# Banana Pi Cheat Sheet

A comprehensive cheat sheet covering Banana Pi single-board computers, including setup, configuration, common commands, and troubleshooting tips.

## Getting Started with Banana Pi

### Initial Setup

**1. Hardware Requirements:**

- Banana Pi board
- MicroSD card (minimum 8GB, Class 10 recommended)
- MicroSD card reader/writer
- Power adapter (5V/2A recommended)
- HDMI cable and monitor
- USB keyboard and mouse
- Ethernet cable (optional, for network connection)

**2. Download Operating System Image:**

- Choose an OS image from the Banana Pi website or a trusted source (e.g., Armbian, Debian, Ubuntu).
- Download the `.img` file.

**3. Flash the OS Image to the MicroSD Card:**

- Use a tool like Balena Etcher, Rufus, or `dd` command-line tool.
- Select the downloaded `.img` file and the MicroSD card as the target.
- Flash the image.

**4. Booting the Banana Pi:**

- Insert the MicroSD card into the Banana Pi.
- Connect the HDMI cable to the monitor.
- Connect the USB keyboard and mouse.
- Connect the Ethernet cable (if using).
- Plug in the power adapter to boot the device.

**5. Initial Configuration:**

- Log in using the default username and password (usually `root` and `bananapi` or `1234`).
- Change the default password immediately.
- Configure network settings (if not using DHCP).
- Update the system using `apt update && apt upgrade`.

### Basic Commands

| | |
|---|---|
| `sudo apt update` | Update the package list. |
| `sudo apt upgrade` | Upgrade installed packages. |
| `sudo apt install <package_name>` | Install a new package. |
| `sudo apt remove <package_name>` | Remove a package. |
| `sudo apt autoremove` | Remove automatically all unused packages. |
| `ifconfig` or `ip addr` | Display network interfaces and IP addresses. |

## Networking and SSH

### Configuring Network Interfaces

Edit the `/etc/network/interfaces` file (or `/etc/dhcpcd.conf` for DHCP configuration) to configure static IP addresses, gateway, and DNS servers.

**Example:**

```
auto eth0
iface eth0 inet static
address 192.168.1.100
netmask 255.255.255.0
gateway 192.168.1.1
dns-nameservers 8.8.8.8 8.8.4.4
```

Restart the networking service:

```
sudo systemctl restart networking
```

Alternatively, use `netplan` for network configuration (on systems that use it):

Edit `/etc/netplan/01-netcfg.yaml` and apply the changes:

```
sudo netplan apply
```

### Enabling SSH

SSH (Secure Shell) allows remote access to the Banana Pi.

- Install the SSH server:

```
sudo apt install openssh-server
```

- Enable and start the SSH service:

```
sudo systemctl enable ssh
sudo systemctl start ssh
```

Access the Banana Pi from another computer using an SSH client (e.g., PuTTY, Terminal):

```
ssh username@<banana_pi_ip_address>
```

To disable password authentication and use SSH keys (recommended for security):

- Generate an SSH key pair on the client machine.
- Copy the public key to the Banana Pi using `ssh-copy-id` or manually add it to `~/.ssh/authorized_keys`.
- Disable password authentication in `/etc/ssh/sshd_config` by setting `PasswordAuthentication no` and restart the SSH service.

### Firewall Configuration (UFW)

UFW (Uncomplicated Firewall) is an easy-to-use firewall management tool.

- Install UFW:

```
sudo apt install ufw
```

- Enable UFW:

```
sudo ufw enable
```

Allow SSH connections:
`bash sudo ufw allow ssh`
Allow specific port:
`bash sudo ufw allow 80`

Check UFW status:
`bash sudo ufw status`

## GPIO and Hardware

### Accessing GPIO Pins

Accessing GPIO pins requires proper libraries and permissions.

- **WiringPi:** A popular library for accessing GPIO pins (may not be available on all Banana Pi models).
- **libgpiod:** A modern library for GPIO access using character devices.

**Using libgpiod:**

- Install libgpiod:

```
sudo apt install libgpiod-dev gpiod
```

- Identify the chip and pin number using `gpioinfo`:

```
gpioinfo
```

- Set a GPIO pin as output:

```
gpioset <chip> <pin>=1  # Set high
gpioset <chip> <pin>=0  # Set low
```

- Read the state of a GPIO pin:

```
gpioget <chip> <pin>
```

### Interacting with Hardware

**I2C**

- Install i2c-tools:

```
sudo apt install i2c-tools
```

- Detect I2C devices:

```
sudo i2cdetect -y 1
```

**SPI**

- Enable SPI in `/boot/config.txt` (if necessary).
- Use libraries like spidev (Python) or similar tools to interact with SPI devices.

**Serial (UART)**

- Serial communication is often available on specific GPIO pins.
- Use tools like `minicom` or libraries to communicate over serial.

### Example Python Script (libgpiod)

```
import gpiod
import time

LED_PIN = 18  # Replace with the actual GPIO pin number
CHIP = 'gpiochip0'  # Replace with the correct chip name

# Get the GPIO chip and line
chip = gpiod.Chip(CHIP)
led_line = chip.get_line(LED_PIN)

# Configure the pin as output
led_line.request(consumer='led-blink', type=gpiod.LINE_REQ_DIR_OUT, default_val=0)

try:
    while True:
        led_line.set_value(1)  # Turn on the LED
        time.sleep(1)
        led_line.set_value(0)  # Turn off the LED
        time.sleep(1)
except KeyboardInterrupt:
    led_line.release()
```

## Troubleshooting and Advanced Configuration

### Common Issues and Solutions

**1. Banana Pi Not Booting:**

- Check the MicroSD card for corruption.
- Ensure the OS image is flashed correctly.
- Verify the power supply is adequate (5V/2A recommended).
- Try a different MicroSD card.

**2. No Network Connection:**

- Check the Ethernet cable and router.
- Verify the network configuration (IP address, gateway, DNS).
- Ensure the network interface is enabled.

**3. SSH Connection Refused:**

- Ensure the SSH server is installed and running.
- Check the firewall settings.
- Verify the correct IP address is being used.

**4. GPIO Issues:**

- Verify the correct GPIO pin numbers are being used.
- Ensure the proper libraries are installed and configured.
- Check for permission issues.

### Advanced Configuration

| | |
|---|---|
| Overclocking | • Edit `/boot/config.txt` (if available) to adjust CPU frequency and voltage.<br>• Be cautious, as overclocking can lead to instability and overheating. |
| Kernel Updates | • Update the kernel using `rpi-update` (if available) or by manually building a new kernel.<br>• Ensure compatibility with the hardware and OS. |
| Bootloader Configuration | • The bootloader (e.g., U-Boot) can be configured to customize the boot process.<br>• Modify the bootloader configuration files to change boot parameters and device tree settings. |
| Device Tree Overlays | • Device tree overlays allow customization of the hardware configuration without modifying the base device tree.<br>• Use overlays to enable or disable specific hardware features. |

### Monitoring System Resources

- Use `htop` or `top` to monitor CPU usage, memory usage, and running processes.

```
sudo apt install htop
htop
```

- Use `df -h` to check disk space usage.

```
df -h
```

- Use `vcgencmd measure_temp` (if available) to check the CPU temperature.

```
vcgencmd measure_temp
```