Perfect Display Chain Configuration - Ghetto Blaster

Complete, Precise Configuration Reference

Date: 2025-01-11
System: Raspberry Pi 5 + Waveshare 7.9" HDMI Display (400x1280 native portrait → 1280x400 landscape)

■ Complete Display Chain

```
Hardware: Waveshare 7.9" HDMI Display
  ↓ Native: 400x1280 portrait
  ↓
Boot Configuration (/boot/firmware/cmdline.txt)
  ↓ video=HDMI-A-1:400x1280M@60,rotate=90
  ↓ fbcon=rotate:1
Kernel Framebuffer (FB0)
  ↓ rotate=90: Rotates framebuffer 90° clockwise
  ↓ Result: 1280x400 landscape framebuffer
Console (fbcon=rotate:1)
  ↓ rotate:1: Rotates console 90° clockwise
  ↓ Result: Landscape boot screen
DRM/KMS (Direct Rendering Manager)
  ↓ Exposes rotated framebuffer to userspace
  ↓ Result: 1280x400 landscape available to X11
X11 Server (Xorg)
  ↓ Started via systemd (localdisplay.service)
  ↓ Reads /etc/X11/xorg.conf.d/99-touch-calibration.conf
Touch Input Transformation
  ↓ TransformationMatrix "-1 0 1 0 -1 1 0 0 1"
  ↓ 180° rotation matrix (fixes both X and Y axis swap)
  ↓ Result: Touch coordinates match display orientation
Xrandr (X11 Display Configuration)
  ↓ start-chromium-clean.sh detects mode
  ↓ Sets mode to 400x1280, then rotates left
  ↓ Result: X11 display is 1280x400 landscape
Chromium Browser
  ↓ Launched with --window-size=1280,400
  ↓ --start-fullscreen --kiosk
  ↓ Result: Fullscreen landscape web UI
  ↓
User sees: Landscape moOde interface with correct touch
```

■ Perfect Configuration Checklist

1. Boot Configuration (`/boot/firmware/config.txt`)

Required Settings:

```
[pi5]
dtoverlay=vc4-kms-v3d-pi5,noaudio
hdmi_enable_4kp60=0

[all]
disable_overscan=1
hdmi_group=2
hdmi_mode=87
hdmi_timings=400 0 220 32 110 1280 0 10 10 10 0 0 0 60 0 59510000 0
```

```
hdmi_drive=2
hdmi_blanking=0
hdmi_force_hotplug=1
```

Why:
• `hdmi_group=2` + `hdmi_mode=87` + custom `hdmi_timings`: Defines custom 400x1280@60Hz mode for Pi 5
• `hdmi_timings`: Precise pixel timings matching Waveshare hardware specs
• `hdmi_force_hotplug=1`: Forces HDMI detection even if no EDID
• `noaudio` in dtoverlay: Disables HDMI audio (using I2S/AMP100 instead)

2. Kernel Command Line (`/boot/firmware/cmdline.txt`)

Required Parameters:
```
video=HDMI-A-1:400x1280M@60,rotate=90 fbcon=rotate:1
```

Why:
• `video=HDMI-A-1:400x1280M@60`: Sets framebuffer to hardware native resolution (400x1280 portrait)
• `HDMI-A-1`: HDMI port identifier for Pi 5
• `400x1280M@60`: Native portrait resolution at 60Hz
• `M`: Progressive mode
• `rotate=90`: Rotates framebuffer 90° clockwise at kernel level
• Converts 400x1280 portrait framebuffer → 1280x400 landscape framebuffer
• Applied BEFORE any userspace (X11) sees the display
• This is the foundation - everything else builds on this
• `fbcon=rotate:1`: Rotates console text 90° clockwise
• `1` = 90° clockwise rotation
• `2` = 180° rotation
• `3` = 270° clockwise rotation
• Ensures boot messages and console are readable in landscape

Critical: Both parameters are required:
• `rotate=90` rotates the framebuffer (graphics)
• `fbcon=rotate:1` rotates the console (text)
• Without both, boot screen shows portrait orientation

3. Touch Calibration (`/etc/X11/xorg.conf.d/99-touch-calibration.conf`)

Required Configuration:
```
Section "InputClass"
    Identifier "WaveShare Touchscreen"
    MatchProduct "WaveShare"
    Option "TransformationMatrix" "-1 0 1 0 -1 1 0 0 1"
EndSection
```

Why:
• **Transformation Matrix Format:** `"a b c d e f 0 0 1"`
• This is a 3x3 matrix: `[a b c] [d e f] [0 0 1]`
• Transforms touch coordinates: `[x' y' 1] = [x y 1] × Matrix`
• **Matrix `"-1 0 1 0 -1 1 0 0 1"` means:**
• `x' = -1*x + 0*y + 1 = -x + 1` (flips X axis, then translates)
• `y' = 0*x + -1*y + 1 = -y + 1` (flips Y axis, then translates)
• This is a 180° rotation + translation
• **Why this matrix:**
• Display is rotated 90° clockwise at kernel level
• X11 rotates it another 90° clockwise (left rotation = 90° CCW from hardware)
• Total: Hardware portrait → Kernel landscape → X11 landscape
• Touch hardware reports coordinates in hardware orientation
• After all rotations, touch X/Y axes are swapped (left/right AND top/bottom)
• Matrix `-1 0 1 0 -1 1` flips both axes, correcting the swap

Alternative Matrices (if needed):

• `"0 -1 1 1 0 0 0 0 1"`: 90° CCW rotation (if only one axis swapped)
• `"0 1 0 -1 0 1 0 0 1"`: 270° rotation (if different swap pattern)
• `"-1 0 1 0 1 0 0 0 1"`: Horizontal flip only (if only left/right swapped)

4. X11 Display Configuration (`/usr/local/bin/start-chromium-clean.sh`)

Required xrandr Sequence:

```
# Detect available mode and rotate accordingly
if xrandr | grep -q "400x1280"; then
    xrandr --output HDMI-2 --mode 400x1280 --rotate left 2>&1 || \
    xrandr --output HDMI-1 --mode 400x1280 --rotate left 2>&1
elif xrandr | grep -q "1280x400"; then
    xrandr --output HDMI-2 --mode 1280x400 --rotate normal 2>&1 || \
    xrandr --output HDMI-1 --mode 1280x400 --rotate normal 2>&1
fi
```

Why:
• **Sequence matters:** Must check what mode is available first
• **If 400x1280 available:** Set to hardware native, then rotate left
• `--rotate left` = 90° counterclockwise rotation
• Hardware 400x1280 portrait → X11 1280x400 landscape
• **If 1280x400 available:** Kernel already rotated it, just set to normal
• Kernel `rotate=90` already converted to landscape
• X11 just needs to use the rotated mode as-is
• **HDMI-1 vs HDMI-2:** Pi 5 can have either, script tries both

Critical: The xrandr rotation happens AFTER kernel rotation:
• Kernel: 400x1280 portrait → 1280x400 landscape (framebuffer)
• X11: Detects 1280x400 or sets 400x1280 and rotates → 1280x400 (X11 display)
• Result: Both layers show 1280x400 landscape

5. Chromium Launch Configuration

Required Flags:

```
chromium-browser \
    --kiosk \
    --no-sandbox \
    --user-data-dir=/tmp/chromium-data \
    --window-size=1280,400 \
    --window-position=0,0 \
    --start-fullscreen \
    --noerrdialogs \
    --disable-infobars \
    --disable-session-crashed-bubble \
    --disable-restore-session-state \
    --disable-web-security \
    --autoplay-policy=no-user-gesture-required \
    --check-for-update-interval=31536000 \
    --disable-features=TranslateUI \
    http://localhost
```

Critical Flags Explained:
• `--window-size=1280,400`: **MUST match landscape dimensions**
• Tells Chromium to create window at landscape size
• Without this, Chromium might use portrait dimensions
• `--start-fullscreen`: Ensures fullscreen mode
• `--kiosk`: Removes UI chrome, prevents exit
• `--no-sandbox`: Required on Pi 5 (security trade-off for compatibility)
• **NO `--disable-gpu`**: Let KMS + Pi GPU handle rendering
• GPU acceleration works correctly with KMS
• Disabling GPU can cause white screen issues

Window Management (xdotool):

```
xdotool windowsize $WINDOW 1280 400
xdotool windowmove $WINDOW 0 0
xdotool windowraise $WINDOW
```

• Ensures window is exactly 1280x400

• Positions at top-left corner
• Brings to front

6. moOde Database Configuration

Required Setting:

```
UPDATE cfg_system SET value='portrait' WHERE param='hdmi_scn_orient';
```

Why:
• moOde needs to know the **hardware** orientation (portrait)
• moOde's internal logic uses this for layout decisions
• Software rotation (xrandr) handles the visual rotation
• This is a moOde-specific setting, separate from actual display rotation

---

■ Display Chain Flow (Detailed)

Layer 1: Hardware

• **Waveshare 7.9" HDMI Display**
• Native resolution: 400x1280 portrait
• Connected to Pi 5 HDMI-A-1 port

Layer 2: Boot Configuration (config.txt)

• Defines custom HDMI mode: 400x1280@60Hz
• Sets pixel timings matching hardware
• Forces HDMI detection

Layer 3: Kernel Framebuffer (cmdline.txt)

• `video=HDMI-A-1:400x1280M@60`: Creates 400x1280 framebuffer
• `rotate=90`: Rotates framebuffer 90° clockwise
• **Result:** Kernel sees 1280x400 landscape framebuffer
• `fbcon=rotate:1`: Rotates console 90° clockwise
• **Result:** Boot screen text is landscape

Layer 4: DRM/KMS (Direct Rendering Manager)

• Kernel exposes rotated framebuffer via DRM
• X11 queries DRM for available modes
• **Result:** X11 sees 1280x400 mode available

Layer 5: X11 Server

• Starts via `localdisplay.service`
• Reads touch calibration from `/etc/X11/xorg.conf.d/99-touch-calibration.conf`
• Applies transformation matrix to touch input
• **Result:** Touch coordinates match display orientation

Layer 6: Xrandr (X11 Display Configuration)

• `start-chromium-clean.sh` runs xrandr commands
• Sets display mode and rotation
• **Result:** X11 display is 1280x400 landscape

Layer 7: Chromium Browser

• Launched with `--window-size=1280,400`
• Creates window at landscape size
• Loads `http://localhost` (moOde web UI)
• **Result:** Fullscreen landscape web interface

Layer 8: User Interaction

• Touch input → Transformation matrix → Correct coordinates
• Display shows landscape moOde interface
• **Result:** Perfect touch and display alignment

■ Why This Configuration Works

The Multi-Layer Rotation Strategy

Problem: Hardware is portrait (400x1280), but we want landscape (1280x400) display AND correct touch.

Solution: Rotate at multiple layers, each handling a different aspect:

1. **Kernel Level (`rotate=90`):**
• Rotates framebuffer early in boot process
• Ensures boot screen is landscape
• Foundation for all higher layers

2. **Console Level (`fbcon=rotate:1`):**
• Rotates text console separately
• Boot messages readable in landscape

3. **X11 Level (xrandr `--rotate left`):**
• Handles X11 display rotation
• Works with kernel rotation to ensure correct mode
• Adapts to what kernel provides

4. **Touch Level (TransformationMatrix):**
• Corrects touch coordinates after all rotations
• Hardware reports in native orientation
• Matrix transforms to match final display orientation

5. **Application Level (`--window-size=1280,400`):**
• Chromium creates window at correct size
• Prevents scaling issues
• Ensures fullscreen works correctly

Why Not Just One Rotation?

• **Kernel rotation alone:** Boot screen OK, but X11 might not detect rotated mode correctly
• **X11 rotation alone:** Boot screen portrait, X11 landscape (inconsistent)
• **Both kernel + X11:** Consistent landscape at all layers

• **Touch matrix:** Required because touch hardware doesn't rotate automatically

The "Forum Solution" Approach

This configuration follows the moOde forum solution pattern:
• Hardware configured as portrait (its native orientation)
• Software rotation converts to landscape
• Application told to render at landscape size
• All layers synchronized to same orientation

■ Configuration Files Summary

File	Purpose	Key Setting
`/boot/firmware/config.txt`	HDMI mode definition	`hdmi_timings=400 0 220 32 110 1280...`
`/boot/firmware/cmdline.txt`	Kernel framebuffer rotation	`video=HDMI-A-1:400x1280M@60,rotate=90 fbcon=rotate:1`
`/etc/X11/xorg.conf.d/99-touch-calibration.conf`	Touch coordinate transformation	`TransformationMatrix "-1 0 1 0 -1 1 0 0 1"`
`/usr/local/bin/start-chromium-clean.sh`	X11 rotation + Chromium launch	`xrandr --rotate left` + `--window-size=1280,400`
moOde database	Hardware orientation hint	`hdmi*scn*orient = 'portrait'`

■ Verification Commands

```
# Check kernel cmdline (boot rotation)
cat /proc/cmdline | grep -o 'video=[^ ]*'
cat /proc/cmdline | grep -o 'fbcon=rotate:[0-9]*'

# Check framebuffer size
cat /sys/class/graphics/fb0/virtual_size

# Check X11 display size
DISPLAY=:0 xdpyinfo | grep dimensions

# Check xrandr output and rotation
DISPLAY=:0 xrandr --query | grep "HDMI-1\|HDMI-2"

# Check touch calibration
cat /etc/X11/xorg.conf.d/99-touch-calibration.conf

# Check Chromium window size
DISPLAY=:0 xwininfo -root -tree | grep "moOde Player"

# Check Chromium process args
ps aux | grep chromium | grep window-size

# Test touch coordinates (requires xinput)
DISPLAY=:0 xinput list
DISPLAY=:0 xinput list-props <touch-device-id> | grep TransformationMatrix
```

Expected Outputs:
• Kernel cmdline: `video=HDMI-A-1:400x1280M@60,rotate=90 fbcon=rotate:1`
• Framebuffer: `1280,400` (after rotation)
• X11 dimensions: `1280x400`
• xrandr: `HDMI-1 connected primary 1280x400+0+0 left` (or `normal` if kernel already rotated)
• Touch matrix: `"-1 0 1 0 -1 1 0 0 1"`
• Chromium window: `1279x399+0+0` (1px difference is normal)

• Chromium args: `--window-size=1280,400`

■■ Common Issues and Solutions

Issue: Boot Screen Still Portrait

Symptom: Boot messages/text appear in portrait orientation
Cause: Missing `fbcon=rotate:1` in cmdline.txt
Fix: Add `fbcon=rotate:1` to `/boot/firmware/cmdline.txt`

Issue: Touch Left/Right Swapped

Symptom: Touching left side triggers right side action
Cause: Wrong transformation matrix
Fix: Use matrix `"-1 0 1 0 -1 1 0 0 1"` (180° rotation)

Issue: Touch Top/Bottom Also Swapped

Symptom: Both X and Y axes are swapped
Cause: Need 180° rotation matrix (both axes)
Fix: Matrix `"-1 0 1 0 -1 1 0 0 1"` fixes both

Issue: White Screen After Boot

Symptom: Display is white/illuminated but no content
Cause: Chromium started before web server ready, or GPU disabled
Fix:
• Ensure `start-chromium-clean.sh` waits for web server (already does)
• Remove `--disable-gpu` flag if present
• Check `/var/log/chromium-clean.log` for errors

Issue: Only 1/3 of Screen Visible

Symptom: Content appears scaled wrong, only portion visible
Cause: SCREEN_RES or window-size set to portrait dimensions
Fix: Ensure `--window-size=1280,400` (landscape, not `400,1280`)

Issue: Display Upside Down

Symptom: Content appears rotated 180°
Cause: Wrong xrandr rotation direction
Fix: Change `--rotate left` to `--rotate right` (or vice versa)

■■ Build System Integration

Custom Build Scripts

`imgbuild/moode-cfg/stage3*03-ghettoblaster-custom*02-display-cmdline.sh`:
• Automatically adds `video=...` and `fbcon=rotate:1` to cmdline.txt
• Runs during image build process
• Ensures boot screen rotation is configured

`moode-source/etc/X11/xorg.conf.d/99-touch-calibration.conf`:
• Included in custom build
• Touch calibration pre-configured
• No manual setup needed

`moode-source/usr/local/bin/start-chromium-clean.sh`:
• Included in custom build
• Handles X11 rotation and Chromium launch
• Started by `localdisplay.service`

Deployment

All display configuration is included in custom builds:
1. Boot config (config.txt) → `moode-source/boot/firmware/config.txt.overwrite`
2. Kernel cmdline → Build script adds rotation parameters
3. Touch calibration → `moode-source/etc/X11/xorg.conf.d/99-touch-calibration.conf`
4. Chromium script → `moode-source/usr/local/bin/start-chromium-clean.sh`

No manual configuration needed - everything is automated in the build process.

■ Related Documentation

• `DISPLAY_CONFIG_WORKING.md` - Original working configuration reference
• `docs/AUDIO_CHAIN_PERFECT_CONFIG.md` - Audio chain documentation (companion doc)
• `docs/COMPLETE_AUDIO_CHAIN_CONFIGURATION.md` - Detailed audio configuration

Status: ■ Complete and Working
Last Verified: 2025-01-11
System: Raspberry Pi 5 + Waveshare 7.9" HDMI Display