I2C Configuration Guide for Raspberry Pi 4 with HiFiBerry and Waveshare Display

The good news: Your hardware setup has **no inherent I2C bus conflicts**. The HiFiBerry boards and Waveshare DSI display use completely separate I2C buses by design, eliminating the primary concern when combining audio HATs with DSI displays.

Why there's no conflict

The Raspberry Pi 4 architecture separates these I2C communication channels onto different physical buses. The **HiFiBerry AMP100 and ASP** communicate exclusively over **i2c-1** (the GPIO header I2C bus on pins 3/5), appearing as `/dev/i2c-1` in Linux. Meanwhile, the **Waveshare 7.9" display** uses its own dedicated **i2c_csi_dsi** bus through the DSI connector's built-in I2C lines (pins 11/12), appearing as `/dev/i2c-10`. These buses operate independently without address space overlap or electrical interference.

Complete /boot/firmware/config.txt configuration

Here's the exact configuration for your hardware combination:

```
# Disable onboard audio (required for HiFiBerry)
dtparam=audio=off

# Graphics driver (required for modern DSI displays)
dtoverlay=vc4-kms-v3d,noaudio

# HiFiBerry AMP100 configuration
dtoverlay=hifiberry-amp100,automute
force_eeprom_read=0

# Waveshare 7.9" DSI display with touch
dtoverlay=vc4-kms-dsi-waveshare-panel,7_9_inch

# Optional: Enable GPIO I2C for manual access (not required for HiFiBerry)
# dtparam=i2c_arm=on

# Optional: Adjust I2C speed if needed (default 100kHz usually works)
# dtparam=i2c_arm_baudrate=100000
```

Critical notes:

- **Do NOT add** `dtparam=i2c_vc=on` - the Waveshare overlay automatically enables the DSI I2C bus
- The `force_eeprom_read=0` parameter is essential for Linux kernel 5.4+ to prevent HAT EEPROM conflicts
- The `automute` parameter prevents amplifier popping sounds during playback transitions
- If using Moode Audio, configure HiFiBerry through the web interface (System → Configure → Audio), which automatically manages these overlays

I2C bus architecture explained

Your hardware uses three distinct I2C buses:

i2c-0: HAT EEPROM bus

- **Physical location:** GPIO0/GPIO1 (pins 27/28)
- **Purpose:** HAT identification via EEPROM
- **Used by:** Raspberry Pi firmware during boot
- **Address:** 0x50 (EEPROM on HiFiBerry)
- **Enable method:** Automatic via firmware
- **User access:** Not typically needed

i2c-1: GPIO I2C bus (i2c_arm)

- **Physical location:** GPIO2/GPIO3 (pins 3/5 on 40-pin header)
- **Purpose:** General I2C communication with HATs
- **Used by:** HiFiBerry AMP100 and ASP boards
- **Devices present:**
  - **TAS5756M amplifier:** 0x4c (appears as "UU" when driver loaded)
  - **ADAU1701 DSP:** 0x34 (on ASP add-on)
  - **EEPROM:** 0x50 (may appear as "UU")
- **Enable method:** `dtparam=i2c_arm=on` or automatic via HiFiBerry overlay
- **Features:** Built-in 1.8kΩ pull-up resistors to 3.3V

i2c-10: DSI/CSI I2C bus (i2c_csi_dsi)

- **Physical location:** DSI connector pins 11 (SCL) and 12 (SDA)
- **Purpose:** Display panel and touch controller communication
- **Used by:** Waveshare 7.9" display
- **Devices present:**
  - **Panel controller:** 0x45
  - **GT911 touch controller:** 0x14
- **Enable method:** Automatic via `vc4-kms-dsi-waveshare-panel` overlay
- **Features:** Electrically isolated from GPIO I2C, separate BSC0 controller

Additional buses (informational)

- **i2c-20, i2c-21:** HDMI DDC buses (created by vc4-kms-v3d for EDID reading)
- **i2c-22:** Parent controller for i2c-0 and i2c-10 (do not use directly)

Device tree overlay parameters

Waveshare display overlay parameters

The `vc4-kms-dsi-waveshare-panel` overlay supports these key parameters:

Basic usage:

```
dtoverlay=vc4-kms-dsi-waveshare-panel,7_9_inch
```

Available parameters:

- **`7_9_inch`** - Configures for 7.9" model (400×1280 resolution, GT911 touch at 0x14)
- **`dsi0`** - Use DSI0 port instead of default DSI1 (for CM4 or Pi 5)
- **`i2c1`** - Force touch to use GPIO I2C (i2c-1) instead of DSI I2C (requires physical 4-pin header connection)
- **`touchscreen-inverted-x`** - Flip X axis for rotation
- **`touchscreen-inverted-y`** - Flip Y axis for rotation
- **`touchscreen-swapped-x-y`** - Swap axes for 90°/270° rotation

Example for 90° rotation:

```
dtoverlay=vc4-kms-dsi-waveshare-panel,7_9_inch,touchscreen-inverted-
y,touchscreen-swapped-x-y
```

HiFiBerry AMP100 overlay parameters

The `hifiberry-amp100` overlay offers these options:

- **`automute`** - Automatically mute amplifier when no audio playing (recommended)
- **`24db_digital_gain`** - Allow up to 24dB digital gain (default limits to 0dB to prevent clipping)
- **`slave`** - Force AMP100 into slave mode (Pi as I2S master)
- **`leds_off`** - Disable onboard indicator LEDs
- **`mute_ext_ctl=<val>`** - Enable hardware mute control in ALSA mixer

Example with options:

```
dtoverlay=hifiberry-amp100,automute,leds_off=true
```

Waveshare display DIP switch settings

The Waveshare 7.9" display has a DIP switch on the back that controls I2C mode selection:

I2C0 Mode (RECOMMENDED for your setup)

- **Switch position:** Set to "I2C0"
- **Communication path:** Via DSI cable pins 11/12
- **Linux bus:** `/dev/i2c-10` (i2c_csi_dsi)
- **Power:** Separate 5V supply required via header or USB-C
- **Advantages:**
  - No GPIO I2C cable needed
  - No conflict with HiFiBerry I2C
  - Cleaner installation
  - Uses dedicated DSI I2C bus
- **config.txt:** Standard overlay without `i2c1` parameter

I2C1 Mode (NOT recommended for your setup)

- **Switch position:** Set to "I2C1"
- **Communication path:** Via 4-pin header to GPIO pins 3/5
- **Linux bus:** `/dev/i2c-1` (i2c_arm)

- **Power:** Via 4-pin header connection
- **Disadvantages:**
    - **Shares I2C bus with HiFiBerry** (potential for conflicts)
    - Requires physical cable connection to GPIO header
    - Address collision possible with HiFiBerry devices
- **config.txt:** Requires `i2c1` parameter in overlay

For your hardware combination, use I2C0 mode exclusively. This eliminates any possibility of I2C address conflicts or bus contention between the display and audio hardware.

HiFiBerry I2C address mapping

The HiFiBerry hardware uses these confirmed I2C addresses on the i2c-1 bus:

Device	7-bit Address	8-bit Address	Function	Detection
TAS5756M (AMP100)	0x4c	0x98/0x99	Class-D amplifier control	Shows as "UU" when driver active
ADAU1701 (ASP DSP)	0x34	0x68/0x69	DSP processor control	Shows as "UU" when driver active
EEPROM	0x50	0xA0/0xA1	HAT configuration data	May show as "UU" or "50"

Important notes:

- Addresses shown as "UU" in `i2cdetect` output indicate a kernel driver is actively using that device (not an error)
- The ADAU1701 can support addresses 0x34-0x37 via ADDR pins, but HiFiBerry ASP uses 0x34
- These addresses are on i2c-1 only and do not conflict with the display on i2c-10

GT911 touch controller specifications

The Waveshare display uses a Goodix GT911 capacitive touch controller:

I2C Communication:

- **Addresses:** 0x14 or 0x5D (address selected during reset via INT pin level)
- **Default on Waveshare:** 0x14
- **Bus:** i2c-10 (i2c_csi_dsi) when using I2C0 mode
- **Speed:** Up to 400kHz, 100kHz recommended
- **Protocol:** Standard I2C with 16-bit register addressing

Touch capabilities:

- Up to 5 simultaneous touch points
- Configurable resolution (set to 400×1280 for 7.9" model)
- Interrupt-driven operation
- Optional firmware configuration file

Driver integration: The Linux `goodix` driver automatically handles:

- I2C address selection (if reset GPIO provided)
- Touch coordinate reporting via input subsystem
- Multi-touch protocol B
- Power management

Verification commands after configuration

After applying the configuration and rebooting, verify proper operation:

Check I2C buses are present

```
i2cdetect -l
```

Expected output:

```
i2c-1   i2c       bcm2835 (i2c@7e804000)          I2C adapter
i2c-10  i2c       i2c-10-mux (chan_id 0)          I2C adapter
i2c-20  i2c       brcmstb-hdmi-i2c                I2C adapter
i2c-21  i2c       brcmstb-hdmi-i2c                I2C adapter
i2c-22  i2c       bcm2835 (i2c@7e205000)          I2C adapter
```

Scan HiFiBerry I2C bus

```
sudo i2cdetect -y 1
```

Expected output:

```
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- UU -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- UU -- -- --
50: UU -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

- **UU at 0x34:** ADAU1701 DSP (ASP board)
- **UU at 0x4c:** TAS5756M amplifier (AMP100)
- **UU or 50 at 0x50:** HAT EEPROM

Scan display I2C bus

```
sudo i2cdetect -y 10
```

Expected output:

```
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- 14 -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

```
40: -- -- -- -- -- 45 -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

- **14 at 0x14:** GT911 touch controller
- **45 at 0x45:** Display panel controller

Verify audio device

```
aplay -l
```

Expected output:

```
**** List of PLAYBACK Hardware Devices ****
card 0: sndrpihifiberry [snd_rpi_hifiberry_dac], device 0: HiFiBerry DAC
HiFi pcm5102a-hifi-0 [HiFiBerry DAC HiFi pcm5102a-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

Check touch input device

```
ls -l /dev/input/by-path/*event-touch*
```

Expected output:

```
lrwxrwxrwx 1 root root 10 Nov  7 12:34 /dev/input/by-path/platform-
fef00700.dsi-event-touch -> ../event0
```

Verify device tree overlays loaded

```
dtoverlay -l
```

Expected output:

```
Overlays (in load order):
0:  vc4-kms-v3d
1:  hifiberry-amp100
2:  vc4-kms-dsi-waveshare-panel
```

Check kernel messages for errors

```
dmesg | grep -E "(i2c|goodix|hifiberry|tas5756|adau1701)"
```

Look for successful device initialization messages and absence of errors.

Test display and touch

```
# Check for touch events
sudo evtest /dev/input/event0
# Touch the screen - you should see coordinate output
```

Moode Audio specific considerations

If using Moode Audio, the configuration process is streamlined through the web interface:

Initial HiFiBerry configuration

1. Navigate to **Menu → Configure → Audio**
2. Select **"HiFiBerry Amp2"** from the I2S audio device dropdown (Moode uses this profile for AMP100)
3. Click **SET** - system automatically reboots
4. Verify under **MPD options** that Audio device shows "I2S audio device"

Adding the DSI display

1. Connect display and verify DIP switch is set to I2C0
2. Manually add to `/boot/firmware/config.txt`:
3. `dtoverlay=vc4-kms-dsi-waveshare-panel,7_9_inch`
4. Reboot the system
5. Navigate to **Menu → Configure → System**
6. Enable **"Local display"** option
7. Set screen blank timeout as desired

Important Moode considerations

Automated configuration: Moode automatically manages the HiFiBerry device tree overlay when you select it via the web interface. The system writes appropriate `dtoverlay=hifiberry-amp100` to config.txt and configures ALSA/MPD automatically.

Manual additions: You must manually add the Waveshare display overlay. Moode's automation doesn't cover DSI displays yet, but the two configurations coexist without conflict.

Volume control: For AMP100, start at low ALSA volume (20-30%) for safety. Moode properly handles volume scaling - 100% ALSA = 0dB (no clipping in current versions).

Power supply: The AMP100 can power the Pi through its GPIO header. Ensure your external supply (12-30V) to the AMP100 provides adequate current (3-4A minimum) to handle both the Pi and amplifier under load, especially with the display connected.

DSP programming: If using the ASP add-on, program it using the HiFiBerry DSP Toolkit (`pip install hifiberry-dsp`) via SSH. The ADAU1701 at address 0x34 on i2c-1 is accessible after boot.

Known issues and troubleshooting

Issue: Display touch controller not detected

Symptoms: No `/dev/input/event*` for touch, i2cdetect shows nothing at 0x14

Solutions:

1. Verify DIP switch is set to I2C0 mode
2. Check display has adequate power (400-490mA required)
3. Ensure `vc4-kms-dsi-waveshare-panel` overlay is loaded: `dtoverlay -l`
4. Check kernel messages: `dmesg | grep goodix`
5. Verify DSI cable is fully seated in both connectors
6. Try power cycling the entire system (not just reboot)

Issue: HiFiBerry not producing audio

Symptoms: `aplay -l` shows no devices, or playback produces no sound

Solutions:

1. Verify `dtparam=audio=off` is present in config.txt
2. Ensure `dtoverlay=hifiberry-amp100` is loaded: `dtoverlay -l`
3. Check `force_eeprom_read=0` is set (critical for kernel 5.4+)
4. Verify AMP100 power supply is connected and adequate
5. Check speaker connections and amplifier mute status
6. In Moode, verify I2S device is selected and volume is not at zero
7. Check kernel messages: `dmesg | grep -i hifiberry`

Issue: Display shows image but touch doesn't work

Symptoms: Display functions but touch input not registering

Solutions:

1. Verify GT911 appears at 0x14: `sudo i2cdetect -y 10`
2. Check touch input device exists: `ls /dev/input/by-path/*touch*`
3. Test with `sudo evtest` to see if events are generated
4. May have 64-second delay on first boot (GT911 searches for config file)
5. Check for device tree overlay parameters if rotation is needed

Issue: I2C bus conflicts or "UU" everywhere

Symptoms: i2cdetect shows "UU" at all addresses, or devices not responding

Solutions:

1. "UU" is normal when kernel drivers are active - not an error
2. To see actual addresses, temporarily blacklist drivers or boot without overlays
3. If genuine conflicts exist, verify you're using I2C0 mode on display (not I2C1)
4. Check no other I2C devices are connected to GPIO pins 3/5
5. Verify config.txt doesn't have conflicting i2c overlays

Issue: Moode web interface shows "MPD output error"

Symptoms: Intermittent audio dropouts, I2S sync errors in dmesg

Solutions:

1. Disable Bluetooth if not needed: `dtoverlay=disable-bt` in config.txt
2. Update to latest Moode version
3. Verify adequate power supply (3A+ for Pi 4)
4. Check for firmware updates: `sudo rpi-update`
5. Try different USB cables if using USB storage

Issue: Display power or instability problems

Symptoms: Lightning bolt icon, random reboots, display flickering

Solutions:

1. Use quality 5V 3A+ power supply
2. If AMP100 powers the Pi, ensure external supply to AMP100 is adequate
3. Keep combined draw of Pi + display + AMP100 idle current within supply capacity
4. Consider separate power supply for display if issues persist
5. Check all ground connections are solid

Best practices summary

For optimal operation of this hardware combination:

1. **Use I2C0 mode** on the Waveshare display DIP switch - this is non-negotiable for conflict-free operation
2. **Configure HiFiBerry through Moode's web interface** if using Moode Audio - let automation handle audio configuration
3. **Add display overlay manually** to config.txt - it won't conflict with Moode's audio configuration
4. **Use adequate power supply** - 3-4A minimum, consider letting AMP100 power the Pi through GPIO header
5. **Build incrementally** - configure audio first, verify operation, then add display
6. **Document working configuration** - save your final config.txt for reference
7. **Avoid manual I2C parameters** - don't add `dtparam=i2c_vc=on` as it's unnecessary and handled automatically
8. **Monitor for errors** - check `dmesg` and `/var/log/moode.log` after configuration changes

The separation of I2C buses in your hardware setup means these components are designed to work together. Following this configuration guide should result in a stable system with functioning audio, display, and touch control operating simultaneously without conflicts.