

## DJI Phantom 2 Vision+ Plus Blank Screen Corrupt NAND fix

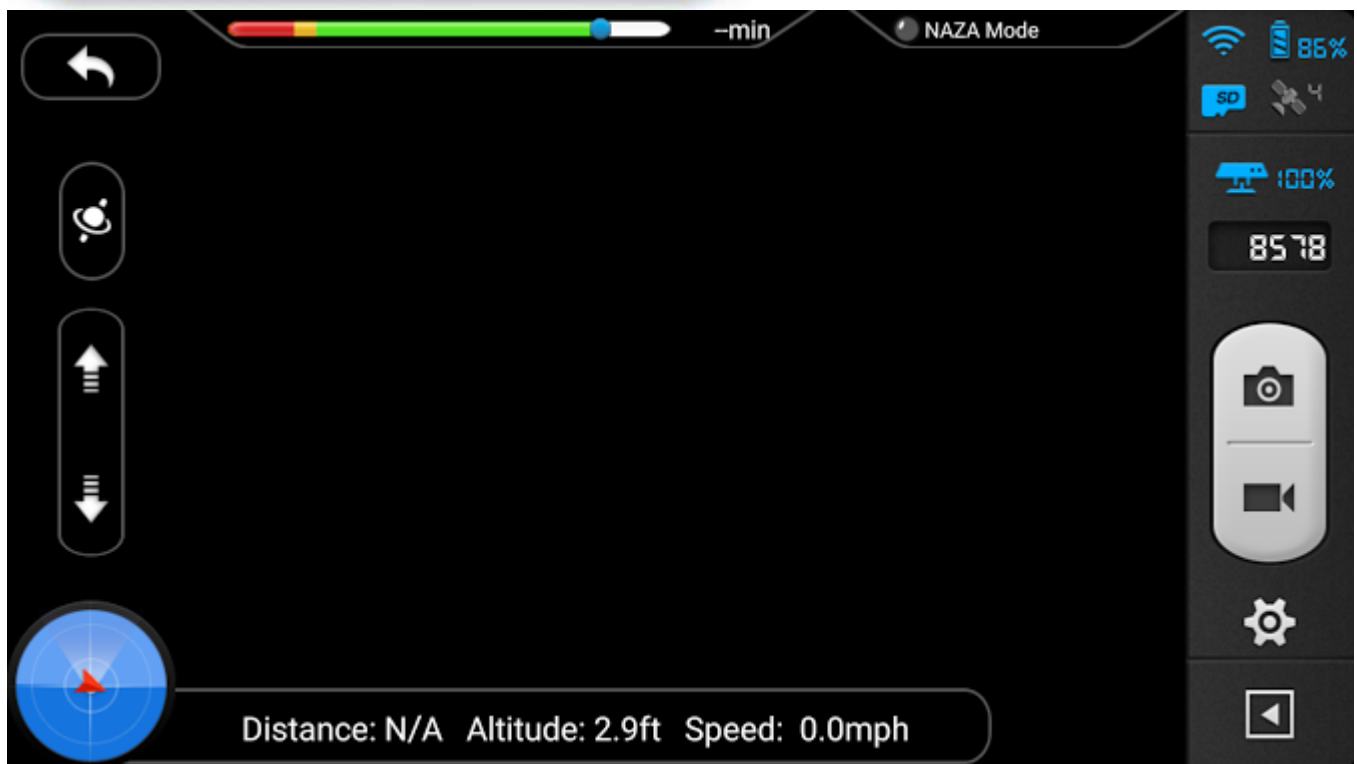
### DJI Phantom 2 Vision+ Vision Plus Blank or Black Screen Fix

This brief tutorial aims to help people fix the black screen / blank screen issue that affects many DJI Phantom 2 Vision+ Plus quadcopters out there. There are tons of forum posts about people who store their drone for a few months and turn it on to find that there is no streaming video to the app. Or others had everything working fine, then they update the quad only to lose the ability to view live streaming FPV footage while flying.

Credit goes out to Gauch0, andrew\_by, and rmhome on the PhantomPilots.com "[lightbridge firmware PROBLEM \(and solution\) - firmware version not found by dji assistant tool](#)" thread.

If these instructions were helpful and saved you some money, please feel free to **donate** a little something as a way to say thanks:

<https://www.paypal.me/TGutmann>



**Problem:** DJI Phantom 2 Vision+ works fine, has a wifi connection to the app, transmits telemetry and GPS status fine, but no live FPV video streaming while flying

**Hypotheses:** wifi module (air side, inside the drone) is bad

**Solutions:**

- Buy new module (\$150-\$250 on eBay)
- Reflash wifi module firmware to fix / reflash corrupt NAND memory chips

### **What you'll need:**

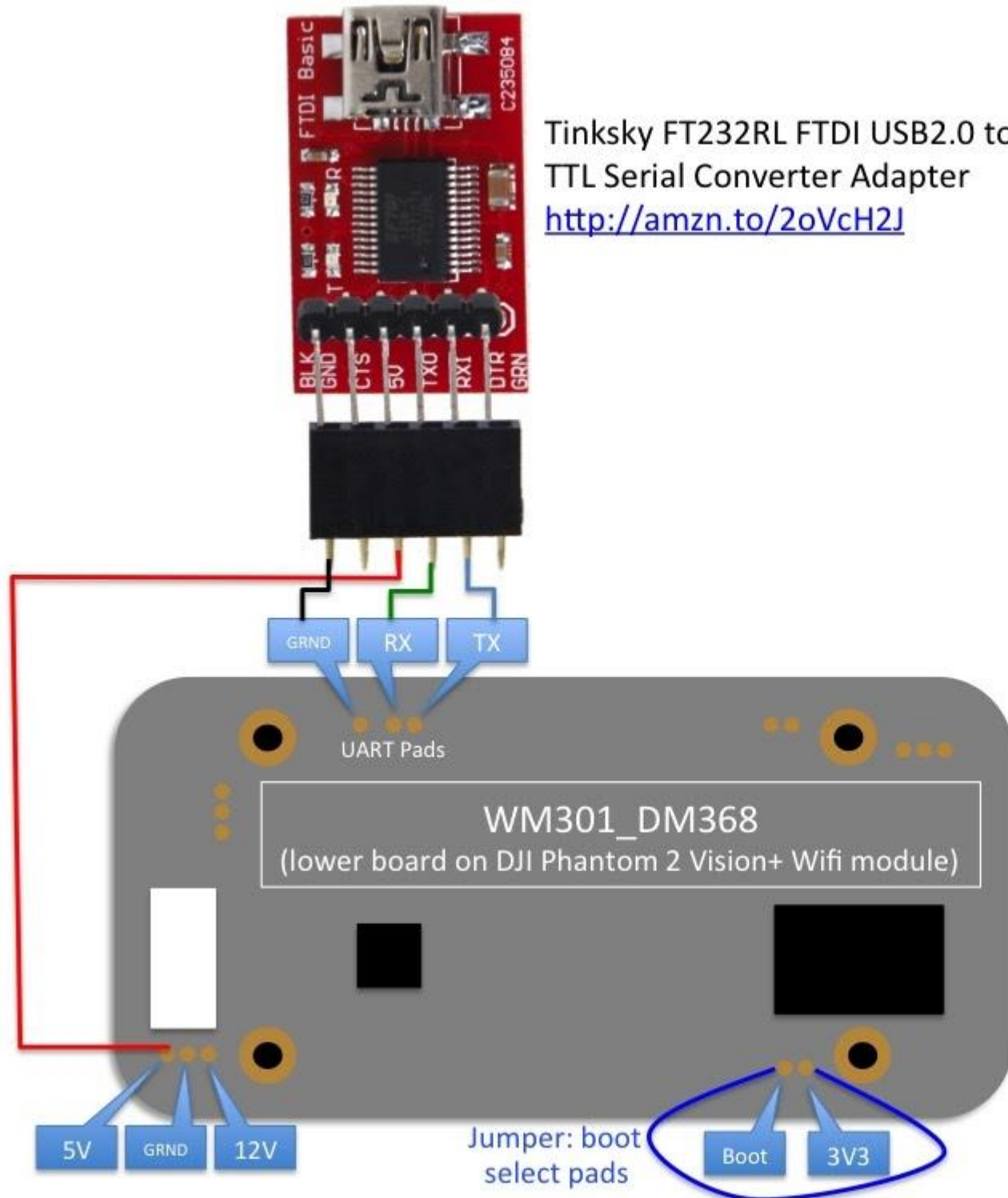
1. Tinksky FT232RL FTDI USB2.0 to TTL Serial Converter Adapter  
<http://amzn.to/2oVcH2J>
2. [Download the Flash utility and binaries for phantom 2 vision plus wifi module firmware](#)
3. [Tera Term](#) or similar serial terminal software (allows you to see results from the serial connection you are going to make between the wifi board and your computer)
4. Soldering iron
  1. solder
  2. small wires to connect the USB TTL adapter to the pads on the wifi board
5. Screw drivers to extract the wifi module from the quad

### **Section 1: Hardware Process**

- Open the quad, extract the wifi module, open the wifi module, remove the lower pc board (labeled WM301\_DM368)
- This [Youtube video by Palek Desai](#) walks you through all the above steps, if you run into any snags
- Take the lower board out, it looks like this:



- Connect up the USB2.0 to TTL Serial Converter as follows:



- Tips: use a small blade or small file to rough up the pads on the pc board so solder will flow and make a good connection to your wiring. And pre-solder the pads and wires before soldering the two together for best results.

## Section 2: Flashing procedure stage 1 (on a Window machine)

1. Plug the TTL serial converter into a USB port on your computer

1. Determine what COM port the device is assigned  
(<https://tnp.uservoice.com/knowledgebase/articles/172101-determining-the-com-port-of-a-usb-to-serial-adapte> )
2. Open up Tera Term or similar terminal application, connect to the TTL serial converter by selecting the serial connection
  1. In Tera Term you goto Setup menu > Serial Port ...
  2. Choose the correct COM port that was assigned to your adapter, and choose "no flow control" and 115200 for the COM port speed
  3. Jumper BOOT and 3V3 pads of the wifi module PC board
  4. Connect the board to the TTL Serial adapter
1. Note: I had some issues with the board not booting and the terminal program not displaying text from the board if I didn't connect specifically in this order: TTL adapter to USB port, then open Tera Term on the correct serial port, then connect the PC board to the TTL serial adapter (ie power the board)
5. If you did everything correctly you'll see "BOOT ME BOOT ME BOOT ME" start displaying in the terminal window
6. Now leave everything connected and close out Tera Term
7. Open a Command Prompt and navigate to the folder where you put the [Flash utility and binaries for phantom 2 vision plus wifi module firmware](#)
8. **Update on 8-31-2017:** The below steps #9 and #10 are optional 90% of the time. I've now refreshed the firmware on over 20 units and only needed steps 9 and 10 once. I recommend skipping these steps and moving on to section 3, only follow steps 9 and 10 if section 3 and onward doesn't fix your problem.
9. ~~Execute this command: sfh\_DM36x.exe -nandflash -v -p "COMXX" dm6467\_297\_ubl.img -u-boot-modifiedByGaucho.img~~
  1. ~~Note: replace "COMXX" with "COM3" if your device is on COM port #3. Quotes around COM are fine in the command~~
10. ~~It will start flashing some stuff and likely pop out and start saying "BOOT ME BOOT ME BOOT ME" again. At this point I hit CTRL-C which exits the utility. From there I kicked off the same command you see in step 8, above. Eventually all the flashing will succeed and you'll see something like this:~~

```
C:\Windows\system32\cmd.exe
Target: Writing header and image data to Block 0x00000015, Page 0x00000000
Target: Erasing block 0x00000016 through 0x00000016.
Target: Writing header and image data to Block 0x00000016, Page 0x00000000
Target: Erasing block 0x00000017 through 0x00000017.
Target: Writing header and image data to Block 0x00000017, Page 0x00000000
Target: Erasing block 0x00000018 through 0x00000018.
Target: Writing header and image data to Block 0x00000018, Page 0x00000000
Target: Protecting the entire NAND flash.
Target: DONE
Sending the Application image
Waiting for SENDIMG sequence...
Target: SENDIMG
SENDIMG received. Returning ACK and header for image data...
ACK command sent. Waiting for BEGIN command...
Target: BEGIN
BEGIN command received.
100% [ ████████████████████████████████████████████████████████████████ ]
Image data sent...

Waiting for DONE...
Target: DONE
DONE received. All bytes of image data received...
Target: Writing APP to NAND flash
Target: Unprotecting blocks 0x00000019 through 0x00000032.
Target: Number of blocks needed for header and data: 0x0x00000003
Target: Attempting to start in block number 0x0x00000019.
Target: Erasing block 0x00000019 through 0x0000001B.
Target: Writing header and image data to Block 0x00000019, Page 0x00000000
Target: Erasing block 0x0000001C through 0x0000001E.
Target: Writing header and image data to Block 0x0000001C, Page 0x00000000
Target: Erasing block 0x0000001F through 0x00000021.
Target: Writing header and image data to Block 0x0000001F, Page 0x00000000
Target: Erasing block 0x00000022 through 0x00000024.
Target: Writing header and image data to Block 0x00000022, Page 0x00000000
Target: Erasing block 0x00000025 through 0x00000027.
Target: Writing header and image data to Block 0x00000025, Page 0x00000000
Target: Erasing block 0x00000028 through 0x0000002A.
Target: Writing header and image data to Block 0x00000028, Page 0x00000000
Target: Erasing block 0x0000002B through 0x0000002D.
Target: Writing header and image data to Block 0x0000002B, Page 0x00000000
Target: Erasing block 0x0000002E through 0x00000030.
Target: Writing header and image data to Block 0x0000002E, Page 0x00000000
Target: Protecting the entire NAND flash.
Target: DONE
Target: DONE
Operation completed successfully.
```

Note: if you don't see "Operation Completed Successfully" then you need to get the device to get into the "BOOT ME" loop again and run the script again (steps 4-8)

You're not done yet.



### Section 3: Flashing procedure stage 2

1. From there I rinse and repeat but flash another part of the NAND...
2. Plug the TTL serial converter into a USB port on your computer
1. Determine what COM port the device is assigned  
(<https://tnp.uservoice.com/knowledgebase/articles/172101-determining-the-com-port-of-a-usb-to-serial-adapte> )
3. Open up Tera Term or similar terminal application, connect to the TTL serial converter by selecting the serial connection
  1. In Tera Term you goto Setup menu > Serial Port ...
  2. Choose the correct COM port that was assigned to your adapter, and choose "no flow control" and 115200 for the COM port speed
  4. Jumper BSEL and 3V3 pads of the wifi module PC board
  5. Connect the board to the TTL Serial adapter
  1. Note: I had some issues with the board not booting and the terminal program not displaying text from the board if I didn't connect specifically in this order: TTL adapter to USB port, then open Tera Term on the correct serial port, then connect the PC board to the TTL serial adapter (ie power the board)
6. If you did everything correctly you'll see "BOOT ME BOOT ME BOOT ME" start displaying in the terminal window
7. Now leave everything connected and close out Tera Term
8. Open a Command Prompt and navigate to the folder where you put the [Flash utility and binaries for phantom 2 vision plus wifi module firmware](#)
9. Execute this command: **sfh\_DM36x.exe -nandflash -v -p "COMXX" ubl1\_editedByGaucho.img u-boot\_modifiedByGaucho.img**
  1. Note: replace "COMXX" with "COM3" if your device is on COM port #3. Quotes around COM are fine in the command
10. It will start flashing some stuff and likely pop out and start saying "BOOT ME BOOT ME BOOT ME" again. At this point I hit CTRL-C which exits the utility. From there I kicked off the same command you see in step 9, above. Eventually all the flashing will succeed and you'll see something like this:



#### Section 4: Check that you're good to go

1. Disconnect the wifi board from the TTL adapter, disconnect the TTL from the USB port, exit command prompt
2. Un-jumper BSEL from 3v3 on the wifi board (disconnect the two pads)
3. Connect the TTL adapter to the USB port
4. Open Tera Term, connect to the appropriate COM port: Setup > Serial Port > COMXX, Flow control:none, Speed 115200
5. Plug the TTL adapter to the wifi board
6. You should see results like this:

```
DM36x initialization passed!
UBL Product Vesion : DJI-UBL-1.0-rc2
Dji UBL Version: 1.51(Jun 19 2014 - 01:14:42)
Booting Catalog Boot Loader
BootMode = NAND
Starting NAND Copy...
Valid magicnum, 0xA1ACED66, found in block 0x00000019.
    DONE
Jumping to entry point at 0x81080000.
```

```
U-Boot Product Vesion : DJI-Uboot-1.0-rc2
U-Boot 2010.12-rc2-svn539-Dji (Feb 13 2014 - 04:32:46)
Cores: ARM 297 MHz
DDR:    270 MHz
I2C:    ready
DRAM:   128 MiB
NAND:   128 MiB
Bad block table found at page 65472, version 0x01
Bad block table found at page 65408, version 0x01
*** Warning - bad CRC, using default environment
```

```
Net:    Ethernet PHY: GENERIC @ 0xff
DaVinci-EMAC
Hit any key to stop autoboot:  1      0
```

```
Loading from nand0, offset 0x4a0000
Image Name:   Linux-2.6.32.17-davinci1
Created:      2014-04-09  12:21:58 UTC
Image Type:   ARM Linux Kernel Image (uncompressed)
Data Size:    3823424 Bytes = 3.6 MiB
Load Address: 80008000
Entry Point:  80008000
## Booting kernel from Legacy Image at 80700000 ...
Image Name:   Linux-2.6.32.17-davinci1
Created:      2014-04-09  12:21:58 UTC
Image Type:   ARM Linux Kernel Image (uncompressed)
Data Size:    3823424 Bytes = 3.6 MiB
Load Address: 80008000
Entry Point:  80008000
```



Loading Kernel Image ... OK  
OK

Starting kernel ...

```
[ 0.000000] Kernel Product Version : DJI-Kernel-1.0-rc2
[ 0.000000] Linux version 2.6.32.17-davinci1 (root@ubuntu)
(gcc version 4.3.3 (Sourcery G++ Lite 2009q1-203) ) #6 PREEMPT
Wed Apr 9 05:21:55 PDT 2014
[ 0.000000] CPU: ARM926EJ-S [41069265] revision 5
(ARMv5TEJ), cr=00053177
[ 0.000000] CPU: VIVT data cache, VIVT instruction cache
[ 0.000000] Machine: DaVinci DM36x EVM
[ 0.000000] Memory policy: ECC disabled, Data cache
writeback
[ 0.000000] DaVinci dm36x_rev1.2 variant 0x8
[ 0.000000] Built 1 zonelists in Zone order, mobility
grouping off. Total pages: 12192
[ 0.000000] Kernel command line: console=ttyS0,115200n8 rw
dm365_imp.oper_mode=0
video=davincifb:vid0=0,0:vid1=0,0:osd0=0,0:osd1=0,0 mem=48MB
davinci_enc_mgr.ch0_output=COMPOSITE
davinci_enc_mgr.ch0_mode=pal ubi.mtd=2,2048 root=ubi0:rootfs
rootfstype=ubifs ip=off lpj=1077248
[ 0.000000] PID hash table entries: 256 (order: -2, 1024
bytes)
[ 0.000000] Dentry cache hash table entries: 8192 (order:
3, 32768 bytes)
[ 0.000000] Inode-cache hash table entries: 4096 (order: 2,
16384 bytes)
[ 0.000000] Memory: 48MB = 48MB total
[ 0.000000] Memory: 44784KB available (3456K code, 310K
data, 112K init, 0K highmem)
[ 0.000000] SLUB: Genslabs=11, HWalign=32, Order=0-3,
MinObjects=0, CPUs=1, Nodes=1
[ 0.000000] Hierarchical RCU implementation.
[ 0.000000] NR_IRQS:245
[ 0.000000] Console: colour dummy device 80x30
[ 0.000000] Calibrating delay loop (skipped) preset value..
215.44 BogoMIPS (lpj=1077248)
[ 0.000000] Mount-cache hash table entries: 512
[ 0.000000] CPU: Testing write buffer coherency: ok
[ 0.000000] DaVinci: 8 gpio irqs
[ 0.000000] NET: Registered protocol family 16
[ 0.090000] EVM: tvp5146 SD video input
[ 0.160000] bio: create slab <bio-0> at 0
[ 0.170000] DM365 IPIPE initialized in Continuous mode
[ 0.170000] usbcore: registered new interface driver usbfs
[ 0.170000] usbcore: registered new interface driver hub
[ 0.170000] usbcore: registered new device driver usb
[ 0.180000] vpss vpss: dm365_vpss vpss probed
```

```

[ 0.180000] vpss vpss: dm365_vpss vpss probe success
[ 0.180000] dm365_afew_hw_init
[ 0.180000] ch0 default output "COMPOSITE", mode "PAL"
[ 0.180000] VPBE Encoder Initialized
[ 0.190000] cfg80211: Using static regulatory domain info
[ 0.190000] cfg80211: Regulatory domain: US
[ 0.190000] (start_freq - end_freq @ bandwidth),
(max_antenna_gain, max_eirp)
[ 0.190000] (2402000 KHz - 2472000 KHz @ 40000 KHz), (600
mBi, 2700 mBm)
[ 0.190000] (5170000 KHz - 5190000 KHz @ 40000 KHz), (600
mBi, 2300 mBm)
[ 0.190000] (5190000 KHz - 5210000 KHz @ 40000 KHz), (600
mBi, 2300 mBm)
[ 0.190000] (5210000 KHz - 5230000 KHz @ 40000 KHz), (600
mBi, 2300 mBm)
[ 0.190000] (5230000 KHz - 5330000 KHz @ 40000 KHz), (600
mBi, 2300 mBm)
[ 0.190000] (5735000 KHz - 5835000 KHz @ 40000 KHz), (600
mBi, 3000 mBm)
[ 0.190000] cfg80211: Calling CRDA for country: US
[ 0.190000] LogicPD encoder initialized
[ 0.190000] Switching to clocksource timer0_1
[ 0.200000] musb_hdrc: version 6.0, cppi-dma, host, debug=0
[ 0.220000] musb_hdrc: USB Host mode controller at fec64000
using DMA, IRQ 12
[ 0.220000] musb_hdrc musb_hdrc: MUSB HDRC host driver
[ 0.220000] musb_hdrc musb_hdrc: new USB bus registered,
assigned bus number 1
[ 0.220000] usb usb1: configuration #1 chosen from 1 choice
[ 0.220000] hub 1-0:1.0: USB hub found
[ 0.220000] hub 1-0:1.0: 1 port detected
[ 0.220000] NET: Registered protocol family 2
[ 0.220000] IP route cache hash table entries: 1024 (order:
0, 4096 bytes)
[ 0.220000] TCP established hash table entries: 2048
(order: 2, 16384 bytes)
[ 0.220000] TCP bind hash table entries: 2048 (order: 1,
8192 bytes)
[ 0.220000] TCP: Hash tables configured (established 2048
bind 2048)
[ 0.220000] TCP reno registered
[ 0.220000] NET: Registered protocol family 1
[ 0.220000] RPC: Registered udp transport module.
[ 0.220000] RPC: Registered tcp transport module.
[ 0.220000] RPC: Registered tcp NFSv4.1 backchannel
transport module.
[ 0.230000] JFFS2 version 2.2. (NAND) ?? 2001-2006 Red Hat,
Inc.
[ 0.230000] msgmni has been set to 87
[ 0.240000] alg: No test for stdrng (krng)

```

```

[    0.240000] io scheduler noop registered (default)
[    0.320000] DM365 IPIPEIF probed
[    0.320000] imp serializer initialized
[    0.320000] davinci_previewer initialized
[    0.330000] davinci_resizer initialized
[    0.330000] Serial: 8250/16550 driver, 2 ports, IRQ sharing
disabled
[    0.330000] serial8250.0: ttyS0 at MMIO 0x1c20000 (irq =
40) is a 16550A
[    0.740000] console [ttyS0] enabled
[    0.750000] serial8250.0: ttyS1 at MMIO 0x1d06000 (irq =
41) is a 16550A
[    0.760000] brd: module loaded
[    0.770000] NAND device: Manufacturer ID: 0x2c, Chip ID:
0xf1 (Micron NAND 128MiB 3,3V 8-bit)
[    0.780000] Creating 4 MTD partitions on "davinci_nand.0":
[    0.780000] 0x000000000000-0x0000004a0000 : "bootloader"
[    0.790000] 0x0000004a0000-0x000000e00000 : "kernel"
[    0.800000] 0x000000e00000-0x0000007f0000 : "filesystem"
[    0.810000] 0x000000000000-0x000000800000 : "all"
[    0.820000] davinci_nand davinci_nand.0: controller rev.
2.3
[    0.830000] UBI: attaching mtd2 to ubi0
[    0.830000] UBI: physical eraseblock size:    131072 bytes
(128 KiB)
[    0.840000] UBI: logical eraseblock size:      126976 bytes
[    0.850000] UBI: smallest flash I/O unit:      2048
[    0.850000] UBI: sub-page size:                512
[    0.850000] UBI: VID header offset:            2048 (aligned
2048)
[    0.860000] UBI: data offset:                  4096
[    1.390000] UBI: attached mtd2 to ubi0
[    1.400000] UBI: MTD device name:              "filesystem"
[    1.400000] UBI: MTD device size:              113 MiB
[    1.410000] UBI: number of good PEBs:          904
[    1.410000] UBI: number of bad PEBs:           0
[    1.420000] UBI: max. allowed volumes:         128
[    1.420000] UBI: wear-leveling threshold:      4096
[    1.430000] UBI: number of internal volumes:    1
[    1.430000] UBI: number of user volumes:        1
[    1.440000] UBI: available PEBs:                0
[    1.440000] UBI: total number of reserved PEBs: 904
[    1.450000] UBI: number of PEBs reserved for bad PEB
handling: 9
[    1.450000] UBI: max/mean erase counter: 2/1
[    1.460000] UBI: image sequence number: 465088957
[    1.460000] UBI: background thread "ubi_bgt0d" started, PID
308
[    1.470000] console [netcon0] enabled
[    1.470000] netconsole: network logging started
[    1.480000] i2c /dev entries driver

```

```
[ 1.480000] Linux video capture interface: v2.00
[ 1.490000] vpfe_init
[ 1.490000] vpfe-capture: vpss clock vpss_master enabled
[ 1.500000] vpfe-capture vpfe-capture: v4l2 device
registered
[ 1.510000] vpfe-capture vpfe-capture: video device
registered
[ 1.550000] EVM: switch to tvp5151 cvbs video input
[ 1.550000] vpfe-capture vpfe-capture: v4l2 sub device
tvp5150 registered
[ 1.560000] EVM: switch to it6604 hdmi video input
[ 1.570000] vpfe-capture vpfe-capture: v4l2 sub device
cat6023 register fails
[ 1.570000] vpfe_register_ccdc_device: DM365 ISIF
[ 1.580000] DM365 ISIF is registered with vpfe.
[ 1.590000] Trying to register davinci display video
device.
[ 1.590000] layer=c1067800,layer->video_dev=c1067964
[ 1.600000] Trying to register davinci display video
device.
[ 1.600000] layer=c1067c00,layer->video_dev=c1067d64
[ 1.610000] davinci_init:DaVinci V4L2 Display Driver V1.0
loaded
[ 1.620000] watchdog watchdog: heartbeat 1 sec
[ 1.620000] TCP cubic registered
[ 1.630000] NET: Registered protocol family 17
[ 1.630000] lib80211: common routines for IEEE802.11
drivers
[ 1.640000] Clocks: disable unused mmcsd0
[ 1.640000] Clocks: disable unused mmcsd1
[ 1.640000] Clocks: disable unused spi0
[ 1.640000] Clocks: disable unused spi1
[ 1.640000] Clocks: disable unused spi2
[ 1.640000] Clocks: disable unused spi3
[ 1.640000] Clocks: disable unused spi4
[ 1.640000] Clocks: disable unused pwm0
[ 1.640000] Clocks: disable unused pwm1
[ 1.640000] Clocks: disable unused pwm2
[ 1.640000] Clocks: disable unused pwm3
[ 1.640000] Clocks: disable unused timer1
[ 1.640000] Clocks: disable unused timer3
[ 1.640000] Clocks: disable unused emac
[ 1.640000] Clocks: disable unused voice_codec
[ 1.640000] Clocks: disable unused asp0
[ 1.640000] Clocks: disable unused rto
[ 1.640000] Clocks: disable unused mjcp
[ 1.710000] nnnn===0
[ 1.720000] davinci_emac_probe: using random MAC addr:
06:aa:d2:d3:a4:79
[ 1.730000] emac-mii: probed
[ 1.770000] UBIFS: recovery needed
```

```
[ 1.810000] UBIFS: recovery completed
[ 1.820000] UBIFS: mounted UBI device 0, volume 0, name
"rootfs"
[ 1.820000] UBIFS: file system size: 110342144 bytes
(107756 KiB, 105 MiB, 869 LEBs)
[ 1.830000] UBIFS: journal size: 9023488 bytes (8812
KiB, 8 MiB, 72 LEBs)
[ 1.840000] UBIFS: media format: w4/r0 (latest is
w4/r0)
[ 1.840000] UBIFS: default compressor: lzo
[ 1.850000] UBIFS: reserved for root: 0 bytes (0 KiB)
[ 1.860000] VFS: Mounted root (ubifs filesystem) on device
0:13.
[ 1.860000] Freeing init memory: 112K
Runing /etc/init.d/rcS
/opt/dji/bin/sshenable: line 2: ./eth0config: not found
sulogin: no password entry for root
login[413]: root login on 'ttyS0'
#Process /etc/profile...
root@Dji-Pro:~#
```

**But if you see the below results, the NAND is still corrupt:**

```
DM36x initialization passed!
UBL Product Vesion : DJI-UBL-1.0-rc2
Dji UBL Version: 1.51(Jun 19 2014 - 01:14:42)
Bootimg Catalog Boot Loader
BootMode = NAND
Starting NAND Copy...
Valid magicnum, 0xA1ACED66, found in block 0x00000019.
Valid magicnum, 0xA1ACED66, found in block 0x0000001B.
Valid magicnum, 0xA1ACED66, found in block 0x0000001D.
Valid magicnum, 0xA1ACED66, found in block 0x0000001F.
No valid boot image found!
NAND Boot failed.
Aborting...
```

### **Section 5: desolder, reassemble and go fly**

1. If you have the good results, boot success on the board, then
2. desolder everything,
3. reassemble the wifi module - it doesn't hurt to use some good heatsink paste when reassembling
4. put it into the bird,
5. fire it up
6. go fly!



I hope this was helpful. I'm OKSTUV on the phantompilots.com forums, but all the hard work was done by: Gaucho, andrew\_by, and rmhome on the PhantomPilots.com "[lightbridge firmware PROBLEM \(and solution\) - firmware version not found by dji assistant tool](#)" thread.

If these instructions were helpful and saved you some money, please feel free to **donate** a little something as a way to say thanks:

<https://www.paypal.me/TGutmann>



Made with the new Google Sites, an effortless way to create beautiful sites.

[Report abuse](#)

