Manufacturing Test Tool User Guide

THIS DOCUMENT IS A RESTRICTED DOCUMENT.

ONLY SPECIFICALLY AUTHORISED PEOPLE ARE PERMITTED TO SEE IT.

IF IN DOUBT, ASK AN OFFICER OF DISPLAYLINK FOR GUIDANCE

This document's classification is:

Customers Only

This means: It has been cleared for release outside the company strictly to recipients who are listed on a signed and agreed Non Disclosure Agreement with DisplayLink which is in force at the time of release by DisplayLink.

If you are not such a person please RETURN THIS DOCUMENT FORTHWITH to:

DisplayLink UK Limited
140 Cambridge Science Park
Milton Road
Cambridge CB4 0GF
UK

Document Analysis

Title	Manufacturing Test Tool User Guide
Document number	NR-113872-TM
Date produced	19 May 2017
Version	Issue 30

Contents

1. Introduction

This document describes how to use the manufacturing test tool (MTT) created by DisplayLink. It allows a customer to write their own test and programming tool for their PCB. The MTT can be used with the DL-3xxx, DL-41xx, DL5xxx and DL-6xxx ASIC based device.

The MTT can also be used to program the SPI Flash for the DL-3xxx, DL-41xx, DL5xxx and DL-6xxx ASIC. The MTT cannot be used for initial programming, as the ASIC will not boot from an empty Flash device.

Throughout this manual, the device being tested is referred to as the UUT (unit under test).

This User Guide applies to version R8.0M2 or later of the MTT.

2. Installation

The Manufacturing Test Tool zip file can be found at NR-130703-LS. Unzip the files and put them all into the same directory. The files include:

- **dl_mfg_test.exe** – the command-line executable of the test tool.
- **example.bat,**
  example_DL3950.bat
  example_DL3xxx_DL5xxx.bat,
  example_DL6xxx.bat
  example_all_DL3xxx_DL5xxx.bat,
  example_DL3xxx_DL5xxx_tests.bat – example batch files which calls the command-line executable.
- **HourGlass24.bmp, leaves.bmp** – example bitmap file to be displayed during the tests.

Make sure that the board under test is already programmed with an appropriate firmware image and config file.

3. Preparing to use the Manufacturing tool

3.1. Windows 7 – Windows 10 Threshold: Stop the DisplayLink Service

Do not run the DisplayLink service at the same time as the Manufacturing Test Tool.

If the DisplayLink service is running, it may:

- automatically update the firmware on the device;
- prevent the MTT from running some tests.

3.2. Windows 10 Anniversary update onwards: Install MTT USB driver

From Windows 10 Anniversary update, the DisplayLink video driver is a Windows user mode driver. To be able to use the Manufacturing Test Tool on Windows 10 Anniversary update onwards, a different USB driver (USBIO) must be installed. The DisplayLink USBIO driver is available from the DisplayLink extranet as document number NR-132500-LS.

The DisplayLink USB device must be updated to use USBIO to run the Manufacturing tests using the Windows Device Manager, but this will prevent the display output appearing as part of the Windows desktop. Once the Manufacturing Tool tasks are complete, the USB driver can be switched back to the DisplayLink video driver in Windows Device Manager.

To use the USBIO drivers, using the Windows Device Manager:

1. Select “View->Show Hidden Devices”
2. In the “Display Adapters” section, there will be several entries related to the DisplayLink product (2 for a single device product, 4 for a dual device product). Right-click on one of these entries, and select “Update driver”.
3. Select “Browse my computer for driver software.”
4. Select “Let me pick from a list of available drivers on my computer.”
5. Click the “Have Disk” button.
6. Browse to the directory where you extracted the USBIO drivers, and select “DisplayLinkUSBIOMTT.inf”, then click “Next”.
7. Repeat steps 2-6 for the other entry in the “Display Adapters” section.
8. Unplug and re-plug the USB connection of the device.
9. Use the Manufacturing Test Tool to put the device into MTT mode, and repeat steps 1 to 8.

To revert to using the standard IDD drivers, uninstall all the entries in the “USB Display Adapters” section of the Windows Device Manager, then unplug and re-plug the USB connection.

This switching process can be automated through scripts. Please contact your DisplayLink FAE for more information.

3.2.1. Using the Manufacturing Test Tool on the Production Line with Windows 10 Anniversary Update onwards

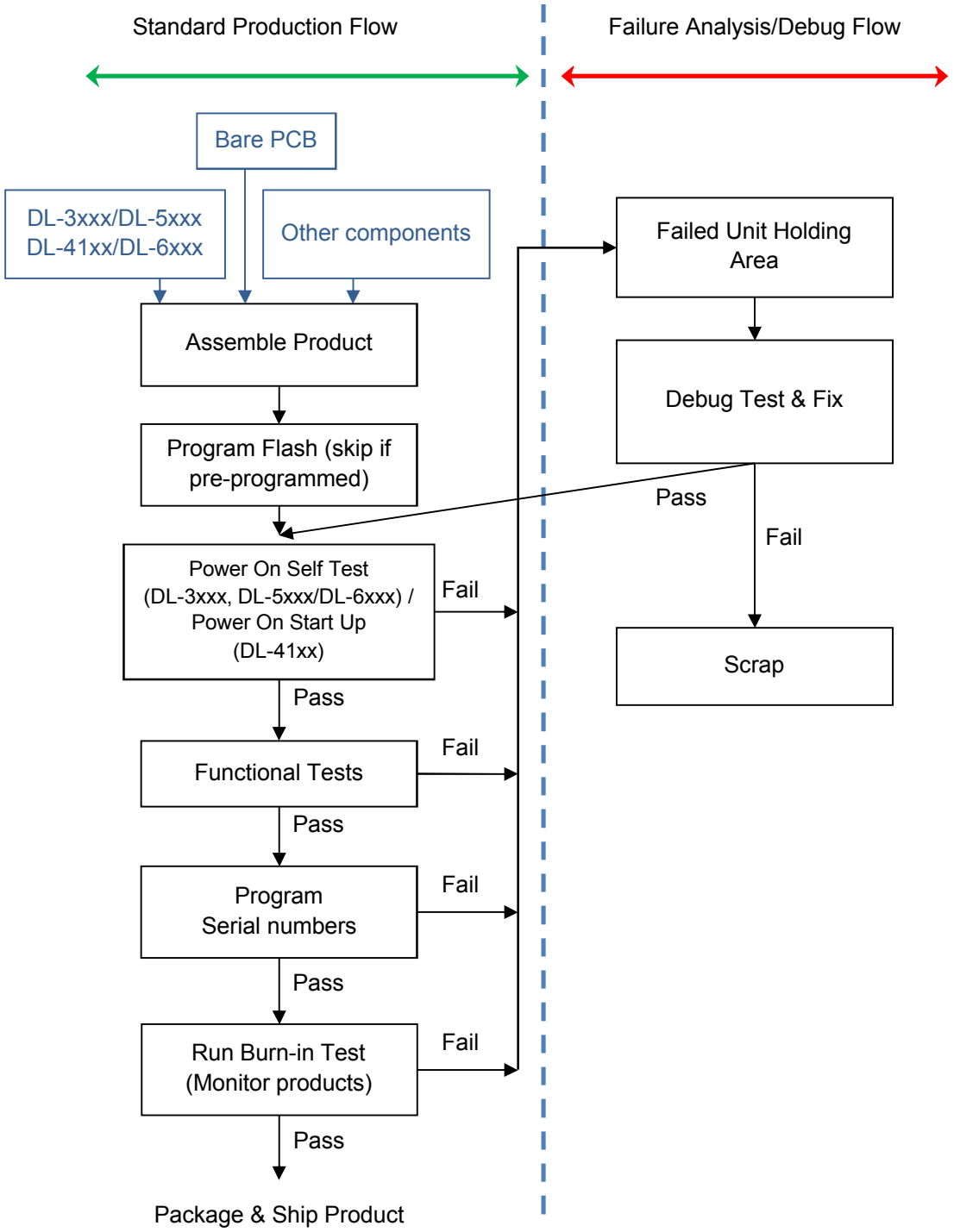
If there is no requirement to display the Windows desktop, for example on a production line, you don’t need to install the DisplayLink host software package with the graphics driver; it is only necessary to install the USBIO drivers. This means that you won’t need to switch drivers using the Windows device manager. We recommend you do not connect the computer to the Internet to avoid Windows Update installing the video driver.

3.2.2. Future versions

The solutions described above are expected to be temporary. A future release of the Manufacturing Test Tool is planned to rely on the graphics driver introduced with Windows 10 Anniversary Update, without the need to switch drivers.

4. Production Line Flow

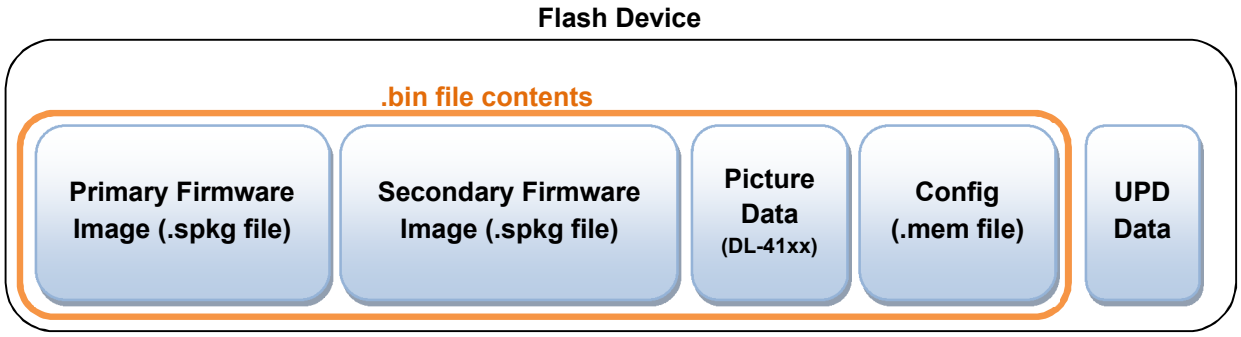
The MTT has been designed for the following production line flow. The whole test process should take around 15 seconds per board if the Flash is pre-programmed (could be quicker or slower, depending on which tests are run).



5. Flash Areas

The Flash contains the following areas:

- Primary firmware image (.spkg file)
- Secondary firmware image (.spkg file)
- Picture data (splash and no-USB pictures, .mem file) (only for DL-41xx)
- Config (.mem file)
- UPD (unique per-device) data



Firmware is delivered to customers from the firmware web builder. This generates a **.bin** file, containing the primary and secondary firmware images and the config, but no UPD data. This .bin file must be programmed using an external programmer when the board is manufactured.

- The *primary firmware image* is the one that is usually used by the device. The host software will automatically update the primary firmware image, to ensure compatibility between the host software and the firmware. It can also be upgraded over USB using the MTT if required.
- The *secondary / backup firmware image* is only used if the primary image is corrupted by an interruption during firmware upgrade. It cannot be changed by the host software. It can only be changed either by the MTT or by reprogramming the whole Flash with a new .bin file.
- The *Picture data* is used to hold contain background tiles and foreground images for the splash and no-USB images. It can only be changed by the MTT or by reprogramming the whole Flash with a new bin file.
- The *config* (.mem) file generated from the web builder can be upgraded separately over USB, using the MTT. However, the config must remain compatible with the primary and secondary firmware images. To ensure this, **it is safest to reprogram the whole .bin file (from the web builder) when updating a config.**
- The *UPD data* is programmed using the MTT. It can contain the USB serial number, MAC address (DL-3xxx/DL-5xxx/DL-6xxx) and/or the EDID (for embedded monitor products) and gamma curve lookup tables (LUT) (DL-41xx)

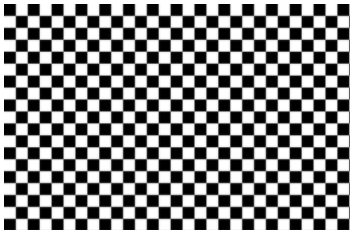
6. Power-On Self Test (POST) (DL-3xxx/DL-5xxx/DL-6xxx only)

At power-on, the device will check the Flash for a serial number. If there is no serial number in Flash, the device will run a power-on self test (POST) of the UUT, and then, DL-3xxx/DL-5xxx/DL-6xxx will enter video burn-in mode.[1]

The initial .bin file from the web builder does not contain a serial number in UPD, so a newly-assembled device will always start in manufacturing test mode. To work in normal mode, you must program the UPD serial number, using the 'upd' command.

The burn-in test cycles through a fixed sequence of test patterns: a full screen of red, green, blue, white, then black.

The burn-in test will continue until an MTT command is executed, after which a checkerboard test pattern is displayed.



The POST consists of:

- **DDR data and address test**: check the pins for opens and shorts.
- **Video test**: display checkerboard pattern on all configured outputs. If device has an embedded EDID, the display resolution will be taken from the EDID data. Otherwise, a default display resolution of 1024x768@60 will be set.
- **Ethernet loopback** test: if the firmware config. includes Ethernet, send a test packet over Ethernet and check that it is received back correctly.
- **Audio loopback** test: if the firmware config. includes an audio input (microphone) and an audio output (headphones), loop all audio received back to the audio output.
- The Flash and ARC processor are also implicitly tested.

This allows basic functional testing before connecting USB.

[1] DL-6xxx devices will also start in MTT mode if there is no OTP data (blank OTP) on the device. Use the 'info otp' MTT command to verify the OTP data. If the OTP is blank, then the device must be returned to DisplayLink.

Connections

The video test assumes that monitors are connected to all configured video outputs. The video test pattern cannot be automatically verified; a tester must check that the display is correct.

The Ethernet loopback test assumes that an Ethernet loopback connector is fitted, which shorts together the data pins such that any packet sent out will be received back again.

To use the audio loopback test, connect a microphone and speakers (or a headset). Speak into the microphone, and check that the speakers play the sound correctly.

POST results

The results of the POST are available in two ways:

- Via a USB command (see ‘post’ command in section 8).
- Via a configurable POST LED. The POST LED will turn on if the DDR or Ethernet POST tests fail.

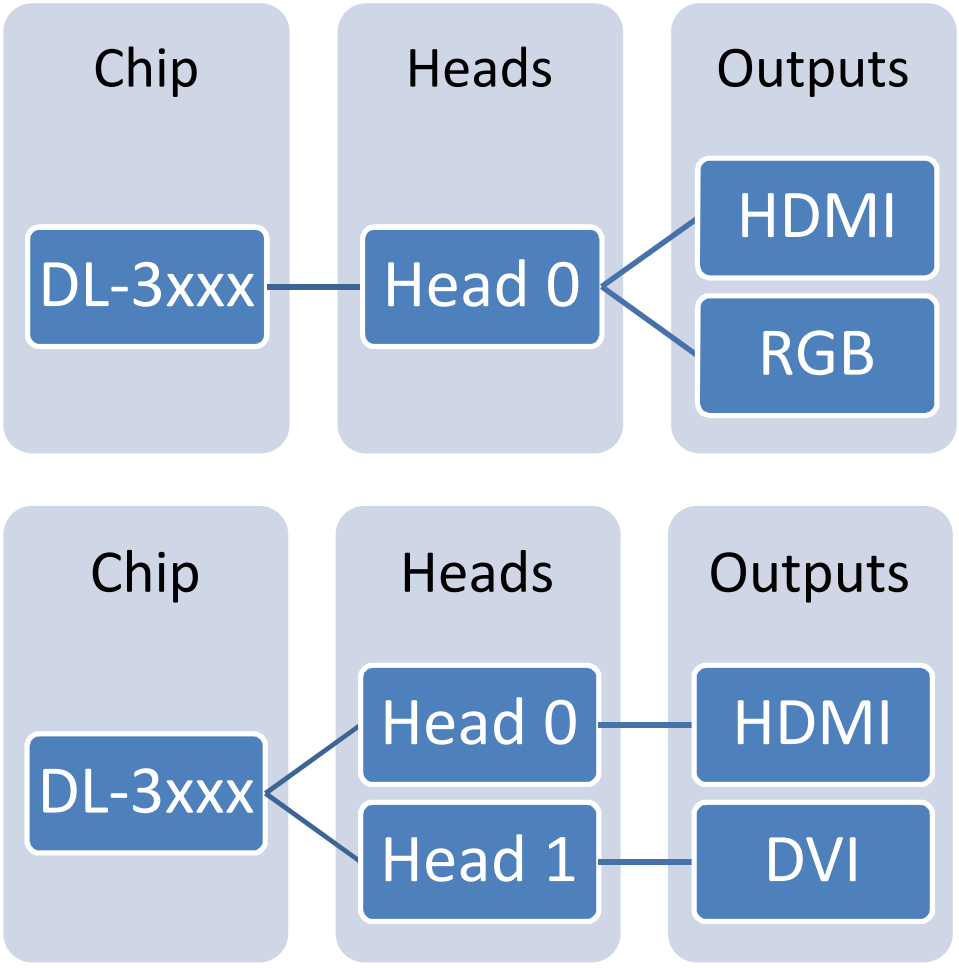
7. Video output configuration

The MTT can display various test patterns on the video outputs. It can also set the video mode and configure the dual-head operation.

DL-3xxx/DL-5xxx/DL-6xxx devices can support dual-head video. This means that two video outputs can be enabled at the same time, with the same or different images.

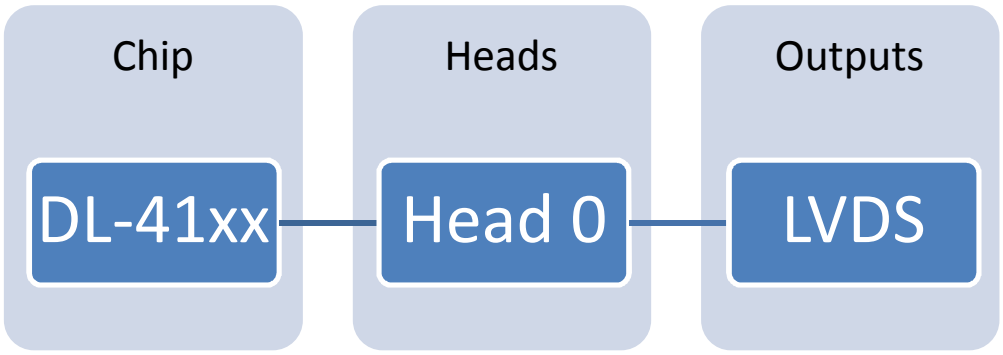
The image sources are called ‘heads’. For DL-3xxx and DL-5xxx, each video output (HDMI, DVI, DAC, etc.) can be assigned to head 0, head 1, or no head.

Example configurations:

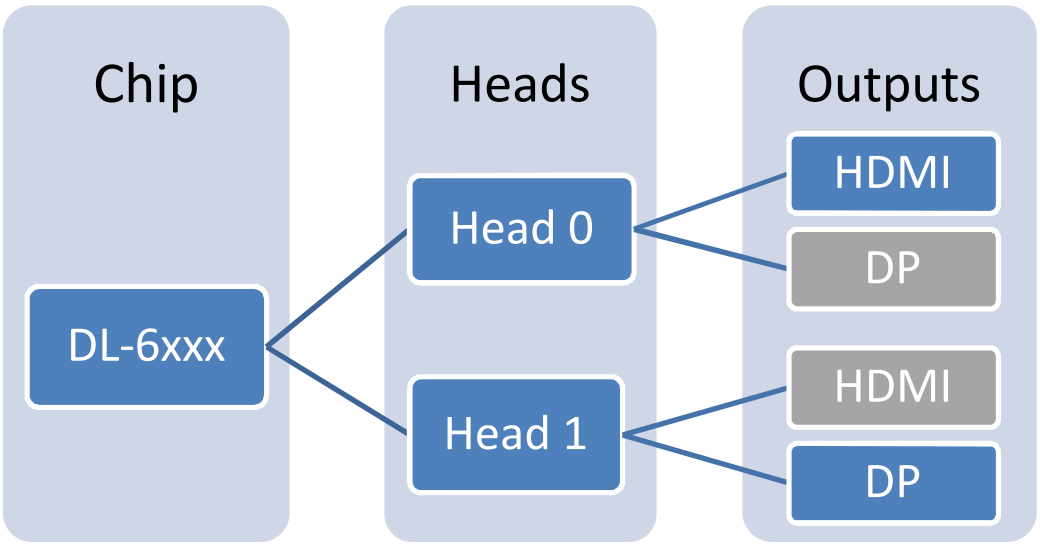


DL-41xx devices only support single-head video. This means that only one video output is available. The video output (LVDS) is assigned to head 0.

Configuration:



DL-6xxx devices can support dual-head video. This means that two video outputs can be enabled at the same time, with the same or different images. Only one output at a time can be active per head.



8. DL-3xxx/DL-41xx/DL-5xxx/DL-6xxx Command Line Tests

After the POST or Power On Start Up has run, customers can (optionally) test the UUT more thoroughly by running functional tests over USB, using the command line interface for the MTT. To run these tests, the UUT must be connected to a test PC via USB.

If communication with the UUT fails at any point, the message “A device in mtt mode could not be located” will be displayed.

List MTT Commands

Syntax: **dl_mfg_test --help**
dl_mfg_test -h

Displays all the available command-line options. Does not attempt to communicate with the UUT.

Version Number

Syntax: **dl_mfg_test --version**
dl_mfg_test -v

Prints out the MTT version number. Does not attempt to communicate with the UUT.

Example output:

```
# dl_mfg_test -v
DisplayLink manufacturing tests - version 7.1
```

Manufacturing Test Mode

Syntax: **dl_mfg_test mtt**

If there is no serial number in Flash, the UUT will automatically start in MTT mode. If a serial number has been programmed into Flash, the UUT can still be put into MTT mode by running the ‘mtt’ command. This will send a message to the UUT to make it restart in Manufacturing Test Mode.

Also for DL6xxx SKU types, if there is no OTP data, the UUT will automatically start in MTT mode.

Normal Mode

Syntax: **dl_mfg_test reboot**

Reboots device and returns to normal operating mode. Note that this will not “force” normal mode, so the device will still enter MTT mode if its condition demands it – for example, if the UPD is not programmed.

Handling Multiple Devices

If multiple devices are connected to the test PC, they can be individually accessed by MTT commands using the --***dev*** argument. Only one C*ompound Device* may be attached, however – see next page for more details about Compound Devices.

List all connected devices

Syntax: **dl_mfg_test --dev list**

Example output (with two devices connected):

```
# dl_mfg_test --dev list
4300_HWPE001ESDB-014
4307_000800110011117
```

Run command on named device

Syntax: **dl_mfg_test <test command & arguments> test --dev <dev_name>**
<dev_name> = USB device ID, followed by _ (underscore), followed by USB serial number (as seen in output from "--dev list" command).

Example:

```
dl_mfg_test info flash --dev C300_M00000048
```

Device name changes in MTT mode

When a device is put into MTT mode, its device name will change, as the serial number from Flash is ignored, and the chip serial number is used as the USB serial number instead – example:

```
# dl_mfg_test --dev list
4300_HWPE001ESDB-014            [Not in MTT mode]
4307_000800110011117            [Not in MTT mode]

# dl_mfg_test --dev 4300_HWPE001ESDB-014 mtt

# dl_mfg_test --dev list
4300_HWPE001DL5900-00.283.B4D   [In MTT mode]
4307_000800110011117            [Not in MTT mode]
```

Video Tests

Configure the video mode

(**--head** option only available for DL-3xxx/DL-5xxx/DL-6xxx)

Syntax: **--mode <width> <height> <refresh> [--head <head>]**
-m <width> <height> <refresh> [--head <head>]

<width>	The width of the display in pixels
<height>	The height of the display in pixels
<refresh>	The refresh rate in Hz

Syntax: **--detailedmode <htotal > <haddr> <hsync> <hfrontporch> <hpol> <vtotal> <vaddr> <vsync> <vfrontporch> <vpol> <clock> [--head <head>]**

<htotal>	The total horizontal time
<haddr>	The horizontal addressable video time
<hsync>	The horizontal sync width
<hfrontporch>	The horizontal front porch
<hpol>	The horizontal polarity: 1 for positive, 0 for negative
<vtotal>	The total vertical time
<vaddr>	The vertical addressable video time
<vsync>	The vertical sync width
<vfrontporch>	The vertical front porch
<vpol>	The vertical polarity: 1 for positive, 0 for negative
<clock>	The pixel clock (kHz)

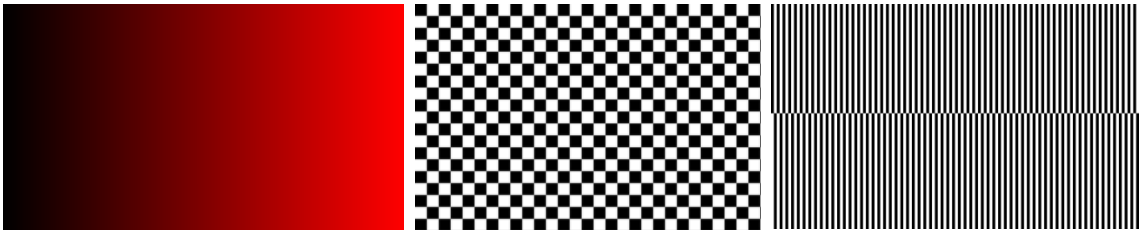
The MTT will check that the resolution requested is supported by the UUT connected and (for the 'mode' command) that the mode requested is a standard mode.

If the mode is supported, the video registers are configured to give the specified resolution.
This can be used on its own or as a switch with a video test pattern or bitmap command (see below).

Note: DL-6xxx devices support 4k resolution, but only at a refresh rate of 30 Hz in manufacturing test mode.

N.B. If the UUT has an embedded EDID, the EDID resolution will be used as the default resolution for the MTT.

Test patterns



Syntax	Test
dl_mfg_test video cr r	Red colour ramps repeating across the screen
dl_mfg_test video cr g	Green colour ramps repeating across the screen
dl_mfg_test video cr b	Blue colour ramps repeating across the screen
dl_mfg_test video gr 1	Grey ramps repeating left to right
dl_mfg_test video gr 2	Grey ramps repeating top to bottom
dl_mfg_test video bw 1	Black and white vertical bars, with opposite phase for the bottom half of the screen
dl_mfg_test video bw 2	Black and white checkerboard
dl_mfg_test video black	All black pixels
dl_mfg_test video white	All white pixels

This test is designed for visual inspection for pass/fail. Also, if there is an error communicating with the UUT, 1 is returned. Note that these tests are not affected by changes in the gamma curve data in the UPD, nor are they affected by changes to contrast or dither.

Note
- for DL-3xxx/DL-5xxx devices: All test patterns are sent to head 0. Head 1 does not support test patterns, only bitmap images.
- for DL-6xxx devices, the test pattern is sent to both heads, and will appear on all currently active outputs.

Display a bitmap

Syntax: **dl_mfg_test video bitmap <filename> [--head <head>]***

<filename> The bitmap filename, including path if necessary
<head> The head which should display the bitmap; not supported for DL-41xx devices

Displays a bitmap in the centre of the display, surrounded by black. If the bitmap is too large for the display, the edges will be cropped. The bitmap file must be a valid .BMP file whose colour depth is 24 bits per pixel.

This test is designed for visual inspection for pass/fail. The --mode and --head switches are optional. If the display mode is not specified, the current display mode will be used. If the head is not specified, head 0 is the default.

Examples:

Display image on head 0 at 1024x768@60:
```
# dl_mfg_test video bitmap leaves.bmp -m 1024 768 60 --head 0
```

Display image on head 1 at current resolution:
```
# dl_mfg_test video bitmap leaves.bmp --head 1
```

Display image on head 0 at current resolution:
```
# dl_mfg_test video bitmap leaves.bmp
```

Display an image that is too small:
```
# dl_mfg_test.exe video bitmap dl\image\ChristmasTree.bmp
Image is 640x480, display is 1366x768
Image is too small, centering on screen
```

Display an image that is too large:
```
# dl_mfg_test.exe video bitmap dl\image\GreenDotsFullscreen.bmp
Image is 1920x1080, display is 1366x768
Image is too wide and too tall, clipping to 1366x768
```

Adjust the screen resolution to match the image:
```
# dl_mfg_test.exe video --mode 640 480 60 bitmap dl\image\ChristmasTree.bmp
Image is 640x480, display is 640x480
```

Firmware Information

Syntax: **dl_mfg_test info firmware**

Reports the firmware version of the primary firmware image in Flash. If 'M' appears at the end, this is a development firmware build. It also reports whether the device can boot from DDR.

Example output:
```
# dl_mfg_test info firmware
Firmware Version: 6.1.8.44268
DDR upgrade support: Enabled
```

Backup Firmware Information

Syntax: **dl_mfg_test info backup**

Reports the firmware version of the backup firmware image in Flash. If 'M' appears at the end, this is a development firmware build.

Example output:
```
# dl_mfg_test info backup
Backup Firmware Version: 6.1.7.44043
```

OTP Information

Syntax: **dl_mfg_test info otp**

Reports public data stored in the OTP (one-time programmable) memory.

Example output:
```
# dl_mfg_test info otp
OTP Serial Number: 0001234
OTP SKU: 00001
OTP MAC: 00:0F:9D:00:05:C3
```

HDCP Information

Syntax: **dl_mfg_test info hdcp**

Displays the following information from the device:
- Whether HDCP is enabled in config
- Whether HDCP is enabled for HDMI outputs.
- Whether HDCP is enabled for DisplayPort outputs.
- Whether HDCP2.x is supported on the upstream link.

Output types that are not supported on a given device will report disabled.

Example output:

```
# dl_mfg_test info hdcp
HDCP status:
-------------
Configuration:      enabled
HDMI:               enabled
DisplayPort:        enabled
Upstream HDCP2.x:   disabled
```

If the firmware build doesn't report HDCP status (older firmware versions won't), the command will be rejected as below:

```
# dl_mfg_test info hdcp
HDCP status not reported by this firmware release
```

When interpreting the data above please note:

- Older firmware does not support report Upstream HDCP2.x and will report disabled.
- The Configuration field reports whether SKU and Config supports HDCP and if keys are present.
- DL-6xxx may report Configuration enabled even if -N part.
- DisplayPort refers to external DP501 device on DL-3xxx and DL-5xxx; on DL-6xxx the internal output.
- On DL-3xxx and DL-5xxx, DisplayPort status is independent of the other fields – because the fitted external DP501 may or may not have HDCP capability (we noted some products which expect HDCP capability have DP501 parts that do not support HDCP, and vice-versa).
- For HDCP2.x support, Upstream HDCP2.x must report enabled.

Flash Information

Syntax: **dl_mfg_test info flash**

Displays the following information from Flash memory:

- primary image firmware version
- backup image firmware version
- configuration information
- UPD (Unique Per-Device) data: serial number, MAC address and EDID header (if present)

Example output:

```
# dl_mfg_test info flash
Firmware Type:      FflyMoni
Release Level:      0
Version:            7.1.11
SVN Version:        50384M
Build Date:         2012-11-26
Time:               16:38:46
Backup FW Version: 7.1.10

*** Encrypted configuration ***
Name:               DisplayLink_ChiM
```

```
Date:               Fri 16 Nov 2012
Min FW Version:     7.1.6

*** UPD Data ***
USB Serial No:      E00021
HDCP Flags:         0
MAC address:        00:0F:9D:00:05:C3

Active Embedded EDID (from UPD)...
00 FF FF FF FF FF FF 00
0D AF 90 15 34 12 00 00
1F 13 01 03 80 22 13 78
0A D1 F5 93 5D 59 90 26
1D 50 54 21 08 00 81 C0
01 01 01 01 01 01 01 01
01 01 01 01 01 01 60 1D
56 BC 50 00 26 30 1E 3F
4C 00 58 C2 10 00 00 18
00 00 00 FE 00 4E 31 35
36 42 36 2D 4C 30 44 0A
20 20 00 00 00 FE 00 43
4D 4F 0A 20 20 20 20 20
20 20 20 20 00 00 00 FF
00 31 32 33 34 0A 20 20
20 20 20 20 20 20 00 63

EDID serial number : 0x00001234 (4660)
EDID serial string : ChiMei panel Multi 12345
EDID mfg year:week : 2009:31
```

Video information

Syntax: **dl_mfg_test info video**

For DL-41xx it displays the preferred video mode indicated by the current active EDID, and the video mode as last set by a **--mode**, **--detailedmode** or **video rgb** command. For DL-3xxx/DL-5xxxxx/DL-6xxx it displays current video mode for all heads. Note that the "preferred mode" is the "post-scaler" mode, the native mode for the panel.

Example output (DL-41xx):

```
# dl_mfg_test info videoActive EDID prefers Video Mode 1366x768 @ 60Hz
(1554,1366,63,30,0,806,768,12,4,0,75200)
Currently using Video Mode 640x480 @ 60Hz (800,640,96,8,0,525,480,2,2,0,25180)
```

Example output (DL-3xxx/DL-5xxx/DL-6xxx):

```
# dl_mfg_test info video
Head 0:
Currently using Video Mode 640x480 @ 60Hz (800,640,96,8,0,525,480,2,2,0,25180)
Head 1:
Currently using Video Mode 1366x768 @ 60Hz (1554,1366,63,30,0,806,768,12,4,0,75200)
```

The video mode given (1366x768 @ 60Hz) corresponds to the values needed for a "**--mode**" command for the same effect. The values in parentheses correspond to the values needed for a "**--detailed**" command for the same effect.

EDID information

Syntax: **dl_mfg_test info edid**

For DL-41xx devices, it displays the content of the current active EDID. For DL-3xxx/DL-5xxx/DL-6xxx devices it displays EDID from all available outputs.

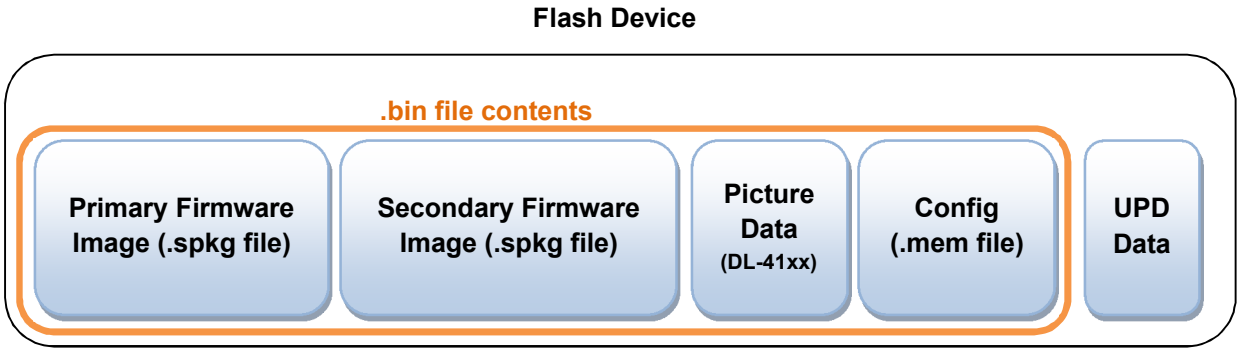
Example output:

```
# dl_mfg_test info edid

Active Embedded EDID (from UPD)...

EDID DATA (v1.3)
=========
Monitor Type : CMO1590 ()
Monitor ID   : 1234
Manufactured : 2009.31
Serial Number: 1234

Display Parameters
==================
Input type      : Digital
Color bit depth : Undefined
Video interface : Undefined
```

```
Max image size  : 34x19cm (15.3" diagonal, 16:9 ratio)
Gamma           : 2.200

Feature Support
===============
Preferred Mode    : yes
Monitor Standby   : no
Monitor Suspend   : no
GTF Timing Support: no
sRGB Color Space  : no
Active Off        : no
Color Encodings   : RGB 4:4:4 & YCrCb 4:4:4

No Monitor Range Limits specified
Preferred Mode : 1366x768x24@60

Detailed Timing 1366x768x24@60
===============
Pixel Clock     : 75200000
hActive         : 1366
hBlanking       : 188
hFrontPorch     : 30
hSyncPulseWidth: 63
hBorder         : 0
hBackPorch      : 95
vActive         : 768
vBlanking       : 38
vFrontPorch     : 4
vSyncPulseWidth: 12
vBackPorch      : 22
vBorder         : 0
flags           : 0x18
             Progressive
      Normal display; no stereo
      syncSignal1 = 3 = DigitalSeparate
      syncSignal2 = 0 = HSyncNeg:VSyncNeg
ImageSize       : 344x194mm (15.5" diagonal, 16:9 ratio)
VideoTiming(1554 1366 0 63 95 30 NEG, 806 768 0 12 22 4 NEG, 75.200MHz,
48.391KHz, 60.039Hz, Reduced Blanking)

Display Modes
=============
640x480x24@60
800x600x24@60
1024x768x24@60
1280x720x24@60
1366x768x24@60

No EDID extension blocks found
Checksum Valid : 0x63
```

Program the Flash Memory

See section 5 for details on the Flash memory areas.



Program Firmware

Syntax:**dl_mfg_test program <filename>**

<filename>	The firmware file (*.spkg), including path if necessary.

Restrictions for compound devices

This command cannot be used unless all component devices are separately set into MTT mode. Please refer to the section 'Handling Compound Devices' for more information.
It is not possible to program each component device separately. If the **--dev** option is used then all component devices will be programmed and not just the one specified.

Program Backup Firmware

Syntax: **dl_mfg_test program backup <filename>**

<filename>	The backup firmware file (*.spkg), including path if necessary.

Restrictions for compound devices

The same restrictions apply here that are specified for 'Program Firmware'.

Program Config

Syntax: **dl_mfg_test config <filename_1> [<filename_2>]**

<filename_x>	The config file (*.mem), including path if necessary. Maximum two files (device dependent, second filename is not supported for DL-41xx).

All *config files* now specify a **minimum firmware version.** The MTT program will check the primary **and** backup firmware image versions. If this compatibility check fails the *config file* will **not** be programmed.

Restrictions for compound devices

- This command cannot be used unless all component devices are separately set into MTT mode. Please refer to the section 'Handling Compound Devices' for more information.
- It is not possible to program each component device separately.
- **filename_1** will always be assigned to the DL-x9x1 device and **filename_2** will always be assigned to the DL-x9x5 device.

Program UPD

Syntax:**dl_mfg_test upd**	**erase**	
	-s <usb_serial_1> [<usb_serial_2>]	
	-a <mac>	(only DL-3xxx/5xxx/6xxx)
	-e <edid_file>	(only DL-3xxx/5xxx/41xx)
	-n <edid_hex_ser_no>	(only DL-3xxx/5xxx/41xx)
	-x <edid_ser_string>	(only DL-3xxx/5xxx/41xx)
	-d <edid_date>	(only DL-3xxx/5xxx/41xx)
	gamma load <index> <csv_file>	(only DL-41xx)
	gamma use <index>	(only DL-41xx)

<usb_serial_n>	Maximum two strings (device dependant, second string not supported for DL-41xx). USB serial number (up to 20 character string for DL-3xxx/DL-5xxx/DL-6xxx and 123 character string for DL-41xx).
<mac>	MAC address
<edid_file>	EDID file (binary)
<edid_hex_ser_no>	EDID hex serial number (1-8 digit hexadecimal string)
<edid_ser_string>	EDID serial string (1-13 character text string)
<edid_date>	EDID year/week date (*yyyy:ww*)
<index>	Index of the gamma curve to write (0-4)
<csv_file>	CSV file containing the lookup data

- **erase** erases the entire content of the UPD. If given, it must be the first command after **upd**.
- **-s** or **--serial** sets the USB serial number to **<usb_serial>**
- **-a** or **--mac** sets the MAC address
- **-n** or **--edid-serial** sets the EDID serial number to **<edid_hex_ser_no>**
- **-x** or **--edid-monitor** sets the EDID serial string to **<edid_ser_string**
- **-d** or **--edid-date** sets the EDID manufacturing date to **<edid_date>**, specified as year/week in the form `yyyy:ww`
- **gamma load** writes a gamma curve lookup table (LUT) to the UPD. When a new LUT is written to the device (at the end of the **upd** command) it takes effect immediately with no need for other user intervention.
- **gamma use <index>** sets the gamma curve with the given index (0-4) as the current gamma curve.

These commands work by reading the current UPD from the device, amending it, and then writing it back again. More than one UPD command can be given on the same command line, and the requested changes will be made in the same write. The order of commands on the command line does not affect the order in which alterations to the UPD are made.

Note for DL-41xx devices:

A **upd** command line may contain multiple **gamma load** and/or **gamma use** commands. If more than one **gamma use** command is given on the command line, then the **gamma use** that occurs last will be used. If one or more **gamma load** commands are given, but there is no **gamma use** command, then the lowest numbered curve being set will be used as the current gamma curve.

Examples:

Make gamma curve 1 current:
```
# dl_mfg_test upd gamma use 1
```

Set gamma curve 0 and make gamma curve 1 current:
```
# dl_mfg_test upd gamma use 1 gamma load 0 lut0.csv
```

Set gamma curves 0 and 1, and make gamma curve 1 current:
```
# dl_mfg_test upd gamma use 1 gamma load 0 lut0.csv gamma load 1 lut1.csv
```

Set gamma curves 0 and 1, and make gamma curve 0 current:
```
# dl_mfg_test upd gamma load 0 lut0.csv gamma load 1 lut1.csv
```

Note that the video test pattern commands use display mechanisms that are not affected by changes to the gamma settings. The **video rgb (**see DL-41xx Command Line Tests**)** command is specifically provided for displaying changes to the gamma curve configuration.

Sets the UUT's USB serial number. Optionally, also sets the MAC address and/or an embedded EDID file.

N.B. This command will **erase all existing UPD data**. This means that the serial number, MAC address and EDID must all be set at the same time, not separately.

Q: Which EDID serial number should I use?
A: The DisplayLink host software will identify a monitor by the EDID serial string (-x), if it has one. If no serial string is set, either in the original EDID file or via MTT, the software will identify the device by the date of manufacture (-d) and hex serial number (-n).

So, **it is preferable to set the EDID serial string (-x)**.

Restrictions for compound devices

- This command cannot be used unless all component devices are separately set into MTT mode. Please refer to the section 'Handling Compound Devices' for more information.
- It is not possible to program each component device separately.
- ***usb_serial_1*** will always be assigned to the DL-x9x1 device and ***usb_serial_2*** will always be assigned to the DL-x9x5 device.
- Both serial numbers must be unique.
- It is possible to specify ***usb_serial_1*** and *not* ***usb_serial_2*****.** In this case a second serial number is generated. The generated serial number will be of the format: ***usb_serial_1 + "B"*****.** Generating ***usb_serial_2*** will fail if ***usb_serial_1*** is at the maximum length. In this case the user must specify ***usb_serial_2***.
- The ***–a*** option must be combined with the ***–s*** option. No other option combination is valid. Only one MAC address can be specified. The MAC address will be written to the appropriate component device regardless of the one specified by the *--dev* option.

9. DL-3xxx/DL-5xxx/DL-6xxx Command Line Tests

Note: The following tests cannot be run on a DL-41xx device.

Power-On Self Test Results

Syntax: **dl_mfg_test post**

Displays the results of the power-on self test.

Example output:

```
# dl_mfg_test post
POST Results:
-------------
DDR data test:          PASS
DDR address test:       PASS
Ethernet loopback test: PASS
```

Handling Compound Devices

Compound devices include two component devices. Some examples of compound devices are the DL-3950 (which is a combination of a DL-3951 and a DL-3955) and the DL-5910 (which is a combination of a DL-5911 and a DL-5915). For most MTT commands, the component devices are treated separately, but for programming commands a compound device is controlled as a whole.

Testing

A compound device will appear as two separate devices when the '***--dev list'*** command is used. These component devices must be set into MTT mode separately, as described in 'Handling Multiple Devices' above.

Non-programming MTT commands must be sent to each component device separately, using the --dev argument to specify which device to test.

Programming

Compound devices are always programmed together. To program one, you can specify either component device using the --dev argument, and the MTT will program both component devices at the same time. This applies to:

- Program Firmware
- Program Backup Firmware
- Program Config
- Program UPD

Please see each command for specifics of programming compound devices.

Multiple compound devices

Multiple compound devices should not be connected to the PC at the same time, as it will be unclear which devices are associated with each component device pair. However, multiple DL-3x00 or DL-5x00 devices can be connected at the same time as one compound device.

Burn-in mode

Syntax: **dl_mfg_test burn-in**

This will cycle through a fixed sequence of test patterns: a full screen of red, green, blue, white, then black. The burn-in test will continue until the UUT is power-cycled.

Video Tests

Assign a head to a video output

Syntax: **dl_mfg_test video map <output> <head>**

This enables the video output, using the image from the specified head.
The output value must be 0, 1, or 2* – the outputs are in the order listed in the configuration file.
The head value must be 0 or 1.

* This command applies to the DL-3500, DL-3700, DL-3900, DL-5700 and DL-5900 only. For DL-6xxx, see the “video enable” command instead.

Enable a video output

Syntax: **dl_mfg_test video enable <output>**

The output number must be 0, 1, 2 or 3* – the outputs are in the order listed in the configuration file.
This will connect the output to the video head and enable the output.

* Output 3 is valid for DL-6xxx.

Disable a video output

Syntax: **dl_mfg_test video disable <output>**

The output number must be 0, 1, 2* or 3* – the outputs are in the order listed in the configuration file.
This will disconnect the output from the video head and disable the output.

* Output 2 is valid for DL-3500, DL-3700, DL-3900, DL-5700, DL-5900 and DL-6xxx.
* Output 3 is valid for DL-6xxx.

Video head blanking

Syntax: **dl_mfg_test video blank <head> <on/off>**

The head number must be 0 or 1. Use “on” to turn the head on, and “off” to put it into blanking mode.

NOTE: The blank command will have no effect if the output is displaying a *test pattern* on a DL-3xxx/5xxx device.

Notes for usage of video disable/enable and head blanking commands with DL-6xxx

- Only one output can be enabled at the same time on each head.
- After power up, only outputs with connected sinks are enabled. Enabling other outputs will require usage of the video tests commands.
- After enabling an output it might be necessary to disable blanking on the assigned head.

Example usage (enable video on outputs 1,3 with previously 0,2 enabled):

```
# dl_mfg_test video disable 0
# dl_mfg_test video enable  1
# dl_mfg_test video disable 2
# dl_mfg_test video enable 3
# dl_mfg_test video blank 0 off
# dl_mfg_test video blank 1 off
```

DPMS power down

Syntax: **dl_mfg_test video power <head> <suspend/standby/off/on>**

The head number must be 0 or 1.
The head can be put into 4 different DPMS power modes:

- Suspend: VSync disabled, HSync enabled
- Standby: VSync enabled, HSync disabled
- Off: VSync disabled, HSync disabled
- On: VSync enabled, HSync enabled

Get EDID data

Syntax: **dl_mfg_test edid <output_file> <output_num>**
dl_mfg_test edid <output_file>
dl_mfg_test edid refresh <output_num>
dl_mfg_test edid refresh

<output_file> The file to store the EDID data in, including path if necessary.
<output_num> The output number (0, 1, 2* or 3*). If this is missing, the MTT will read the EDID from UPD instead.

Gets EDID data from the UUT. You can read the EDID for any output, or from UPD (for embedded monitor designs). The EDID data will be saved to a file.

The “refresh” version of this command will force the firmware to re-read the EDID from the output. You can then use the “<output_file>” version of this command to read the updated EDID.

* Output value 2 is valid for DL-3500, DL-3700, DL-3900, DL-5700, DL-5900 and DL-6xxx.
* Output value 3 is valid for DL-6xxx.

DP PHY tests

Compliance Test Patterns

Syntax: **dl_mfg_test dp phytest <pattern> <link_rate> <swing> <pre-emph>**
Syntax: **dl_mfg_test dp phytestdevice <pattern> <link_rate> <swing> <pre-emph>**

"phytest"	This command is used if CTS test equipment is connected to the DP output during the test.
"phytestdevice"	This command is used if a real DP sink is connected to the DP output during the test.

<pattern>	The PHY test pattern: 0. No pattern 1. D10.2 2. Symbol error count 3. PRBS7
<link_rate>	The required link rate (1.62G, 2.7G or 5.4G). *Note: Only DL-6xxx supports 5.4G.*
<swing>	The main link swing setting (0-3).
<pre-emph>	The main link pre-emphasis setting (0-3).

Inter-pair Skew Test

Syntax: **dl_mfg_test dp skewtest <custom_pattern> <link_rate> <swing> <pre-emph>**

<custom_pattern>
The custom PHY test pattern can be any user defined pattern with an exact length of 10 bytes. The custom PHY pattern should be specified in a hexadecimal string format. Some examples are:

"0x55555555555555570"

"0x11111111111111111"

"0xAAAAAAAAAAAAAAAA" etc…

<link_rate> The required link rate (1.62G, 2.7G or 5.4G). <swing> The main link swing setting (0-3).
<pre-emph> The main link pre-emphasis setting (0-3).

DDR Tests

Data bus test

Syntax: **dl_mfg_test ddr data**

Tests the DDR data bus for open and short circuits. If there is an error, 1 is returned.

Address bus test

Syntax: **dl_mfg_test ddr address <size>**
<size> Size of DDR in MB

Tests the DDR address bus for open and short circuits. If there is an error, 1 is returned.

Checkerboard test

Syntax: **dl_mfg_test ddr checker**

Tests the DDR with checkerboard data packets looking for “at speed” faults. The data pattern used is:

FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000
FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000
0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF
0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF

alternating with:

0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF
0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF
FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000
FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000FFFF0000

If there is an error, 1 is returned.
Warning: this test can take 15 seconds.

Byte Count data test

Syntax: **dl_mfg_test ddr byte**

This test uses an inverting 16-bit word count for the test data pattern. If there is an error, 1 is returned.

Antiphase test

Syntax: **dl_mfg_test ddr anti <n> start**
dl_mfg_test ddr anti <n> stop

<n> is the DDR data pin to test.

Tests for cross-talk issues by toggling one data pin in antiphase with the rest of the data bus (e.g. alternating 0x00000001 & 0xFFFFFFFE to test data pin 0).

Once started, this test will keep running until the user stops it.

Audio Tests

Loopback test

Syntax: **dl_mfg_test audio <on|off>**
Enables or disables the audio loopback. The loopback will copy any sound from the audio input to the audio outputs(s).

If there is an error, 1 is returned.

Volume test

Syntax: **dl_mfg_test audio volume <channel> <level>**
Sets the audio volume.

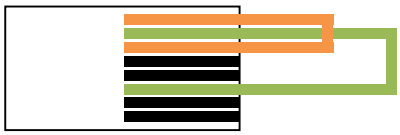
<channel> = audio channel 1 or 2
<level> = audio volume level 0-127

Ethernet Tests

Loopback test

Syntax: **dl_mfg_test eth loop**

Sends an Ethernet packet and checks that it has been received correctly. Assumes that an Ethernet loopback connector is fitted, which shorts together the data pins such that any packet sent out will be received back again.



Pin 1 (Tx+) to pin 3 (Rx+)
Pin 2 (Tx-) to pin 6 (Rx1)

If there is an error, 1 is returned.

Phy test modes

Syntax: **dl_mfg_test eth mode <n>**
<n> = test mode number, 0-9 (modes 7, 8, 9 DL-6xxx only)

Sets the RealTek Ethernet PHY into various modes needed for QualiPHY compliance testing. The test modes are as follows. Please contact RealTek for any further information on these tests.

0: Normal mode (Tests disabled)
1: 1G Test Mode 1 – Transmit Jitter Test
2: 1G Test Mode 2 – Transmit Jitter Test (MASTER mode)
3: 1G Test Mode 3 – Transmit Jitter Test (SLAVE mode)
4: 1G Test Mode 4 – Transmit Distortion Test
5: 100M – MLT-3 Output A Test (forced MDI-X mode)
6: 100M – MLT-3 Output B Test (forced MDI mode)
7: 10M – Diff Voltage / TP_IDL / Jitter
8: 10M – Harmonic Pattern 0xff
9: 10M – Harmonic Pattern 0x00

Power save control

Syntax: **dl_mfg_test eth powersave <on/off>**

Used to turn off the PHYs power saving mode when not connected. This is not a useful operating mode in normal service, but is useful for ethernet interoperability testing. In normal operation, powersaving is always enabled.

GPIO Tests

Enable an output pin

Syntax: **dl_mfg_test gpio <n> enable**
<n> = GPIO pin number 0-31 (0-15 for DL-6xxx)

Enables output on the specified GPIO pin number. If there is an error, 1 is returned.

Disable an output pin

Syntax: **dl_mfg_test gpio <n> disable**
<n> = GPIO pin number 0-31 (0-15 for DL-6xxx)

Disables output on the specified GPIO pin number. If there is an error, 1 is returned.

Change output value

Syntax: **dl_mfg_test gpio <n> output <0/1>**
<n> = GPIO pin number 0-31 (0-15 for DL-6xxx)

Drives the output value (0 or 1) on the specified GPIO pin number. If there is an error, 1 is returned.

Read a pin

Syntax: **dl_mfg_test gpio <n> read**
<n> = GPIO pin number 0-31 (0-15 for DL-6xxx)

Reads the specified GPIO pin number. The pin value is returned as 0 or 1. If there is an error, 1 is returned.

10. DL-41xx Command Line Tests

The following tests only apply to DL-41xx devices. They cannot be run on DL-3xxx/DL-5xxx/DL-6xxx devices.

Burn-in mode

The device may be put into burn-in mode using the **burn-in** command. The command does not invoke burn-in mode immediately, but instead, if successful, stores the parameters and sets a flag in the Flash which is detected when the device next powers-up. At this next power-up the flag (in Flash) will be reset and the video burn-in cycle will commence and continue indefinitely until power-cycled. After this power-cycling the device will now not be in burn-in mode; if burn-in mode is required again, it must be invoked once more from the command line. Additionally, the command also configures the burn-in images, backlight brightness and dwell time. Note that the backlight-brightness value specified by this command relates to the device's possible PWM output range and not to the values set by the 'Backlight Configuration Descriptor' defined in the configuration file.
Eight separate burn-in images, in .mem file format, are distributed with the MTT. Each represents a solid colour (white, black, red, green, blue, yellow, cyan and magenta). The user can select up to six images to be displayed during burn-in. Note that burn-in images and picture .mem files, as used by the 'program picture' command, are not interchangeable and can only be used with their intended mode of operation.

Syntax: **dl_mfg_test burn-in <backlight-brightness> <dwell-time> <burn-in file> …**

<backlight-brightness>:	range 0-255, where 0 = Off, 255 = maximum brightness.
<dwell-time>:	range 1-65535 in seconds.
<burn-in file>:	Burn-in image file (.mem). Minimum = 1. Maximum = 6. Duplicate files are allowed.

For example:

```
# dl_mfg_test burn-in 128 5 red.mem blue.mem
Would cause approximate mid-brightness, load two images (red and blue) to be
cycled through and dwell 5 seconds per image.
```

Errors:
The command will return fail if any of the parameters are out of range or missing.

Video Tests

Video raster mode

A flat colour at maximum intensity may be written to the display using the **video rgb** command. This is primarily intended for displaying changes to the gamma curves being used on the device.
Syntax: **dl_mfg_test video rgb <red>,<green>,<blue>**

<red>, **<green>** and **<blue>** must all be values in the range 0-255.

For example:
```
# dl_mfg_test video rgb 255 0 0
```

would cause the display to show red.

This command works by first selecting a video mode of 640x480 (to enable display on the screen), and then putting the device into raster mode, with the selected colour. This is actually displayed at the native resolution as described in the EDID. All other commands affecting the display implicitly exit raster mode. The **--mode** and **--detailedmode** switches may not be used with this command.

Backlight PWM

Setting the backlight PWM brightness and frequency

Syntax: **dl_mfg_test pwm set <brightness> [<frequency>]**

<brightness> Requested PWM brightness value. Decimal number in range 0 – 255.
<frequency> Requested PWM frequency in Hz. Valid range is 1 – 100000. Optional, default = leave unchanged.

Brightness values map linearly to the PWM duty cycle, so that a brightness value of 0 corresponds to 0% and 255 corresponds to 100%. The firmware will take account of PWM polarity if it exists in the config, so that a value of 255 will always mean maximum observed brightness whether or not the PWM is specified as inverted.

Below a certain threshold (3052 Hz) the PWM frequency is constrained to be a multiple of 10Hz. If a frequency below this is specified, it will be rounded up to the nearest 10Hz.

If the backlight is off when the “set” command is run and the frequency parameter is omitted, the frequency used will be the default value taken from the config. If there is no config present, an error will occur.

Examples:
```
# dl_mfg_test pwm set 255
# dl_mfg_test pwm set 64 48000
```

Getting the backlight PWM brightness and frequency

Syntax: **dl_mfg_test pwm get**

Example output:

```
# dl_mfg_test pwm get
PWM brightness: 255
PWM frequency: 48000
```

Program Picture

Syntax: **dl_mfg_test picture <filename>**

<filename> The picture file (*.mem), including path if necessary.

The picture filename contains one or more of:

- Splash screen background
- Splash screen image
- No-USB background
- No-USB image

The pictures are written to flash memory.

At the moment, picture .mem files are created by DisplayLink on request, from bitmap (.bmp) files. Note that any picture .mem files created for use with version 7.1 or older are not compatible with version 7.2. Additionally, burn-in images (.mem files), as used by the 'burn-in' command, are not compatible with this mode of operation.

Erase Picture

Syntax: **dl_mfg_test picture erase**

This command erases the splash and No-USB pictures stored in flash memory.

11. Example batch files

Some example batch files are included with these manufacturing tests. These show how to run the tests from the command line, program the UUT and report errors.

The batch files are only examples, and can be edited as needed to fit different circumstances. Some of the batch files include auto-incrementing serial numbers and MAC addresses.

DL-3xxx/DL-41xx/DL-5xxx/DL-6xxx batch files

- **example.bat**

This batch file will run the following test sequence:

Version number
Print the version number of the MTT.

Stop the DisplayLink service
This MUST be done before running any tests.

Video tests
Display a range of test patterns in turn, prompting the user to check the image visually.

Program serial number
Program the UUT with the USB serial number. ***N.B. Each device must have a unique USB serial number.***

Repeat tests
Allow the user to test/program another device

SKU-specific batch files

- **example_DL3xxx_DL5xxx_tests.bat**

This batch file will run the following test sequence:

Version number
Print the version number of the MTT.

Stop the DisplayLink service
This MUST be done before running any tests.

POST results
Print the results of the Power-On Self Test.

DDR tests
The data and address bus tests, checking for shorts and opens on those signals.

Video tests
Display a range of test patterns in turn, prompting the user to check the image visually.

- **example_DL3xxx_DL5xxx.bat**

This batch file will display the version number, run the tests from *example_DL3xxx_DL5xxx_tests.bat* and then program the UUT with the USB serial number and MAC address
N.B. Each device must have a unique USB serial number.

- **example_all_DL3xxx_DL5xxx.bat**

This batch file will loop round all connected devices and run the example.bat file on each device. It will also give the user the option to test/program more devices by allowing them to disconnect/connect.

- **example_DL3950.bat**

This batch file shows the differences in syntax for programming the DL-3950 compound device. It will also run the tests from example_DL3xxx_DL5xxx_tests.bat on each component device. Please refer to the section 'Handling Compound Devices' for more information.

- **example_DL6xxx.bat**

This batch file shows the example commands for DL6xxx.
This batch file will run the following test sequence:

Version number
Print the version number of the MTT.

Stop the DisplayLink service
This MUST be done before running any tests.

Video tests
Display a range of test patterns in turn, prompting the user to check the image visually.

Program serial number
Program the UUT with the USB serial number. ***N.B. Each device must have a unique USB serial number.***

Repeat tests
Allow the user to test/program another device.