Waveshare 3.5inch RPi LCD (G)

Setup Guide for Raspberry Pi OS

Kernel 6.12.x | Debian Trixie | panel-mipi-dbi driver

1. Overview

This guide documents the complete setup process for the Waveshare 3.5inch RPi LCD (G) display on Raspberry Pi OS. The (G) variant uses the ST7796S display driver chip and XPT2046 resistive touch controller, both communicating over SPI. Unlike older Waveshare LCD variants, the (G) uses the modern panel-mipi-dbi kernel framework instead of the legacy LCD-show scripts.

2. Hardware

2.1 Display Specifications

Specification	Value
Display Driver	ST7796S
Touch Controller	XPT2046 (resistive)
Resolution	320 x 480 pixels
Panel Type	IPS, 262K colors
Interface	SPI (display + touch share bus)
Connection Options	40-pin GPIO header or GH1.25 11-pin cable
Backlight Control	GPIO18 (PWM capable)

2.2 Wiring - GH1.25 11-Pin Cable to Raspberry Pi

> NOTE: Only 8 wires are needed for display-only (no touch). MISO, T_CS, T_IRQ are touch-only.

Cable Pin	Signal	Raspberry Pi
1	VCC	Pin 1 - 3.3V (NOT 5V)
2	GND	Pin 6 - GND
3	CLK	Pin 23 - GPIO11 (SPI0 SCLK)
4	MOSI	Pin 19 - GPIO10 (SPI0 MOSI)

5	LCD_CS	Pin 24 - GPIO8 (SPI0 CE0)
6	DC	Pin 15 - GPIO22
7	RST	Pin 13 - GPIO27
8	BL	Pin 12 - GPIO18 (backlight)
9	MISO	Pin 21 - GPIO9 (SPI0 MISO)
10	T_CS	Pin 26 - GPIO7 (SPI0 CE1)
11	T_IRQ	Pin 11 - GPIO17

3. Software Setup

3.1 Prerequisites

- Raspberry Pi OS (Bullseye or newer, 32-bit or 64-bit)
- Kernel version 6.1.21 or higher (check with: uname -r)
- Internet connection for downloading firmware

Verify kernel support before proceeding:

```
modinfo panel-mipi-dbi 2>/dev/null && echo FOUND || echo NOT FOUND
```

3.2 Step 1 - Install ST7796S Firmware

The panel-mipi-dbi driver requires a firmware file containing the ST7796S initialization sequence:

```
cd /tmp
wget https://files.waveshare.com/wiki/common/St7796s.zip
unzip St7796s.zip
sudo cp st7796s.bin /lib/firmware/
```

Verify:

```
ls -la /lib/firmware/st7796s.bin
```

NOTE: The file should be approximately 117 bytes. It starts with the 'MIPI DBI' magic header.

3.3 Step 2 - Configure /boot/firmware/config.txt

Open the boot configuration file:

```
sudo nano /boot/firmware/config.txt
```

Add the following lines at the end of the file, under the [all] section:

```
dtparam=spi=on
dtoverlay=mipi-dbi-spi,speed=48000000
dtparam=compatible=st7796s\0panel-mipi-dbi-spi
dtparam=width=320,height=480,width-mm=49,height-mm=79
dtparam=reset-gpio=27,dc-gpio=22,backlight-gpio=18
```

```
dtoverlay=ads7846,speed=2000000,penirq=17,xmin=300,ymin=300,xmax=3900,ymax=3800
,pmin=0,pmax=65535,xohms=400
extra_transpose_buffer=2
```

NOTE: On older Raspberry Pi OS (Bullseye), the config file may be at /boot/config.txt instead.

3.4 Step 3 - Reboot and Verify

```
sudo reboot
```

After reboot, verify the framebuffer and driver loaded:

```
ls /dev/fb*
dmesg | grep -i panel-mipi
```

Expected output:

```
/dev/fb0  /dev/fb1
[drm] Initialized panel-mipi-dbi 1.0.0 for spi0.0 on minor 0
[drm] fb1: panel-mipi-dbid frame buffer device
```

4. Test Pattern

4.1 Enable Backlight

The backlight is a simple on/off GPIO (not PWM). Turn it on via sysfs:

```
echo 1 | sudo tee /sys/class/backlight/backlight_gpio/brightness
```

4.2 Python Test Script

Install PIL and create the test pattern script:

```
sudo apt-get install -y python3-pil
```

Create ~/lcd_test.py with the following content:

```
import struct
from PIL import Image, ImageDraw

FB, WIDTH, HEIGHT = '/dev/fb1', 320, 480
img = Image.new('RGB', (WIDTH, HEIGHT))
draw = ImageDraw.Draw(img)
colors = ['red','green','blue','yellow','cyan','magenta','white','black']
bar_h = HEIGHT // len(colors)
for i, c in enumerate(colors):
    draw.rectangle([0, i*bar_h, WIDTH, (i+1)*bar_h], fill=c)
draw.text((10, 10), 'LCD OK', fill='white')
pixels = list(img.convert('RGB').getdata())
buf = bytearray()
for r,g,b in pixels:
    rgb565 = ((r&0xF8)<<8)|((g&0xFC)<<3)|(b>>3)
    buf += struct.pack('<H', rgb565)
open('/dev/fb1','wb').write(buf)
```

Run it:

```
python3 ~/lcd_test.py
```

5. Autostart at Boot

Create a systemd service to run the display script automatically at boot:

```
sudo nano /etc/systemd/system/waveshare-lcd.service
```

Service file content:

```
[Unit]
Description=Waveshare 3.5inch LCD (G) Init
After=multi-user.target

[Service]
Type=oneshot
ExecStart=/usr/bin/python3 /home/pi/lcd_test.py
RemainAfterExit=yes
User=pi

[Install]
WantedBy=multi-user.target
```

Enable and start:

```
sudo systemctl daemon-reload
sudo systemctl enable waveshare-lcd
sudo systemctl start waveshare-lcd
```

Check status:

```
sudo systemctl status waveshare-lcd
```

6. Automated Setup Script

A bash script (waveshare_lcd_setup.sh) is provided that performs all steps automatically:

```
sudo bash waveshare_lcd_setup.sh
```

The script performs the following actions:

- Checks for panel-mipi-dbi kernel support
- Downloads and installs the ST7796S firmware file
- Appends LCD configuration to /boot/firmware/config.txt
- Installs python3-pil dependency
- Creates the test pattern Python script
- Creates and enables the systemd service

7. Troubleshooting

Symptom	Solution
/dev/fb1 missing	Check config.txt has dtparam=spi=on and mipi-dbi-spi overlay. Reboot.
Backlight on, screen black	Check MOSI (Pin 19), DC (Pin 15), RST (Pin 13) wiring. Verify st7796s.bin is in /lib/firmware/.
Screen black, no backlight	Check BL wire to Pin 12 (GPIO18). Run: echo 1 | sudo tee /sys/class/backlight/backlight_gpio/brightness
panel-mipi-dbi NOT FOUND	Kernel too old. Need 6.1.21+. Run: sudo rpi-update
VCC connected to 5V	Move VCC to Pin 1 (3.3V). 5V will damage the display.
modetest works but fb stays black	DRM CRTC not active. Use Python PIL to write directly to /dev/fb1 instead.

8. Useful Diagnostic Commands

Command	Purpose
`uname -r`	Check kernel version
`modinfo panel-mipi-dbi`	Verify driver availability
`ls /dev/fb*`	Check framebuffer devices
`dmesg | grep -i panel-mipi`	Check driver init messages
`ls /sys/class/backlight/`	Check backlight control
`cat /sys/class/graphics/fb1/virtual_size`	Check framebuffer dimensions
`modetest -M panel-mipi-dbi -c`	Check DRM connector status
`lsmod | grep spi`	Verify SPI module loaded