■ Network IP Address Guide - Ghettoblaster

Last Updated: 2025-01-12
Status: ■ Complete Guide

■ Table of Contents

■ Device Identification

Ghettoblaster (Raspberry Pi 5)

- **Hostname:** `ghettoblaster` / `ghettoblaster.local`
- **Alternative:** `moodepi5.local`
- **User:** `andre` (SSH: `pi` or `andre`)
- **SSH Password:** `0815`
- **Device Type:** Raspberry Pi 5 with moOde Audio

Mac (Development Machine)

- **Hostname:** Varies
- **Purpose:** Development, SD card configuration, SSH access

---

■ IP Address Reference

■■ *CRITICAL: NEVER USE .101*

192.168.1.101 = **FOREIGN ROUTER** (NOT the Pi!)
- ■ **NEVER** use this IP for SSH or connections
- ■ This is a router/gateway, NOT the Ghettoblaster
- ■ Always verify IP before connecting

■ *Valid Ghettoblaster IP Addresses*

Scenario 1: Direct Ethernet (Mac ↔ Pi)

- **IP:** `192.168.10.2`

- **Gateway:** `192.168.10.1` (Mac)
- **Subnet:** `192.168.10.0/24`
- **When:** USB-C Ethernet connection, static configuration
- **SSH:** `ssh andre@192.168.10.2` or `ssh pi@192.168.10.2`

Scenario 2: WiFi - "nam yang 2" Network

- **IP:** `192.168.1.159` (DHCP assigned, may vary)
- **Network:** `192.168.1.0/24`
- **SSID:** `NAM YANG 2`
- **Password:** `1163855108`
- **Auto-Connect:** ■ Enabled
- **SSH:** `ssh andre@192.168.1.159` or `ssh andre@ghettoblaster.local`

Scenario 3: WiFi - "The Wing Hotel"

- **IP:** DHCP assigned
- **SSID:** `The Wing Hotel`
- **Password:** `thewing2019`
- **Auto-Connect:** ■ Enabled (priority 50)

Scenario 4: WiFi - Other Networks

- **IP:** DHCP assigned (varies by network)
- **Method:** DHCP
- **Auto-Connect:** Depends on configuration

Scenario 5: Ethernet via Router

- **IP:** `192.168.1.100` (if configured)
- **Network:** `192.168.1.0/24`
- **Method:** DHCP or Static

■ Network Scenarios

Scenario A: Direct Ethernet Connection (Development)

```
Mac (192.168.10.1) ←→ USB-C Ethernet ←→ Pi (192.168.10.2)
```

Use Case: SD card configuration, development, direct access
SSH: `ssh andre@192.168.10.2`
Status: Static IP, always available when connected

Scenario B: WiFi - "nam yang 2" (Current)

```
Router (192.168.1.1) ←→ WiFi ←→ Pi (192.168.1.159)
```

Use Case: Normal operation, internet access
SSH: `ssh andre@192.168.1.159` or `ssh andre@ghettoblaster.local`
Status: DHCP, auto-connect enabled

Scenario C: WiFi - Hotel Network

```
Hotel Router ←→ WiFi ←→ Pi (DHCP)
```

Use Case: Travel, hotel WiFi
SSH: Use hostname `ghettoblaster.local` or find IP via `nmcli`

Status: Auto-connect enabled (priority 50)

Scenario D: Multiple Interfaces Active

```
Ethernet (192.168.10.2) + WiFi (192.168.1.159)
```

Use Case: Both connections active
SSH: Either IP works, WiFi preferred for internet
Status: Both active, routing determines which is used

■ Connection Methods

Method 1: SSH via IP Address

```
# Direct Ethernet
ssh andre@192.168.10.2
# Password: 0815

# WiFi - nam yang 2
ssh andre@192.168.1.159
# Password: 0815

# Alternative user
ssh pi@192.168.10.2
```

Method 2: SSH via Hostname (mDNS)

```
# Preferred method (works if mDNS available)
ssh andre@ghettoblaster.local
ssh andre@moodepi5.local
```

Method 3: Find IP Address

```
# From Mac, scan network
nmap -sn 192.168.1.0/24 | grep -B 2 "ghettoblaster"

# From Pi itself
ip addr show
nmcli device status
hostname -I
```

Method 4: SSH with Password (non-interactive)

```
# Using sshpass (if installed)
sshpass -p '0815' ssh andre@192.168.1.159

# Or use SSH keys (recommended for automation)
```

■ Testing Procedures

Test 1: Verify Current IP Address

```
# On Pi
ssh andre@192.168.1.159 'hostname -I && ip addr show | grep "inet "'
```

Expected Output:

- WiFi IP: `192.168.1.159` (or similar)
- Ethernet IP: `192.168.10.2` (if connected)

Test 2: Verify Network Connectivity

```
# Ping test
ping -c 3 192.168.1.159
ping -c 3 ghettoblaster.local

# SSH connectivity
ssh andre@192.168.1.159 'echo "Connection successful"'
```

Test 3: Verify WiFi Connection

```
# Check WiFi status
ssh andre@192.168.1.159 'nmcli device status'
ssh andre@192.168.1.159 'nmcli connection show --active'
```

Expected:
- `wlan0` connected to `nam-yang-2`
- Auto-connect enabled

Test 4: Verify All Network Interfaces

```
# List all interfaces and IPs
ssh andre@192.168.1.159 'ip -4 addr show | grep -E "inet |^[0-9]"'
```

Test 5: Verify DNS Resolution

```
# Test hostname resolution
ping -c 1 ghettoblaster.local
nslookup ghettoblaster.local
```

■ Troubleshooting

Problem: Cannot Connect via IP

Solution:
1. Verify IP is correct (NOT .101!)
2. Check if device is on same network
3. Try hostname instead: `ghettoblaster.local`
4. Check firewall: `sudo ufw status`

Problem: WiFi Not Connecting

Solution:

```
# Check WiFi status
nmcli device status
nmcli device wifi list

# Restart NetworkManager
sudo systemctl restart NetworkManager

# Reconnect manually
sudo nmcli device wifi connect "NAM YANG 2" password "1163855108"
```

Problem: Multiple IPs Confusion

Solution:

```
# List all IPs
hostname -I

# Check which interface is active
ip route show default

# Check connection priority
nmcli connection show | grep priority
```

Problem: SSH Connection Refused

Solution:

```
# Check SSH service
sudo systemctl status ssh

# Check if SSH is enabled
sudo systemctl enable ssh
sudo systemctl start ssh

# Check SSH port
sudo netstat -tlnp | grep :22
```

Problem: Wrong IP Address

Solution:

1. Always verify IP before connecting
2. Use `hostname -I` on Pi to get current IP
3. Use `nmap` to scan network
4. Use hostname (`ghettoblaster.local`) instead

■ Quick Reference Commands

Find Current IP

```
# On Pi
hostname -I
ip addr show | grep "inet "

# From Mac
ping -c 1 ghettoblaster.local
nmap -sn 192.168.1.0/24
```

Connect via SSH

```
# Preferred (hostname)
ssh andre@ghettoblaster.local

# Direct IP (if known)
ssh andre@192.168.1.159
ssh andre@192.168.10.2
```

Network Status

```
# All interfaces
nmcli device status

# Active connections
nmcli connection show --active

# WiFi networks
nmcli device wifi list
```

Configure WiFi

```
# Connect to network
sudo nmcli device wifi connect "SSID" password "PASSWORD"

# Create persistent connection
sudo nmcli connection add type wifi con-name "NAME" ifname wlan0 ssid "SSID" autoconnect yes wifi-se
```

■ Verification Checklist

Before considering network setup complete:

- [ ] Can connect via SSH using hostname (`ghettoblaster.local`)
- [ ] Can connect via SSH using IP address
- [ ] WiFi auto-connects on boot
- [ ] IP address is stable (or DHCP working)
- [ ] Internet connectivity works (if needed)
- [ ] All network interfaces show correct status
- [ ] No .101 IP confusion
- [ ] Documentation updated with current IPs

■ Current Configuration Status

Last Verified: 2025-01-12

Active Connections:

- ■ WiFi: `nam-yang-2` (NAM YANG 2) - `192.168.1.159`
- ■ Ethernet: Available (192.168.10.2 when connected)
- ■ Auto-Connect: Enabled for WiFi

SSH Access:

- ■ User: `andre` (password: `0815`)
- ■ Hostname: `ghettoblaster.local`
- ■ IP: `192.168.1.159` (current WiFi)

Remember: Always verify IP addresses before connecting. Never use `.101` - that's a router, not the Pi!