Or use Advanced IP Scanner (free download)  
# Scan range: 192.168.1.1 - 192.168.1.254
```

Initial SSH Connection

Connect and Update

```
bash  
  
# SSH to your Pi (replace with actual IP)  
ssh pi@192.168.1.100  
  
# First login will show fingerprint warning - type 'yes'  
# Enter the password you set during flashing  
  
# Update system immediately  
sudo apt update && sudo apt upgrade -y  
  
# Configure Pi settings  
sudo raspi-config
```

Essential raspi-config Settings

1. Change User Password → [Set new secure password]
2. Network Options → N2 Wireless LAN → [Configure if needed]
3. Interfacing Options → P2 SSH → Enable
4. Advanced Options → A1 Expand Filesystem
5. Advanced Options → A3 Memory Split → 16 (headless setup)
6. Localisation → I1 Change Locale → [Your locale]
7. Localisation → I2 Change Timezone → [Your timezone]

Verification and Testing

System Information


```
bash
```

```
# Check OS version
```

```
cat /etc/os-release
```

```
# Check hardware info
```

```
vcgencmd get_mem arm
```

```
vcgencmd measure_temp
```

```
vcgencmd measure_volts
```

```
# Check network connectivity
```

```
ping -c 4 google.com
```

Network Configuration

```
bash
```

```
# Check IP configuration
```

```
ip addr show
```

```
# Check WiFi status
```

```
iwconfig wlan0
```

```
# Check SSH status
```

```
sudo systemctl status ssh
```

Install Development Tools

```
bash
```

```
# Update package list
```

```
sudo apt update
```

```
# Install essential development packages
```

```
sudo apt install -y git python3-pip python3-venv sqlite3 tcpdump dnsutils
```

```
# Install network monitoring tools
```

```
sudo apt install -y nmap iftop htop
```

```
# Test Python
```

```
python3 --version
```

```
pip3 --version
```

Troubleshooting Common Issues

SSH Connection Problems

Issue: "Connection refused"

```
bash

# Solution 1: Verify SSH is enabled
# Re-flash SD card and ensure ssh file is created

# Solution 2: Check IP address
nmap -sn 192.168.1.0/24

# Solution 3: Try ethernet connection
# Connect Pi directly to router with ethernet cable
```

Issue: "Permission denied"

```
bash

# Solution: Verify username and password
ssh pi@192.168.1.100
# Default user is 'pi', use password set during flashing
```

WiFi Connection Problems

Issue: Pi not connecting to WiFi

```
bash

# Check wpa_supplicant.conf syntax
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf

# Reconfigure WiFi
sudo raspi-config
# Network Options → Wireless LAN

# Restart networking
sudo systemctl restart dhcpcd
```

SD Card Issues

Issue: "No space left on device"

```
bash
```

```
# Expand filesystem
```

```
sudo raspi-config
```

```
# Advanced Options → Expand Filesystem
```

```
sudo reboot
```

```
# Check available space
```

```
df -h
```

Issue: SD card corruption

```
bash
```

```
# Prevention: Always shutdown properly
```

```
sudo shutdown -h now
```

```
# Recovery: Re-flash SD card if corruption occurs
```

Security Hardening (Post-Setup)

SSH Key Authentication

```
bash
```

```
# On your computer, generate SSH key
```

```
ssh-keygen -t ed25519 -C "pi-monitor-key"
```

```
# Copy public key to Pi
```

```
ssh-copy-id pi@192.168.1.100
```

```
# Test key authentication
```

```
ssh pi@192.168.1.100
```

Disable Password Authentication

```
bash
```

```
# Edit SSH configuration
```

```
sudo nano /etc/ssh/sshd_config
```

```
# Change these lines:
```

```
# PasswordAuthentication no
```

```
# PubkeyAuthentication yes
```

```
# Restart SSH service
```

```
sudo systemctl restart ssh
```

Basic Firewall Setup

```
bash
```

```
# Install and configure UFW
```

```
sudo apt install ufw
```

```
sudo ufw default deny incoming
```

```
sudo ufw default allow outgoing
```

```
sudo ufw allow ssh
```

```
sudo ufw allow 53 # DNS
```

```
sudo ufw enable
```

Next Steps

After successful flashing and SSH setup:

1. **Set static IP address** for your Pi
2. **Install project dependencies** (Python packages, databases)
3. **Configure DNS server software** (dnsmasq or custom)
4. **Set up development environment** (Git, code editor)
5. **Begin network monitoring project** development

Your Raspberry Pi is now ready for headless development via SSH!