

# Deep Research: Initial Setup Script and Overwrite Mechanisms

Date: 2026-01-08

Status: ■ Complete Analysis

---

## ■ THE INITIAL SETUP SCRIPT

**\*\*Location:\*\***

`'imgbuild/moode-cfg/stage3_03-ghettoblaster-custom_00-run-chroot.sh'`

This is the **critical initial setup script** that runs during the pi-gen-64 build process. It executes **inside the chroot environment** during Stage 3 of the build, after moOde is installed but before the image is finalized.

---

## ■ WHAT THIS SCRIPT DOES

**\*\*1. User Creation (Lines 22-206)\*\***

- Creates user `andre` with UID 1000 (CRITICAL - moOde requires UID 1000)
- Removes `pi` user to prevent conflicts
- Sets password (from `test-password.txt` or default `4512`)
- Adds user to sudoers with NOPASSWD
- Verifies UID is exactly 1000

**Why Critical:**

- moOde's `worker.php` checks for user with UID 1000
- If UID is wrong → moOde shows "System doesn't contain a user ID" error
- This runs **during build**, ensuring user exists before first boot

**\*\*2. Custom Overlay Compilation (Lines 215-240)\*\***

- Compiles `ghettoblaster-ft6236.dts` → `dtbo` (touchscreen)
- Compiles `ghettoblaster-amp100.dts` → `dtbo` (audio)
- Uses `dtc` (Device Tree Compiler)
- If `dtc` not available → overlays compiled on first boot

**Why Important:**

- Overlays must be compiled before boot
- If missing → hardware won't work (no touchscreen, no audio)

**\*\*3. worker.php Patch Application (Lines 243-259)\*\***

- Applies `worker-php-patch.sh` to `/var/www/daemon/worker.php`
- Patches the `chkBootConfigTxt()` overwrite mechanism
- **CRITICAL:** This prevents `config.txt` from being overwritten!

### **The Patch:**

```
# Checks if patch already applied
if ! grep -q "GhettoBlaster: display_rotate=0" "$WORKER_FILE"; then
    bash "$PATCH_SCRIPT"
fi
```

### **What the Patch Does:**

- Disables `chkBootConfigTxt()` overwrite mechanism
- Hardcodes `\$status = 'Required headers present'
- Prevents `sysCmd('cp ... config.txt ...')` from executing

### ***\*\*4. Service Installation & Enablement (Lines 330-723)\*\****

- Enables `fix-audio-chain.service`
- Enables `audio-optimize.service`
- Enables `localdisplay.service`
- Enables SSH services (multiple methods)
- Enables network services
- Disables cloud-init (prevents boot delays)
- Disables NetworkManager-wait-online (prevents boot blocking)

### ***\*\*5. Cloud-init Removal (Lines 401-461)\*\****

- **Comprehensive removal** using 6 methods:
  1. Remove package (`apt-get remove --purge cloud-init`)
  2. Remove files (`rm -rf /etc/cloud`, `/var/lib/cloud`, etc.)
  3. Create systemd override
  4. Mask cloud-init.target
  5. Mask all cloud-init services
  6. Disable via systemctl

### **Why Critical:**

- Cloud-init can cause 2-5 minute boot delays
- Can interfere with network configuration
- Can cause username conflicts

### ***\*\*6. Network Configuration (Lines 642-666)\*\****

- Creates `/etc/wpa\_supplicant/wpa\_supplicant.conf`
- Pre-configures WiFi: SSID "Martin Router King", PSK "06082020"
- Enables wpa\_supplicant service

### ***\*\*7. Hostname Configuration (Lines 669-687)\*\****

- Sets hostname to "GhettoBlaster"
- Updates `/etc/hostname` and `/etc/hosts`

---

## **■ RELATIONSHIP TO OVERWRITE MECHANISMS**

### ***\*\*The 5 Overwrite Mechanisms (from `ALL\_OVERWRITE\_MECHANISMS\_FOUND.md`):\*\****

#### ***\*\*1. worker.php - chkBootConfigTxt() (RUNTIME)\*\****

- \*\*Status:\*\* ■ \*\*PATCHED BY THIS SCRIPT\*\*
- \*\*Location:\*\* `moode-source/www/daemon/worker.php` lines 105-121
- \*\*Fix:\*\* Applied via `worker-php-patch.sh` (line 246-259)
- \*\*Result:\*\* Overwrite mechanism disabled

**\*\*2. *export-image/prerun.sh - rsync (BUILD TIME)***

- \*\*Status:\*\* ■ \*\*FIXED\*\* (separate fix in export-image/prerun.sh)
- \*\*Location:\*\* `imgbuild/pi-gen-64/export-image/prerun.sh` line 73
- \*\*Fix:\*\* `rsync --exclude config.txt` + explicit copy after rsync
- \*\*Result:\*\* config.txt not overwritten during export

**\*\*3. *stage1/00-boot-files/00-run.sh - install (BUILD TIME)***

- \*\*Status:\*\* ■ \*\*FIXED\*\* (separate fix in stage1)
- \*\*Location:\*\* `imgbuild/pi-gen-64/stage1/00-boot-files/00-run.sh` line 14-15
- \*\*Fix:\*\* Uses `config.txt.overwrite` if available
- \*\*Result:\*\* Custom config.txt used from start

**\*\*4. *stage3/03-ghettoblaster-custom/00-run.sh (BUILD TIME)***

- \*\*Status:\*\* ■ \*\*ACTIVE\*\* (this is the fix mechanism)
- \*\*Location:\*\* `imgbuild/pi-gen-64/stage3/03-ghettoblaster-custom/00-run.sh` line 40-41
- \*\*Action:\*\* Copies `config.txt.overwrite` → `config.txt`
- \*\*Result:\*\* Custom config.txt applied after Stage 1

**\*\*5. *stage3/03-ghettoblaster-custom/00-deploy.sh (BUILD TIME)***

- \*\*Status:\*\* ■ \*\*ACTIVE\*\* (deploy-time fix)
- \*\*Location:\*\* `imgbuild/pi-gen-64/stage3/03-ghettoblaster-custom/00-deploy.sh` line 22-23
- \*\*Action:\*\* Copies `config.txt.overwrite` → `config.txt` during deploy
- \*\*Result:\*\* Custom config.txt applied during deployment

---

## ■ DEEP ANALYSIS: HOW OVERWRITE PROTECTION WORKS

**\*\*Build-Time Protection (Multi-Layer):\*\***

```

Stage 1: Boot Files Installation
■■> stage1/00-boot-files/00-run.sh
    ■■> Checks for config.txt.overwrite
        ■■> If exists → Uses config.txt.overwrite ■
        ■■> If missing → Uses default config.txt ■■

Stage 3: GhettoBlaster Custom Components
■■> 00-run.sh (before chroot)
    ■■> Copies config.txt.overwrite → config.txt ■
    ■■>
    ■■> 00-run-chroot.sh (inside chroot) ← THIS SCRIPT
        ■■> Creates user 'andre' ■
        ■■> Compiles overlays ■
        ■■> Applies worker.php patch ■ ← CRITICAL!
        ■■> Enables services ■

Export: Image Finalization
■■> export-image/prerun.sh
    ■■> rsync --exclude config.txt ■
    ■■> Explicit copy of config.txt.overwrite ■

```

## **\*\*Runtime Protection (After Boot):\*\***

```
Boot Sequence:  
1. Kernel reads /boot/firmware/config.txt (custom version) ■  
2. Systemd starts services  
    ■■> first-boot-setup.service  
        ■■> first-boot-setup.sh  
            ■■> Compiles overlays (if needed)  
            ■■> Applies worker.php patch (if needed)  
            ■■> Ensures user 'andre' exists  
  
3. moOde worker.php starts  
    ■■> chkBootConfigTxt() ← PATCHED!  
        ■■> Returns 'Required headers present' (hardcoded)  
            ■■> NO OVERWRITE! ■  
---
```

## **■ WHY DISPLAY SETTINGS STILL GET LOST**

### **\*\*The Problem:\*\***

Even though `worker.php` overwrite is disabled, display settings can still be lost because:

1. **\*\*moOde's `updBootConfigTxt()` function\*\*** can still modify `'config.txt'`:
  - Called when user changes settings in Web UI
  - Uses `'sed'` to modify specific lines
  - **Does NOT manage display settings** (`hdmi_cvt`, `hdmi_mode`, etc.)
  - These settings can be lost if `'config.txt'` is edited incorrectly
2. **\*\*cmdline.txt\*\*** is NOT protected:
  - `'worker.php'` can modify `'cmdline.txt'` for DSI displays
  - HDMI settings (`'video=HDMI-A-1:...'`) are not managed by moOde
  - Can be lost or overwritten
3. **\*\*config.txt.overwrite\*\*** template may not have correct settings:
  - If moOde restores from template → wrong settings applied
  - Template must match our custom display configuration

### **\*\*The Solution (New):\*\***

Created `persist-display-config.service` that:

- Runs **AFTER** moOde's `worker.php` finishes
- Restores display settings in `'config.txt'`
- Restores `'cmdline.txt'` HDMI settings
- Ensures `'config.txt.overwrite'` has correct settings

## **■ COMPLETE PROTECTION MATRIX**

| Protection Layer                   | Build-Time | Runtime | Status  |
|------------------------------------|------------|---------|---------|
| Stage 1: Use config.txt.overwrite  | ■          | -       | Active  |
| Stage 3: Copy config.txt.overwrite | ■          | -       | Active  |
| Export: Exclude from rsync         | ■          | -       | Active  |
| worker.php: Disable overwrite      | ■          | ■       | Patched |

| persist-display-config.service | - | ■ | New |

---

## ■ KEY FILES REFERENCED

### ***\*\*Build-Time Scripts:\*\****

1. `imgbuild/moode-cfg/stage3\_03-ghettoblaster-custom\_00-run-chroot.sh` ← \*\*THIS SCRIPT\*\*
2. `imgbuild/pi-gen-64/stage1/00-boot-files/00-run.sh` - Initial config.txt install
3. `imgbuild/pi-gen-64/stage3/03-ghettoblaster-custom/00-run.sh` - Copy config.txt.overwrite
4. `imgbuild/pi-gen-64/export-image/prerun.sh` - Export-time protection

### ***\*\*Runtime Scripts:\*\****

1. `moode-source/usr/local/bin/first-boot-setup.sh` - First boot setup
2. `moode-source/usr/local/bin/worker-php-patch.sh` - worker.php patch
3. `scripts/display/persist-display-config.sh` - Display persistence fix (NEW)

### ***\*\*Configuration Files:\*\****

1. `moode-source/boot/firmware/config.txt.overwrite` - Template for config.txt
2. `moode-source/www/daemon/worker.php` - moOde worker (patched)
3. `moode-source/www/inc/common.php` - Contains `updBootConfigTxt()` function

---

## ■ SUMMARY

### ***\*\*What the Initial Setup Script Does:\*\****

1. ■ \*\*Creates user 'andre'\*\* with correct UID 1000
2. ■ \*\*Compiles custom overlays\*\* (FT6236, AMP100)
3. ■ \*\*Applies worker.php patch\*\* - \*\*CRITICAL FOR OVERWRITE PROTECTION\*\*
4. ■ \*\*Enables services\*\* (audio, display, network, SSH)
5. ■ \*\*Removes cloud-init\*\* (prevents boot delays)
6. ■ **Configures network** (WiFi pre-config)
7. ■ **Sets hostname** (GhettoBlaster)

### ***\*\*How It Relates to Overwrite Mechanisms:\*\****

- \*\*Build-Time:\*\* Ensures `config.txt.overwrite` is used and copied correctly
- \*\*Runtime:\*\* Patches `worker.php` to prevent overwrite
- \*\*Gap:\*\* Display settings can still be lost via `updBootConfigTxt()` modifications
- \*\*Solution:\*\* `persist-display-config.service` restores settings after moOde modifications

### ***\*\*The Complete Protection Chain:\*\****

Build-Time:

Stage 1 → Uses config.txt.overwrite ■  
Stage 3 → Copies config.txt.overwrite → config.txt ■

```
Export → Excludes config.txt from rsync ■  
Runtime:  
  worker.php → Patched (no overwrite) ■  
  persist-display-config.service → Restores display settings ■  
---
```

## ■ DEEP RESEARCH FINDINGS

### ***\*\*1. The Initial Setup Script is CRITICAL\*\****

- Without it, user 'andre' wouldn't exist
- Without it, worker.php would overwrite config.txt
- Without it, overlays wouldn't be compiled
- \*\*This script is the foundation of the entire custom build\*\*

### ***\*\*2. Multi-Layer Protection is Necessary\*\****

- Build-time protection alone isn't enough
- Runtime protection is also needed
- Display settings need special handling (not managed by moOde)

### ***\*\*3. The Overwrite Problem is Complex\*\****

- Not just one mechanism, but \*\*5 different mechanisms\*\*
- Each needs different protection strategy
- Display settings need \*\*separate persistence mechanism\*\*

### ***\*\*4. The Solution is Layered\*\****

- \*\*Layer 1:\*\* Build-time (config.txt.overwrite)
- \*\*Layer 2:\*\* Runtime (worker.php patch)
- \*\*Layer 3:\*\* Post-runtime (persist-display-config.service)

---  
**Status: ■ COMPLETE DEEP RESEARCH - ALL MECHANISMS UNDERSTOOD**