



## CFAM800480B0-050TC RPi 3/B+ Compatibility Modification

The Crystalfontz CFAM800480B0-050TC display requires a minor hardware and software modification to ensure the display restarts correctly when a reboot command is issued from the **Raspberry Pi 3/B+**. The modifications required to ensure the display reboots correctly are outlined below.

*A small description of the issue:*

*When restarting, using a Pi 3/B+, colors sequentially flash on the display rather than the display restarting. This is a result of the display entering test mode.*

*Summary of solution:*

*Control the display's RESET during the reboot cycle.*

**Note: this modification is not required on the Raspberry Pi 4 and CM4.**

### Part 1: Hardware modifications

Solder a header to J3 on the back of the display board as indicated in Figure 1 below:

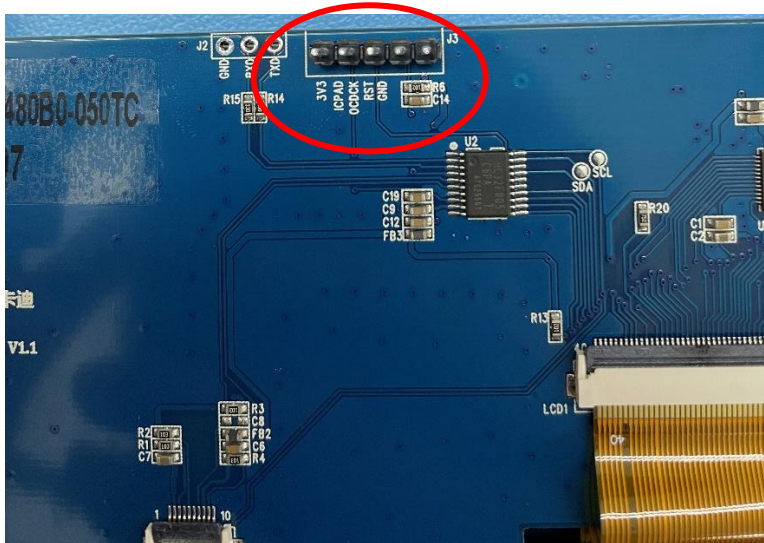


Figure 1: CFAM800480B0-050TC J3 position

Ensure the bottom of the header does **NOT** make contact with the metallic casing of the display (underneath the blue board). Shorting the pins will permanently damage the display. Figure 2, below, shows the point of contact to be avoided:

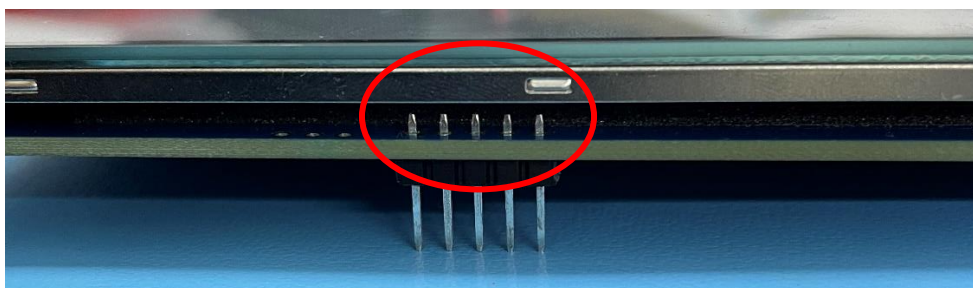


Figure 2: Point of contact to be AVOIDED between display and header



Alternately, solder a single pin header to the RESET pin, as only the RESET pin needs to be controlled. Note the silkscreen labeling corresponds to the order of the pins.

Next, add a jumper between the RESET pin on J3 and any available GPIO on the RPi 3/B+. *I.e.*, connect the RESET pin to GPIO27 (pin 36) on the RPi. Once a connection from the RESET pin to a GPIO has been made, the hardware side modifications are complete.

### **Part 2: Software modifications**

A script that toggles the display's RESET pin needs to be run during restart. This script allows the display to reboot correctly by preventing the display from entering test mode and colored screens from appearing.

The simple code to achieve the required action on the GPIO (connected to the RESET pin in the above hardware modification) is below. The below script simply toggles the RESET pin LOW and invokes a small delay before toggling it HIGH again (change 36 to the correct GPIO connection).

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
GPIO.setup(36, GPIO.OUT)

GPIO.output(36, False)
time.sleep(0.12)
GPIO.output(36, True)

GPIO.cleanup()
```

Next, a configuration file needs to be written for *systemd*. Use the following line in the command prompt, to add information to the “.service” file. For this example, the file is named “rebootfix.service”

```
sudo nano /lib/systemd/system/rebootfix.service
```

Add the following to the newly created file:

```
[Unit]
Description = display reboot fix
DefaultDependencies=no

[Service]
```



```
Type=oneshot
ExecStart=/usr/bin/python /***/***/***/xxxxx.py
TimeoutStartSec=0

[Install]
WantedBy=shutdown.target
```

Replace the asterisk with the appropriate direction and filename.

Use the following lines to tell systemd to start the “rebootfix.service” when the shutdown command is called:

```
sudo systemctl daemon-reload
sudo systemctl enable rebootfix.service
```

Upon running the previous command, a reset will be sent simultaneously to the display while rebooting the RPi via the connected GPIO, preventing the colored screens from appearing.

### **Part 3: Powering off the display entirely (under a normal shutdown scenario)**

To completely turn off the display after shutting down the Raspberry Pi, the display’s VDD line must be pulled LOW. In other words, power must be disconnected from the display, which means the Raspberry Pi has to be disconnected from its power supply.