AN4860
Application note

Introduction to DSI host on STM32 MCUs and MPUs

Introduction

A growing demand for smartphone-like, high level graphical user interfaces in embedded devices is posing big challenges to embedded system designers. The SPI, parallel and RGB interfaces have been widely used so far to make the connection between an MCU (or MPU) and a display. With increasing resolution and refresh rate requirements, a higher number of pins (up to 28 pins in case of the 16.7 million colors displays) and higher pixel clock frequencies are needed. These requests increase the pin count requirement on the controller side and the overall PCB complexity and cost due to board size, routing complexity and skew problems between clock and data.

To address these challenges, STMicroelectronics offers the first MCU products in the market with an integrated MIPI-DSI host. These new STM32 products with DSI host implement a more effective method for connecting to displays. The MIPI-DSI is a high-speed, low pin-count serial interface for displays originally targeted for the mobile industry. The DSI interface has become popular due to its widespread use in mobile phones and tablets, which has driven down the DSI displays costs and made them attractive for other consumer markets.

The STM32 MIPI-DSI host drastically reduces the device's pin count, enabling an easy connection with ubiquitous DSI displays available today in the market. Thanks to its low pin-count and low-power features, the DSI host is the most effective way to connect to displays especially for devices which have stringent constraints on size and power consumption like wearables.

This application note describes the DSI host interface on STM32 MCUs and MPUs and focuses on presenting different operating modes of the DSI host and providing guidelines to choose the best operating mode depending on an application's needs. It also provides practical examples on how to configure the DSI host depending on the operating mode.

Related documents

This application note has to be read in conjunction with below documents available at *www.st.com*:

- *STM32F76xxx and STM32F77xxx advanced Arm®-based 32-bit MCUs* (RM0410)
- *STM32F469xx and STM32F479xx advanced Arm®-based 32-bit MCUs* (RM0386)
- *STM324Rxxx and STM32L4Sxxx advanced Arm®-based 32-bit MCUs* (RM0432)
- *STM32H747/757 advanced Arm®-based 32-bit MCUs (RM0399)*
- *STM32MP157 advanced Arm®-based 32-bit MPUs (RM0436)*
- *STM32U5 Series Arm®-based 32-bit MCUs (RM0456)*
- Related STM32F469/479, STM32F7x8, STM32F7x9, STM32L4R9xx, STM32L4S9xx, STM32H747xI/G, STM32H757xI, STM32MP157A/D, STM32MP157C/F datasheets

Table 1. Applicable products

Type	Product lines
Microcontrollers	STM32F469/479, STM32F7x8/x9, STM32L4R9/S9, STM32H747/757, STM32MP157, STM32U599/5A9, STM32U5F9/5G9

Contents

List of tables

List of figures

1 Standards and references

This section presents the standards and references used in this document.

- MIPI® Alliance specification for display serial interface (DSI)
  v1.1 - November 22, 2011
- MIPI® Alliance specification for display bus interface (DBI-2)
  v2.00 - November 16, 2005
- MIPI® Alliance specification for display command set (DCS)
  v1.1 - November 22, 2011
- MIPI® Alliance specification for display pixel interface (DPI-2)
  v2.00 - September 15, 2005
- MIPI® Alliance specification for stereoscopic display formats (SDF)
  v1.0 - November 22, 2011
- MIPI® Alliance specification for D-PHY
  v1.1 - November 7, 2011

2 DSI host overview

The display serial interface (DSI) is a high-speed serial protocol defined by the MIPI (mobile industry processor interface) Alliance to allow the interface between a display module and a host processor.

The STM32 is the first MCU on the market with an integrated DSI host. The STM32 DSI host provides a highly integrated solution thanks to its internal MIPI D-PHY, a dedicated PLL, and a 1.2 V voltage regulator.

The DSI host provides a high-speed communication interface of up to 2 Gb/s. The DSI host allows the STM32 MCUs and MPUs to interface with a display using only a reduced pin count and without the need of an external bridge. The DSI interface is fully configurable, allowing an easy connection to the DSI compliant displays available today on the market.

Applications can benefit from the easy connection and reduced pin count enabled by the DSI host, which reduces the PCB complexity and the overall system's cost.

The DSI host is deeply integrated with the LCD-TFT display controller (LTDC) to ease the application development and porting.

The STM32 DSI host provides a scalable architecture. Depending on the bandwidth's requirements, the user may choose one or two data lanes.

2.1 Display interfacing

The displays are grouped into two main categories, depending whether they have an internal controller and frame buffer or not:

- Display modules with display controller and frame buffer: they embed a graphic RAM (GRAM) which stores the frame to be displayed and they have a display controller that controls the refresh operation.

  The MCU uses a set of commands to update the display's frame buffer content.

  The display relies on its internal controller and frame buffer to perform the refresh operation without intervention from the MCU.
- Display modules without display controller nor frame buffer: these displays rely on the MCU to send a real time pixel stream of data and video timing information in order to refresh the display.

These different types of displays imply different kinds of display interfacing:

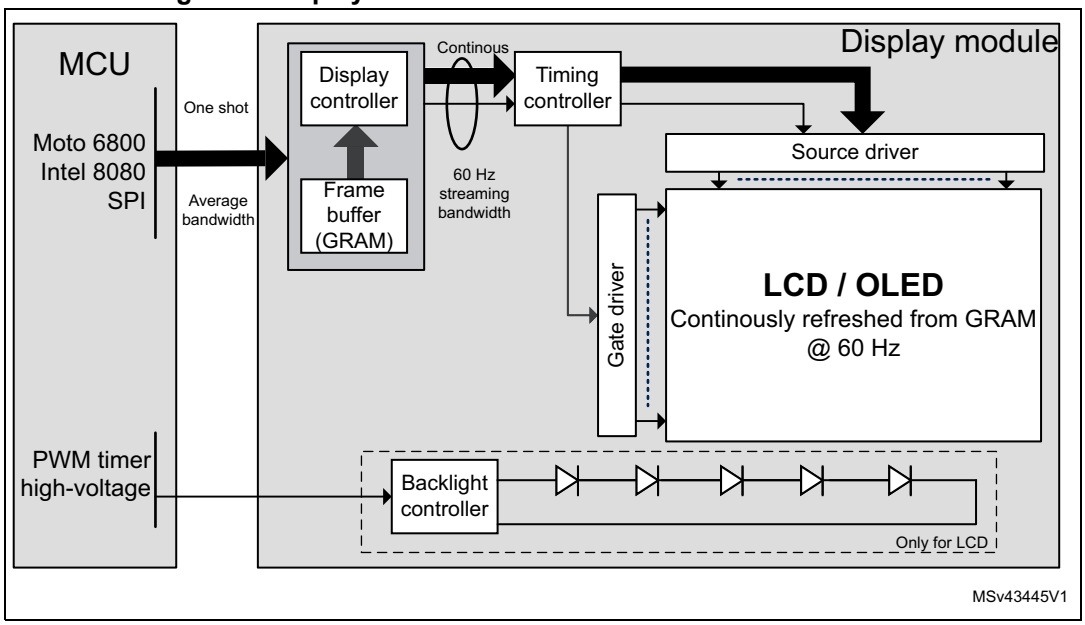
- Interfaces to display modules with controller and frame buffer:
  - SPI interface: uses few pins (up to six pins) but is very slow and not suitable for animations.
  - Parallel interface: uses a parallel bus to send command and data, hence they have a higher bandwidth than the SPI interface.

    Common parallel interfaces are the Motorola 6800 bus and the Intel 8080 bus. The parallel interface requires up to 22 pins (16 data and 6 control signals).

The display controller and the GRAM frame buffer are on the display side. The MCU sends commands to update the display GRAM. The update is done in one shot.

Figure 1 shows the architecture of displays embedding a frame buffer and a display controller with their corresponding interfacing schemes:

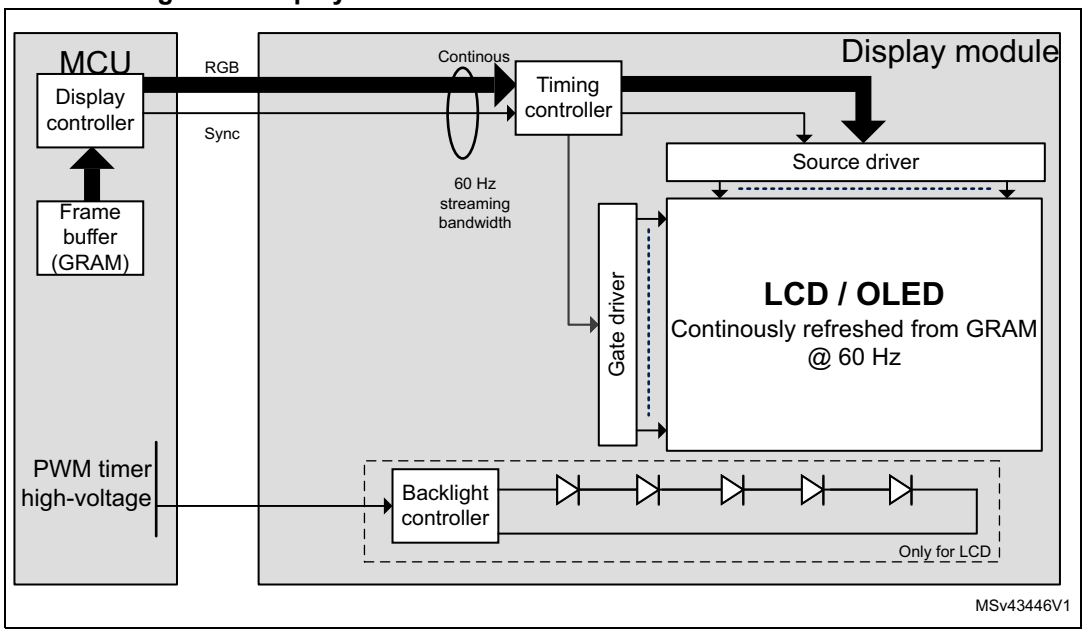
Figure 1. Display architecture with frame buffer and controller



- Interface to display modules without display controller nor frame buffer:
  - RGB interface: The MCU provides both the pixel data and the video timing signals. The RGB interface allows a very good real time performance but requires a high bandwidth on the MCU (or MPU) side in order to feed the display. Also it requires up to 28 pins: 24 RGB data (R[0:7]G[0:7]B[0:7]) and four synchronization signals (PCLK,HSYNC,VSYNC,DATAEN).
    The MCU continuously generates pixel data and video control signals to drive the display. The frame buffer is on the MCU side.

Figure 2 shows the typical architecture of the displays without frame buffer nor display controller:

Figure 2. Display architecture without controller nor frame buffer



2.2 MIPI display specification standards

The MIPI display working group specifies the interconnection between a host processor and a display.

The working group has defined a set of specifications in order to standardize the already existing protocols addressing displays:

- Display bus interface (DBI) covers protocols addressing display modules with display controller and frame buffer. It defines three interface types:
  - Type A which supports the parallel Motorola 6800 bus.
  - Type B which supports the parallel Intel 8080 bus.
  - Type C which supports the SPI interface.
- Display command set (DCS) specifies the commands to be used with displays supporting the MIPI-DBI interface.
- Display pixel interface (DPI) supports the RGB interfaces targeting displays without controller nor frame buffer.

The MIPI display working group has developed new protocols to drive modern displays in a more optimized way.

To decrease the number of wires between MCU and displays, the MIPI display working group has defined the DSI.

The DSI encapsulates either DBI commands (called command mode) or DPI signals (called video mode) and transmits them to the display in a serial manner. This enables getting interfaced with a standard display using only four or six pins, and achieving the same performance than a DPI.

The DSI allows the support of all possible display architectures (with or without display controller and frame buffer) with one single interface.

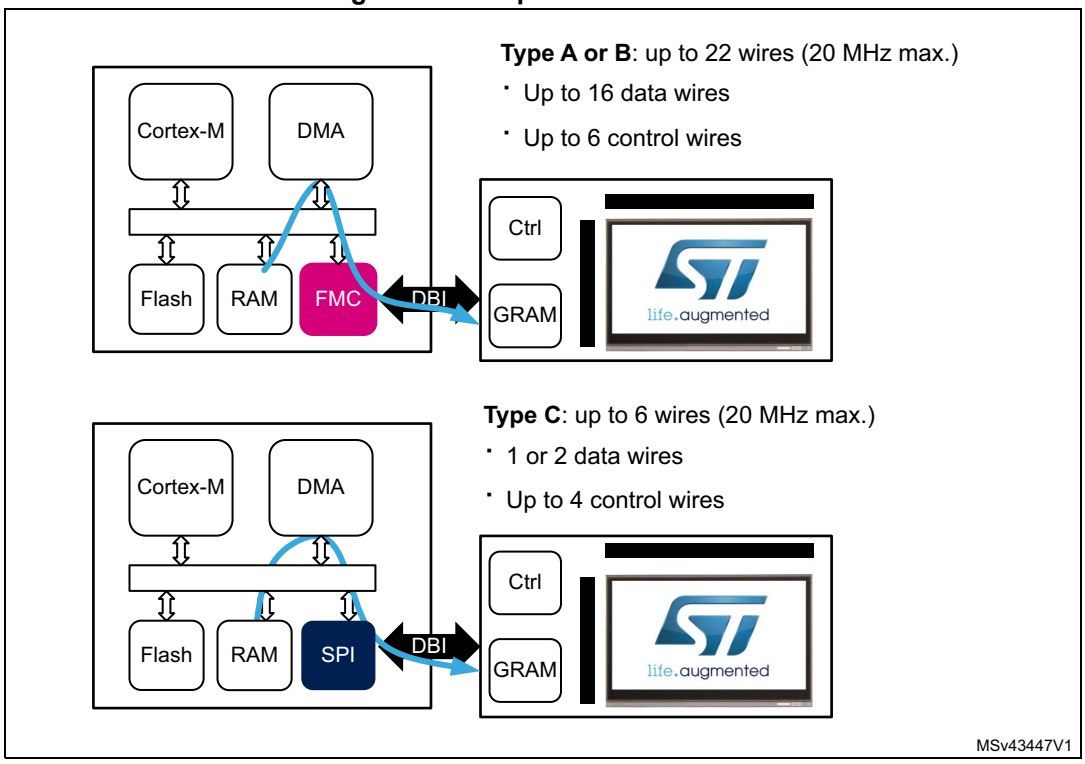
2.3 Display interfaces supported by STM32 products

All the STM32 products support the MIP-DBI type C interface through the SPI interface.

The MIPI-DBI type A and type B interfaces are supported by the STM32 embedding a F(S)MC (flexible synchronous memory controller). Refer to the application note *TFT LCD interfacing with the high-density STM32F10xxx FSMC (*AN2790) for more details on display interfacing through the FSMC.

Figure 3 shows an STM32 microcontroller interfacing with a display using a DBI interface:

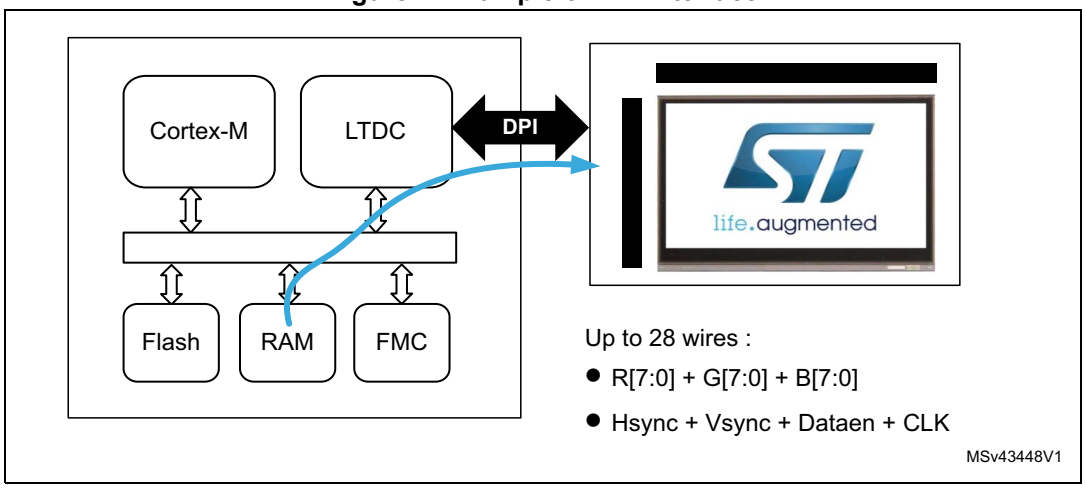
Figure 3. Example of DBI interface



The MIPI-DPI is supported by the STM32 embedding an LTDC (LCD-TFT display controller). Refer to the application note *LCD-TFT display controller (LTDC) on STM32 microcontrollers* (AN4861) for more details on this subject.

Figure 4 shows an STM32 microcontroller interfacing with a display using a DPI interface:

Figure 4. Example of DPI interface

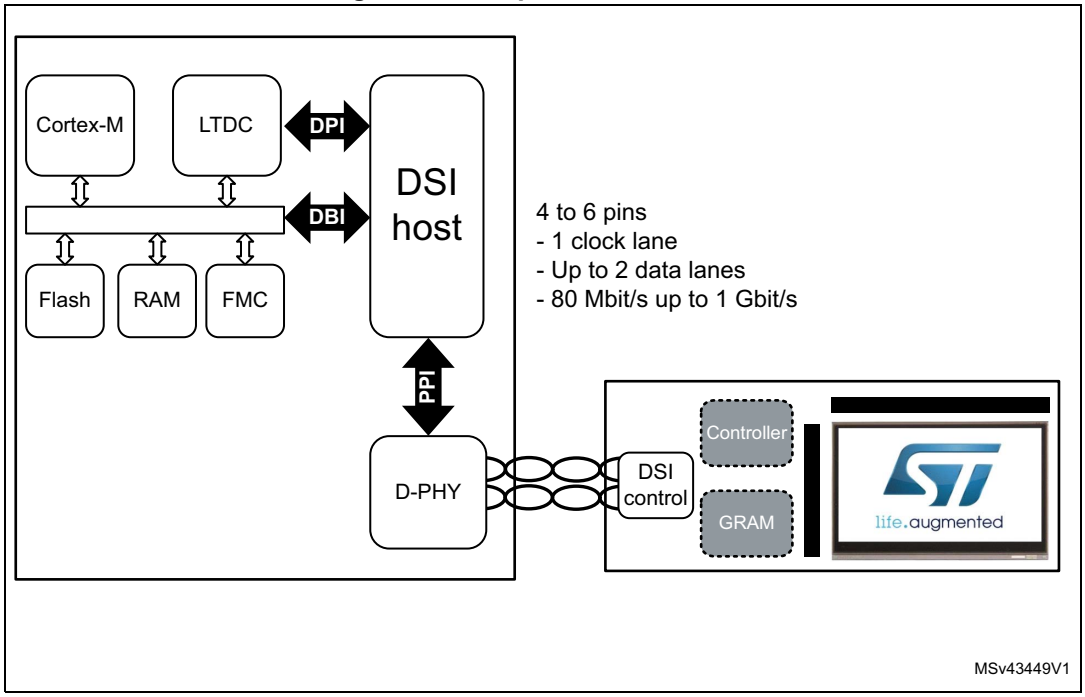


The MIP-DSI is supported by the new STM32 products embedding a DSI host.

The DSI interface gives the possibility to interface with both types of displays. In case of displays with GRAM and display controller, the DSI host sends commands to refresh the GRAM as in DBI mode. In case of displays without GRAM nor display controller, the DSI host sends a stream of pixel data and video synchronization events as in DPI mode.

Figure 5 illustrates an STM32 microcontroller interfacing with a display using a DSI interface.

Figure 5. Example of DSI interface



The *Table 2* below summarizes the different display interface schemes inside the STM32 products.

Table 2. Display interfacing in STM32 products

Classical display interface	MIPI display interface standard	Display with frame buffer support	Display without framebuffer support	Pin cout	Parallel / serial	STM32 support
Motorola 6800	DBI type A	Yes	No	12~22	Parallel	All STM32 with F(S)MC
Intel 8080	DBI Type B	Yes	No	12~22	Parallel	All STM32 with F(S)MC
SPI	DBI Type C	Yes	No	4~6	Serial	All STM32
RGB	DPI	No	Yes	20~28	Parallel	All STM32 with LTDC
-	DSI	Yes (DSI command mode)	Yes (DSI video mode)	4~6	Serial	All STM32 with DSI

2.4 DSI host availability across STM32 microcontrollers

The DSI host is used in STM32 products in conjunction with other features to enable best in class graphics applications. *Table 3* details the graphics portfolio in STM32 products embedding DSI host.

Table 3. STM32 microcontrollers featuring DSI host

Product	Features								
	FLASH (bytes)	On chip SRAM (bytes)	MIPI-DSI host	LTDC (1)	Quad-SPI (2)	JPEG Codec (3)	Chrom-Art Accelerator (4)	Max AHB frequency (MHz)(5)	Max FMC SRAM and SDRAM frequency (MHz)
STM32F469/479 line	Up to 2M	384k	Yes	Yes	Yes	No	Yes	180	90
STM32F7x8 line STM32F7x9 line	Up to 2M	512k	Yes	Yes	Yes	Yes	Yes	216	100
STM32L4R9xx/S9xx	2M	640k	Yes	Yes	Yes	No	Yes	120	60
STM32H747/H757	2M	1M	Yes	Yes	Yes	Yes	Yes	240	110
STM32U599/5A9 STM32U595/5A5	4M	2.5M	Yes	Yes	Yes	No	Yes	160	80

1. The LTDC is a TFT-LCD display controller. For more details on STM32's LTDC interface, refer to application note AN4861.
2. The Quad-SPI interface allows interfacing with external memories to extend the application's size. For more details on STM32's QSPI interface, refer to application note AN4760.
3. The JPEG codec provides hardware acceleration for JPEG encoding and decoding.
4. Chrom-Art Accelerator® is an ST proprietary 2D graphic acceleration engine.
5. The LTDC fetches graphic data at AHB speed.

2.5 DSI host advantages

The DSI host presents many advantages over other display interfaces.

Table 4. DSI host advantages over other display interfaces

DSI advantages	Comments
Low pin-count	The DSI interface drastically reduces the pin requirements. It can deliver the same transfer rate as an RGB interface using a lower pin count. For example, the 28 pins required by the LTDC are reduced to 4 or 6 pins in DSI.
Reduced PCB complexity	The PCB board complexity is reduced as there are fewer signals to route, removing the need for extra PCB layers.
Low EMI (electromagnetic interference)	The DSI interface generates lower levels of EMI thanks to its differential signaling scheme.
Scalable architecture	The DSI host provides a scalable architecture. Depending on the bandwidth requirements, the user may choose one or two data lanes.
Highly integrated solution	The DSI host has an integrated D-PHY, PLL and voltage regulator, removing the need for external components.
Smart architecture	The DSI host works in conjunction with the LTDC controller, which acts as video engine.
Standardized interface	The DSI host supports all operating modes described in the MIPI-DSI specification. The standardized interface enhances the interoperability with the DSI displays.
Optimized operation modes	The STM32 MCUs DSI host has an optimized mode which can be used with the DSI displays embedding a GRAM (graphic RAM). This mode reduces both bandwidth and memory requirements at the MCU side.
Benefit from ubiquitous DSI displays	The DSI displays are widely adopted in smartphones and tablets. They are becoming very attractive for many other embedded applications.

2.6 DSI host in a smart architecture

The STM32 system architecture consists mainly of 32-bit multilayer bus matrix that interconnects various masters with slaves.

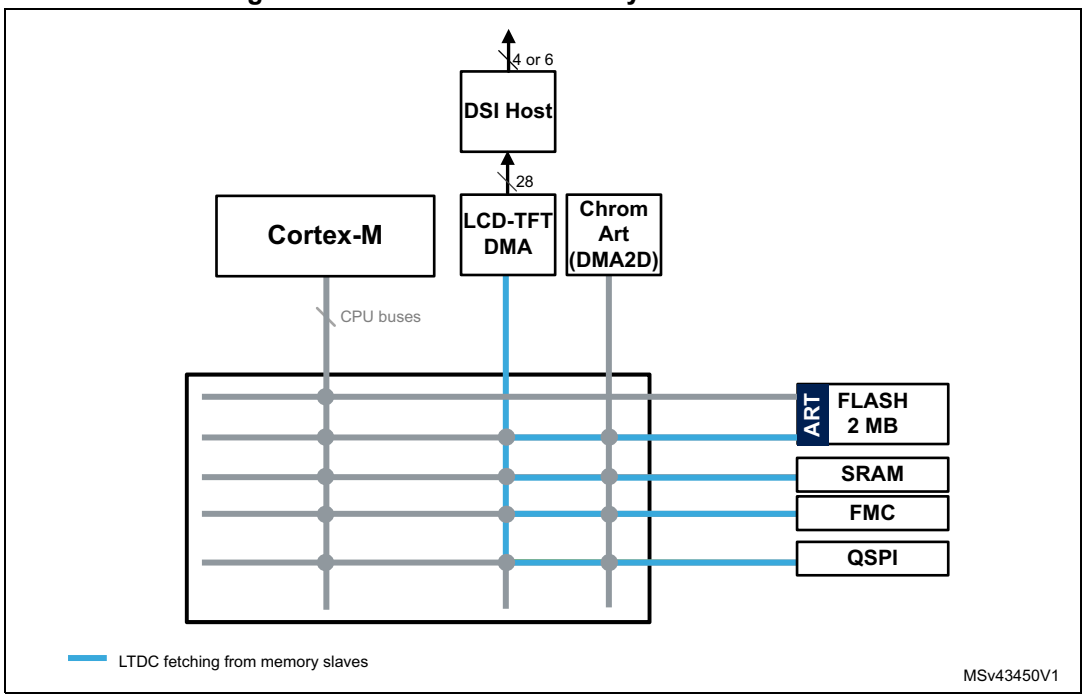
The LTDC is a master on the bus matrix. It fetches graphic data from various memory locations.

The DSI host can be easily integrated by using the LTDC display controller as a video streamer.

The pixel and video timing data from the LTDC are serialized by the DSI host and sent to the display module.

Figure 6 shows a DSI host interconnection in an STM32 system.

Figure 6. STM32 with DSI host system architecture



3 DSI introduction

The DSI specifies the interface between a host processor and a display module. It is built on the existing MIPI Alliance specifications by adopting the pixel formats and the command set specified in the DPI-2, DBI-2, and DCS standards.

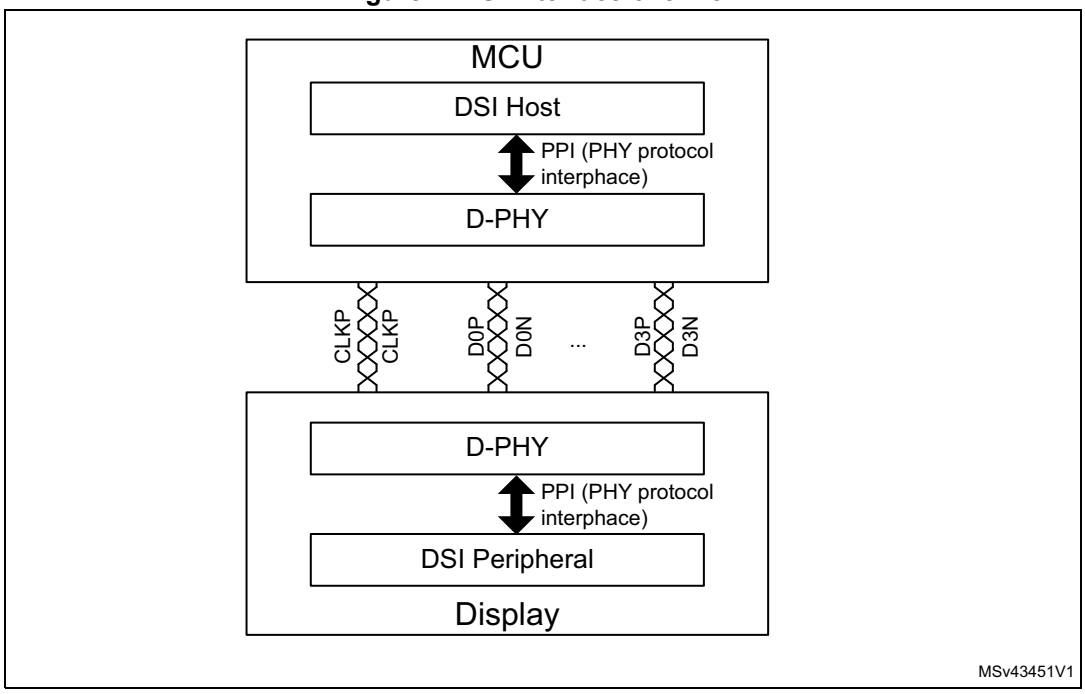
The DSI host sends pixel data, signal events or commands to the display after encapsulating them into DSI packets. The DSI host can read back status or pixel information from the display.

The DSI host transmits the DSI packets in the form of parallel data to the D-PHY through the PHY protocol interface (PPI). The D-PHY serializes the packets and sends them across the serial link.

On the display side, the packets are decomposed into parallel data, signal events and commands.

Figure 7 provides an overview of a DSI interface between a host and a display:

Figure 7. DSI interface overview



3.1 DSI operating modes

The DSI standard defines two basic operating modes for the DSI host and the DSI display: the command mode and the video mode.

3.1.1 Command mode

The command mode refers to an operation in which the transactions take the form of sending commands and data (as defined in the DCS specification) to a display module. A

display module that supports the command mode incorporates a controller and a frame buffer.

The host processor indirectly controls the activity at the display module by sending commands, parameters and data to the display controller.

3.1.2 Video mode

The video mode refers to an operation in which the transfers from the host processor to the display module take the form of a real-time pixel stream. The display module relies on the host processor to provide the image data at a sufficient bandwidth to avoid flicker or other visible artifacts in the displayed image.

3.2 DSI physical layer

The DSI physical layer is based on the MIPI D-PHY specification that describes a source synchronous, high-speed (HS) and low-power (LP) link.

3.2.1 PHY configuration

The link is composed of one double data rate HS clock lane provided by the host processor and of one to four serial data lanes.

A minimal link configuration has one clock lane and one data lane.

Each lane is composed of a pair of wires that are driven in low-swing differential mode during the HS transmissions and in single ended mode during the low-power mode.

The DSI communication can be made in the forward (host to display) or reverse (display to host) direction.

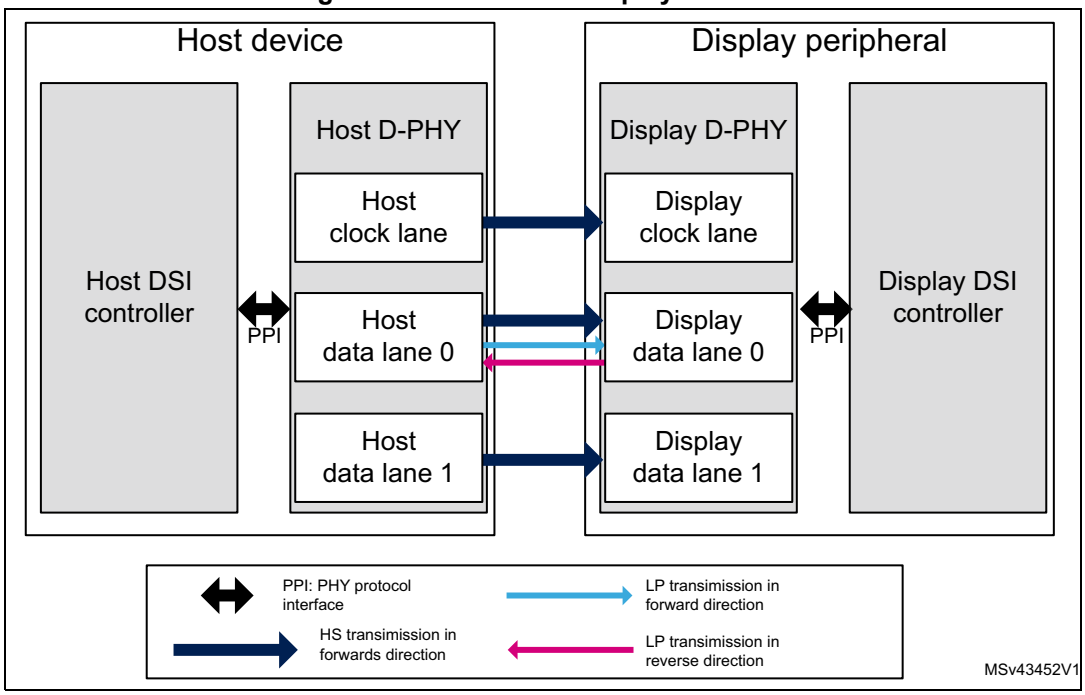
All reverse direction communication (from display to host) is done in low-power mode and using data lane 0 only.

In command mode systems, lane 0 is bidirectional, while in video mode systems it may be either unidirectional or bidirectional.

All other lanes are unidirectional and support a high-speed mode only.

Figure 8 shows an example of a PHY configuration with one clock lane and two data lanes.

Figure 8. DSI host and display interface



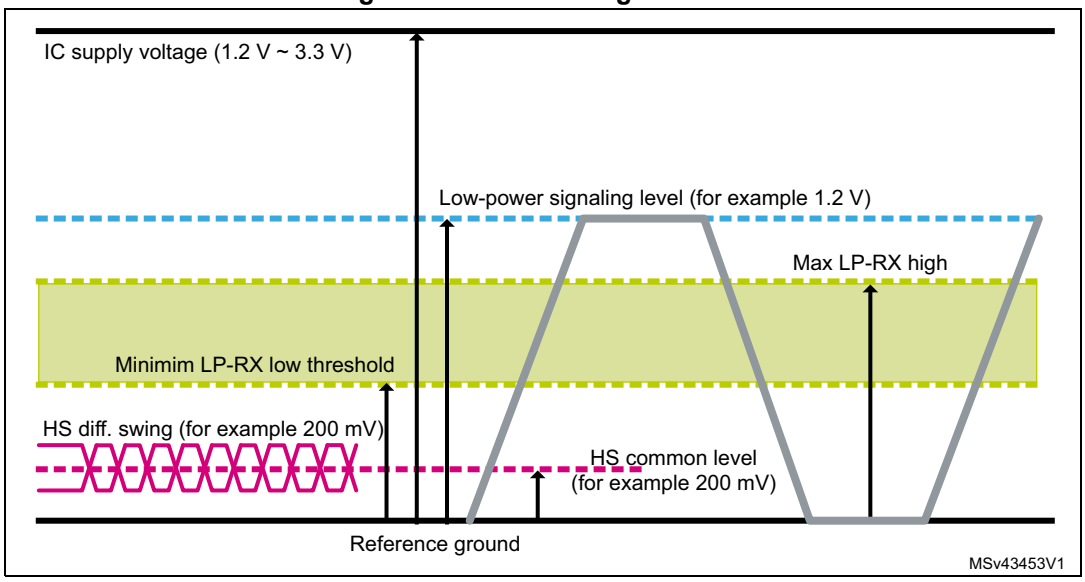
3.2.2 PHY signaling mode

There are two signaling modes in the DSI physical layer:

- High-speed (HS) mode: used for fast data transmission where the lane signals are used in differential mode with speeds up to 1.5 Gbit/s.
- Low-power (LP) mode: used mainly for control purposes. The pair of signals of the lane may be driven independently in a single ended mode with a maximum transfer rate of 10 Mbit/s.

Figure 9 shows an example of HS and LP signal levels in a DSI physical layer:

Figure 9. HS and LP signal levels



Data lane states

The DSI transmitter has one HS transmitter that drives the lines differentially to HS-0 or HS-1 and two LP transmitters that drive each line of the differential pair (D_P and D_N) independently in single ended mode. This results in two possible lane states for the HS transmitter (HS-0 and HS-1) and four lanes states for the LP transmitters (LP-00, LP-10, LP-01, LP-11).

The lane states reflect the mode of operation and are used to transition from one mode to the other (see *Table 5*).

Table 5. Data lane states and operating mode

State code	D_P line level	D_N line level	High-speed burst mode	Low-power control mode	Low-power escape mode
HS-0	HS low	HS high	Differential 0	x	x
HS-1	HS high	HS low	Differential 1	x	x
LP-00	LP low	LP low	x	Bridge	Space
LP-01	LP low	LP high	x	HS-Rqst	Mark-0
LP-10	LP high	LP low	x	LP-Rqst	Mark-1
LP-11	LP high	LP high	x	STOP	(return to stop)

3.2.3 Data lane operating modes

There are three operating modes for data lanes: control, high-speed transmission, and escape.

Control mode

After reset, the data lanes are in control mode (LP-11 stop mode). All other modes start and end to control mode.

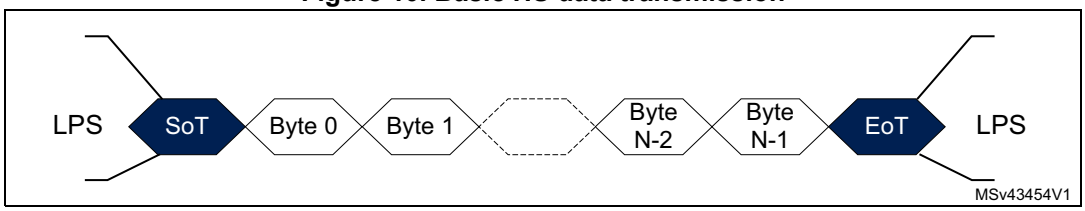
High-speed transmission mode

A high-speed transmission starts with and ends to stop state (LP-11).

To enable synchronization between host and display, a leader and trailer sequences are added. They are removed on the receiver side since they are not part of the actual payload data.

Figure 10 shows a basic HS transmission example.

Figure 10. Basic HS data transmission



1. Start-of-transmission (SOT) procedure

   Upon reception of the HS request (LP-11, LP-01, LP-00) the data lane enters the HS mode.

   The host starts by driving HS-0, then drives HS sync sequence (00011101) to allow the slave synchronization.

   Then the host continues the transmission of the HS data.

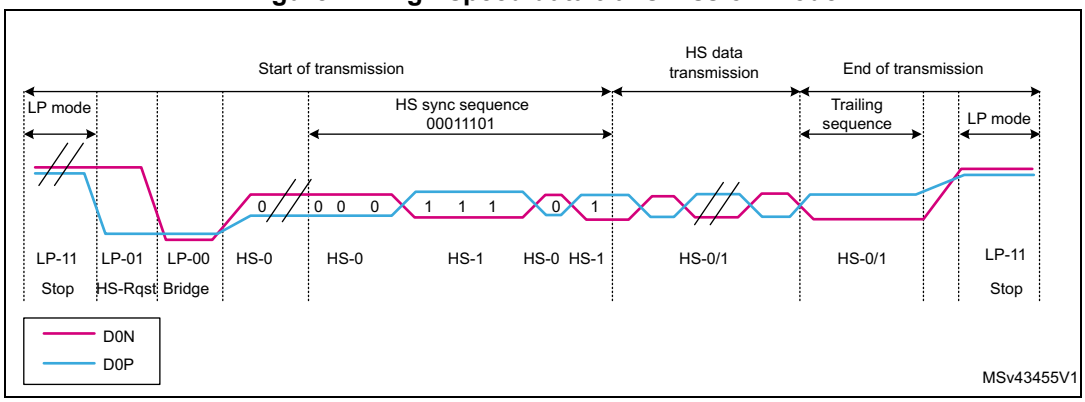
2. End-of-transmission (EoT) procedure

   After the end of the HS burst, the host sends a trailing sequence.

   The trailing sequence is the opposite of the last data bit transmitted: if the last payload bit is HS-0 then the transmitter sends HS-1 as trailing sequence, otherwise it sends HS-0.

   The data lane returns to control mode via the stop state LP-11.

Figure 11. High-speed data transmission mode

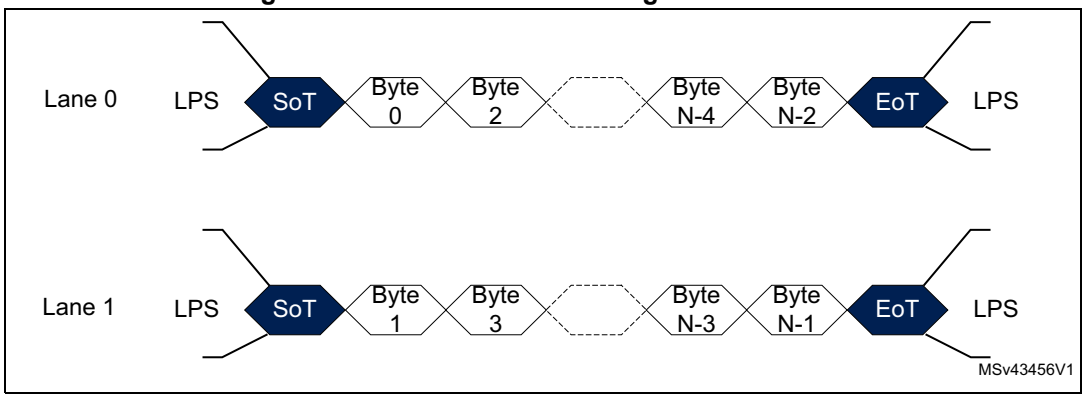


3. Multilane support

   The HS transmission can be done using one or more data lanes.

   Figure 12 shows an example of HS transmission using two data lanes:.

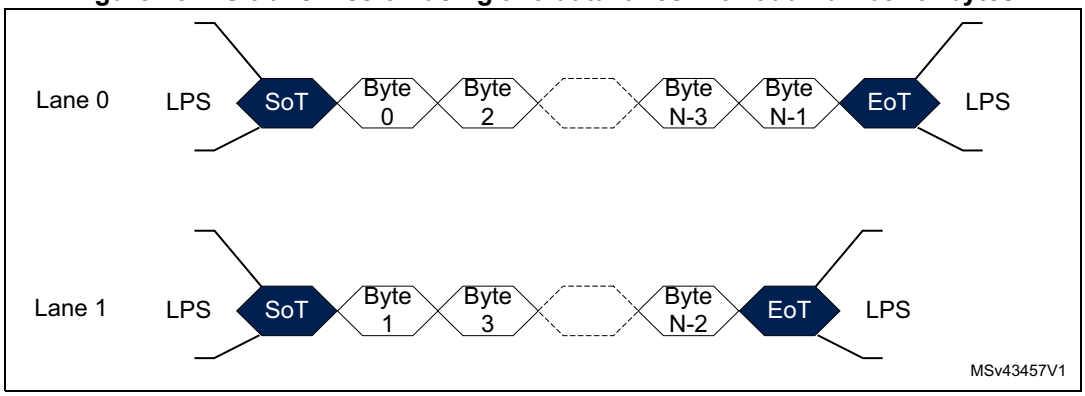
Figure 12. HS transmission using two data lanes



If the number of bytes transmitted is not an integer multiple of the number of lanes, some lanes may complete the HS transmission before the other lanes.

Figure 13 shows an example with two data lanes and odd number of bytes causing lane 1 to complete HS transmission and issue EoT sequence before data lanes 0.

Figure 13. HS transmission using two data lanes with odd number of bytes



Escape mode

The data lane may enter the escape mode via the escape mode request procedure (LP-11,LP-10,LP-00,LP-01,LP-00).

After entering the escape mode, the transmitter sends an 8-bit entry command to indicate the requested action.

Escape entry command may be:

- Low-power data transmission (LPDT)
- Ultra low-power state (ULPS)
- Remote triggers

The *Table 6* states the different entry commands supported in the escape mode:

Table 6. Escape mode commands

Escape mode action	Command type	Entry command pattern
Low-power data transmission	Mode	11100001
Ultra-low-power state	Mode	00011110

Table 6. Escape mode commands (continued)

Escape mode action	Command type	Entry command pattern
Undefined-1	Mode	10011111
Undefined-2	Mode	11011110
Reset-trigger	Trigger	01100010
Tearing effect trigger	Trigger	01011101
Acknowledge trigger	Trigger	00100001
Unknown-5	Trigger	10100000

Note: The low-power data transmission and trigger messages are only supported by lane 0.

The escape mode is exit through the escape mode exit procedure (LP-10, LP-11).

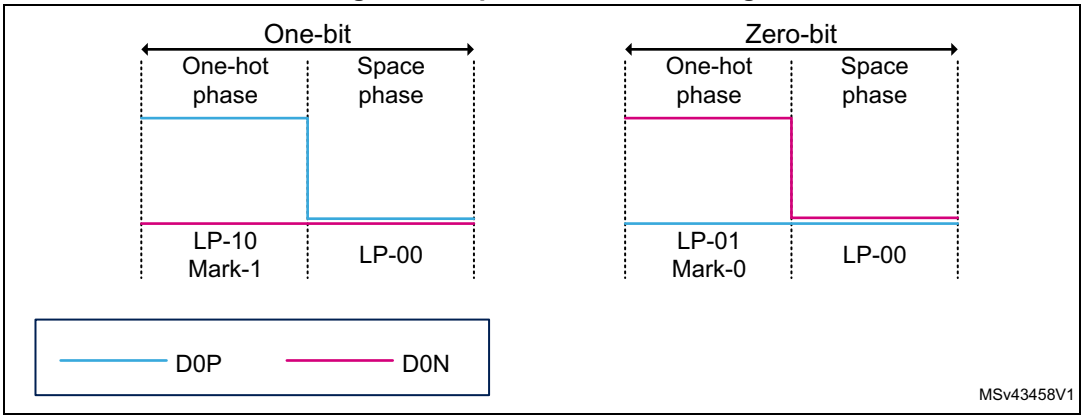
1. Spaced-one-hot coding

   In escape mode, the entry command and the low-power data transmission (LPDT) communication are coded using the spaced-one-hot coding, which means that each mark state is interleaved with a space state.

   Each symbol consists therefore of two parts: a one-hot phase (Mark-0 or Mark-1) and a space phase.

   The spaced-one-hot coding provides high reliability and the ability to extract the clock from the data stream. However, it requires the double of the bandwidth of the data transmitted (see *Figure 14*).

Figure 14. Spaced-one-hot coding



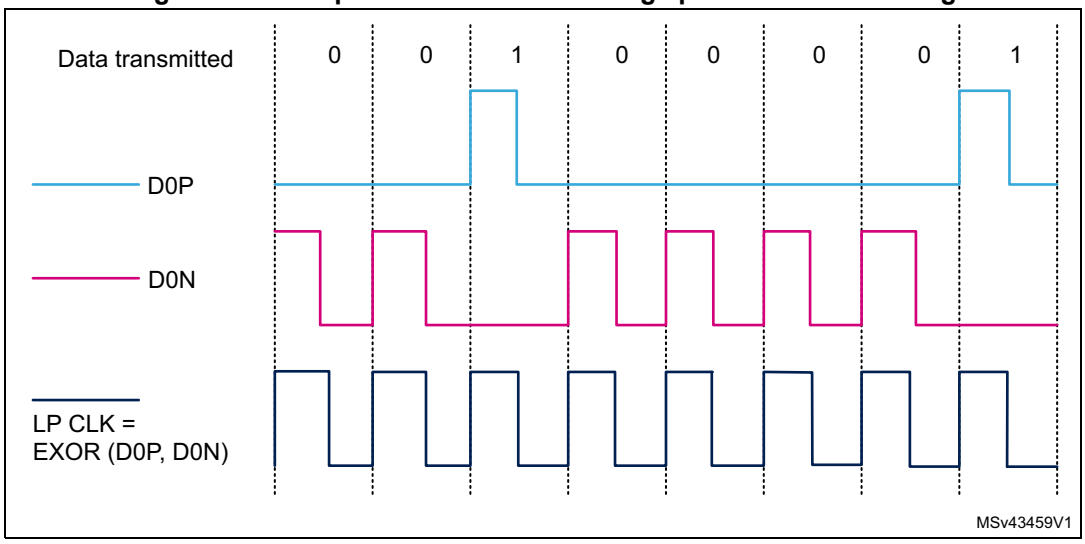
2. Low-power clock

   The transmitter uses a low-power clock signal for the low-power communications, but this clock is not transmitted to the receive side.

   The data is self-clocked by the spaced-one-hot bit encoding, and the receiver can retrieve the clock from the two line signals using an exclusive-OR function.

   Figure 15 shows an example of data transmission using spaced-one-hot encoding. The LP clock is obtained by applying an exclusive-OR function on the two signals (D0P and D0N.).

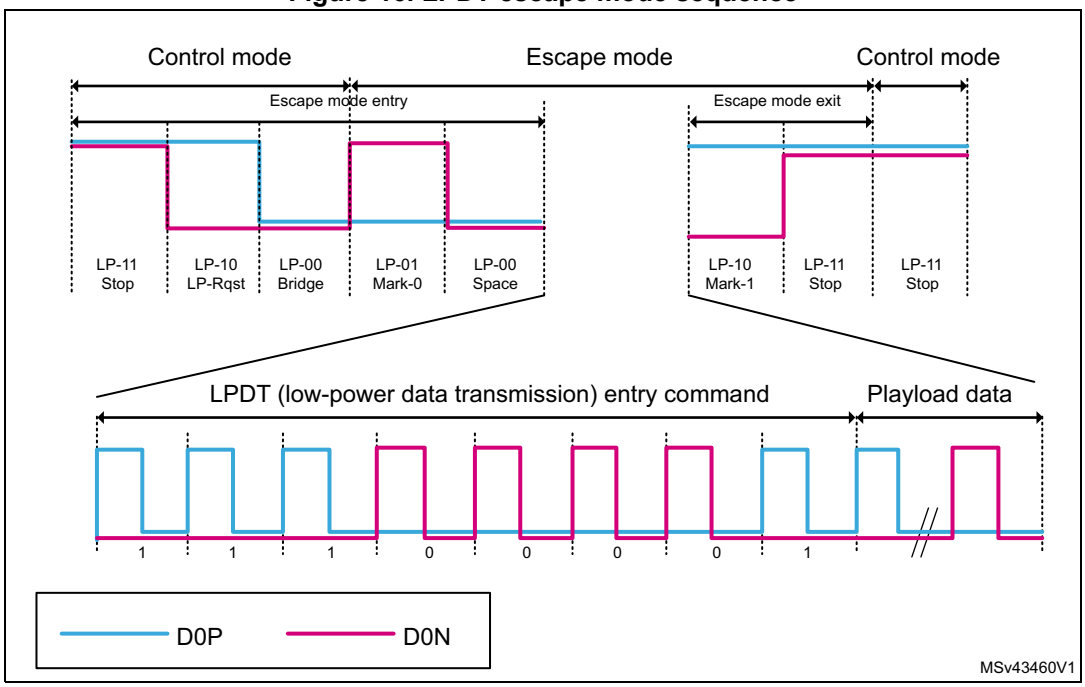
Figure 15. Example communication using spaced-one-hot coding



3. Low-power data transfer

   After the escape mode entry sequence, the transmitter sends an LPDT entry command (11100001) followed by the actual payload data. *Figure 16* shows an LPDT escape mode sequence.

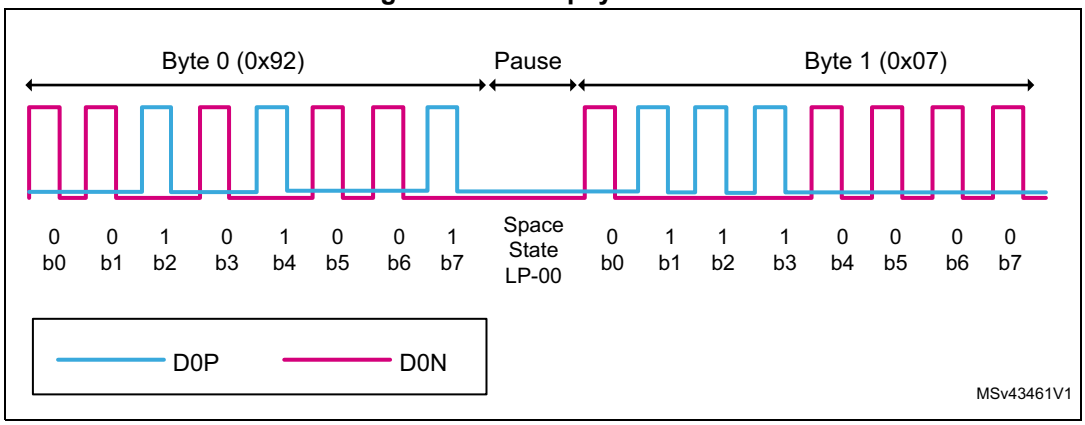
Figure 16. LPDT escape mode sequence



Data is sent in LSB (least significant bit) first, and for a multibyte payload, the least significant byte is transferred first.

During the LPDT, the lane can be paused by maintaining a space state (LP-00) on the lines. *Figure 17* shows an example of an LPDT payload data transmission.

Figure 17. LPDT payload data



4. Ultra low-power state

In order to reduce the power consumption, the DSI host may put the data lanes into the ultra low-power state (ULPS). This is achieved by the following procedure:

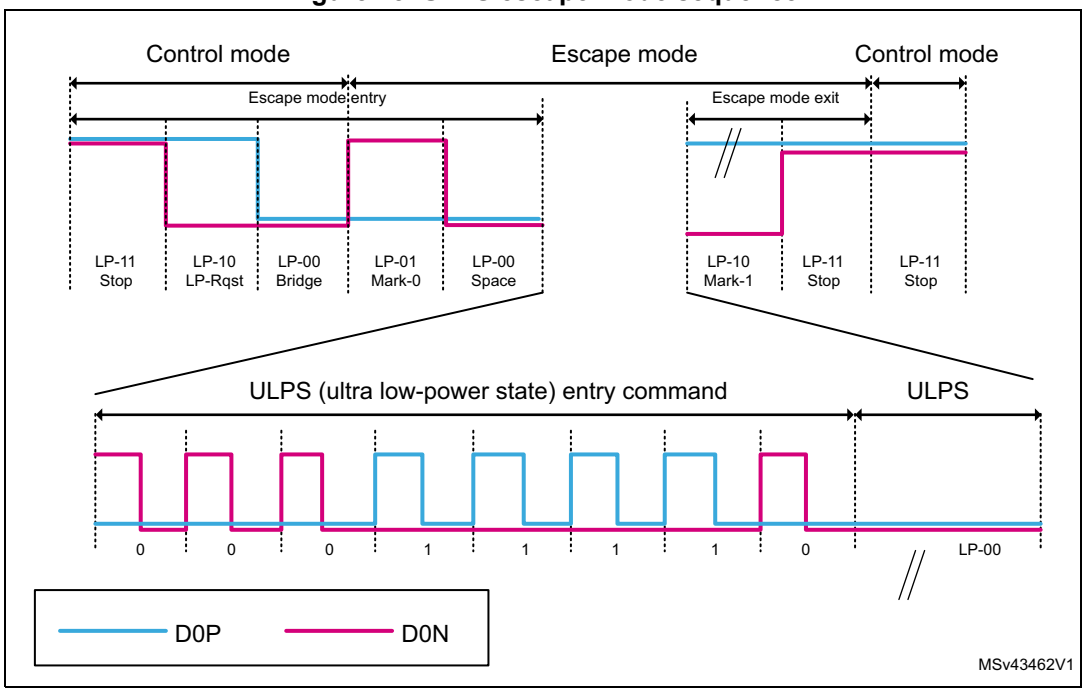
– Enter escape mode (LP-11,LP-10,LP-00,LP-01,LP-00).

– Send ULPS entry command (00011110).

– Keep lane signals into LP-00 state.

The ULPS state is exited with Mark-1 (LP-10) followed by stop state (LP-11). *Figure 18* shows ULPS escape mode sequence.

Figure 18. ULPS escape mode sequence



5. Triggers

The trigger signaling is a messaging system to send a flag to the receiving side. This can be either in the forward or reverse direction.

Three trigger messages are used in DSI:

– Acknowledge trigger: is a message sent by the display to the DSI host to indicate that the last transmissions have been received with no errors. This is a reverse direction communication (see *Figure 19*).

– Tearing effect trigger: used by the display to inform the host about the internal timing. This is a reverse direction communication (see *Figure 20*).

– Reset trigger: sent by the host to reset the display (see *Figure 21*).

Figure 19. Acknowledge trigger example

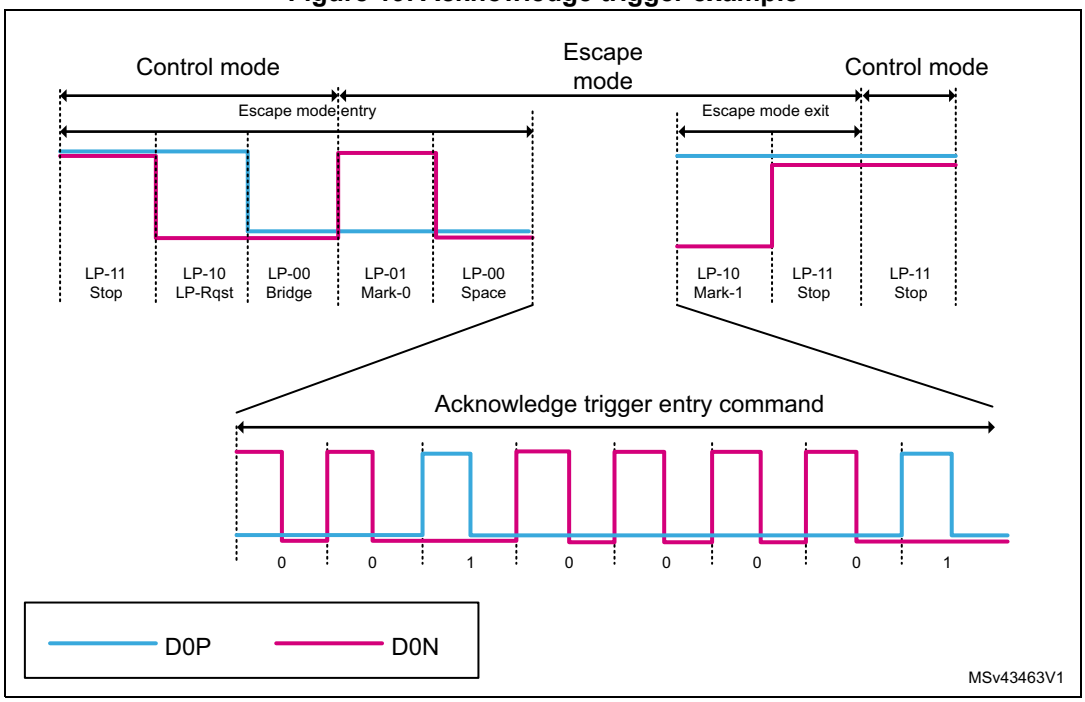


Figure 20. Tearing effect trigger example

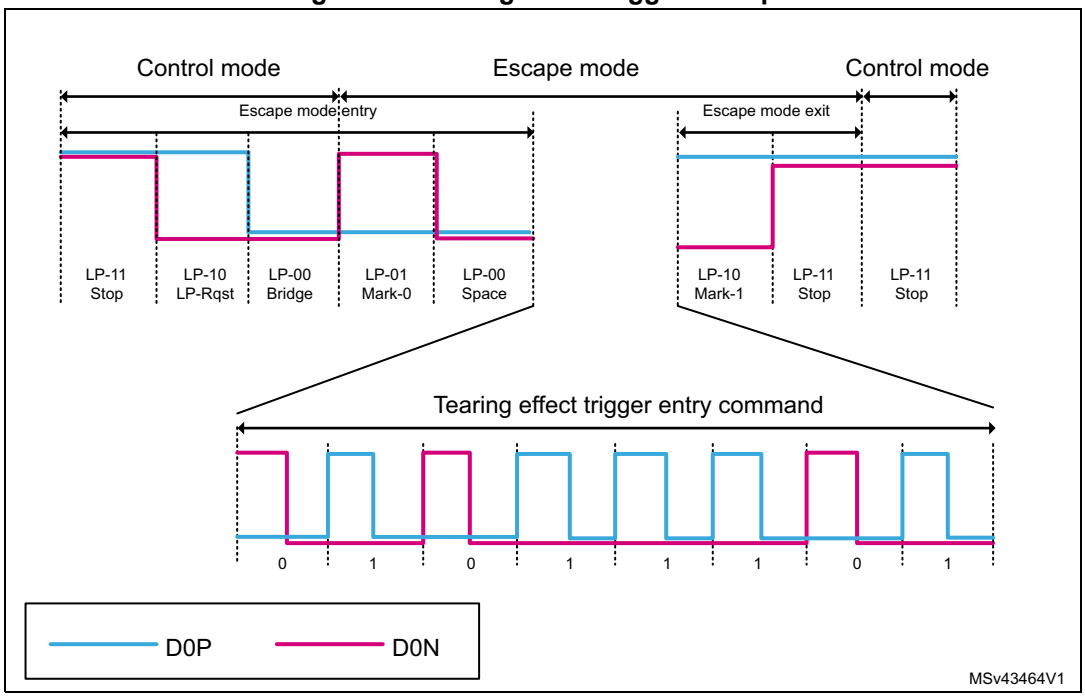
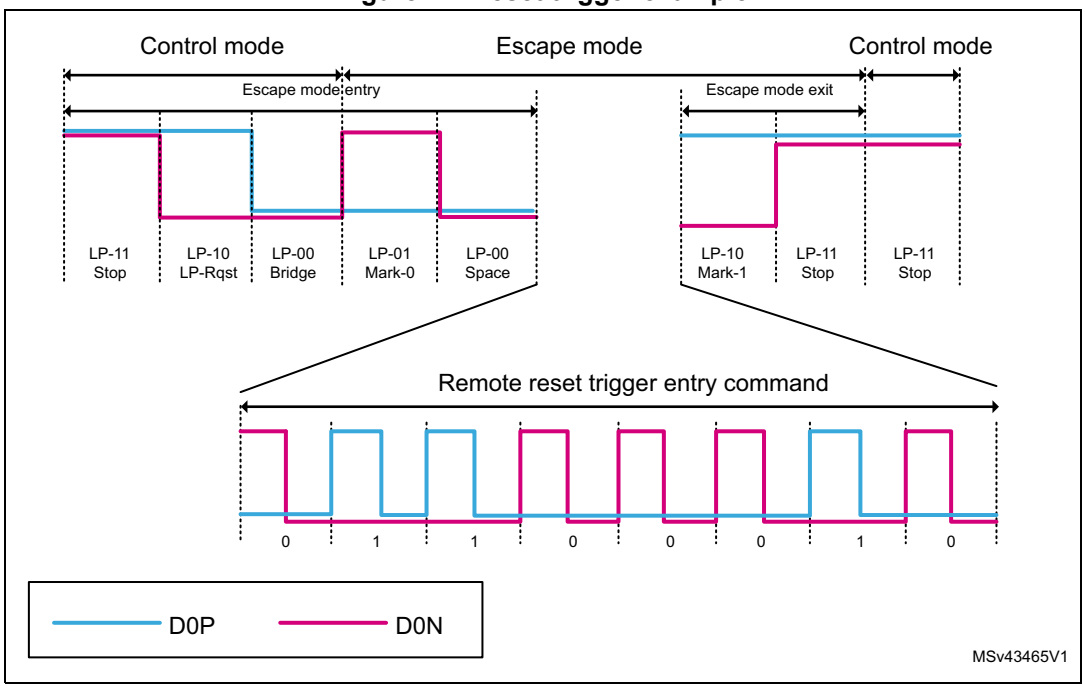


Figure 21. Reset trigger example



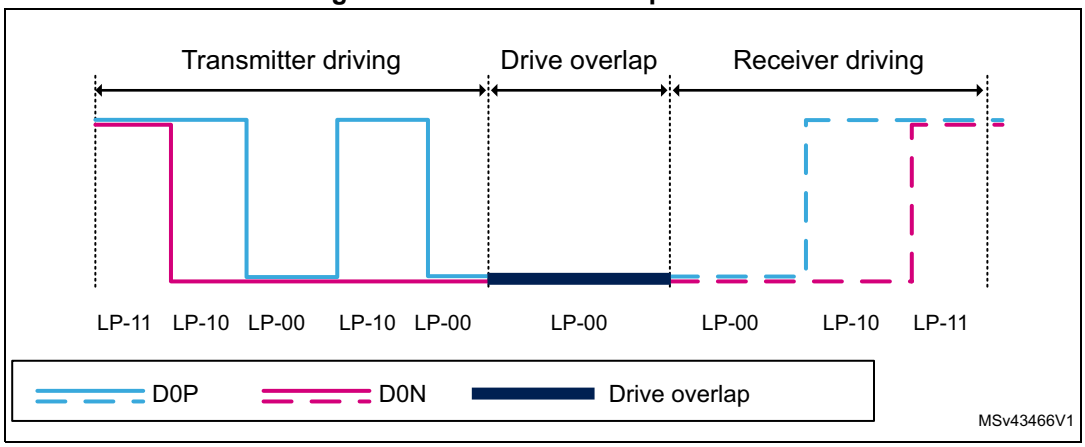
3.2.4 Bidirectional lanes and bus turnaround procedure

The DSI supports bidirectional data links only on data lane 0. To allow a reverse transmission, the data lane direction can be swapped using the bus turnaround (BTA) procedure.

The BTA is started from the stop state (LP-11). After the receiver has the bus ownership, the reverse transmission may begin. Then the receiver must give back the bus ownership to the host through the same turnaround procedure.

Figure 22 shows the BTA sequence.

Figure 22. Bus turnaround procedure



3.2.5 Clock-lane power modes

The DSI CLK lanes can be driven into three different power modes: low-power mode, ultra low-power state and high-speed clock mode.

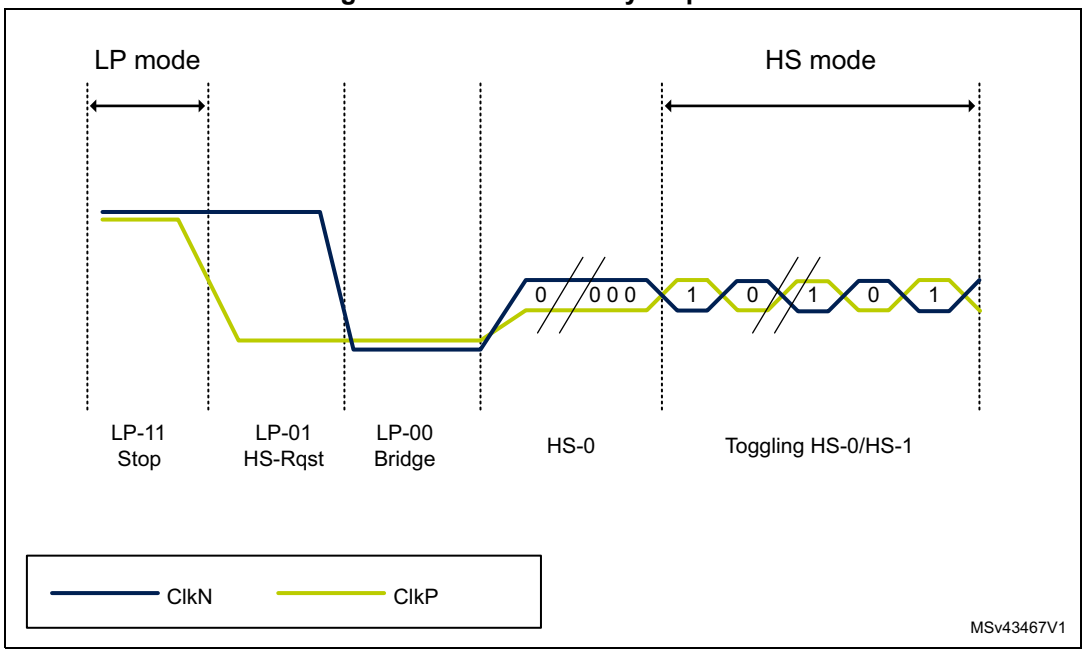
Low-power mode

During the low-power mode, the clock lane is in the LP-11 stop state. All other modes start from and end with the LP mode.

High-speed mode

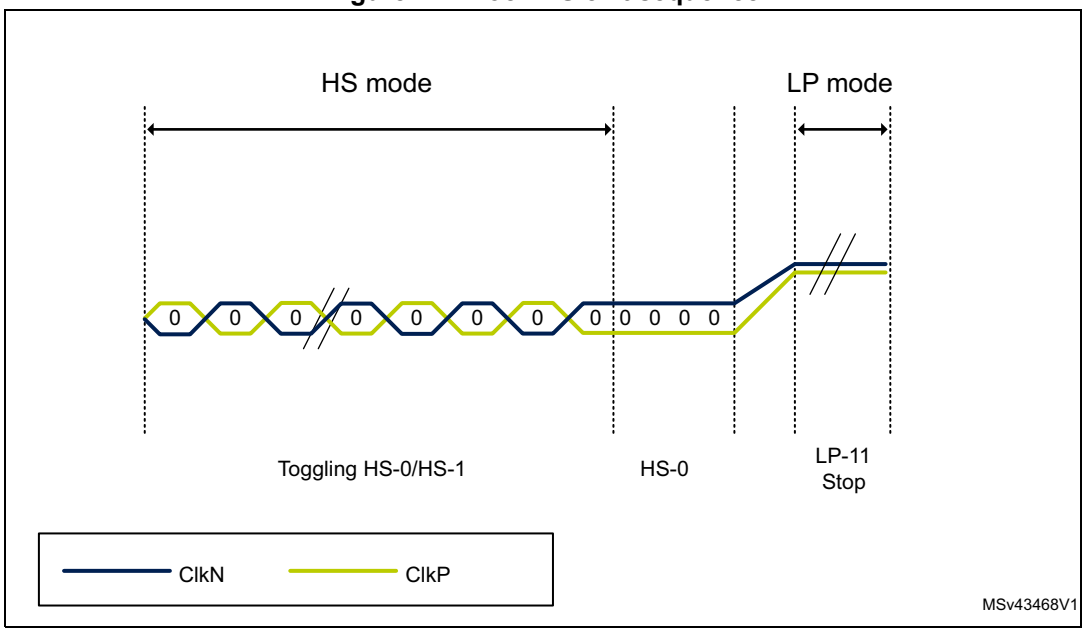
The clock lane enters the HS mode starting from the LP mode by driving an HS entry sequence (LP-11,LP-01,LP-00,HS-0) (see *Figure 23*). After that, the clock lane enters the HS mode and starts toggling HS-0,HS-1.

Figure 23. Clock HS entry sequence



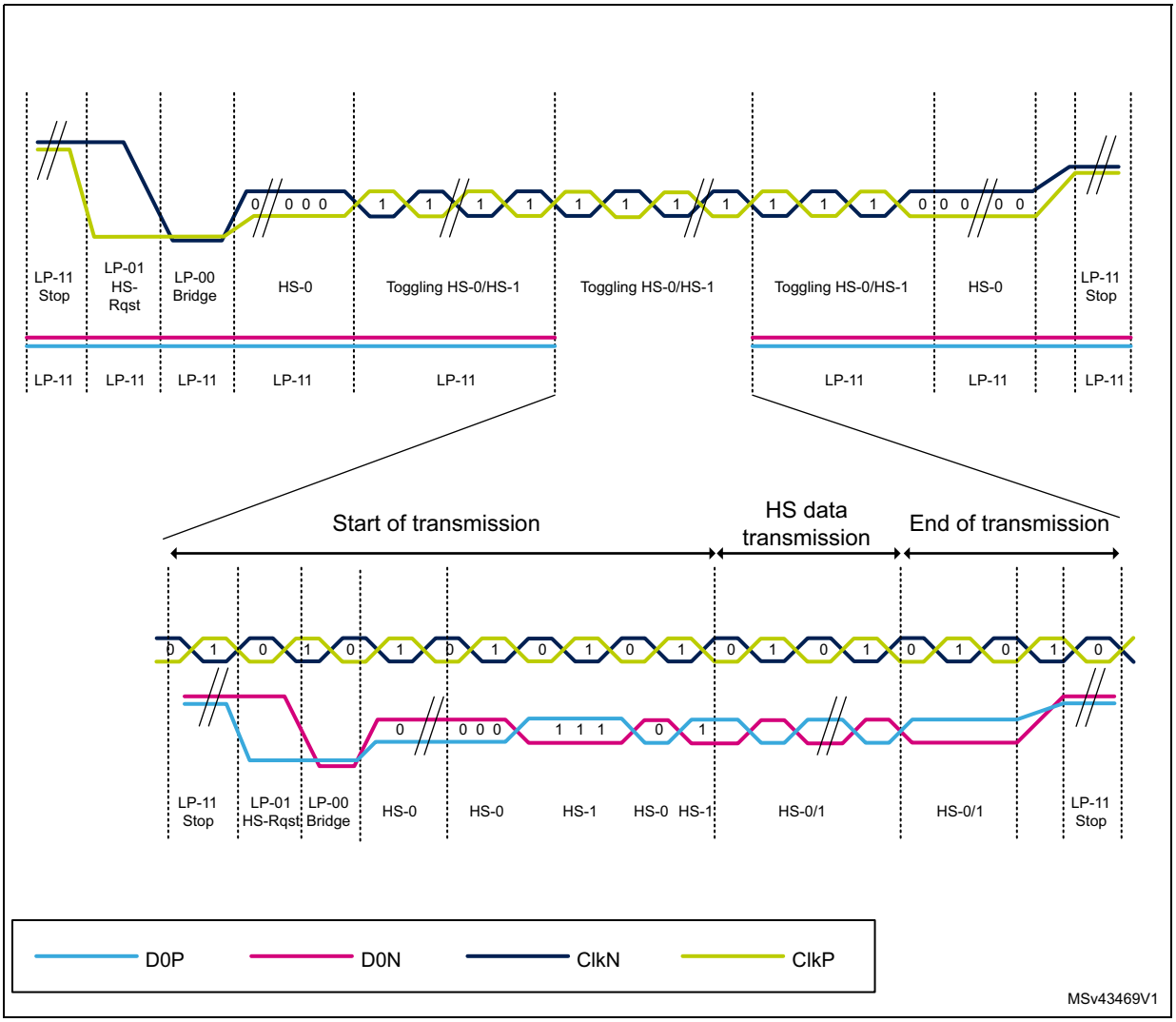
The clock lane leaves the HS mode through the exit sequence (HS-0,LP-11) (see *Figure 24*).

Figure 24. Clock HS exit sequence



The high-speed clock is started before that the high-speed data is sent via the data lanes. The high-speed clock continues clocking after the high-speed data lane has stopped (see *Figure 25*).

Figure 25. Clock and data lanes relationship in HS mode

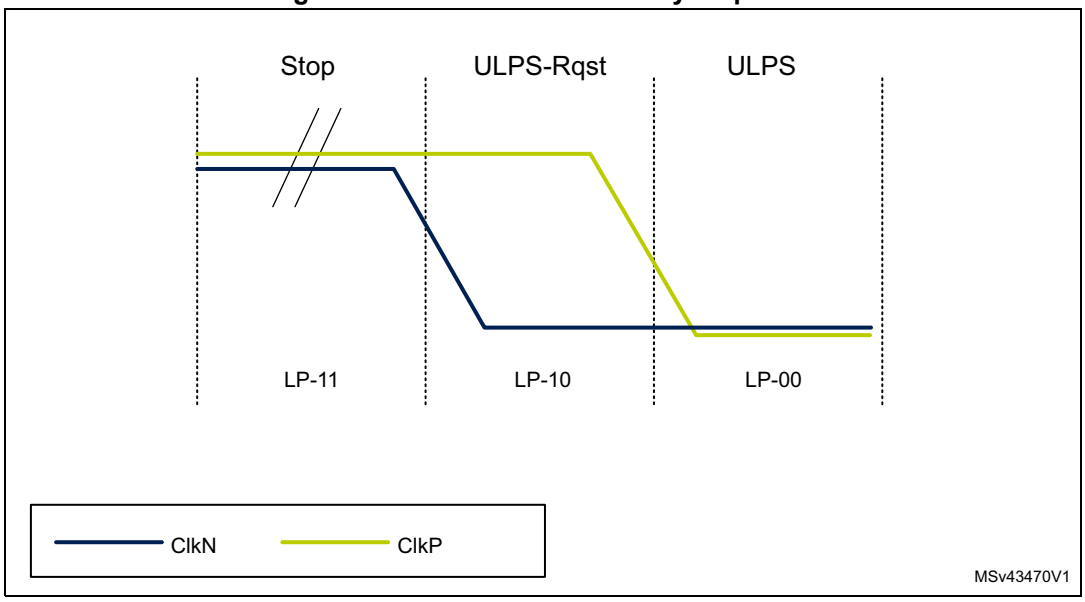


Ultra-low-power state (ULPS)

The data lane supports the escape mode while the clock lane does not support it. The clock lane supports however the ULPS (which is a subset of the escape mode).

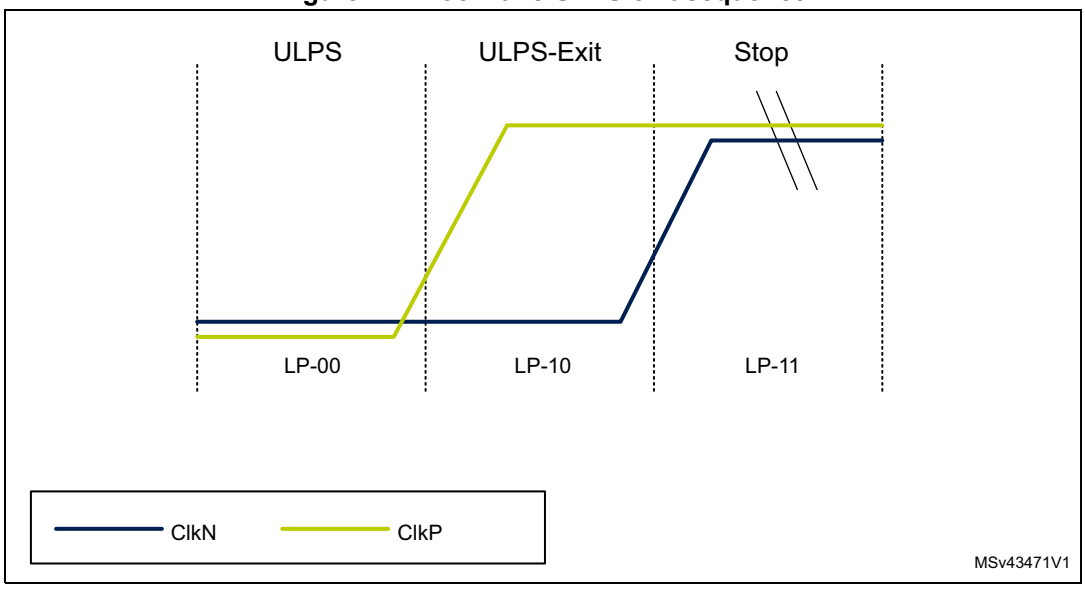
The clock lane may enter the ULPS starting from stop state using the ULPS entry sequence (LP-11,LP-10,LP-00) as shown in *Figure 26*.

Figure 26. Clock lane ULPS entry sequence



The clock lane leaves the ULPS state towards the LPM using the ULPS exit sequence (LP-00, LP-10, LP-11) as shown in*Figure 27*.

Figure 27. Clock lane ULPS exit sequence



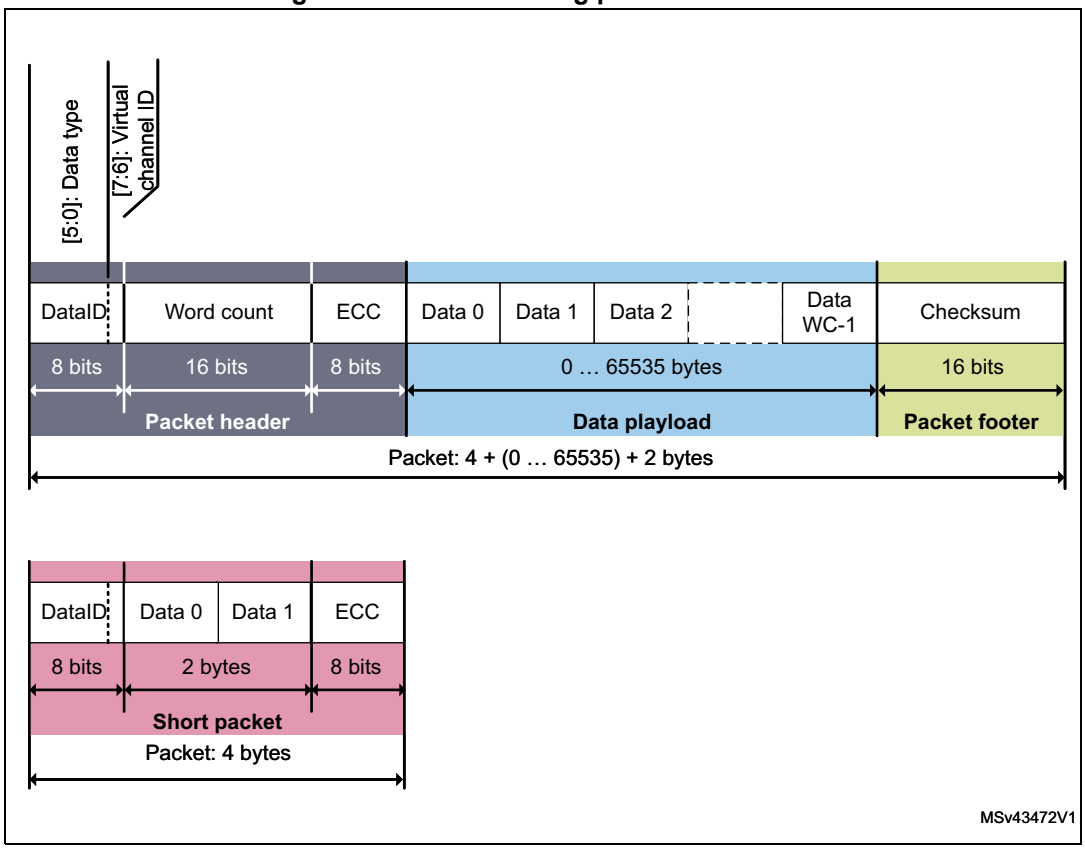
3.3 DSI protocol

The DSI is a packet based protocol. The parallel data and the commands are encapsulated into packets and upended with packet-protocol information and headers.

3.3.1 Packet structure

Two packet structures are defined for the low-level protocol communication: long packets and short packets. Refer to *Figure 28*:

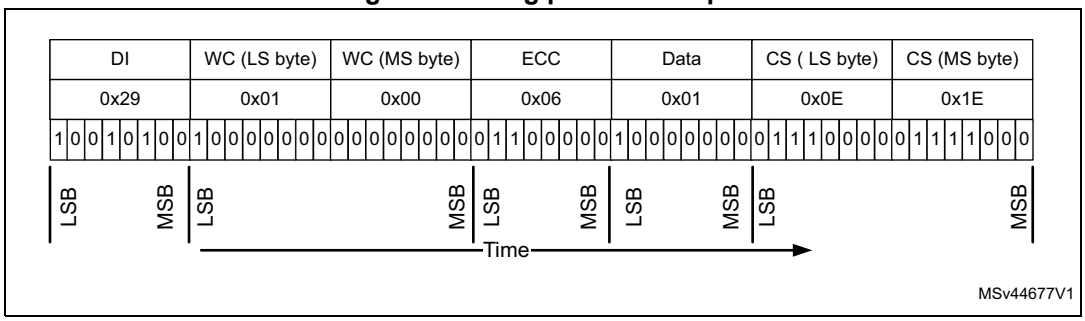
Figure 28. Short and long packet structures



The data is sent in bytes with least significant bit first. For multibyte fields such as word count and checksum, the data is sent with least significant byte first.

Figure 29 shows a long packet example.

Figure 29. Long packet example



Long packet

The long packets are mainly used for large blocks of data transmission such as pixel data. They are composed of three parts: packet header (PH), payload data, and packet footer (PF).

The 32-bit packet header contains:

- 8-bit data ID.
- 16-bit word count that defines the length of the payload data in bytes.
- 8-bit ECC (error-correcting code) to protect the packet header.

The payload data contains application specific data. It is mainly used to convey pixel data, or command parameters. Its length is defined by the word count. It may be between 0 and 65535 bytes in length.

The packet footer consists of a 16-bit checksum (CS). It is calculated by the transmitter and used by the receiver to check whether the data has been received with no errors.

The minimum length of a long packet is 6 bytes with zero payload data.

- Four PH bytes.
- Two PF bytes.

The maximum length is 65541 bytes.

- Four PH bytes
- 65534 payload data
- Two PF bytes.

Short packet

Short packets are formed of four bytes:

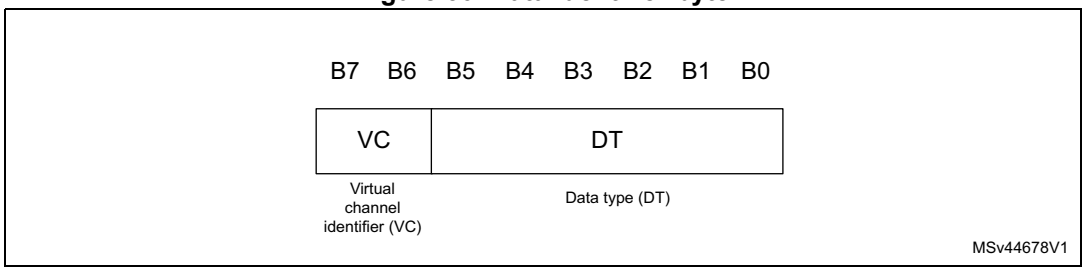
- One byte for data ID.
- Two bytes for command or payload data.
- One byte for ECC.

They are mainly used for short command transmission and for timing sensitive information like video synchronization events.

Data identifier byte

The first byte of any packet is the DI (data identifier) byte. The DI byte is composed of a virtual channel (VC) identifier and a data type (DT) (see *Figure 30*).

Figure 30. Data identifier byte



1. Virtual channel identifier

   A DSI host may serve up to four peripherals with tagged commands or blocks of data, using the virtual channel ID field of the header. The VC identifies the peripheral to which the data is directed.

2. Data type field DT[5:0]

   The data type field specifies if the packet is a long or short packet type. It also specifies the packet format and content of the payload data.

Data protection (ECC and checksum)

The DSI standard provides two data protection mechanisms: ECC and checksum.

1. Error-correcting code

   The error-correcting code (ECC) byte allows single-bit errors to be corrected and 2-bit errors to be detected.

   The ECC represents a robust protection for short packets that are usually used in conveying critical information. It is also a good protection for a long packet header that includes both the data identifier and the word count fields.

2. Checksum

   The payload data in long packets is protected with 16-bit checksum that can only indicate the presence of one or more errors in the payload.

3.3.2 End of transmission (EoT) packet

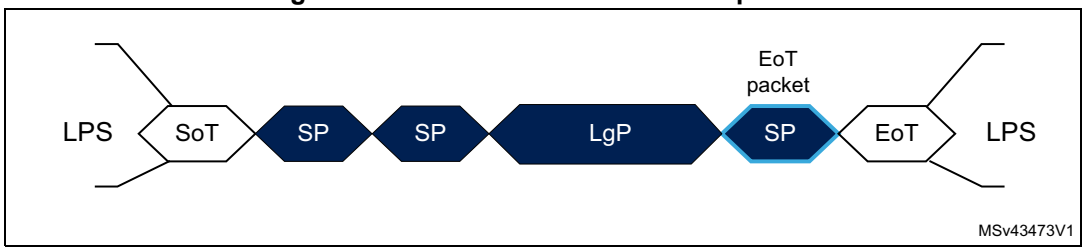
The D-PHY uses an EoT sequence to signal the end of an HS transmission, but this sequence can be interpreted as valid data by the receiver.

In order to add robustness at protocol level, the DSI transmitter can send an EoTp (end of transmission packet) to signal the end of an HS transmission.

This mechanism provides a more robust environment, at the expense of increased overhead (four extra bytes per transmission).

Figure 31 shows an example of HS transmission with EoT packet enabled.

Figure 31. HS transmission with EoT packet



3.3.3 Packet transmission modes

Short and long packets may be transmitted either in HS or LP mode. Also packets sent in HS mode may be split between available data lanes.

Figure 32 shows a short packet transmission in HS mode using one data lane.

Figure 32. Short packet transmission in HS mode using one data lane

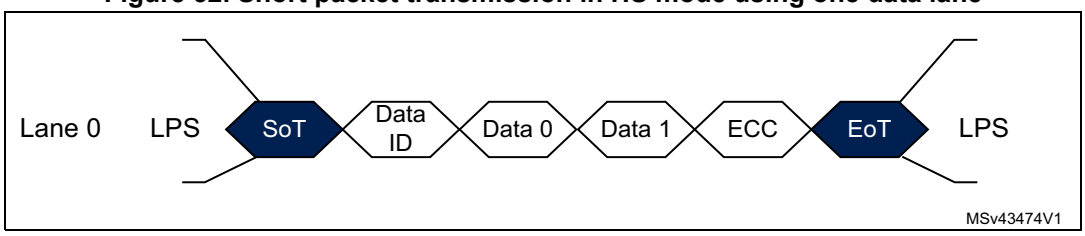


Figure 33 shows a short packet transmission in HS mode using two data lanes.

Figure 33. Short packet transmission in HS mode using two data lanes

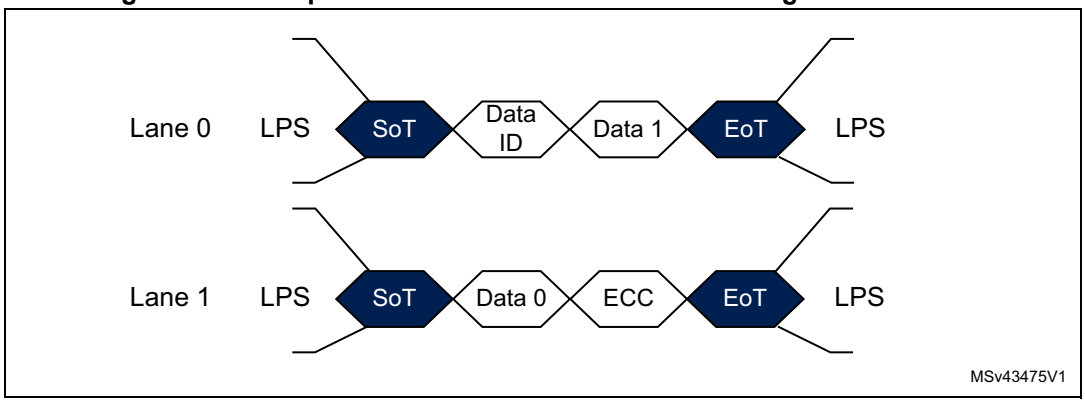


Figure 34 shows a short packet transmission in LP mode.

Note: In LP mode, only the data lane 0 is used for transmission.

Figure 34. Short packet transmission in low-power mode

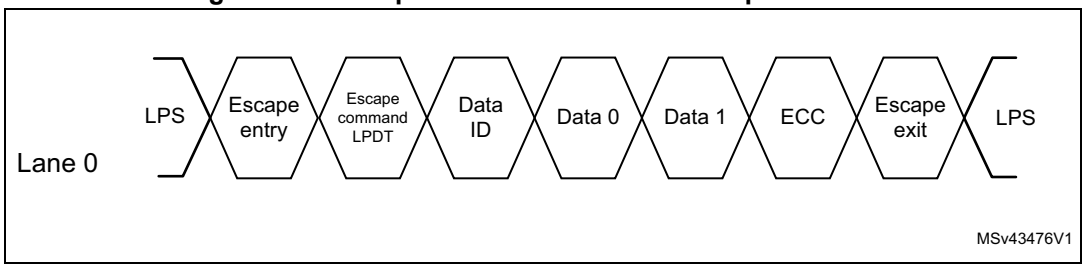
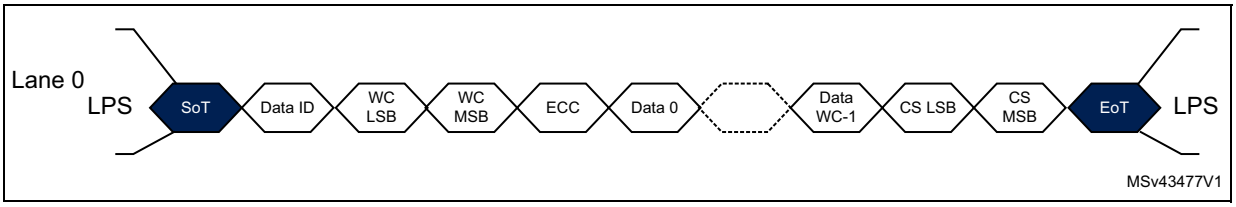


Figure 35 shows a long packet transmission in HS mode using one data lane.

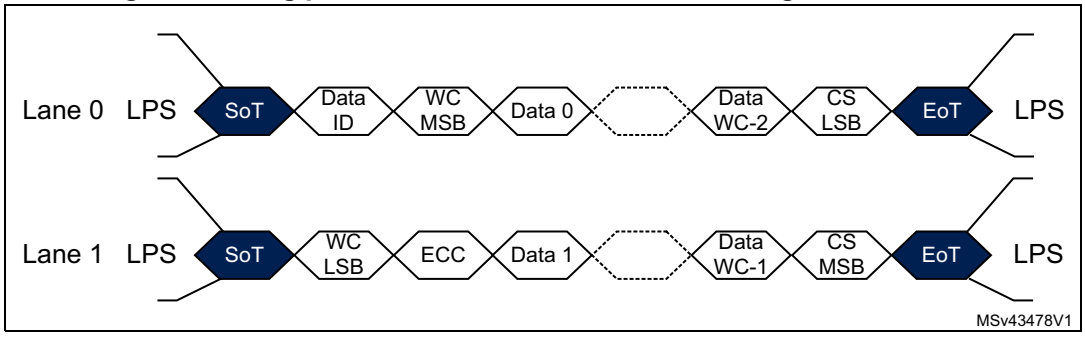
Figure 35. Long packet transmission in HS mode using one data lane



In HS mode, the data transmission can be done using multiple lanes.

Figure 36 shows a long packet transmission in HS mode using two data lanes.

Figure 36. Long packet transmission in HS mode using two data lanes



If the number of payload data is not an integer multiple of the number of lanes, some lanes may complete the HS transmission before the other lanes, sending an EoT one cycle (byte) earlier.

Figure 37 shows an example using two data lanes. In this case, the payload data has an odd number of bytes causing data lane 1 to complete the HS transmission before data lane 0.

Figure 37. Long packet transmission in HS mode using two data lanes with odd payload data

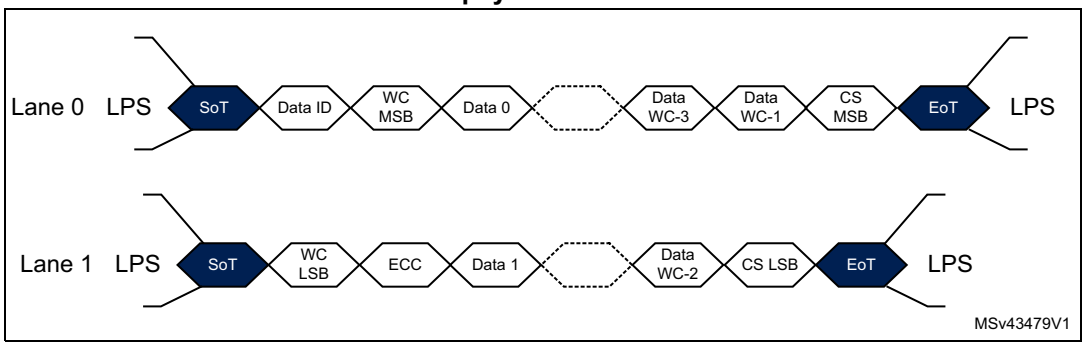
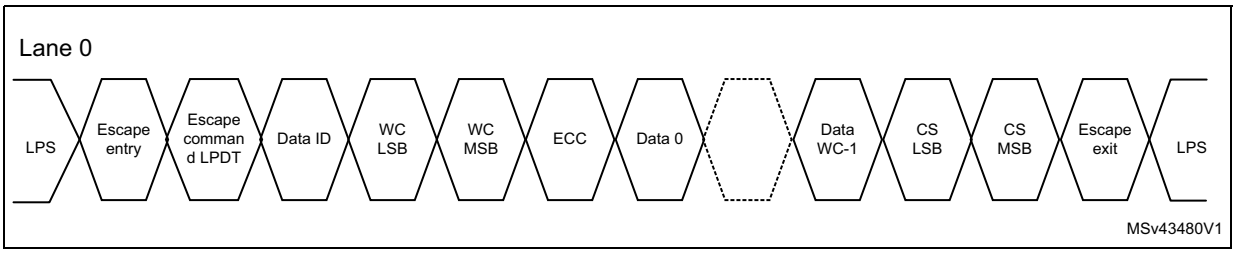


Figure 38 shows a long packet transmission in LP mode.

Figure 38. Long packet transmission in low-power mode



3.3.4 Host to display data types

The host to display data types can be short packets or long packets. They can be either video or command data types. The host to display data types are shown in *Table 7*:

Table 7. Host to display data types

Data type	Description	Packet	DSI mode
0x01	Sync event, V sync start	Short	Video
0x11	Sync event, V sync end	Short	Video
0x21	Sync event, H sync start	Short	Video
0x31	Sync event, H sync end	Short	Video
0x08	End of transmission packet (EoTp)	Short	Both
0x02	Color mode (CM) OFF command	Short	Video
0x12	Color mode (CM) ON command	Short	Video
0x22	Shut down peripheral command	Short	Video
0x32	Turn ON peripheral command	Short	Video

Table 7. Host to display data types (continued)

Data type	Description	Packet	DSI mode
0x03	Generic short write, no parameters	Short	Command
0x13	Generic short write, 1 parameter	Short	Command
0x23	Generic short write, 2 parameters	Short	Command
0x04	Generic read, no parameters	Short	Command
0x14	Generic read, 1 parameter	Short	Command
0x24	Generic read, 2 parameters	Short	Command
0x05	DCS short write, no parameters	Short	Command
0x15	DCS short write, 1 parameter	Short	Command
0x06	DCS read, no parameters	Short	Command
0x37	Set maximum return packet size	Short	Command
0x09	Null packet, no data	Long	Video
0x19	Blanking packet, no data	Long	Video
0x29	Generic long write	Long	Command
0x39	DCS long write/Write_LUT	Long	Command
0x0C	Loosely packed pixel stream 20-bit YCbCr, 4:2:2 Format	Long	Video
0x1C	Packed pixel stream 24-bit YCbCr, 4:2:2 Format	Long	Video
0x2C	Packet pixel stream 16-bit YCbCr, 4:2:2 Format	Long	Video
0x0D	Packet pixel stream 30-bit RGB, 10-10-10 Format	Long	Video
0x1D	Packet pixel stream 36-bit RGB, 12-12-12 Format	Long	Video
0x3D	Packet pixel stream 12-bit YCbCr, 4:2:0 Format	Long	Video
0x0E	Packet pixel stream 16-bit RGB, 5-6-5 Format	Long	Video
0x1E	Packet pixel stream 18-bit RGB, 6-6-6 Format	Long	Video
0x2E	Loosely packed pixel stream 18-bit RGB, 6-6-6 Format	Long	Video
0x3E	Packed pixel stream 24-bit RGB, 8-8-8 Format	Long	Video

Video mode data types

The video mode data types are mainly used to convey synchronization events and pixel data. The synchronization events and the pixel data packets are sent in HS mode since they convey timing critical information.

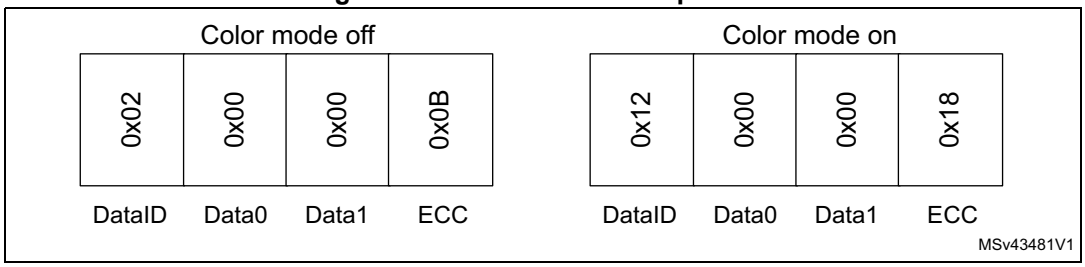
Other video mode data types are the ones used for color mode and shutdown control. They can be sent either in HS or is LP.

Note: All examples shown below are with VC = 0.

1. Shut down and color modes

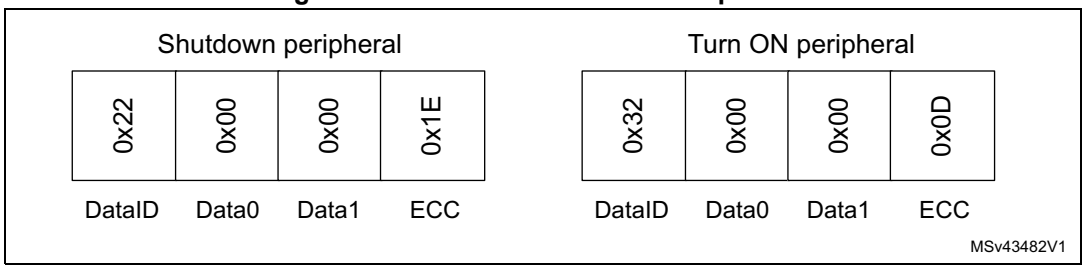
   The color mode commands are short packets that allow to switch the display module between normal mode and low-color mode. The low-color mode is used for power saving purposes (see *Figure 39*).

Figure 39. Color mode video packets



The shutdown and the turn on commands are short packets used to switch on or off the display module (see *Figure 40*).

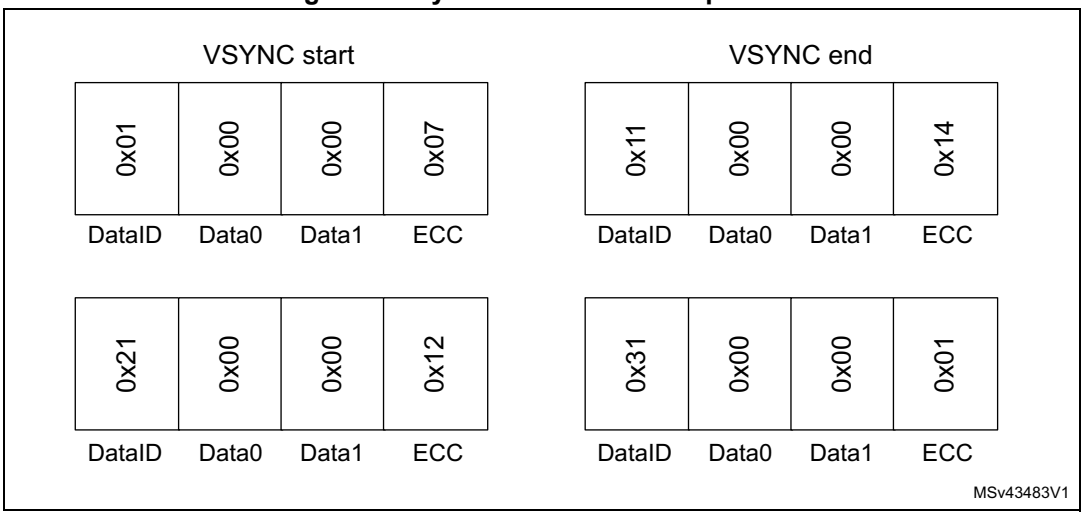
Figure 40. Shutdown/turn-on video packets



2. Synchronization events

   The synchronization events are sent through short packets since short packets are more suitable to convey accurate timing information (see *Figure 41*).

Figure 41. Synchronization event packets



3. Packed pixel streams

The packed pixel stream (PPS) packets are long packets used to transmit RGB image data formatted as pixels to a video mode display module.

The packet consists of the DI byte, a two-byte WC, an ECC byte, a payload of length WC bytes and a two-byte checksum.

The DSI protocol defines packed pixel streams for different color coding:

– Packed pixel stream, 16-bit format. Refer to *Figure 42*.
– Packed pixel stream, 18-bit format. Refer to *Figure 43*.
– Loosely packed pixel stream, 18-bit format. Refer to *Figure 44*.
– Packed pixel stream, 24-bit format. Refer to *Figure 45*.

Figure 42. Packed pixel stream, 16-bit format, data type (0x0E)

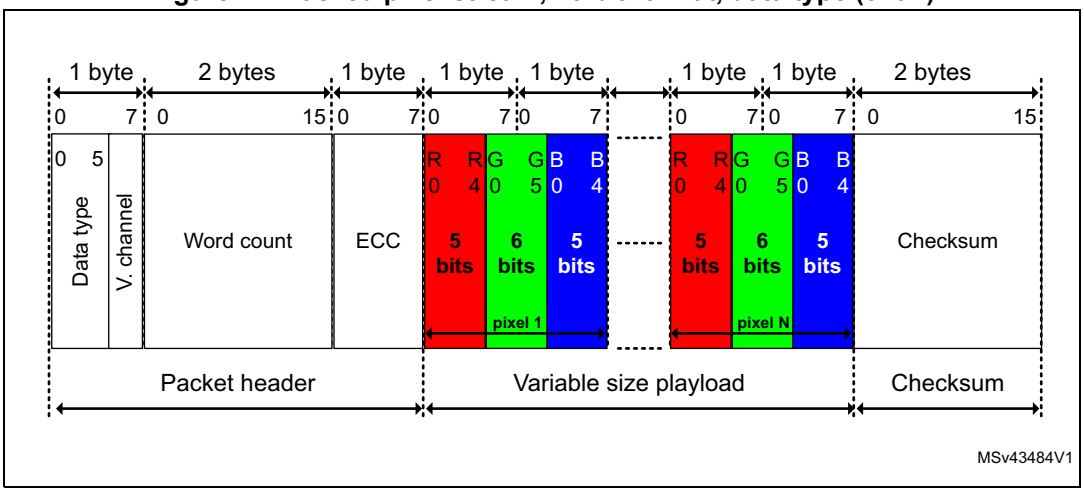


Figure 43. Packed pixel stream, 18-bit format, data type = (0x1E)

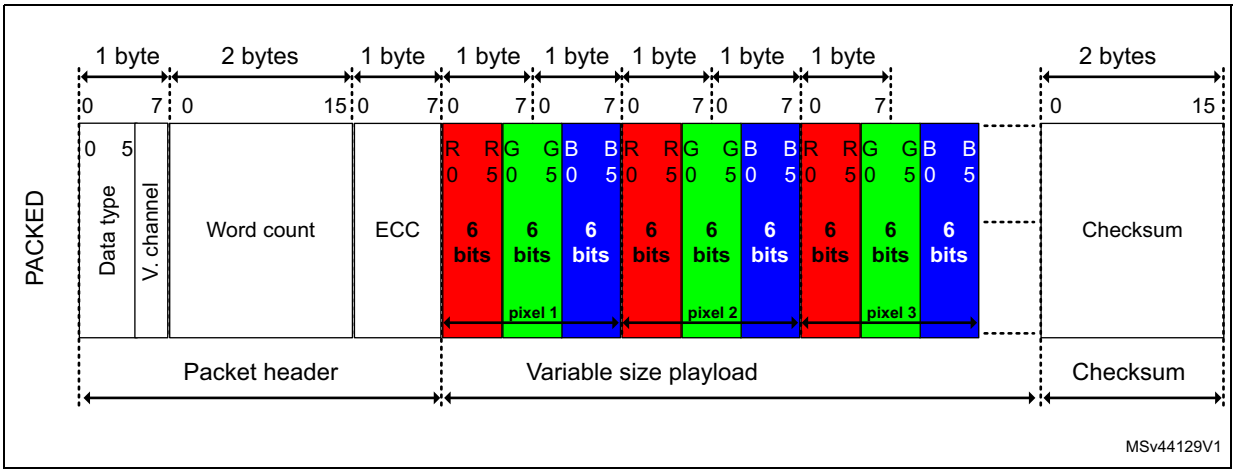


Figure 44. Loosely packed pixel stream, 18-bit format, data type = (0x2E)

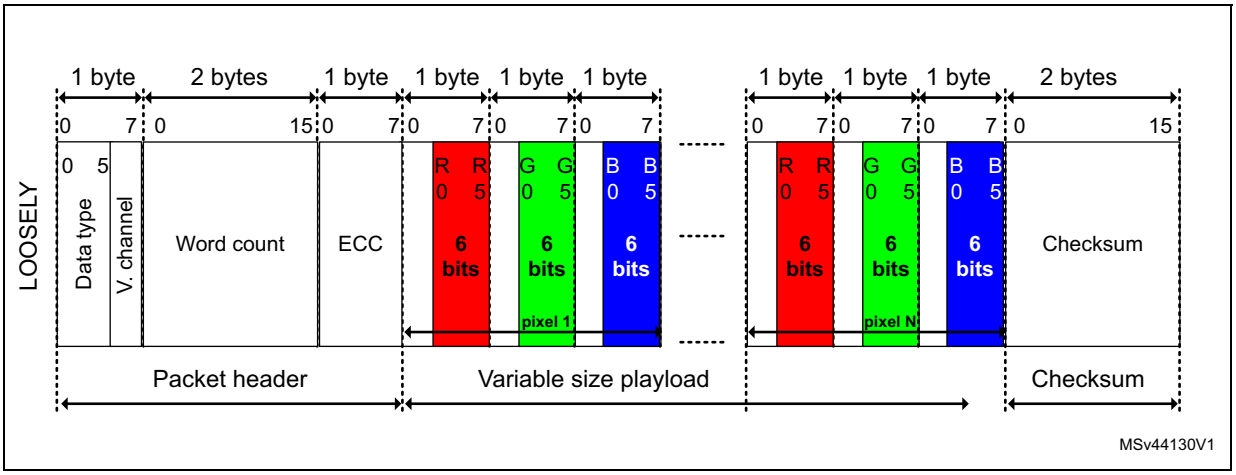
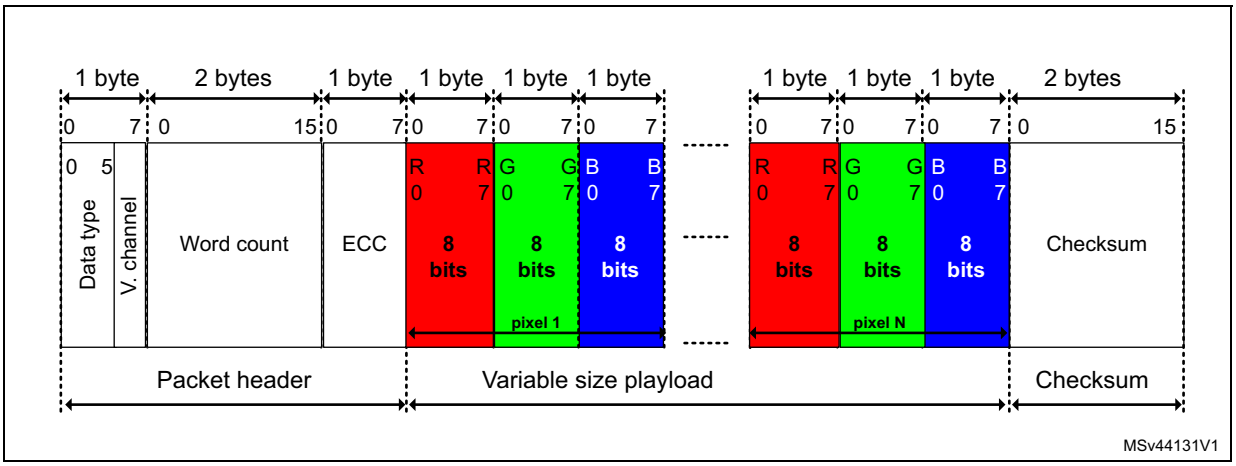


Figure 45. Packed pixel stream, 24-bit format, data type = (0x3E)



Command mode data types

The command mode data types may be sent in HS or LP. They are used to write to and read from the display registers and from the frame buffer.

1. Generic commands

   There are three types of generic commands:

   – Generic short write. Refer to *Figure 46*.
   – Generic long write. Refer to *Figure 47*.
   – Generic read. Refer to *Figure 48*.

Figure 46. Generic short write commands

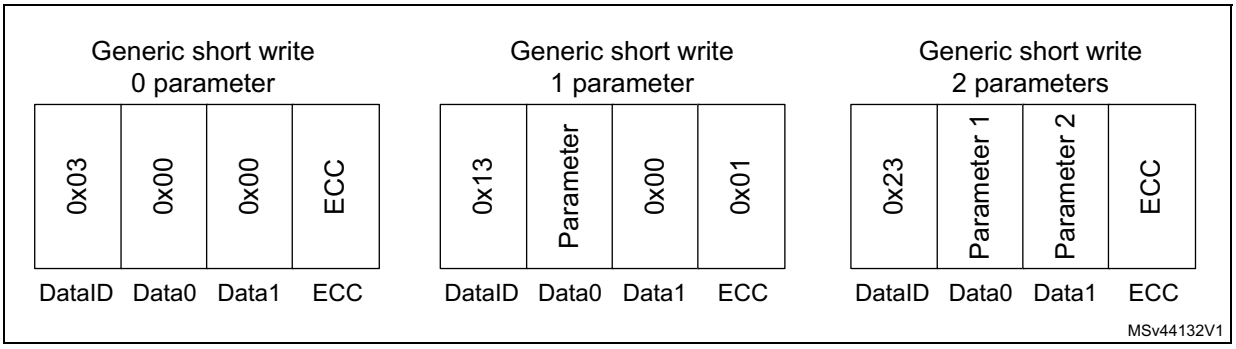


Figure 47. Generic long write commands

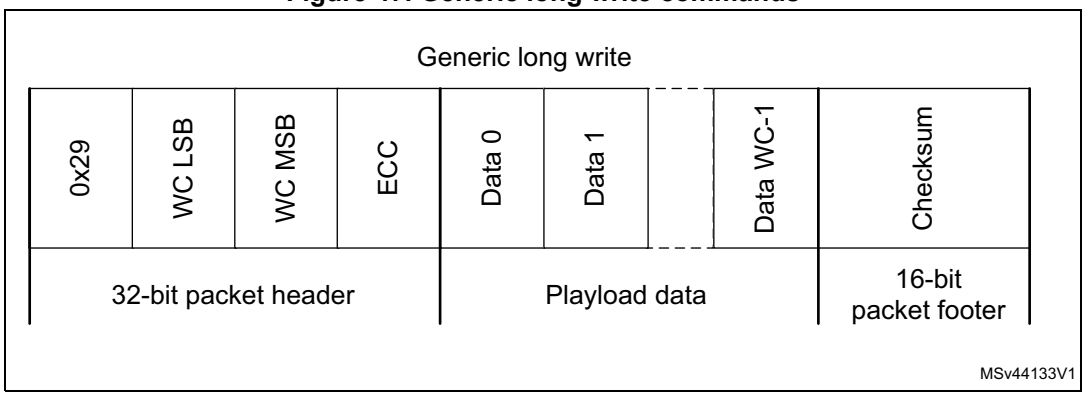
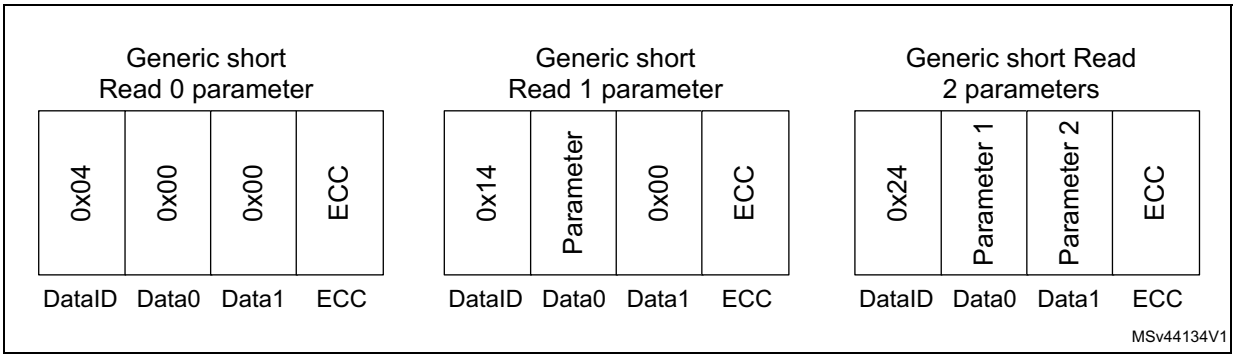


Figure 48. Generic read commands



2. DCS commands

   DCS is a standardized command set defined by the MIPI Alliance intended for the command mode displays. The DCS commands are listed in *Table 8*.

   The DCS commands with a 0 or 1 parameter are sent using short packets, while the DCS commands with more than one parameter are sent using long packets.

   There are three types of DCS commands:

   – DCS short write. Refer to *Figure 49*.

     DCS short write with no parameters is sent using a shot packet with DT 0x05. The DCS command index is placed in the data 0 field. The data 1 field is not used. It is set to 0.

     The DCS short write with one parameter have DT 0x15. The DCS command index is placed in the data 0 field and the parameter is placed in the data 1 field.

   – DCS long write. Refer to *Figure 50*.

   – DCS read. Refer to *Figure 51*.

     The read command must be followed by a BTA to give the bus control to the display, so that the latter can send the response.

     The response may be a DCS short or long read response (see *Table 9*).

Figure 49. DCS short write command

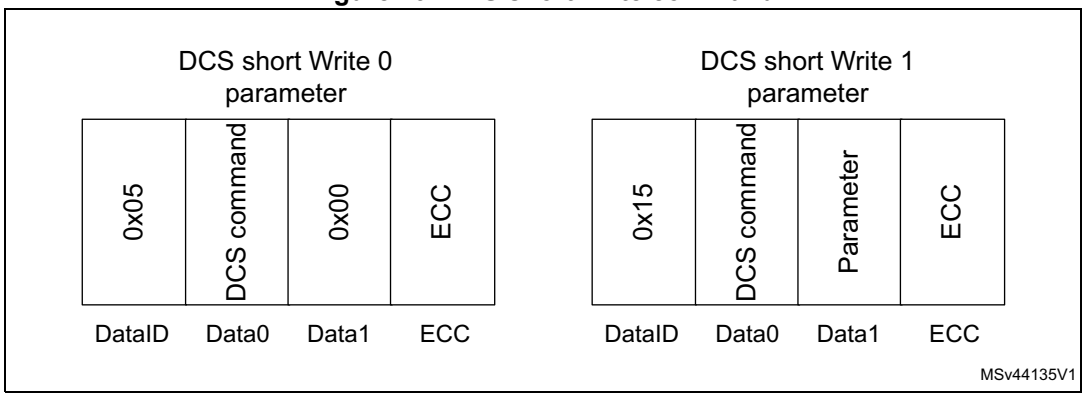


Figure 50. DCS long write command

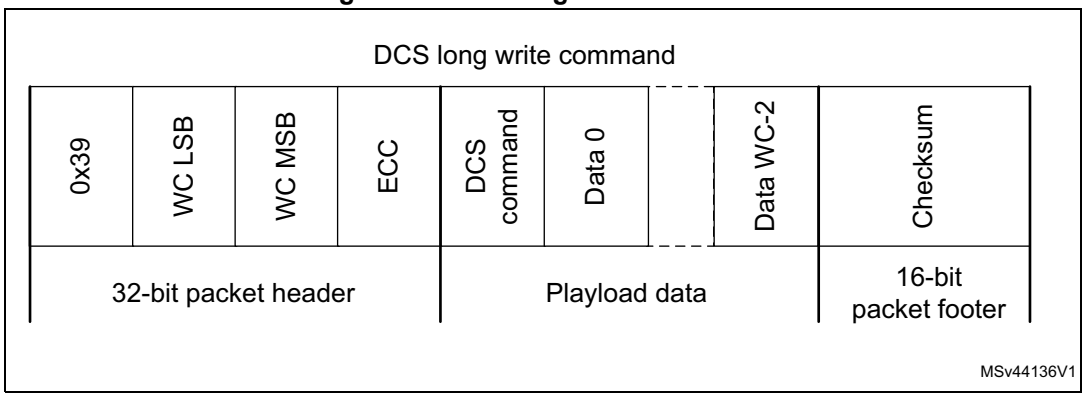
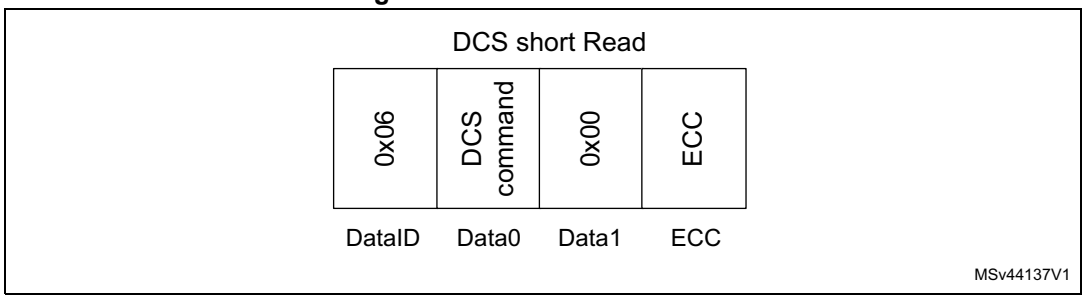


Figure 51. DCS read command



The *Table 8* below shows a list of the DCS commands.

Table 8. DCS command list

Command	Hex code	Description
enter_idle_mode	39h	Reduced color depth is used on the display panel.
enter_invert_mode	21h	Displayed image colors are inverted.
enter_normal_mode	13h	The whole display area is used for image display.
enter_partial_mode	12h	Part of the display area is used for image display.
enter_sleep_mode	10h	Power for the display panel is off.
exit_idle_mode	38h	Full color depth is used on the display panel.
exit_invert_mode	20h	Displayed image colors are not inverted.
exit_sleep_mode	11h	Power for the display panel is on.
get_3D_control	3Fh	Get display module 3D mode.
get_address_mode	0Bh	Get the data order for transfers from the host to the display module and from the frame memory to the display device.
get_blue_channel	08h	Get the blue component of the pixel at (0, 0).
get_diagnostic_result	0Fh	Get peripheral self-diagnostic result.
get_display_mode	0Dh	Get the current display mode from the peripheral.
get_green_channel	07h	Get the green component of the pixel at (0, 0).
get_pixel_format	0Ch	Get the current pixel format.
get_power_mode	0Ah	Get the current power mode.
get_red_channel	06h	Get the red component of the pixel at (0, 0).
get_scanline	45h	Get the current scanline.
get_signal_mode	0Eh	Get display module signaling mode.
nop	00h	No operation.
read_DDB_continue	A8h	Continue reading the DDB from the last read location.
read_DDB_start	A1h	Read the DDB from the provided location.
read_memory_continue	3Eh	Read image data from the peripheral continuing after the last read_memory_continue or read_memory_start.
read_memory_start	2Eh	Transfer image data from the peripheral to the host processor interface starting at the location provided by set_column_address and set_page_address.
set_3D_control	3Dh	3D is used on the display panel.
set_address_mode	36h	Set the data order for transfers from the host to the display module and from the frame memory to the display device.
set_column_address	2Ah	Set the column extent.
set_display_off	28h	Blanks the display device.
set_display_on	29h	Show the image on the display device.

Table 8. DCS command list (continued)

Command	Hex code	Description
set_gamma_curve	26h	Selects the gamma curve used by the display device.
set_page_address	2Bh	Set the page extent.
set_partial_columns	31h	Defines the number of columns in the partial display area on the display device.
set_partial_rows	30h	Defines the number of rows in the partial display area on the display device.
set_pixel_format	3Ah	Defines how many bits per pixel are used in the interface.
set_scroll_area	33h	Defines the vertical scrolling and fixed area on display device.
set_scroll_start	37h	Defines the vertical scrolling starting point.
set_tear_off	34h	Synchronization information is not sent from the display module to the host processor.
set_tear_on	35h	Synchronization information is sent from the display module to the host processor at the start of VFP.
set_tear_scanline	44h	Synchronization information is sent from the display module to the host processor when the display device refresh reaches the provided scanline.
set_VSYNC_timing	40h	Set VSYNC timing.
soft_reset	01h	Software reset.
write_LUT	2Dh	Fills the peripheral lookup table with the provided data.
write_memory_continue	3Ch	Transfer image information from the host processor interface to the peripheral from the last written location.
write_memory_start	2Ch	Transfer image data from the host processor to the peripheral starting at the location provided by set_column_address and set_page_address.

3. Set maximum return packet size

   The *set maximum return packet size* command allows the host processor to limit the size of the response packets coming from a peripheral.

   The initial value of the maximum return packet size is one byte. To get more than one byte, the host has to send this command with the desired size as a parameter before sending a read command.

Note: The maximum return packet size must not exceed the host read FIFO size in order to avoid an overflow.The DSI host read FIFO has 32 x 32-bit words size.

3.3.5 Display to host data types

All the command mode systems support the reverse direction communication. In the other hand, the support for reverse communication is optional for the video mode systems.

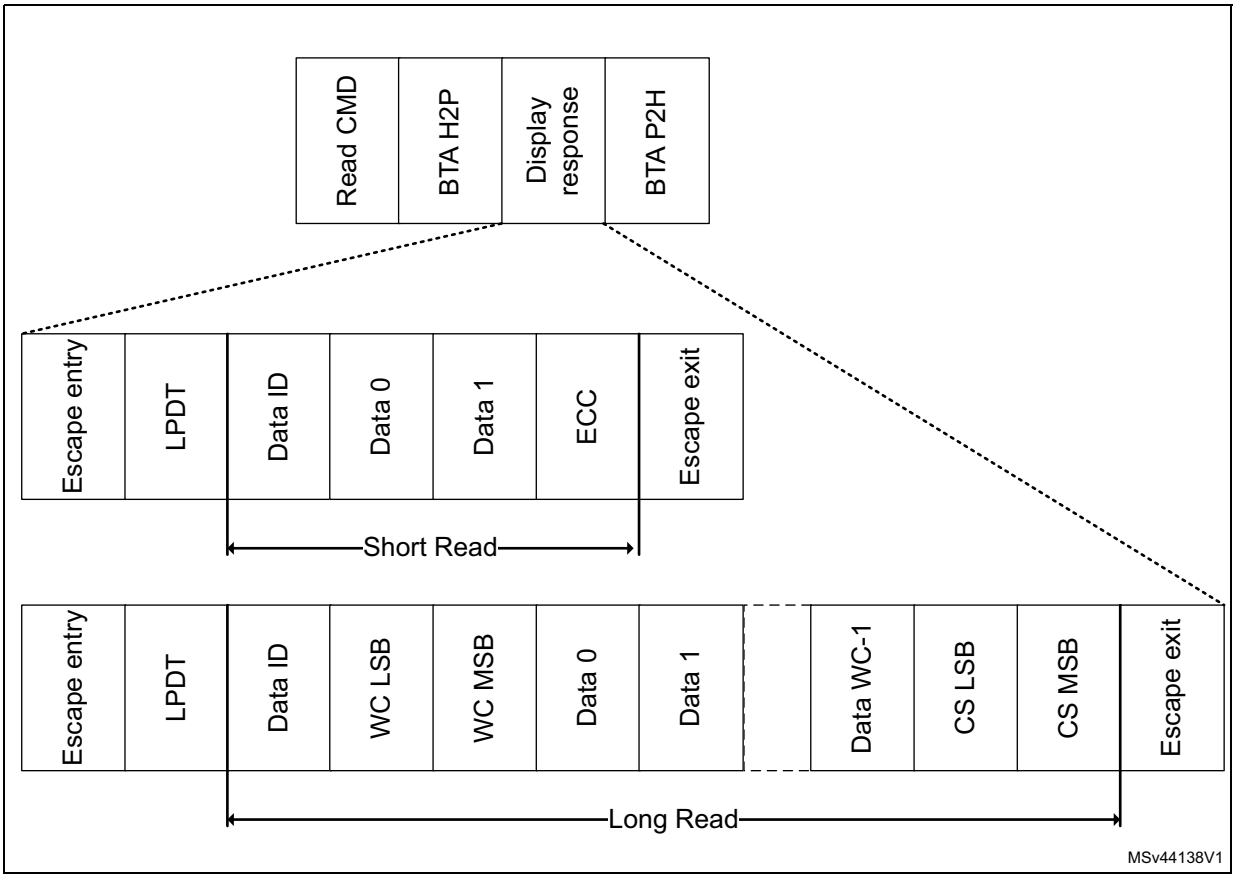
The display to host transmissions use the same short and long packet structures as the forward transmissions.

The display to host communication may only begin after the host gives the bus ownership to the display by using a BTA sequence.

After sending the response, the display gives back the bus control to the host through a BTA sequence.

Figure 52 shows and example of a reverse communication sequence. Note that the display to host transmissions happen in LP mode using the data lane 0.

Figure 52. Reverse communication sequence



The display to host packet types are listed in *Table 9* below.

Table 9. Display to host data types

Data Type	Description	Packet
0x02	Acknowledge and error report	Short
0x08	End of transmission packet (EoTp)	Short
0x11	Generic short read response 1 byte returned	Short
0x12	Generic short read response 2 bytes returned	Short
0x1A	Generic long read response	Long
0x1C	DCS long read response	Long

Table 9. Display to host data types (continued)

Data Type	Description	Packet
0x21	DCS short read response 1 byte returned	Short
0x22	DCS short read response 2 bytes returned	Short

3.3.6 Video mode interface timing

The video mode peripherals require that the pixel data is delivered in real time. The DSI supports several formats for the video mode data transmission:

- Nonburst mode with sync pulses: enables the peripheral to accurately reconstruct the original video timing, including the sync pulse widths. Please refer to *Figure 53*. If there is no sufficient time to switch into LP mode, the time interval in LP is substituted by blanking packets.
- Nonburst mode with sync events: similar to above mode, but for which an accurate reconstruction of sync pulse widths is not required, hence a single sync event is substituted. Refer to *Figure 54*.
- Burst mode: the RGB pixel packets are time-compressed, leaving more time during a scanline for the LP mode (saving power) or for the multiplexing of other transmissions onto the DSI link. Refer to *Figure 55*.

Note: For an accurate reconstruction of timing, the packet overhead (including data ID, ECC, and checksum bytes) should be considered.

The DSI host supports all the three video mode packet sequences. The DSI display supports at least one of the modes described above.

Figure 53. Nonburst mode with sync pulses

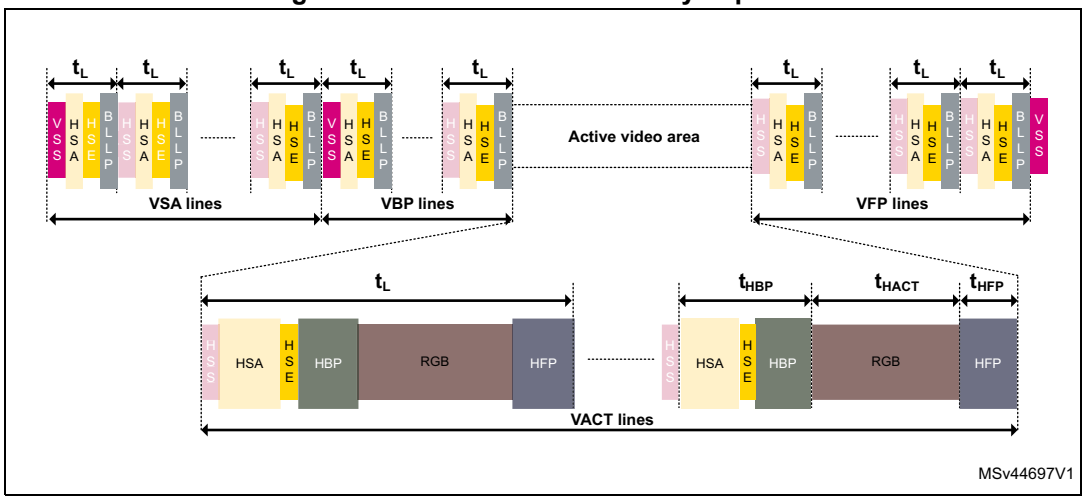


Figure 54. Nonburst mode with sync events

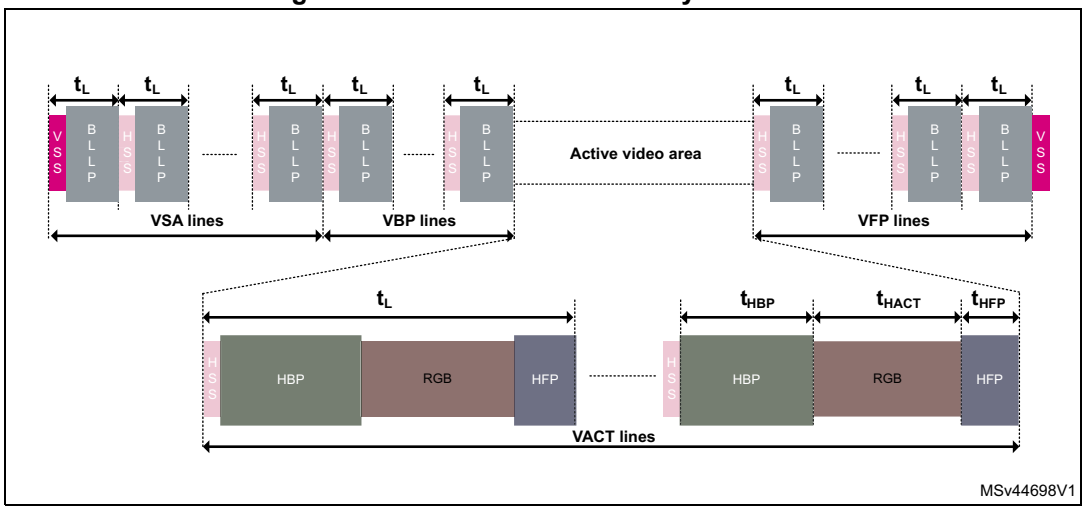
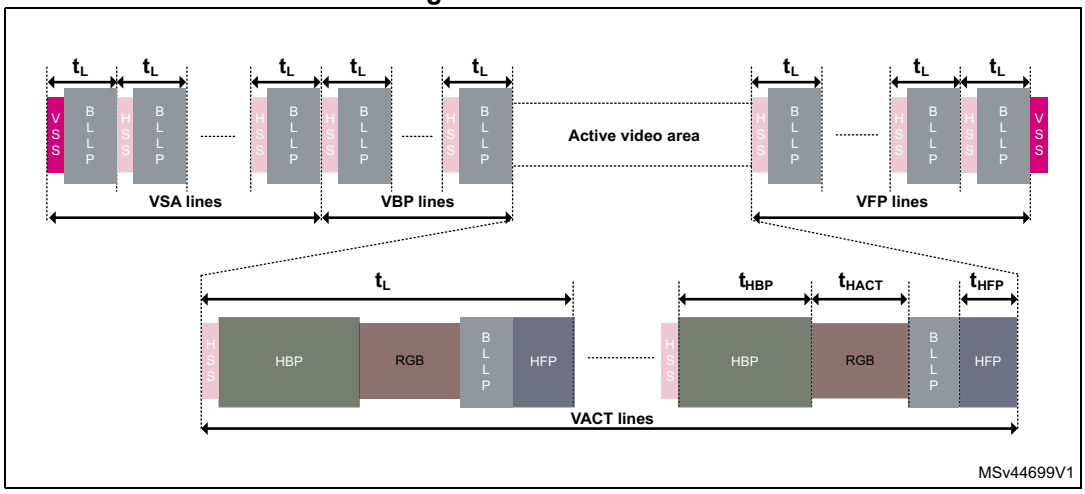


Figure 55. Burst mode



3.3.7 Tearing effect signaling in command mode

The tearing effect (TE) is a visual artifact where a display device shows information from two or more frames in a single screen draw. The artifact occurs when the display GRAM update done by the host is not synchronized with the display's scan process.

The tearing effect signaling is a synchronization mechanism that allows the host to be notified of the timing events at the display side. It is used in command mode systems where the display has its own controller and uses its internal GRAM to refresh the display.

In command mode systems, the display notifies the host through a TE signaling whenever it reaches a specific scanline. This action allows the host to know when to update the display GRAM without causing a tearing effect.

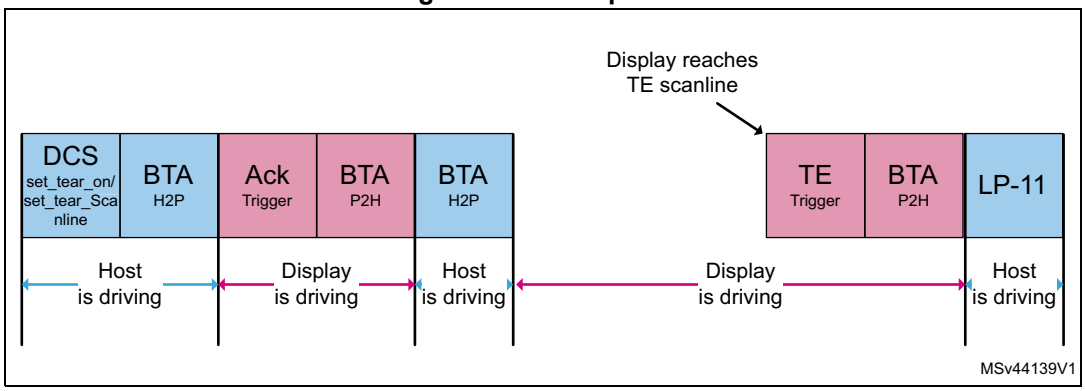
When a TE reporting is required, the DSI host sends set_tear_on or set_tear_scanline DCS commands followed by a double BTA (bus turnaround) procedure to give the bus control to the display.

The double BTA is required because the display's DSI protocol layer responds to the first BTA with an acknowledge trigger or error report and gives back the bus control to the host (as it does not interpret DCS commands).

In order to allow a TE reporting, the host has to perform a second BTA to give the bus control to the display. The display responds with a TE trigger (01011101) as soon as the scanline is reached.

Figure 56 illustrates the tearing effect sequence.

Figure 56. TE sequence



1. BTA H2P is a BTA procedure initiated by the host to give bus control to the peripheral. BTA P2H is a BTA procedure initiated by the peripheral to give bus control to the host.

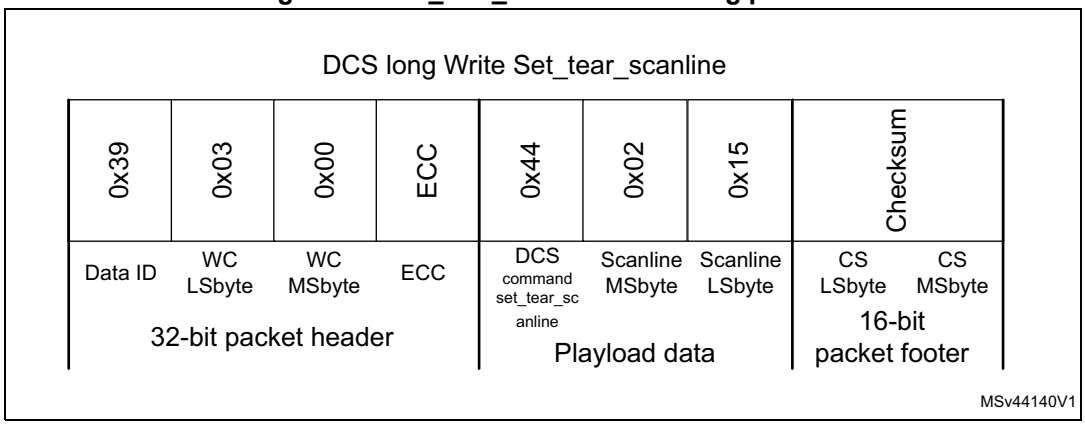
The Set_tear_scanline enables the TE reporting in the same way than set_tear_on, but Set_tear_scanline also defines the scanline at which the TE reporting should happen. The Set_tear_scanline DCS long packet is illustrated in *Figure 57* where the TE reporting is required when the display reaches line 533 (in this example).

Generally set_tear_scanline is used at the initialization phase to program the TE scanline, then the host sends set_tear_on whenever it needs to enable a TE reporting.

The Set_tear_scanline command is a long DCS write command. It takes two parameters that define the TE scanline.

Note: The TE scanline is sent using two bytes, most significant byte first.

Figure 57. Set_tear_scanline DCS long packet



4 DSI host description

This section describes the STM32 DSI host system level architecture and operating modes and provides some guidance to select the appropriate operating mode depending on the application needs.

4.1 DSI system level architecture

The DSI host is deeply integrated with the LTDC. It relies on the LTDC to fetch the pixel data and to provide the video synchronization signals.

The DSI host has two system interfaces:

- The LTDC interface.
  This interface allows the DSI host to capture the pixel data and the video synchronization signals from the LTDC and encapsulates them into DSI packets.
  - The video packets (video synchronization events, pixel packet streams) when in video mode.
  - The memory_write_start (WMS) and memory_write_continue (WMC) DCS commands, when in adapted command mode.
- The APB interface.
  This interface is used for the transmission of DCS and generic command mode packets. These packets are built using the APB register access.
  It can be accessed concurrently in adapted or video mode to transfer DCS or generic packets to the display.

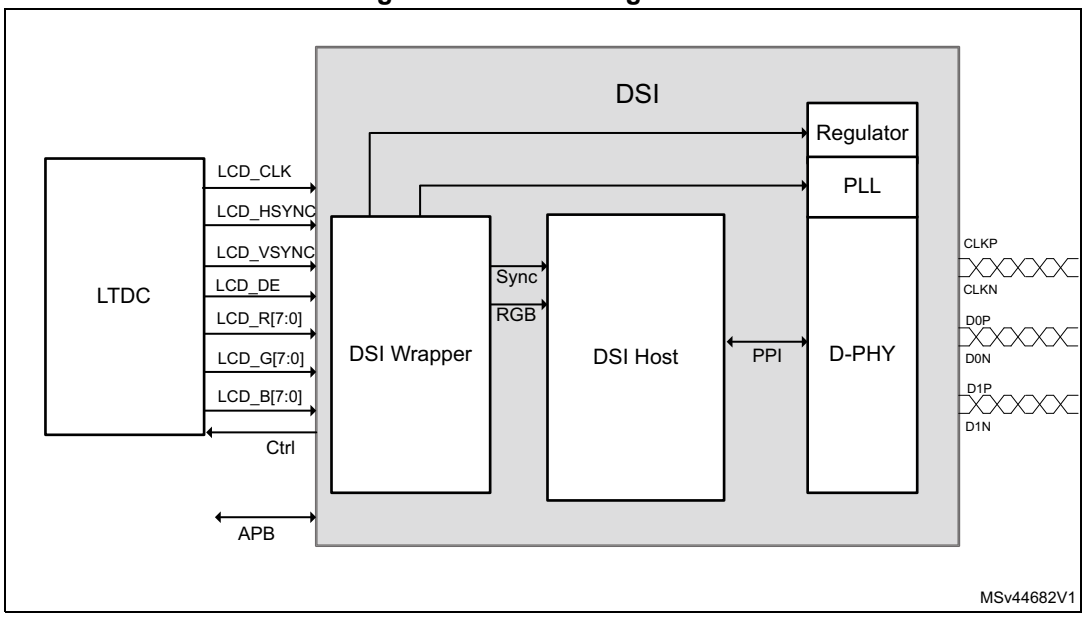
DSI host building blocks

The DSI system building blocks are:

- The DSI Wrapper: interconnects the LTDC to the DSI host to redirect the pixel data and video control signals in either video or adapted command mode.
  The DSI Wrapper also controls the DSI regulator, the DSI PLL and some specific functions of the D-PHY.
- The DSI host controller: builds long or short DSI packets and generates correspondent ECC and CRC codes. The packets are sent in bytes to the D-PHY for serialization.
  In the case where multiple lanes are used, the DSI host also performs data splitting between the available D-PHY lanes.
- The D-PHY: serializes data coming from the DSI host controller and sends it through the serial link.
- The internal PLL: generates the HS clock used by the D-PHY.
- The internal regulator: provides 1v2 supply to the PLL and D-PHY.

The DSI building blocks are shown in *Figure 58*.

Figure 58. DSI building blocks



4.2 Operating modes

The STM32 DSI host supports all the operating modes defined in the DSI specification version 1.1. It supports the command mode and the video mode sequences (burst, nonburst with sync events).

The STM32 DSI host supports also an enhanced command mode, for optimized operation with displays embedding GRAM.

The three operating modes available to convey the graphical data to the display are:

- The video mode.
  It is used to stream over the high-speed link the RGB data and the associated synchronization signals (VSYNC, HSYNC) directly generated by the LTDC.
  The LTDC interface captures the data and synchronization signals and conveys them to the FIFO interfaces.
  Two different streams of data are present on the interface; video control signals and pixel data.
  The DSI host uses both streams of data to build the DSI video packets, which are then transmitted over the DSI link.
  The streaming starts as soon as the DSI host and the LTDC are enabled. This continuous refresh is the best way to get interfaced with a display without graphical RAM.
- The adapted command mode.
  This mode is the smartest way to get interfaced with a display that has its own internal graphical RAM. The DSI host captures only one full frame coming from the LTDC and transforms it into a series of DCS write commands to update the display graphical RAM. This one-shot refresh is automatically done when setting a control bit in the DSI host.
- The APB command mode.
  This mode is used to send commands over the high-speed link for configuration as if it was done using a SPI. The commands are launched using the DSI host APB interface.

4.2.1 Video mode

The DSI host video mode supports the three operating modes defined by the DSI specification.n.

The video mode is used with displays that does not embed a graphic RAM able to hold an entire frame.

The video mode displays rely on the DSI host to provide a continuous pixel stream since they do not have an internal controller.

Nonburst mode with sync pulses

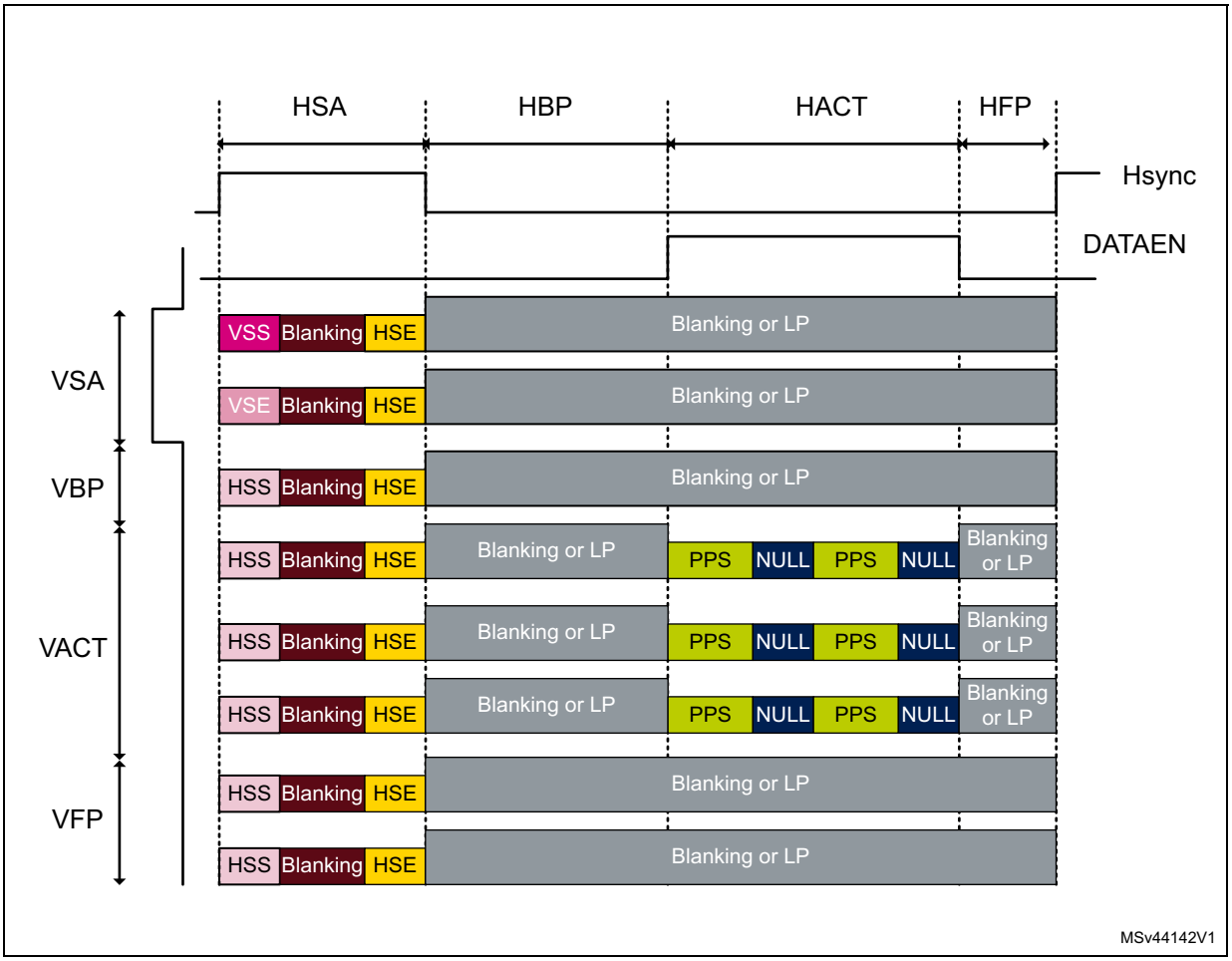
In nonburst mode, the processor uses the partitioning properties of the DSI host to divide the video line transmission into several chunks of pixels and optionally interleaves them with null packets. This is done to match the LTDC interface input pixel bandwidth with the DSI link bandwidth.

In this mode, the host controller and the display do not require a full line of pixel data to be stored. This mode requires only the content of one video packet to be stored which minimizes memory requirements.

This mode enables the peripheral to accurately reconstruct the original video timing, including sync pulse widths.

A typical frame in nonburst mode with sync pulses is shown in *Figure 59*.

Figure 59. Nonburst with sync pulses frame



The HSA (HSYNC active) period is composed of HSS (HSYNC start), blanking and HSE (HSYNC end) packets. The packets in this region are transmitted in high-speed mode and the link does not go to LP during the HSA period.

The DSI host automatically calculates the blanking packet size required to match the timing between the HSS and HSE packets, with the HSA period time.

When the DSI host detects a VSYNC rising edge (supposing VSYNC signal is active high), the DSI host starts an HSA period with a VSYNC start (VSS) packet instead of an HSS packet.

When the VSYNC falling edge is detected, the DSI host sends a VSYNC end (VSE) packet instead of an HSS packet to mark the end of the VSA period.

Other lines inside the VSYNC active (VSA) region are started with an HSS packet.

Outside the vertical active (VACT) period, the link goes to LP after an HSA period until the end of the horizontal line.

In the VACT region, the DSI host conveys an HSA period as described above, then the link either goes to LP or sends a blanking packet for a timing period equal to an horizontal back porch (HBP) period. Then the DSI host sends a PPS (packed pixel stream) packet in one or more chunks with eventually null packets to match the pixel transmission timing with the horizontal active (HACT) period.

Once the HACT period has finished, the DSI host either goes to LP or sends blanking packets for a period equal to a horizontal front porch (HFP) period.

Note: When regions are tagged with blanking or LP, it means that the DSI host may send a high-speed blanking packet for the total period time, or that the DSI host may put the link into low-power mode.

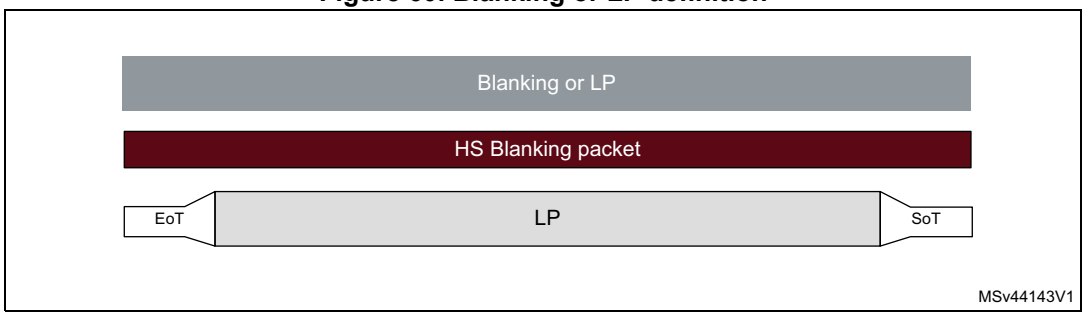
The DSI host can calculate the number of bytes needed inside a blanking packet in order to match the period timing.

In the low-power mode case, the DSI PHY initiates an end of transmission sequence before going to the low-power mode. Then, before starting a new HS transmission, the DSI PHY issues a start of transmission sequence.

The DSI host needs some inputs from the user to know the overhead of the low-power transition (EoT and SOT sequences) to know if switching to low-power is possible during a specific period.

Figure 60 shows the two possible scenarios for blanking or low-power regions.

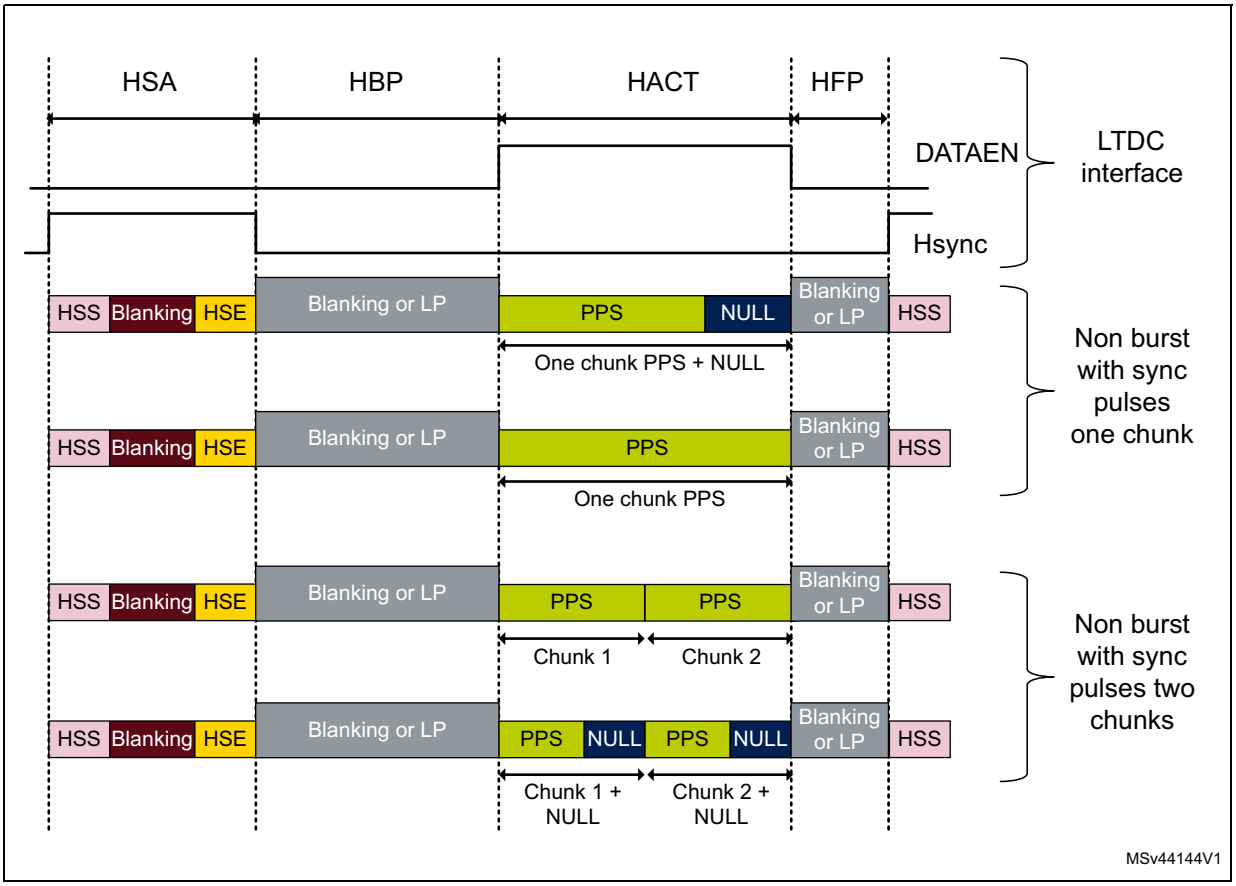
Figure 60. Blanking or LP definition



During the VACT period, there are many different configurations of a horizontal line.

Figure 61 shows different possible configurations of VACT region lines.

Figure 61. VACT region in nonburst mode with sync pulses



During HBP and HFP periods the link may go to LP, or the DSI host sends a blanking packet if there is no sufficient time to transition between HS and LP mode during this period.

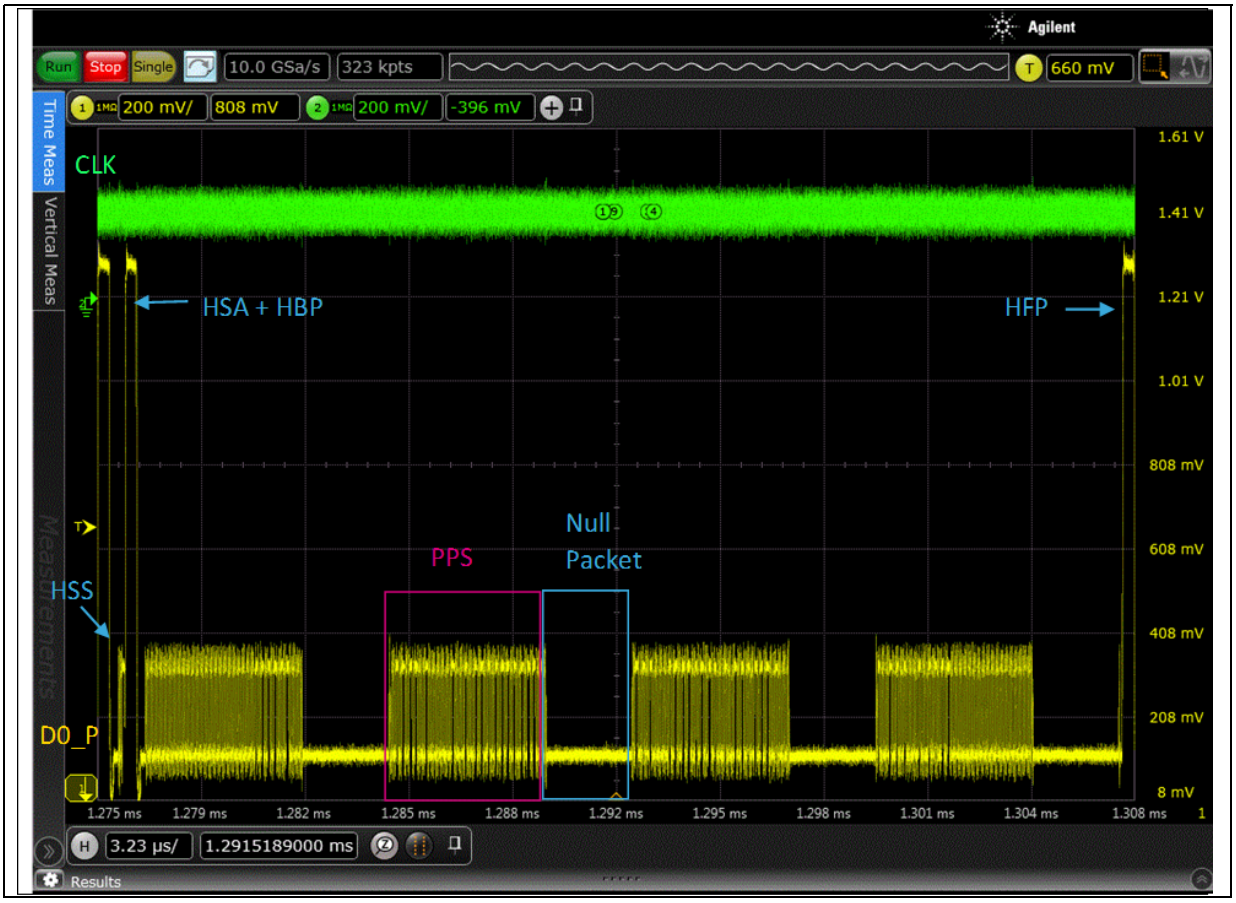
During the HACT period, the DSI host must send pixel data in a timing period that matches the LTCD HACT period. Depending on the DSI and on the pixel clock frequency, the DSI host may send the pixel data using one or more chunks.

Each chunk may contain only a packet pixel stream (PPS) packet or a PPS packet with null packet.

The choice of the number of chunks and the size of null packets in nonburst mode is discussed in *Section 5.2.1: Video mode over LTDC interface*.

Figure 62 shows an active line with four chunks. Each chunk is composed of a PPS packet and a null packet.

Figure 62. Active line with four chunks with null packets configuration



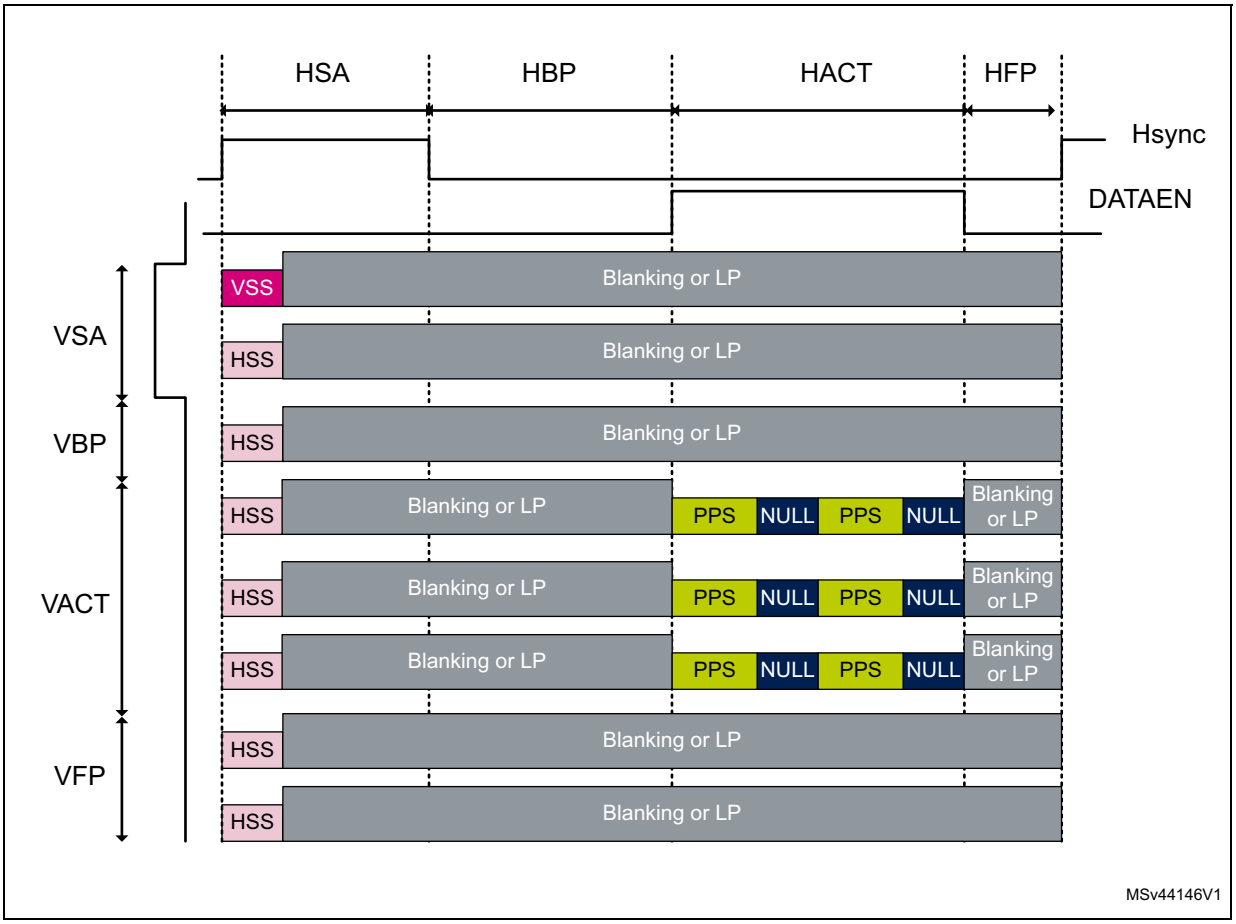
Nonburst mode with sync events

In this mode, an accurate reconstruction of the sync pulse widths is not required, so a single sync event is substituted. In this mode:

- Only the start of each synchronization pulse is transmitted.
- The transmission is done at the same rate than the LTDC.
- If there is no sufficient time to switch into LP mode, the time interval in LP is substituted by blanking packets.

A typical frame in nonburst mode with sync events is shown in *Figure 63*.

Figure 63. Nonburst with sync events frame



The start of the VSA region is determined by a line starting with a VSS packet. All other lines in the frame start with an HSS packet.

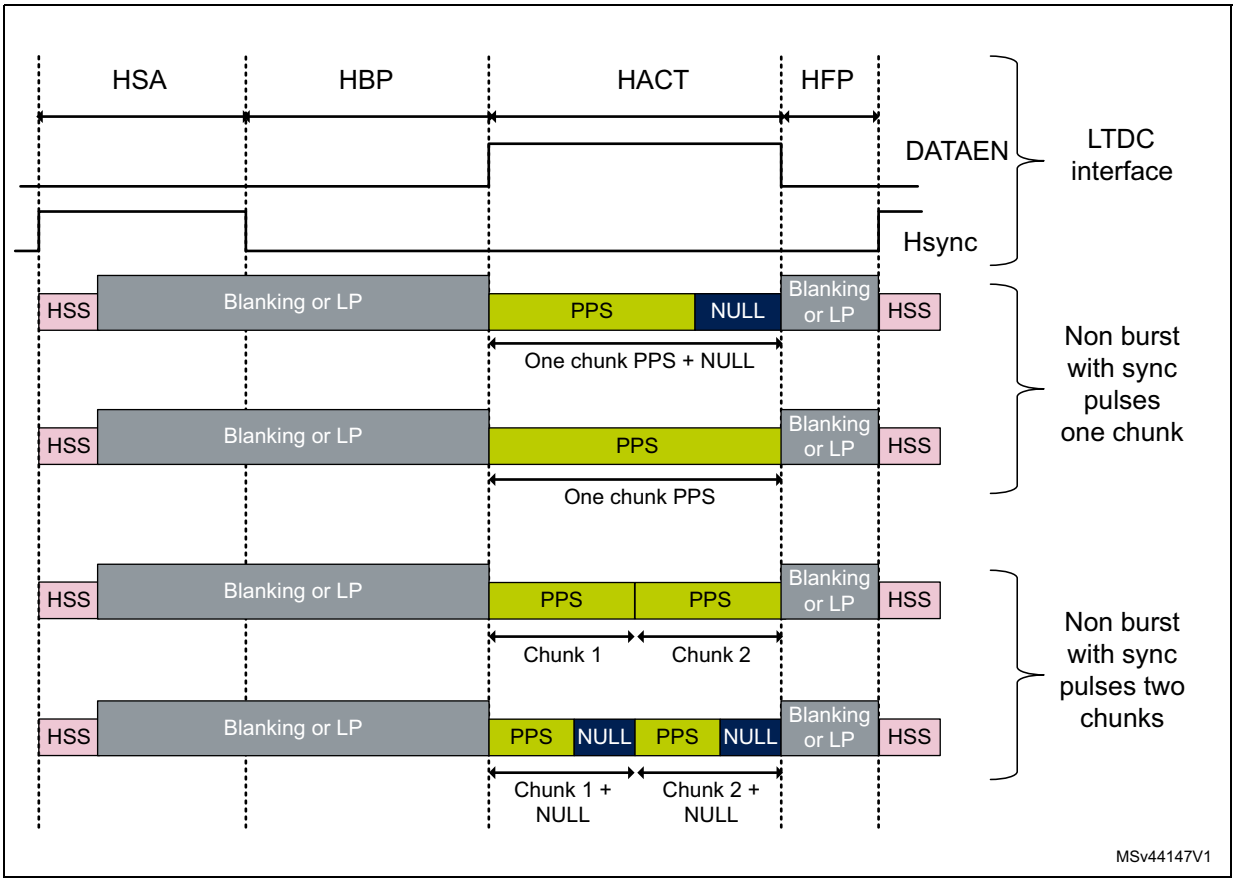
In nonactive regions (VSA,VBP,VFP) the link goes to LP after sending the HSS packet until the next line.

In the VACT region, the DSI host sends an HSS packet than either goes to LP or sends blanking packets until the end of HSA + HBP periods.

The HACT region is same as in the nonburst with sync pulses mode.

Figure 64 shows a VACT region in burst mode with sync events.

Figure 64. VACT region in burst mode with sync events



The VACT region in nonburst mode with sync event differs from the sync pulses only during the HSA period.

The DSI host sends the HSS packet to signal the beginning of an HSA period. Then, the link either goes to LP or the host sends blanking packets until the start of HACT region.

When to use nonburst mode

The nonburst mode provides a better matching of rates for pixel transmission. This mode enables:

– Only a certain number of pixels to be stored in the memory and not requiring a full pixel line (fewer RAM requirements in the DSI host).

– Operations with devices that support only a small amount of pixel buffering (less than a full pixel line).

Burst mode

In the video burst mode, RGB pixel packets are time-compressed, leaving more time during a scanline for LP mode (saving power) or for multiplexing other transmissions onto the DSI link.

In this mode, the entire active pixel line is buffered into a FIFO and transmitted in a single packet with no interruptions. This transmission mode requires that the DSI payload pixel FIFO has the capacity to store a full line of active pixel data inside of it.

This mode is optimally used when the difference between the LTDC bandwidth and the DSI link bandwidth is significant. The burst mode enables the DSI host to quickly dispatch the entire active video line in a single burst of data and then return to low-power mode.

Figure 65 shows a typical frame in video burst mode. The difference versus the nonburst mode with sync events resides in the HACT region. In burst mode, during the HACT region, the RGB data is sent using a single packet at maximum speed then the link goes to LP mode.

Figure 65. Video burst mode frame

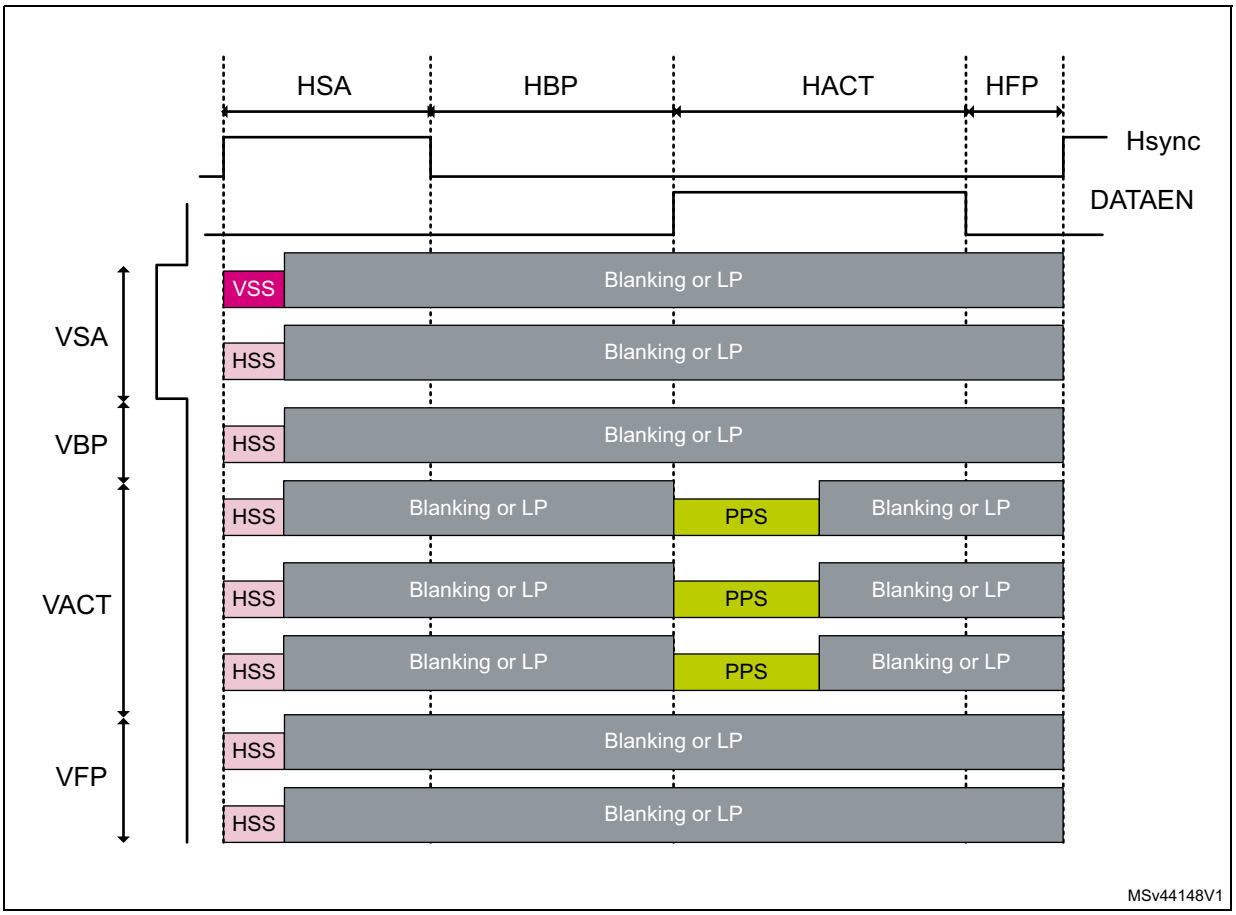
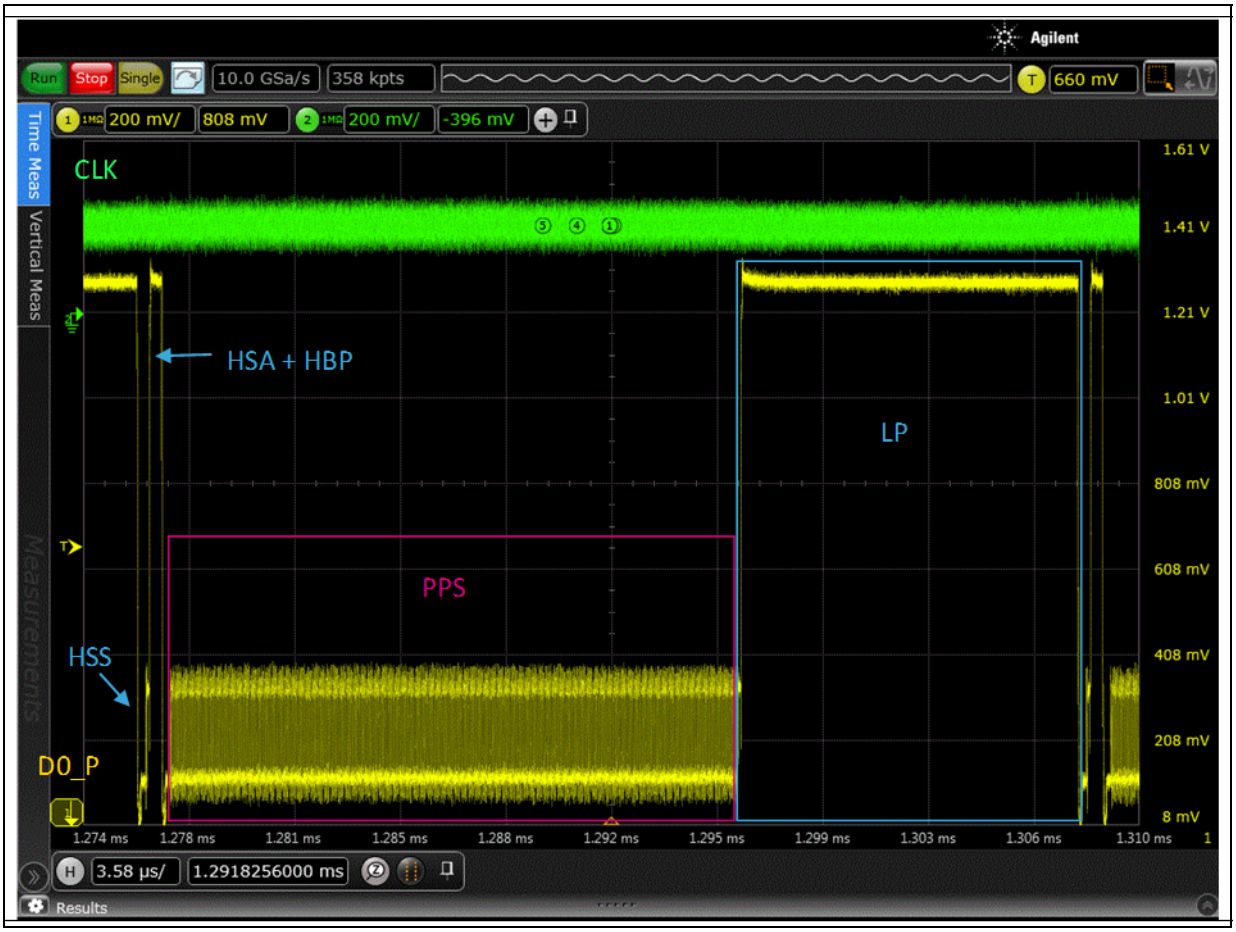


Figure 66 shows a video active line with RGB pixel data sent by using a PPS packet. Since the DSI link bandwidth is higher than the DSI input bandwidth from the LTDC, the link goes to LP mode for a long period.

Figure 66. Active line in burst mode



When use burst mode?

The burst mode is used when:

– The display device supports the reception of a full pixel line in a single packet burst to avoid overflow on the receiving buffer.

– The DSI output bandwidth should be higher than the LTDC interface input bandwidth in a relation that enables the link to go to low-power once per line.

Video mode comparison:

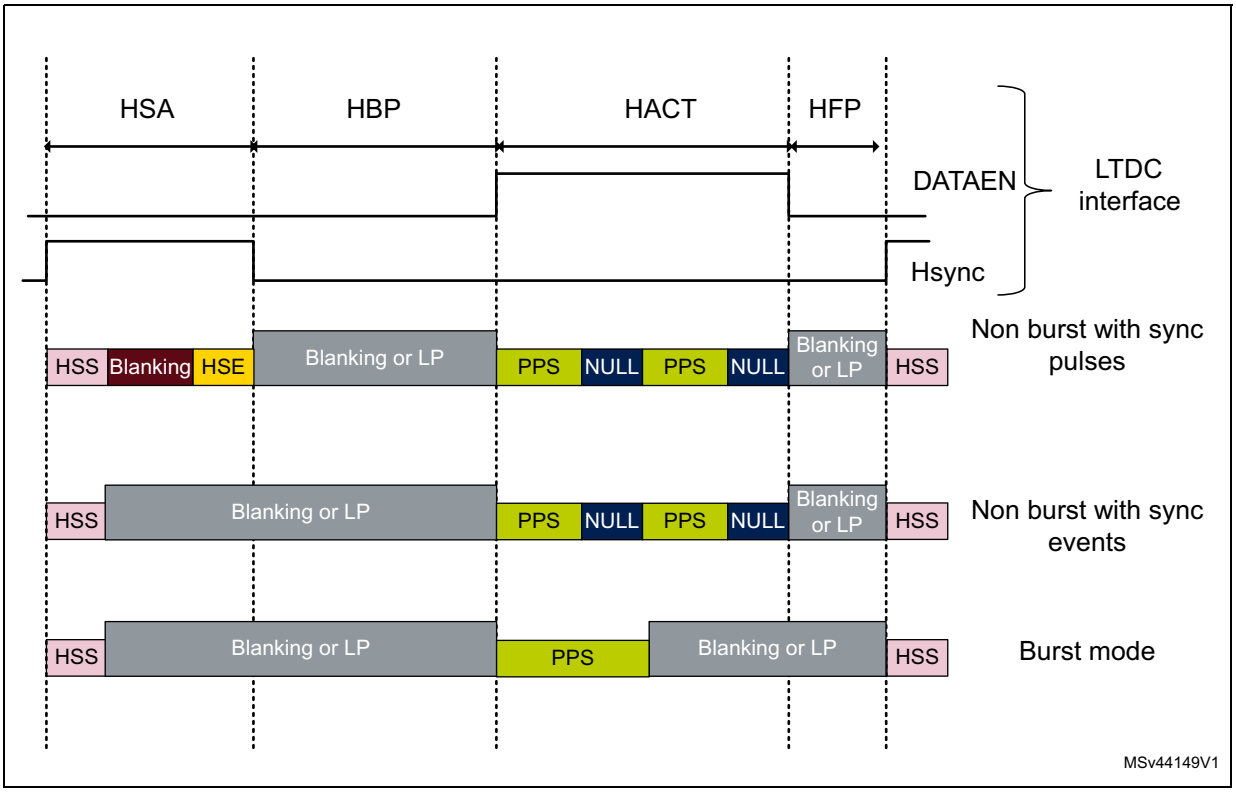
Figure 67 shows a comparison between the three different video mode formats.

The nonburst with sync pulses mode is the most accurate mode to convey video timing information, but it does not enable the option to go to LP during HSA and HACT regions.

The nonburst with sync events mode allows going to LP mode in the HSA region.

The burst mode is the most power efficient mode since it allows going to LP mode even during the HACT region.

Figure 67. Video mode comparison



4.2.2 Adapted command mode

The adapted command mode is a very optimized operating mode to interface with displays having their own graphical RAM.

This mode automatically refreshes the display graphical RAM with the LTDC without any CPU or DMA load.

The GRAM refresh operation is done in conjunction with the LTDC:

- The DSI host controls the LTDC and enables it for one frame.
- The RGB data coming from the LTDC is captured and sent into series of DCS long write command packets to the display.
- Once the graphical RAM is completely refreshed, the DSI host automatically stops the LTDC, and the DSI link goes to low-power mode.

The user controls the refresh operation of the display just by setting one bit when the frame buffer is ready to be sent.

As long as there is no need to update the frame buffer content, the display uses its internal graphic RAM for the refresh operations. This is very useful when the display is infrequently updated, which minimizes the bandwidth utilization on the MCU side and the overall power consumption.

The refresh can be done at the maximum speed of the link in order to reduce the refresh duration. Special attention must be put on the bandwidth requirements on LTDC side.

The user can refresh only part of the frame buffer, allowing faster animation and less bandwidth requirement on MCU side.

The adapted command mode only supports the DCS (display command set), WMS (write_memory_start) and WMC (write_memory_continue) commands. Additional commands such as display configuration commands and tearing effect initialization are required for proper operation. These commands have to be sent through the APB interface.

Example of a display refresh in adapted command mode:

The refresh process is done automatically by the DSI host while in adapted command mode.

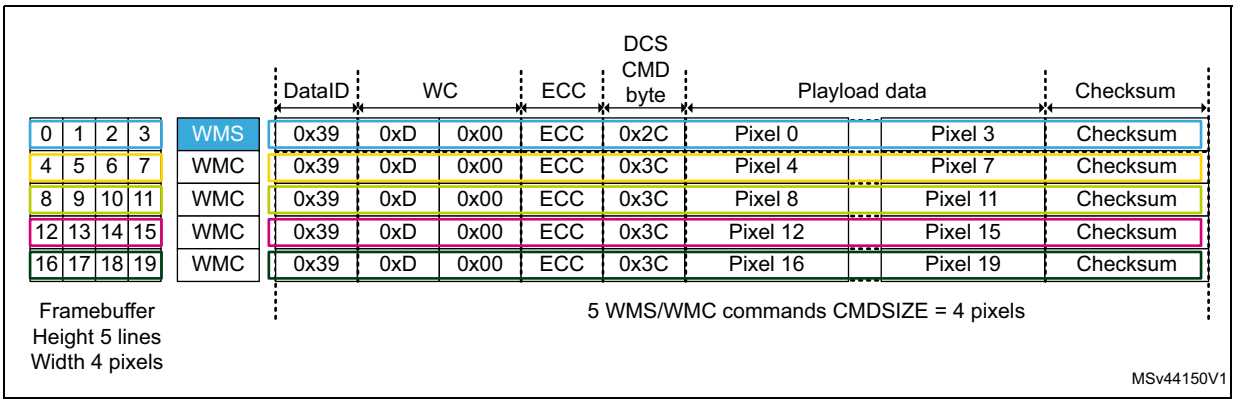
The DSI host enables the LTDC that fetches pixel data from the frame buffer and pass it to the DSI host. The DSI host stores pixel data into a dedicated FIFO, then encapsulates it into WMS and WMC DCS commands.

In the example shown in *Figure 68*, a frame of five lines can be conveyed in five DCS long write commands: one Write_Memory_Start (WMS) and four Write_Memory_Continue (WMC).

The CMDSIZE parameter defines the number of pixels to be transferred in each WMS/WMC command.

In this example, the CMDSIZE is set to be equal to the line size in pixels, so each DCS command encapsulates an entire line made of four pixels.

Figure 68. Full display refresh through WMS/WMC



Example of a partial refresh in adapted command mode

The adapted command mode supports the partial refresh feature that enables to refresh specific portions of the screen without having to send the entire screen.

Resending only part of the screen rather than the entire screen, allows to:

- Reduce the system bandwidth since only a portion of the pixels is fetched from the frame buffer.
- Reduce refresh time allowing for faster animation.
- Reduce power consumption since the DSI link goes to LP mode for longer periods.
- Eventually reduce memory footprint required for frame buffer storage (only part of the frame can be stored).

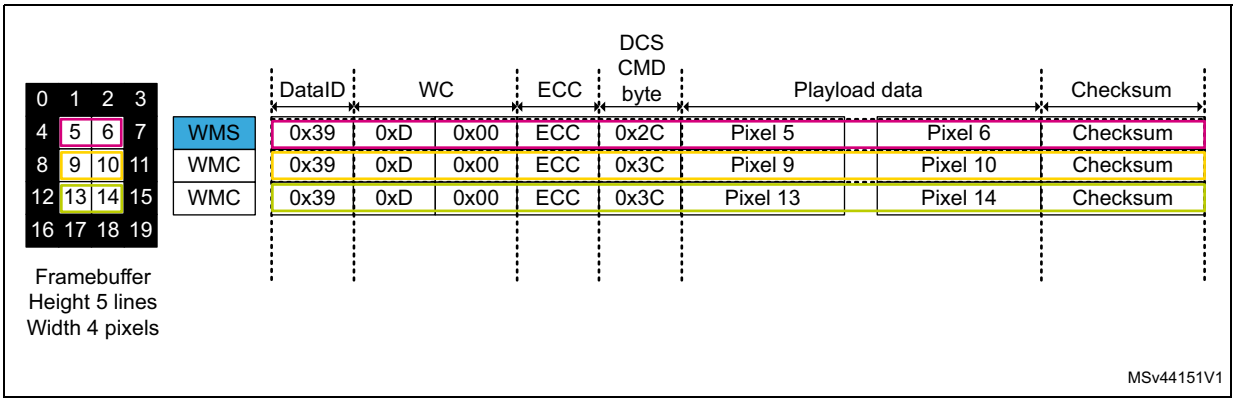
In the example presented in *Figure 69*, only pixels in the center are transmitted.

The CMDSIZE is set to the number of pixels to be sent in each DCS command, which is 2 in this example.

Note: The user may need to send DCS commands over the APB interface in order to select the area to be refreshed on the display side (set_column_address and set_page_address). Also the user has to reprogram the LTDC parameters to select the window of pixels to be fetched.

Please refer to STM32cube command mode examples for more details about required settings.

Figure 69. Display partial refresh



The basic flow of the adapted command mode is shown in *Figure 70*.

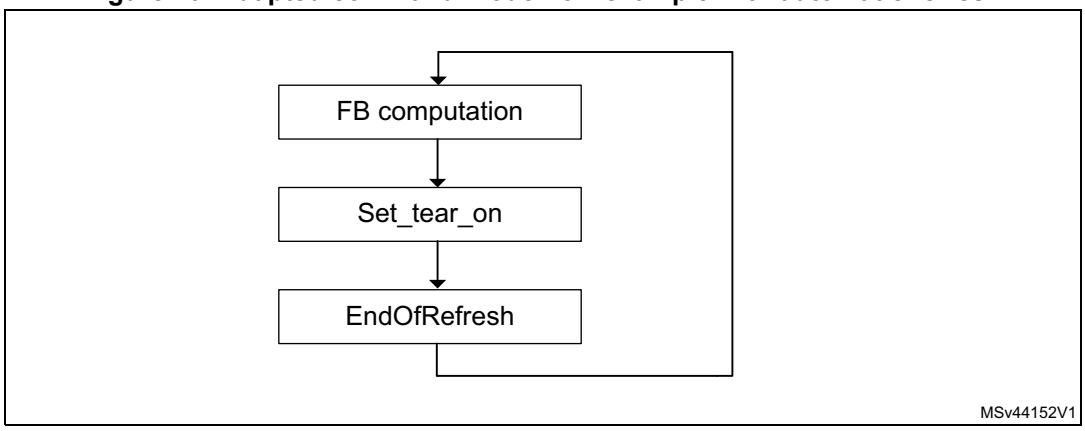
Once the frame buffer is ready, the user sends Set_tear_on command to activate the TE reporting on the display side. When the display reaches the programmed scanline, it sends TE trigger message.

After receiving the TE event, the host automatically refreshes the display by sending WMS/WMC DCS commands if the automatic refresh feature is enabled.

If the automatic refresh feature is not enabled, the refresh in this case is simply done by setting one bit (LTDC_EN bit) in the DSI Wrapper.

When the refresh operation is terminated, an EndOfRefresh event is signaled to the DSI host that may now start the computation of the next frame.

Figure 70. Adapted command mode flow example with automatic refresh



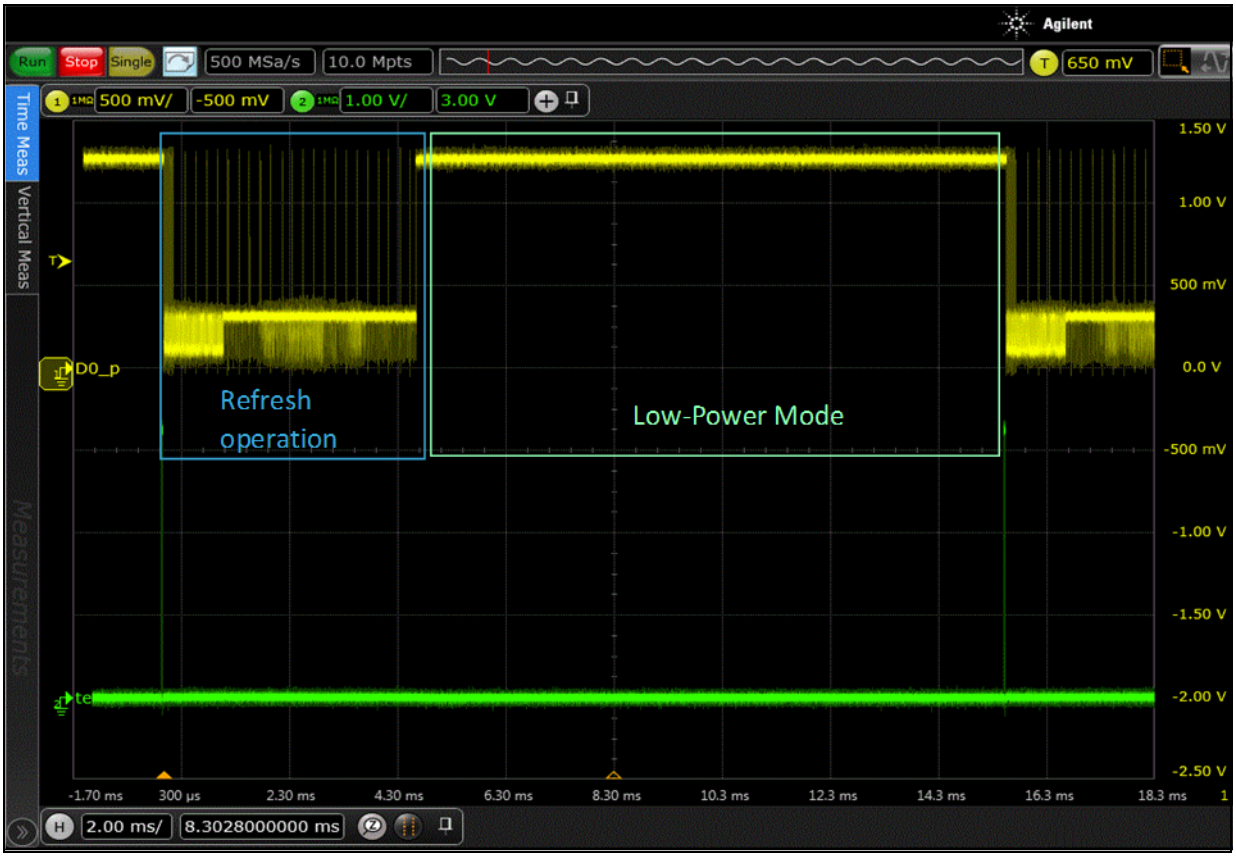
Adapted command mode advantages

The adapted command mode offers many advantages:

- It allows a more integrated solution.
- The image calculation and the GRAM update may not occur at the same time. This avoids concurrency issues between the LTDC and the DMA2D graphic engine when fetching data from the frame buffer.
- One frame buffer sufficient on MCU side.
- The frame buffer can fit in the internal MCU SRAM (a 320 x 320 display at 16 bpp color depth requires about 200 Kbytes for the frame buffer).
- Low-power consumption: as long as there is no need to update the frame, the display uses its graphic RAM for refresh and the link can be put into low-power mode.
- Partial refresh support: the user may choose to update only one portion of the display. This allows to reduce the refresh time, the bandwidth utilization on MCU side, and the power consumption.
- The pixel clock can be put at a maximum frequency that allows to use maximum bandwidth in shorter period.

Figure 71 shows an example of display refresh in adapted command mode. The DSI host sends DCS commands to update the frame, and then goes back to LP mode.

Figure 71. Display refresh in adapted command mode



Tearing effect management

The tearing effect allows a perfect synchronization between the display and the DSI host for refresh operation on displays having their own graphical RAM.

The tearing effect signaling can be done in two ways:

- Over the link without any additional pin.
- Or using an additional pin.

Tearing effect over link

When the tearing effect is done over the link, the DSI host sends a set_tear_on or set_tear_scanline command and gives the control of the bus to the display through a BTA procedure. After the first BTA, the display will respond with acknowledge trigger and give back bus control to the host. The host starts a second BTA procedure to give bus control to the display. Once the programmed scanline is reached by the display, it sends a trigger to the DSI host and gives back the control of the bus to the DSI host. An interrupt can be raised to launch the graphic RAM refresh, or user may choose to do an automatic refresh.

Figure 73 illustrates a tearing effect reporting over a DSI link.

Figure 72. Tearing effect request and response example

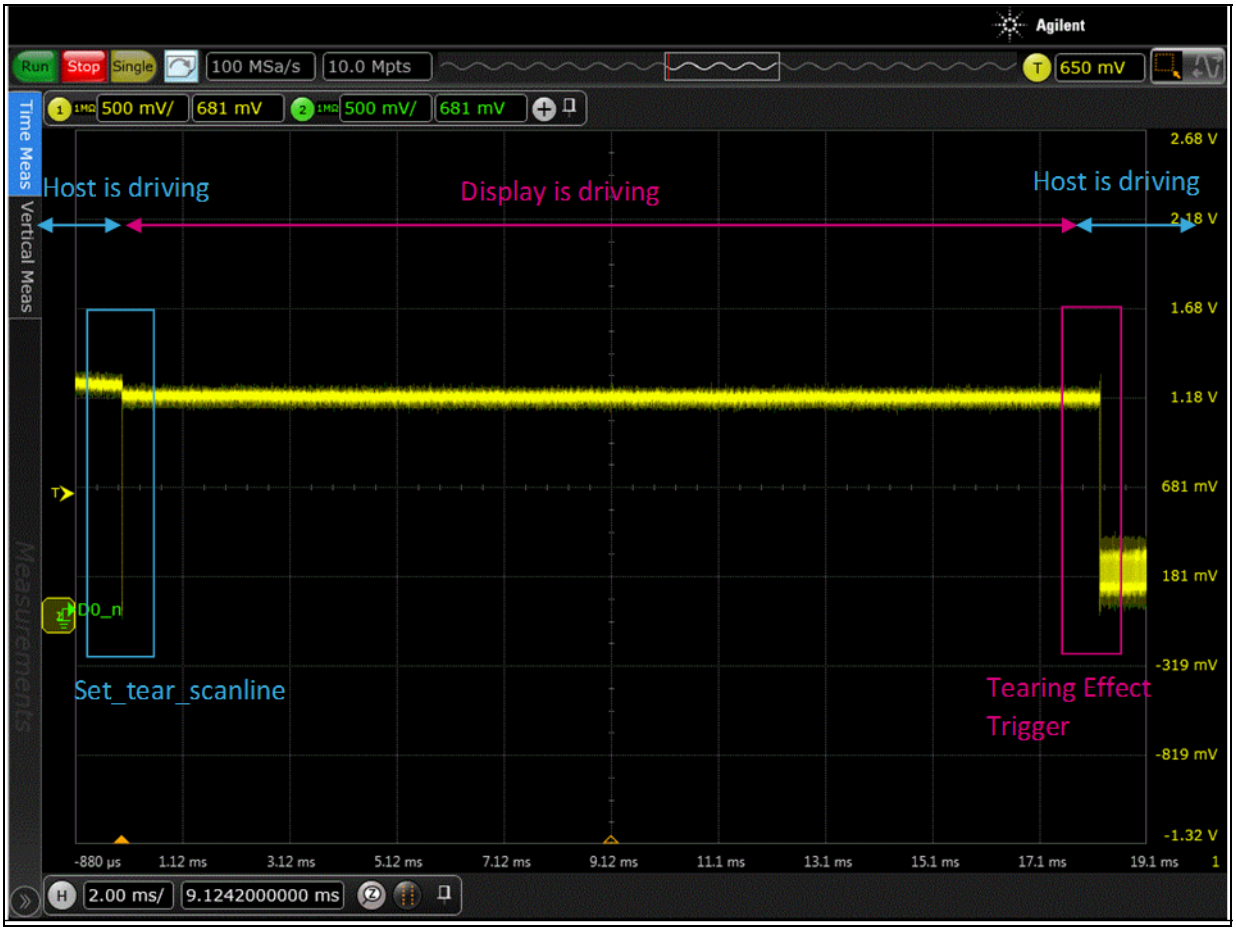
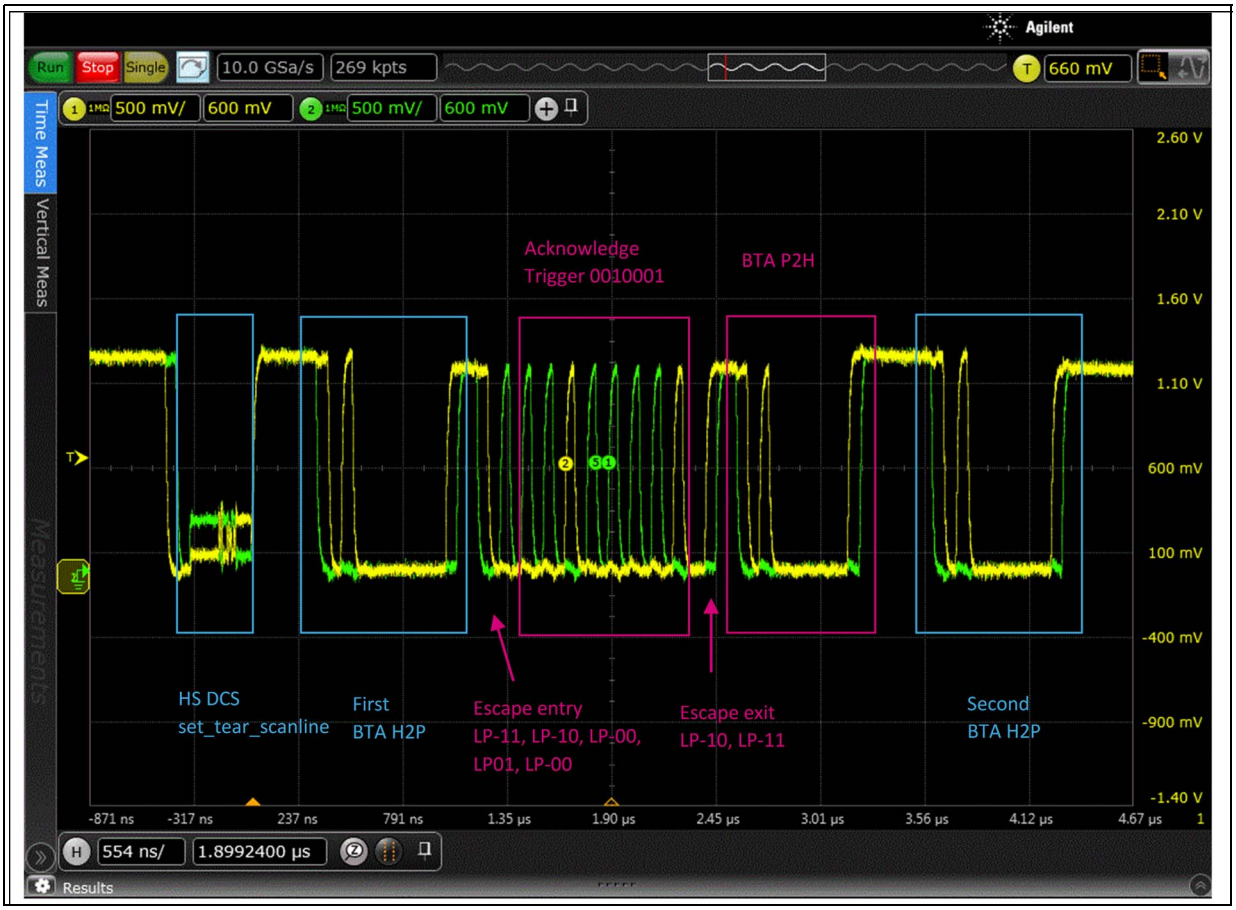


Figure 73 shows a zoom on the Set_tear_scanline command with double BTA.

Figure 73. Set_tear_scanline DCS command with double BTA

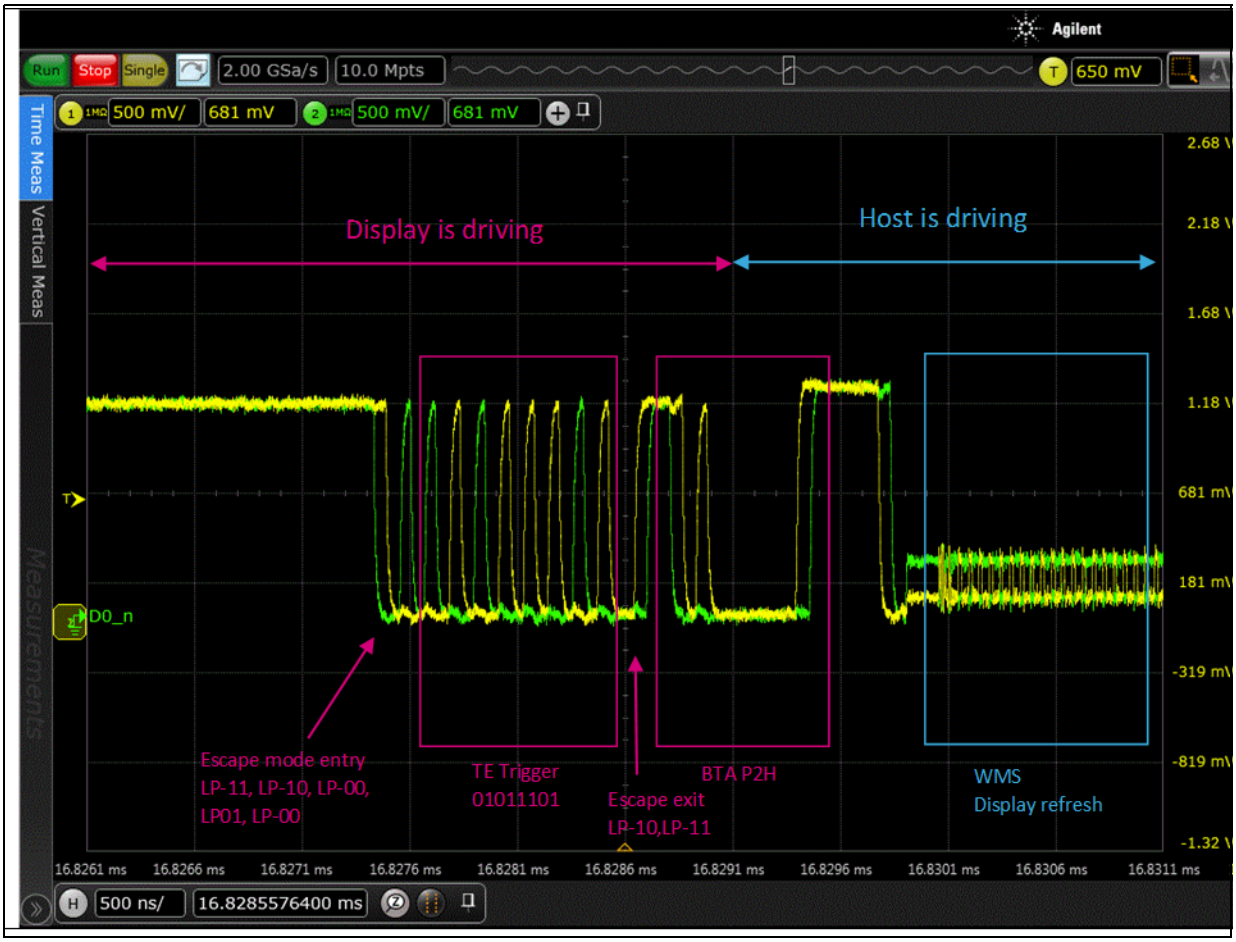


When the display reaches the specified scanline, it sends a tearing effect trigger with a BTA procedure to give back the bus control to the DSI host.

Figure 74 shows a zoom on the tearing effect trigger message.

The DSI host starts the display refresh operation using the WMS/WMC DCS commands.

Figure 74. Tearing effect trigger from display



Tearing effect over pin

When the tearing effect is done over a pin, the DSI host sends the set_tear_on or the set_tear_scanline DCS command. Then the display toggles a dedicated pin to trigger the DSI host when the programmed scanline is reached.

The set_tear_on or set_tear_scanline DCS commands may be sent only once, then the display generates continuous synchronization signal.

Even if an additional pin is required, this mechanism avoids having multiple exchanges over the link between the DSI host and the display.

An interrupt can be raised on the pin toggling to launch the graphic RAM refresh.

Then the DSI host starts a refresh operation by sending the WMS/WMC DCS commands.

The refresh may be done automatically upon detection of a TE interrupt, or enabled by software.

Figure 75 shows a TE reporting through a GPIO. When the display module reaches the programmed scanline, it generates a pulse on the TE pin.

Figure 76 shows a zoom on the tearing effect over a GPIO.

Figure 75. TE over pin

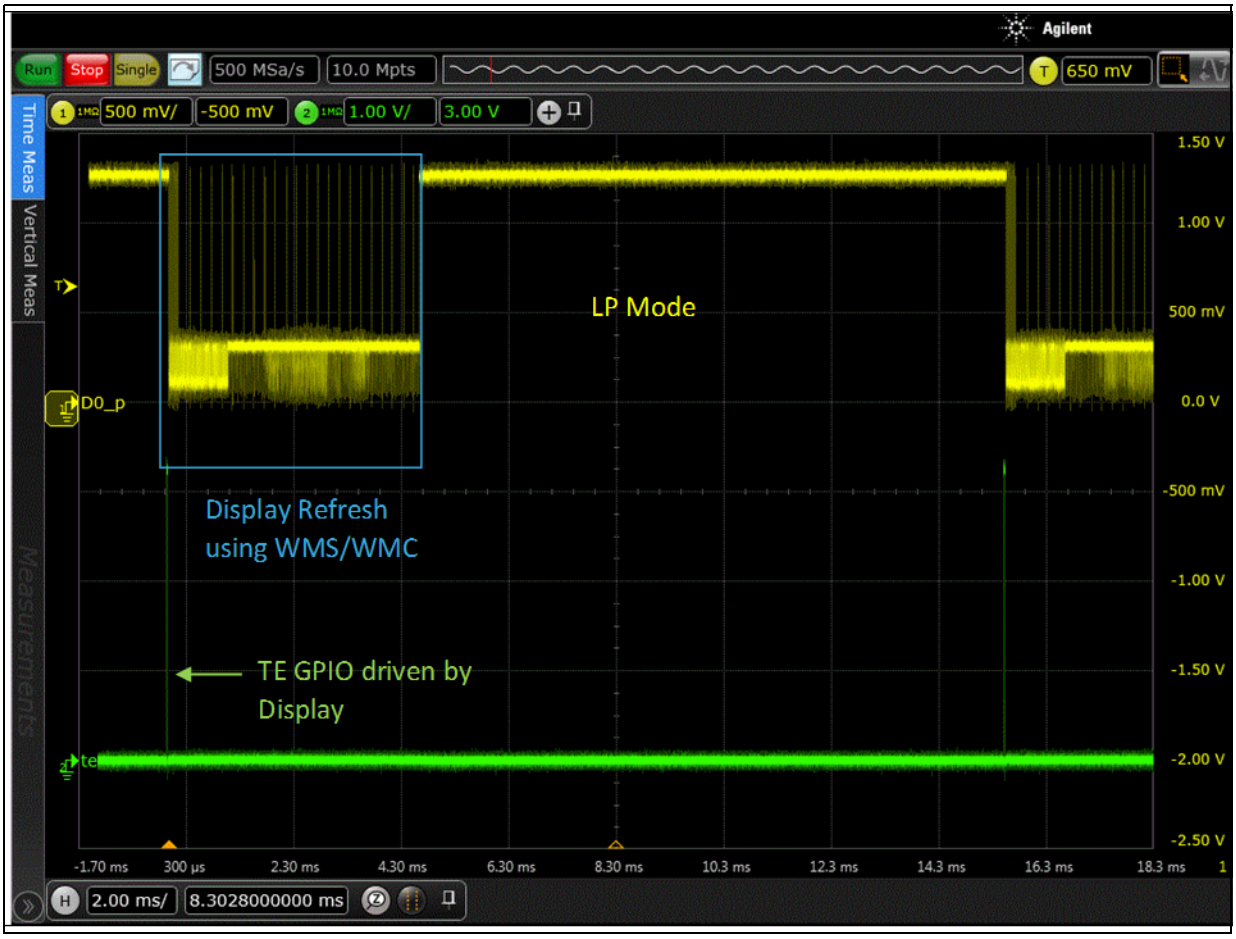
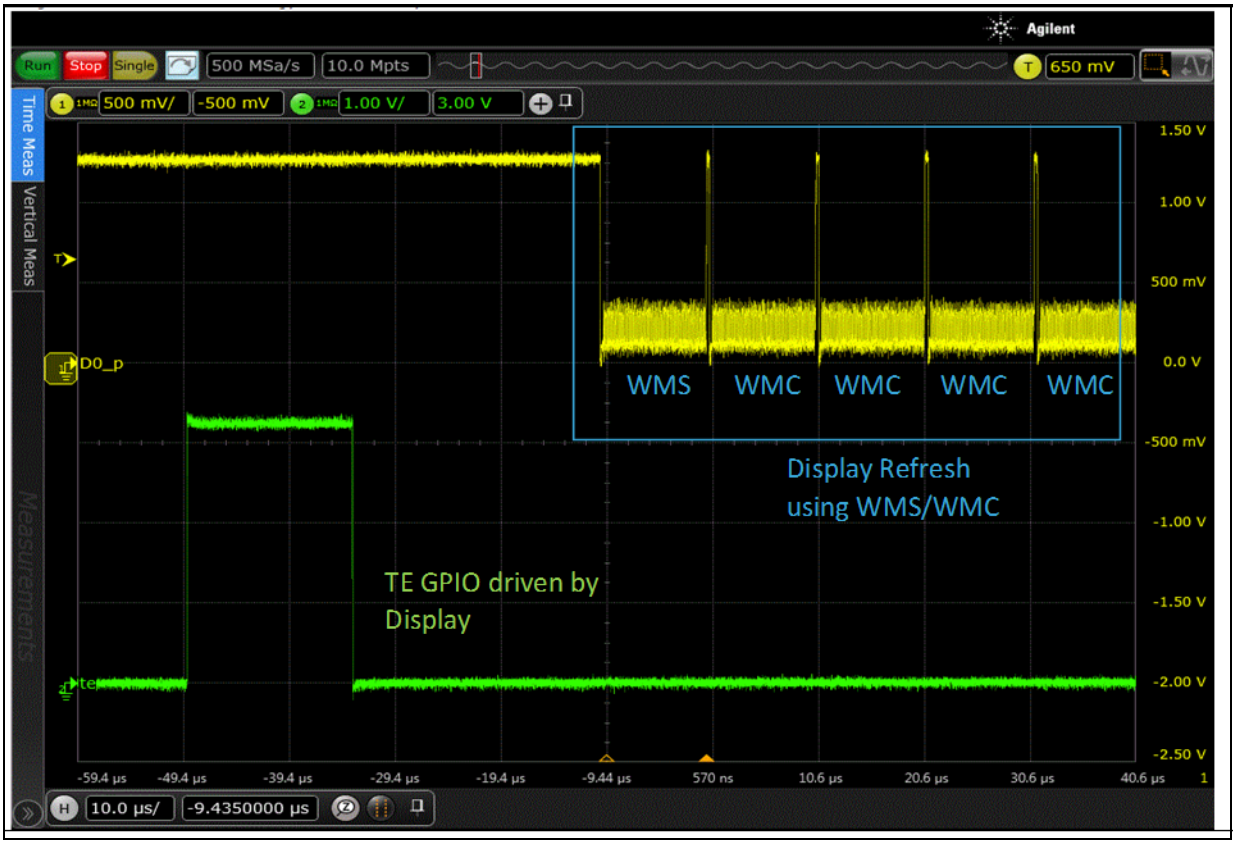


Figure 76. Zoom on tearing effect over pin



4.2.3 APB command mode

The APB command mode is used to send commands through the DSI host APB register interface.

Generic commands or DCS commands can be sent for display configuration at startup or for maintenance operations when the application is running.

All the commands can be sent either in high-speed or in low-power modes as some displays accept only low-power communication at startup.

Commands can also be sent during video streaming. The DSI host scheduler automatically evaluates, according to the programmed timings, if it has the sufficient time to insert a command during a video transmission.

All the commands are fully programmable by software, which makes the DSI host able to support all the standard DCS commands and all the display-dependent custom commands.

When to use the APB command mode?

The APB command mode is used mainly in display initialization phases and for maintenance operations on display. It operates concurrently with the video mode or with the adapted command mode.

4.3 Operating mode choice

The choice between the video mode or the adapted command mode has a big impact on the architecture and the cost of the solution.

From the microcontroller's stand point, the adapted command mode is preferred for a cost optimized solution.

As the video mode does not require a graphical RAM on the display side, this solution is adapted for large displays that often do not bring graphical RAM due to cost optimization. The constraints in terms of bandwidth and memory usage on the MCU (or MPU) side remains the same as the ones for the current LTDC based solutions. Most of the times an external RAM is required for double buffering of the frame buffer.

The adapted command mode requires a display with a graphical RAM. The display may have a slightly higher cost but most of the small displays bellow 480 x 480 embed a graphical RAM. As a consequence, the adapted command mode with small display does not require an external RAM as the frame buffer may be stored in the internal MCU (or MPU) RAM. This reduces considerably the bandwidth issues on the MCU and also the overall BOM cost and solution integration as no external RAM is required.

4.4 DSI interrupts

The DSI host has many interrupts to monitor all the timings and events of the communication. Refer to relevant STM32 reference manual for a detailed description of all the interrupt sources.

The interrupts can be generated either by the DSI Wrapper host or by the DSI host. All the interrupts are merged in one interrupt line going to the interrupt controller.

In addition to the protocol-related interrupts, the DSI host also provides interrupts to manage:

- Regulator events.
- PLL events.
- Tearing effect events (only in command mode).
- End of refresh events (only in command mode).

4.5 Low-power modes

The DSI host is active in run and sleep modes. A DSI host interrupt can cause the device to exit the sleep mode.

In stop mode, the DSI host is frozen and its register content is kept. In standby mode, the DSI host is powered-down and it must be reinitialized afterwards.

Table 10 presents the DSI host low-power modes.

Table 10. DSI host low-power modes

Mode	Description
Run	Active.
Sleep	Active. Peripheral interrupts cause the device to exit the sleep mode.
Stop	Frozen. Peripheral register content is kept.
Standby	Power-down. The peripheral must be reinitialized after exiting standby mode.

5 DSI host configuration

This section describes the low-level hardware registers required for the configuration and use of the DSI host. For complete programming sequences, please refer to the relevant STM32 reference manual.

The configuration process is split into two parts:

- Global initialization: this is common to all operating modes: video mode or adapted command mode.
- Operational mode configuration: this section depends on the chosen operating mode, either video mode or adapted command mode.

Note: The APB command mode is used in parallel with the video mode and the adapted command mode. It is not optimized for a use in standalone to refresh the display, even if it is possible, because it causes high latency. This is why the focus is done only on the video and adapted command operating modes configuration.

5.1 DSI host global configuration

This section describes the DSI host configuration common to all the operating modes. It details regulator, PLL, PHY, flow control and LTDC interface configurations.

5.1.1 DSI regulator configuration

The DSI host has a dedicated 1.2 V internal regulator that provides supply to the D-PHY and to the PLL. *Table 11* shows the required registers to configure the DSI regulator.

Table 11. DSI regulator configuration registers

Description	Register field
Enable regulator	DSI_WRPCR.REGEN
Wait for regulator ready	DSI_WISR.RRS

The DSI regulator providing the 1.2 V is controlled through the DSI Wrapper.

The regulator is enabled setting the REGEN bit of the DSI_WRPCR register.

Once the regulator is ready, the RRIF bit of the DSI_WISR register is set.

The power ON / OFF of the D-PHY is done by directly enabling the 1.2 V regulator.

5.1.2 Clocks configuration

This section shows the configuration of the different clocks required by the DSI host:

- The HS clock using DSI host internal PLL.
- The TX escape clock for LP communication.
- The secondary clock source required in ultra low-power state mode with DSI PLL off.

DSI PLL configuration

The STM32 DSI host has a dedicated DSI PLL controlled through the DSI Wrapper. The DSI PLL configuration registers are described in *Table 12*.

Table 12. DSI PLL configuration registers

Description	Register field
Configure PLL loop division factor	DSI_WRPCR.NDIV
Configure PLL output division factor	DSI_WRPCR.ODF
Configure PLL input division factor	DSI_WRPCR.IDF
Enable PLL	DSI_WRPCR.PLLEN
Wait for PLL locked	DSI_WISR.PLLS

The incoming clock to the DSI PLL is the HSE (high-speed external) oscillator clock.

The PLL output clock is the HS clock fed to the D-PHY. The HS clock is a full-rate clock. It must be in the range between 80 MHz and 500 MHz.

The D-PHY uses the HS clock to generate a half-rate DDR clock that is transmitted to the display using the clock lane. Which means that for 500 Mbit/s per lane rate, the PLL output must be 500 MHz and the transmitted clock lane is a 250 MHz DDR clock.

The D-PHY divides the HS clock by eight to generate the lane_byte_clk, and feed it to the DSI host.

The lane_byte_clk is calculated using the following formula:

$$FVCO = \left(\frac{HSE}{IDF}\right) X\ 2\ X\ NDIV$$

$$Lane\_Byte\_CLK = FVCO / (2 \times ODF \times 8)$$

with FVCO must be in the range of 500 MHz to 1 GHz.

$$Lane\_Byte\_CLK = FVCO / (2 \times ODF \times 8)$$

Refer to product datasheet for FVCO range.

Calculation example

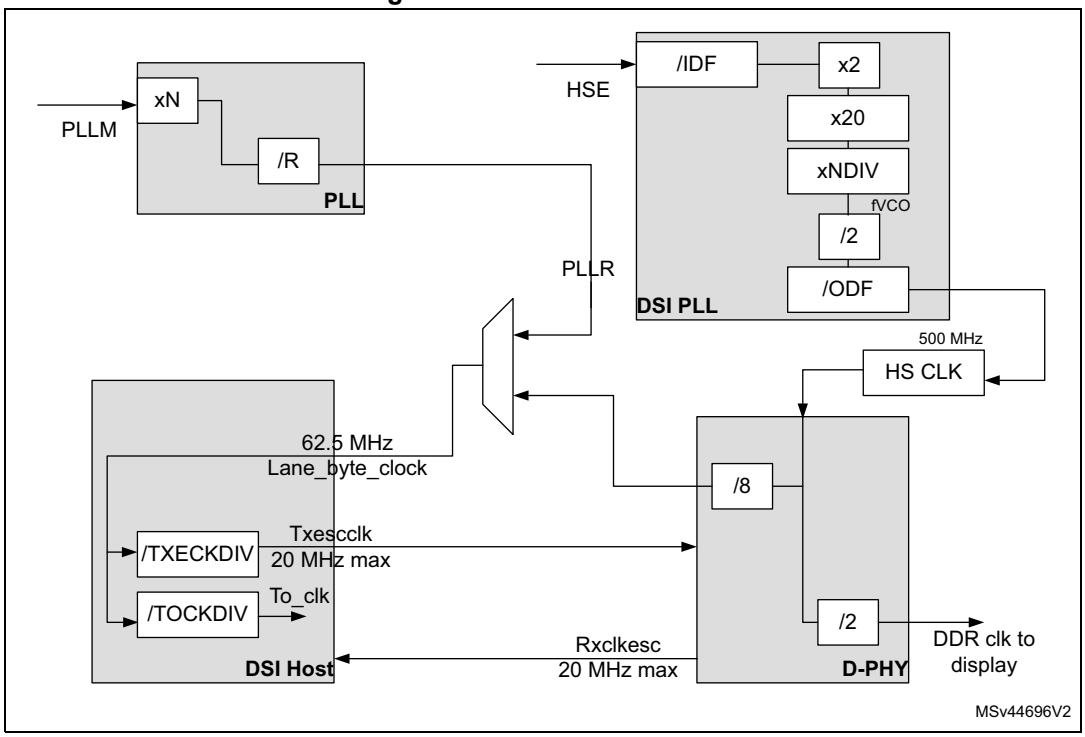
To obtain 500 Mbit/s rate per lane, the DSI PLL must be configured to output 500 MHz HS clock, and the lane_byte_clk fed to the DSI host will be equal to 500 MHz/8 = 62.5 MHz. With a 25 MHz HSE oscillator, 62.5 MHz lane_byte_clk may be obtained using these parameters:

IDF = 1, NDIV = 20, ODF = 1

Lane_byte_clk = ((25/1) x 2 x 20) / (2x1x8) = 62.5 MHz

Figure 77 shows the DSI 500 MHz HS clock, scheme.

Figure 77. DSI clock scheme



TX escape clock configuration

The TX escape clock is used in LP mode. It takes two cycles of tx_esc_clk to transmit one bit in LP mode. *Table 13* shows the register used to configure the TX escape clock.

Table 13. TX escape clock configuration register

Description	Register field
TX escape clock division	DSI_CCR.TXECKDIV

The Tx escape clock is calculated as follows:

TX escape CLK = Lane_byte_CLK / TXECKDIV

Note: TXECKDIV must be programmed to ensure that the TX escape clock is less than 20 MHz. The TX prescaler must be set to a value higher than 2. A prescaler value 0 or 1 disables the generation of the TX escape clock.

Secondary clock source setting

The DSI host lane_byte_clock source can be fed from DSI-PHY or from a specific output (PLLR) of the main PLL. In ultra low-power state, the DSI PLL may be turned off. In this case, the lane_byte_clock source is the PLLR.

Table 14 presents the register bit used to select between the two clock sources.

Table 14. Secondary clock source setting register

Description	Register field
DSI clock source selection	DCKCFGR2.DSISEL

This bit must be cleared when the DSI-PHY is used as the DSI lane_byte_clk source. This is in normal operating mode.

This bit must be set when PLLR is used as DSI lane_byte_clk source in case DSI PLL and DSI-PHY are off.

5.1.3 DSI host PHY parameters

This section presents the D-PHY parameters controlled from the DSI host.

Number of lanes

The DSI host provides a scalable architecture using one or two data lanes. *Table 15* shows the register fields used to program the number of lanes.

Table 15. Number of lanes configuration register

Description	Register field
Set number of active data lanes	DSI_PCONFR.NL

The number of lanes depends on the bandwidth requirements of the application.

Each DSI lane has a maximum of 500 Mbit/s data rate, for a total 1Gbit/s rate in dual data lane mode. Refer to *Section 7: DSI host performance* for more information about how to evaluate the required number of lanes.

PHY clock and digital sections control

The DSI host provides control of the D-PHY digital section and of the clock lane module.

Table 16 shows the register fields used to control the D-PHY clock and the digital section.

Table 16. PHY clock and digital sections control registers

Description	Register field
D-PHY digital section control	DSI_PCTLR.DEN
Enable the D-PHY Clock Lane module	DSI_PCTLR.CKE

The DSI_PCTLR.DEN field allows getting the D-PHY out of reset.

The DSI_PCTLR.CKE must be set to enable the D-PHY clock lane.

Clock lane control

The DSI host provides specific features for clock lane control. *Table 17* presents different register fields used for clock lane control.

Table 17. Clock lane control registers

Description	Register field
Automatic Clock lane Control	DSI_CLCR.ACR
D-PHY Clock Control	DSI_CLCR.DPCC

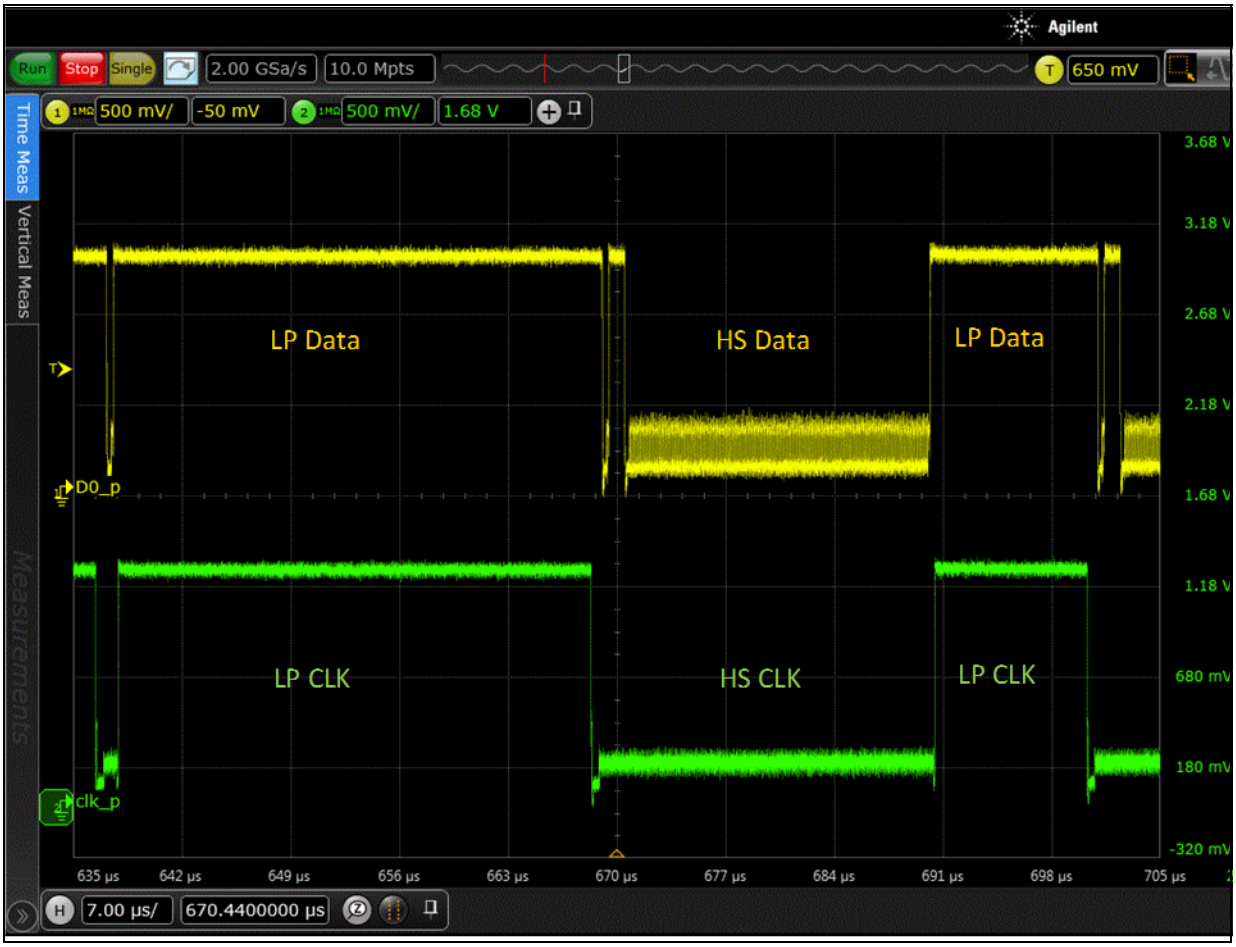
Automatic clock lane control

The user may choose to let the DSI host automatically control when to put the clock lane into LP state.

This is done by setting DSI_CLCR.ACR bit to 1 and DSI_CLCR.DPCC must also be set to 1 to enable the HS mode for the clock.

Figure 78 shows an example with automatic clock lane control enabled. The DSI host automatically stops providing the clock in HS mode and put it in stop state LP-11 when time allows.

Figure 78. Automatic clock lane control

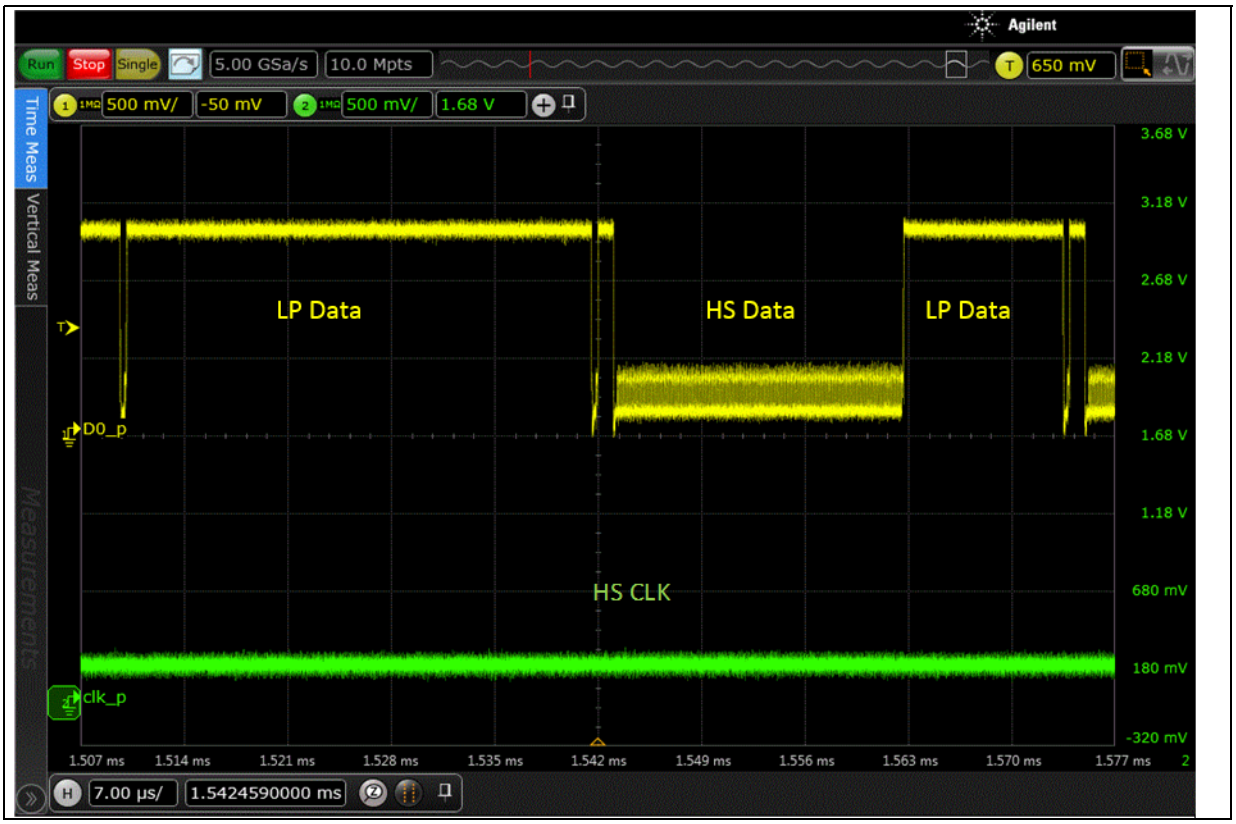


Note: In video mode the DSI host decides if it is possible to send the clock lane into LP mode based on the timing provided in the PHY transition timing (DSI_CLTCR.HS2LP_TIME and

DSI_CLTCR.LP2HS_TIME) which is discussed on Section 5.2.1: Video mode over LTDC interface.

Figure 79 shows an example with automatic clock lane control disabled and clock lane always in HS mode.

Figure 79. Example of automatic clock lane control disabled and clock lane always in HS mode



Manual clock lane control

It is also possible to manually control the clock lane state through the DSI_CLCR.DPCC bit of the DSI host clock lane configuration register.

The clock lane may be manually put into an LP state by clearing DSI_CLCR.DPCC.

5.1.4 DSI Wrapper PHY parameters

This section provides the D-PHY parameters controlled from the DSI Wrapper.

HS bit period setting

The D-PHY requires to know the high-speed clock rate provided by the PLL. This value is indicated to the D-PHY through the DSI Wrapper. *Table 18* presents the register fields inside the DSI Wrapper used to set the HS bit period.

Table 18. HS bit period configuration register

Description	Register field
Set the bit period in HS mode in multiples of 0.25 ns	DSI_WPCR0.UIX4

This field defines the bit period in high-speed mode in multiples of 0.25 ns, and is used as a time base for all the timings managed by the D-PHY.

The unit interval is configured through the DSI_WPCR0.UIX4 field. If this period is not a multiple of 0.25 ns, the driven value should be rounded down.

Note: This field is mandatory. It must be correctly programmed to avoid timing mismatch issues between the DSI host and the PHY.

Example: For a 500 Mbit/s link speed, the HS clock output from the PLL and fed to the PHY is 500 MHz, which has a 2 ns bit period. So, the UIX4 is programmed to 2 / 0.25 = 8.

The unit interval is half the clock period so,

UI= 1 / (2 x 250 MHz) = 2 ns, so UIX4 is programmed to 8.

5.1.5 Protocol flow control

The DSI host offers flow control features including EoTp transmission and reception, ECC and CRC reception, and bus turnaround control.

Note: The ECC and CRC generation in the forward direction is mandatory and always enabled.

Table 19 presents the register fields used for flow control settings.

Table 19. Protocol flow control configuration register

Description	Register field
EoTp transmission enable	DSI_PCR.ETTXE
EoTp reception enable	DSI_PCR.ETRXE
Bus turnaround enable	DSI_PCR.BTAE
ECC reception enable	DSI_PCR.ECCRXE
CRC reception enable	DSI_PCR.CRCRXE

EoTp transmission and reception

Multiple packets may exist within a single HS transfer. The end of transfer is always signaled at the PHY layer using a dedicated EoT sequence. In order to enhance the overall robustness of the system, the DSI defines a dedicated EoT packet (EoTp) at the protocol layer.

This mechanism provides a more robust environment, at the expense of an increased overhead (four extra bytes per transmission).

The EOTp is sent at the end of the HS transmission. After that, the link goes to LP state (refer to *Section 3.3.2: End of transmission (EoT) packet* for more details).

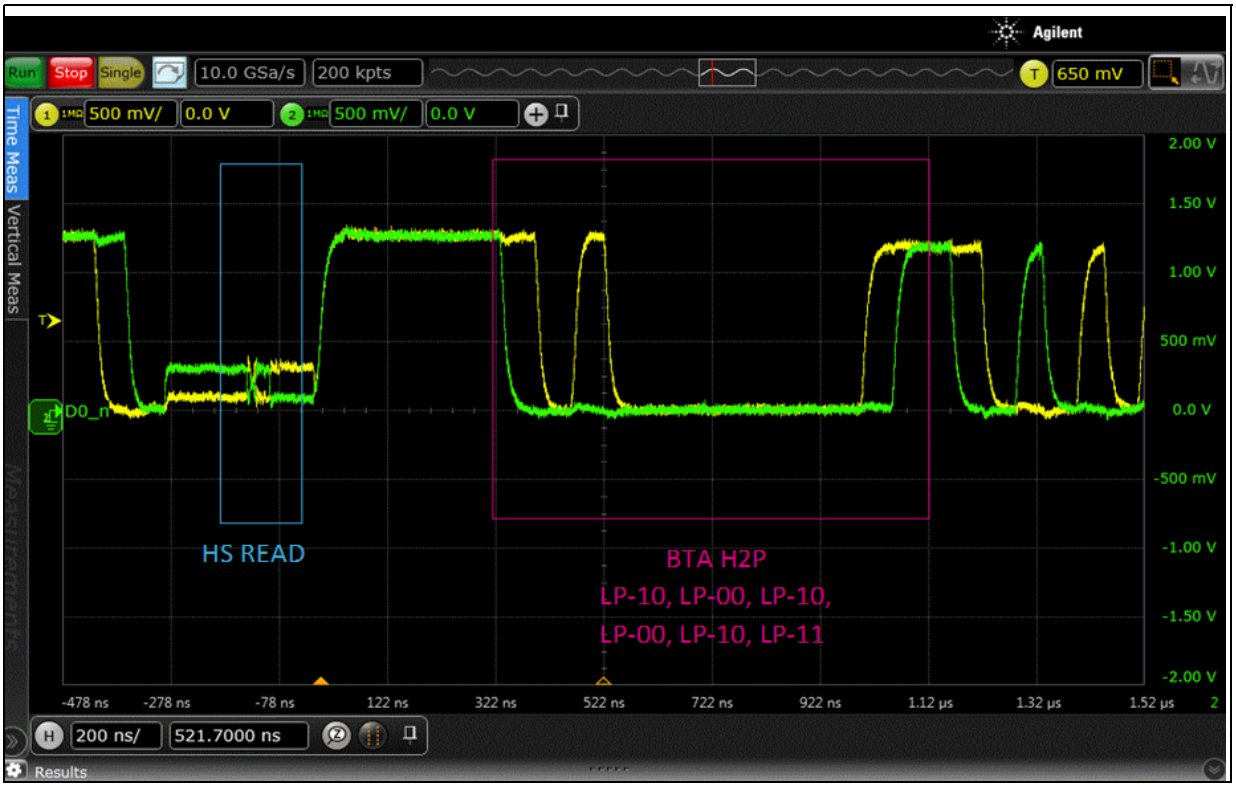
The DSI host supports both the transmission and the reception of the EoTp by setting respectively: DSI_PCR.ETTXE and DSI_PCR.ETRXE

Bus turnaround

The bus turnaround enable is mandatory when the reverse direction communication is required, for example on read, acknowledge and tearing effect requests.

Figure 80 shows an example of a read command with BTA procedure automatically started by the DSI host if DSI_PCR.BTAE is set.

Figure 80. BTA procedure after read command



ECC and CRC reception

The DSI host can check the values of the ECC and CRC fields of the received packets. This is enabled by setting DSI_PCR.ECCRXE and DSI_PCR.CRCRXE respectively.

5.1.6 DSI host LTDC interface configuration

The DSI host LTDC interface allows color coding and video signal polarity control.

Color coding configuration

Table 20 presents the register fields used to control color coding.

Table 20. Color coding configuration registers

Description	Register field
DSI Wrapper color coding configuration	DSI_WCFGR. COLMUX
DSI host color coding configuration	DSI_LCOLCR.COLC
Loosely packed variant to 18-bit configuration	DSI_LCOLCR.LPE

LTDC output is always 24-bit RGB R[7:0]G[7:0]B[7:0].

The DSI Wrapper allows to map LTDC output to DSI host input. The same color format must be selected for the DSI Wrapper and the DSI host. This color format is applied to the DSI output packets.

Available color coding are:

- 000: 16-bit configuration 1
- 001: 16-bit configuration 2
- 010: 16-bit configuration 3
- 011: 18-bit configuration 1
- 100: 18-bit configuration 2
- 101: 24-bit

Different 16-bit and 18-bit configurations have no effect on the DSI output packets. User may choose any of them as long as he programs same configuration in DSI Wrapper (DSI_WCFGR. COLMUX) and DSI host (DSI_LCOLCR.COLC)

If any of the 16-bit configurations is chosen, the DSI host outputs a 565 color coding. If any of the 18-bit configurations is chosen, the DSI host outputs a 666 color coding.

One exception is for the video mode 18-bit configuration, the DSI_LCOLCR.LPE bit selects between loosely and not loosely packet configuration (refer to *Figure 43* and *Figure 44* for not loosely and loosely 18-bit packets).

In video mode 18-bit color format, if the 18-bit loosely packet variant is used, the user must set DSI_LCOLCR.LPE.

Note: In video mode, the display is able to know the color format used thanks to the data type field inside the header of the packed pixel stream (refer to Section : Video mode data types on page 40). In command mode, the host needs to inform the display of the color format using set_pixel_format DCS command.

Note: The DSI host color coding defines how pixels are coded in the DSI packets transmitted through the DSI link. It is different than image source color format. The image source color format must be consistent with the LTDC layer input color format. It is possible to have an image source coded on 16-bits and DSI host output packets coded on 24-bits.

Video control signal polarity

The polarity of the video control signals (HSYNC, VSYNC and data enable) can be controlled through the LTDC interface.

Table 21 shows the register fields for video control signal polarity.

Table 21. Video control signal polarity registers

Description	Register field
Configure the HSYNC polarity	DSI_LPCR.HSP
Configure the VSYNC polarity	DSI_LPCR.VSP
Configure the DATA ENABLE polarity	DSI_LPCR.DEP

The video control signals (HSYNC, VSYNC and DE) polarity can be programmed in the DSI LTDC interface. The programmed value must be consistent with the LTDC control signal polarity settings, meaning that the polarity must be the same in the DSI and LTDC except for the DE signal.

LTDC global control register allows to set the polarity of NOT DE signal. In all DSI modes, LTDC NOT DE polarity must be set to active low and so DSI DE polarity must be active high. The inverse cannot be used.

5.2 DSI operational modes configuration

This section presents the configuration settings specific to the operational mode used. The user can choose either video mode or the adapted command mode.

5.2.1 Video mode over LTDC interface

This section presents the DSI host configuration specific to video mode.

Video mode selection

The user must choose one of the three available video modes. The video mode selection register is shown in *Table 22*.

Table 22. Video mode selection register

Description	Register field
Configure the video mode type	DSI_VMCR.VMT

The three video mode types are:

- 00: nonburst with sync pulses
- 01: nonburst with sync events
- 1x: burst

LP state in video mode

The DSI specification recommends to periodically end the HS transmission and drive the data lanes to the LP state. This enables the PHY synchronization.

The transition to the LP state should take place once per scanline. If not possible, the transition to the LP state must happen once per frame.

PHY transition timing configuration

In video mode, the DSI host requires information on PHY transition timing overhead between HS and LP states. These timings are set in registers described in *Table 23*.

Table 23. PHY transition timing configuration registers

Description	Register field
Clock lane timer	
Configure the maximum time that the D-PHY clock lane takes to go from High-Speed to Low-Power transmission measured in lane byte clock cycles	DSI_CLTCR.HS2LP_TIME
Configure the maximum time that the D-PHY clock lane takes to go from Low-Power to High-Speed transmission measured in lane byte clock cycles	DSI_CLTCR.LP2HS_TIME
Data lane timer	

Table 23. PHY transition timing configuration registers (continued)

Description	Register field
Configure the maximum time that the D-PHY data lane takes to go from High-Speed to Low-Power transmission measured in lane byte clock cycles	DSI_DLTCR.HS2LP_TIME
Configure the maximum time that the D-PHY data lane takes to go from Low-Power to High-Speed transmission measured in lane byte clock cycles	DSI_DLTCR.LP2HS_TIME

These timings allow the DSI host to know the LP to HS and the HS to LP transitions overhead to see if it has enough time to go to LP mode during video blanking periods.

It is mandatory to set LP2HS_TIME and HS2LP_TIME for data lanes in DSI_DLTCR register.

For clock lane, LP2HS_TIME and HS2LP_TIME are to be set in the DSI_CLTCR register, only when “automatic clock lane control” is enabled.

LP2HS_TIME reflects the maximum time required by the PHY to switch between LP and HS, while HS2LP_TIME reflects the maximum time required by the PHY to switch between HS and LP.

The DSI host compares blanking period with total transition time in order to know whether it can switch to LP during a blanking period or not.

If (period timing) > (total transition time), then the DSI host requests the D-PHY to go to LP.

If (period timing) < (total transition time), then the DSI host sends blanking packet in HS mode during that period.

- PHY transition timing:
  - Data lane
    LP2HS= 17 lanebyteclk
    HS2LP= 18 lanebyteclk
  - Clock lane
    LP2HS= 36 lanebyteclk
    HS2LP= 28 lanebyteclk
- Total transition time calculation:
  - Only data lanes go to LP
    Transition time=HS2LP_TIME (DATA) + LP2HS_TIME (DATA)
  - Clock and data lanes go to LP
    Transition time= HS2LP_TIME (clock) + LP2HS_TIME (clock)

Example calculation

This example assumes that:

- HS2LP_TIME = 18 lane_byte clock cycle
- LP2HS_TIME = 17 lane_byte clock cycle
- HFP period = 35 pixel_clock cycles

HS2LP_TIME +LP2HS_TIME = 35 lane_byte clock cycle.

HFP > (HS2LP + LP2HS), so the DSI host goes to LP in the HFP region.

Note: The HS2LP_TIME and LP2HS_TIME values must be carefully set to reflect the maximum time required by the PHY to switch between low-power and high-speed. Otherwise, if the timing provided to the DSI host is less than actual required transition timing, the DSI host may request an LP transition in a period that is shorter than the PHY transition time; which may cause a video timing violation.

The programmed timing must not be much bigger than the actual required timing by the PHY, because the DSI host uses this timing for internal calculation.

LP transitions configuration

Configure the low-power transitions in the DSI_VMCR to define the video periods that are permitted to go to low-power if there is time available to do so.

Table 24 shows the register fields used to program LP settings for each region.

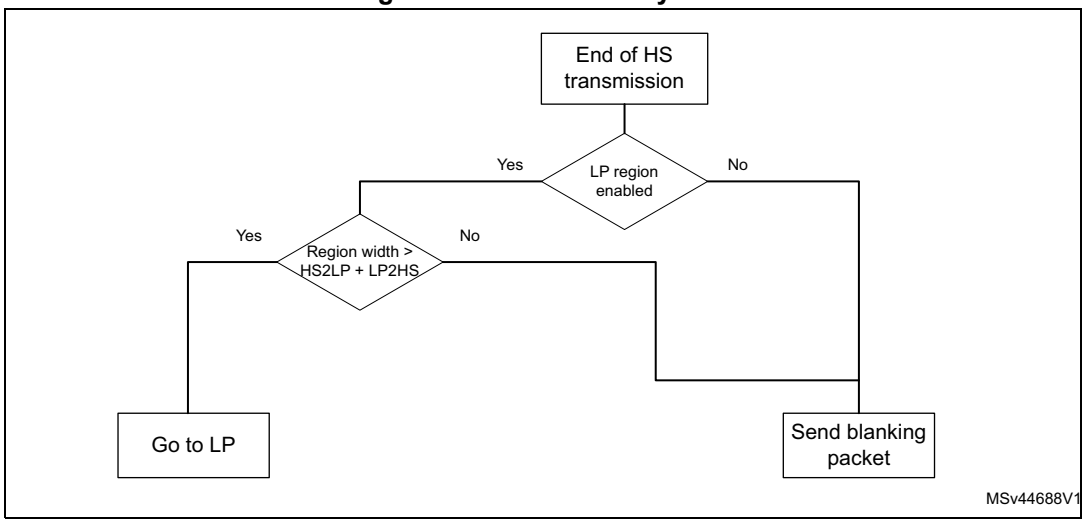
Table 24. LP transitions configuration registers

Description	Register field
Low-power horizontal front-porch enable	DSI_VMCR.LPHFPE
Low-power horizontal back-porch enable	DSI_VMCR.LPHBPE
Low-power vertical active enable	DSI_VMCR.LPVAE
Low-power vertical front-porch enable	DSI_VMCR.LPVFPE
Low-power vertical back-porch enable	DSI_VMCR.LPVBPE
Low-power vertical sync active enable	DSI_VMCR.LPVSAE

Figure 81 shows the basic LP mode entry flow. If the LP entry is disabled in a region, the DSI host sends blanking packets instead of entering the LP mode.

If the LP entry is enabled in one region, the DSI host checks if the period length is long enough to enter and exit LP mode. This is done by comparing the region’s period with the HS2LP and LP2HS transition timings.

Figure 81. LP mode entry flow

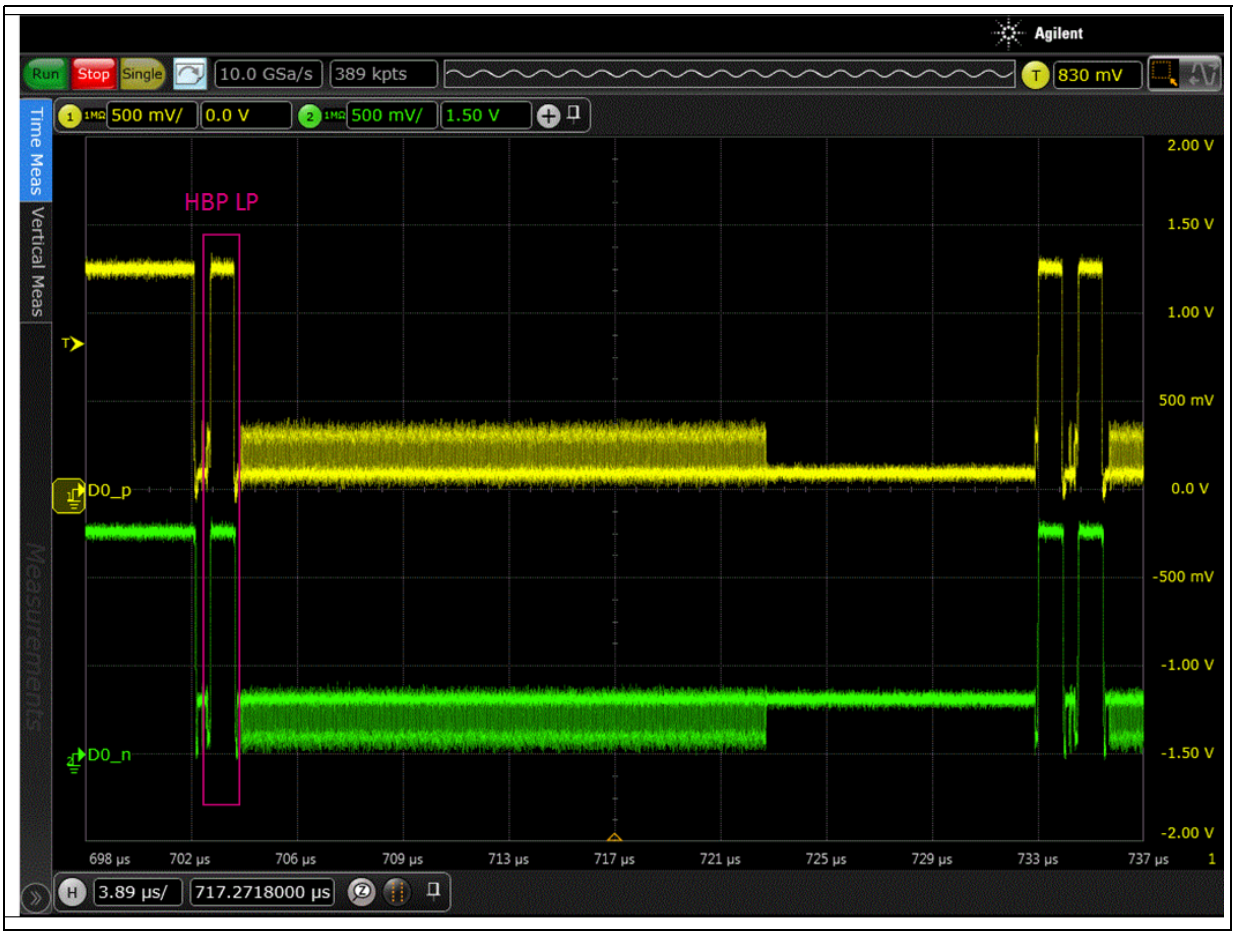


- LP mode disabled in HBP region
  When VMCR.LPHBPE=0 the DSI host sends blanking packet in HS mode during HBP period. Then it sends pixel data using the packed pixel stream packet (see *Figure 82*).

Figure 82. LP disabled in HBP region

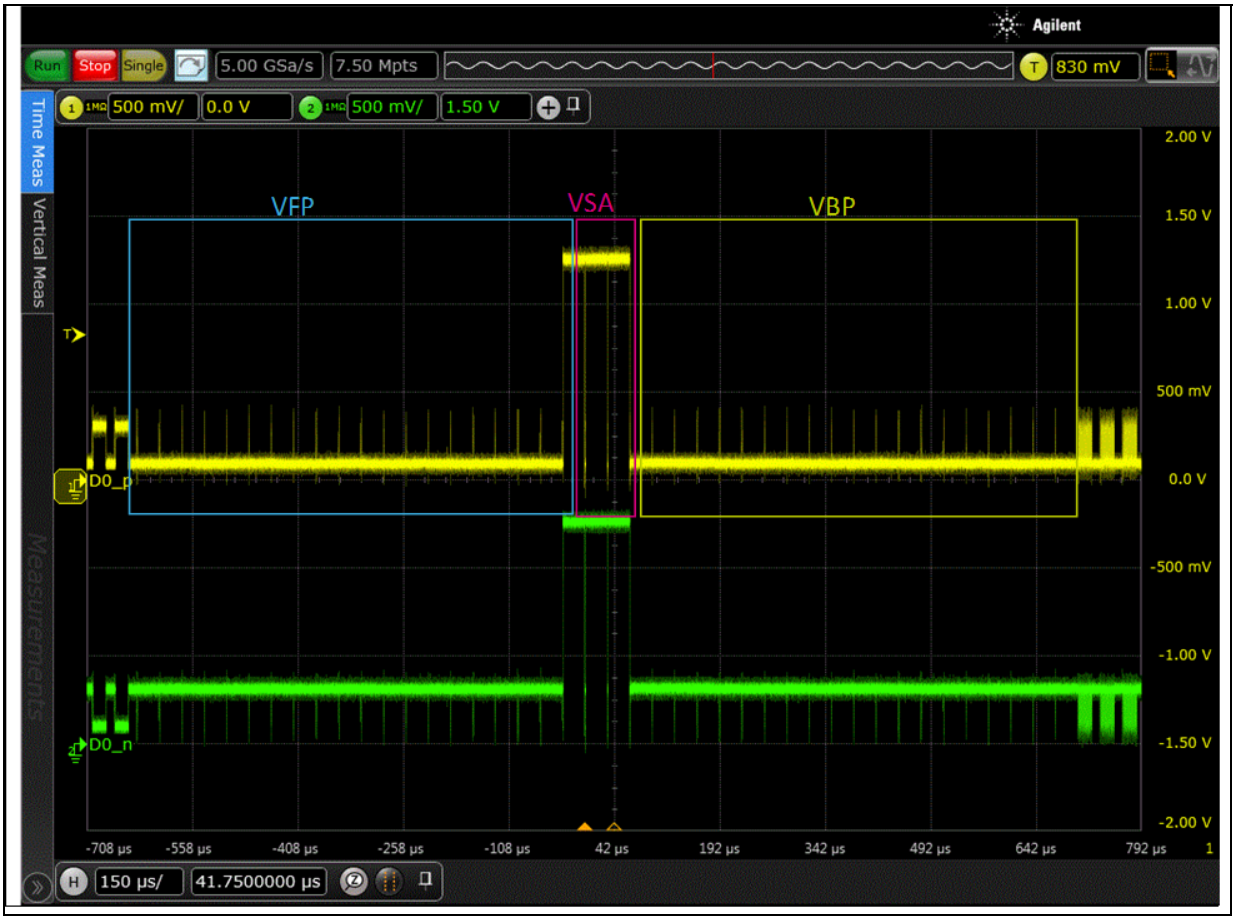
- LP mode enabled in HBP region
  When VMCR.LPHBPE=1, the DSI host goes to LP during HBP period if HBP period >HS2LP+LP2HS. Then it goes back to HS mode and sends pixel data using packed pixel stream packet (see *Figure 83*).

Figure 83. LP enabled in HBP region



- LP mode enabled only in VSA region
  Figure 84 shows a case where LP mode is only enabled in the VSA region.

Figure 84. LP mode enabled only in VSA region

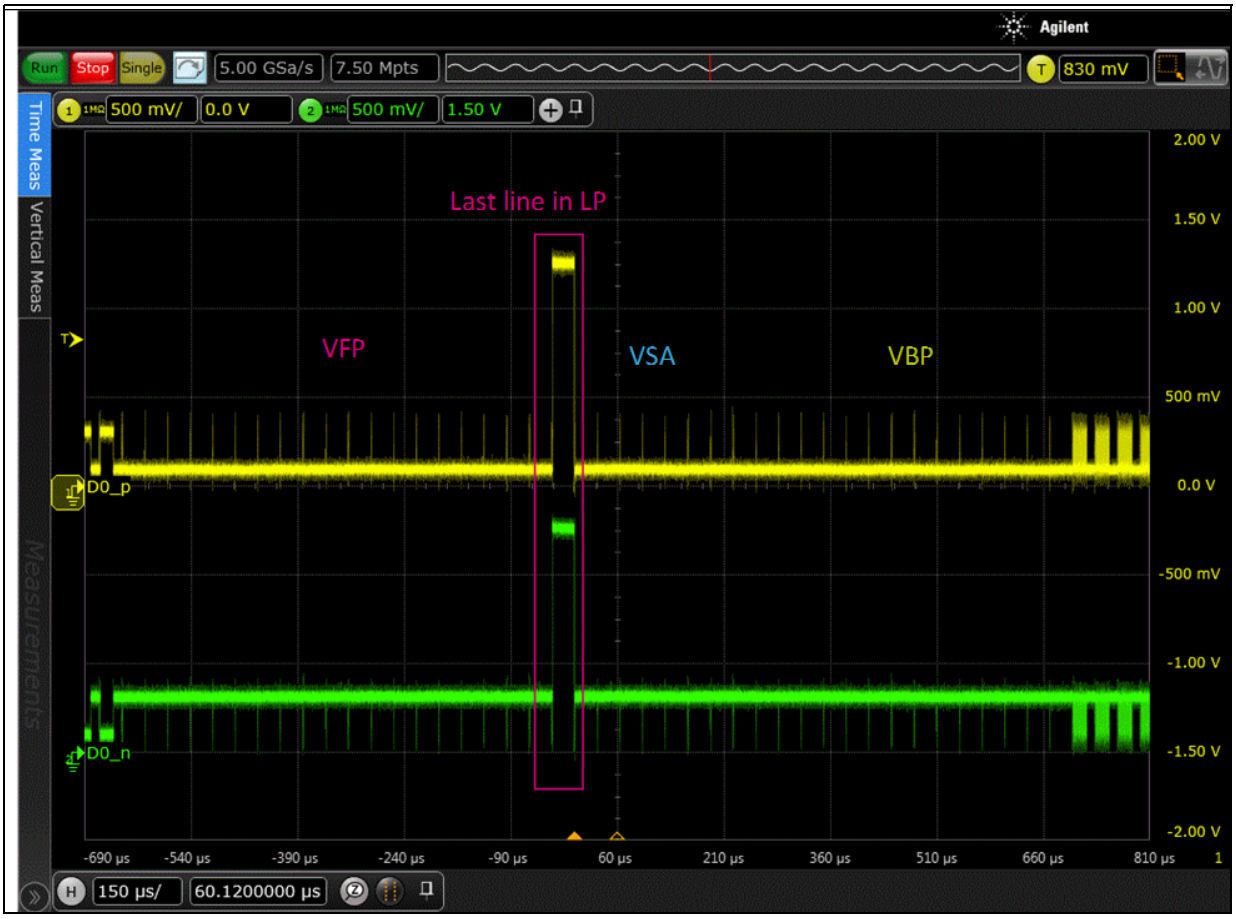


Note: The last line of frame always goes to LP mode, even if LP mode is not enabled for the VFP region. This is to ensure that the DSI link enters the LP mode at least once per frame.

- LP mode disabled for all regions

  The STM32 DSI host ensues that the host goes to LP state at least once per frame. This happens at the last line of the frame even if no region is configured to go to LP in DSI_VMCR register (see *Figure 85*).

Figure 85. Last line in low-power mode



LTDC settings

The DSI host relies on the LTDC to stream the pixel data and video control signals. The LTDC configuration is crucial for the good operation of the DSI host.

- LTDC video timing

  The frame's vertical and horizontal timing values are retrieved from the display datasheet.

  Since the DSI specification recommends entering the LP state at each scanline, the user may choose an horizontal timing in a way that it allows the DSI link to go to LP once per scanline (as long as the timing is compliant with the display timing specification).

  Refer to *Table 25* for a display timing example.

Table 25. Display timing example

Item	Specification			Unit
	Min	Typical	Max	
Vertical timing				
VSA	2	2	63	HS
VBP	20	20	255	HS
VFP	18	20	255	HS
Vertical blanking period	40	42	1024	HS
VACT	-	480	-	HS
Vertical refresh rate	-	60	-	Hz
Horizontal timing				
HSA	1	10	63	PCLK
HBP	3	15	63	PCLK
HFP	4	16	63	PCLK
Horizontal blanking period	8	32	128	PCLK
HACT		800		PCLK
fPCLK	24	26.36	30.74	MHz

The horizontal timings are chosen to allow an LP transition once per scanline.

– Horizontal timing in pixel clock:

HSA = 5, HBP = 35, HACT = 800, HFP = 35

With these HBP and HFP values, the link goes to LP during both HBP and HFP periods as explained in *Section : PHY transition timing configuration on page 83*.

– Vertical timing in lines:

VSA = 2, VBP = 20, VACT = 480, VFP = 20

- LTDC pixel clock setting

The pixel clock is set according to following formula:

pixel_clock = (VSA+VBP+VACT+VFP) x (HSA+HBP+HACT+HFP) x frame rate

Example calculation:

– Horizontal timing in pixel_clock:

HSA = 5, HBP = 35, HACT = 800, HFP = 35

– Vertical timing in lines:

VSA = 2, VBP = 20, VACT = 480, VFP = 20

– Refresh rate = 60 fps.

– pixel_clock = (2+20+480+20) x (5+35+800+35) x 60= 522 x 875 x 60 = 27.4 MHz

Refer to *Section 7: DSI host performance* for more details on the maximum pixel clock supported.

DSI host video timing

It is mandatory to provide the DSI host with display video timing information. *Table 26* shows the DSI host registers used to configure video timing.

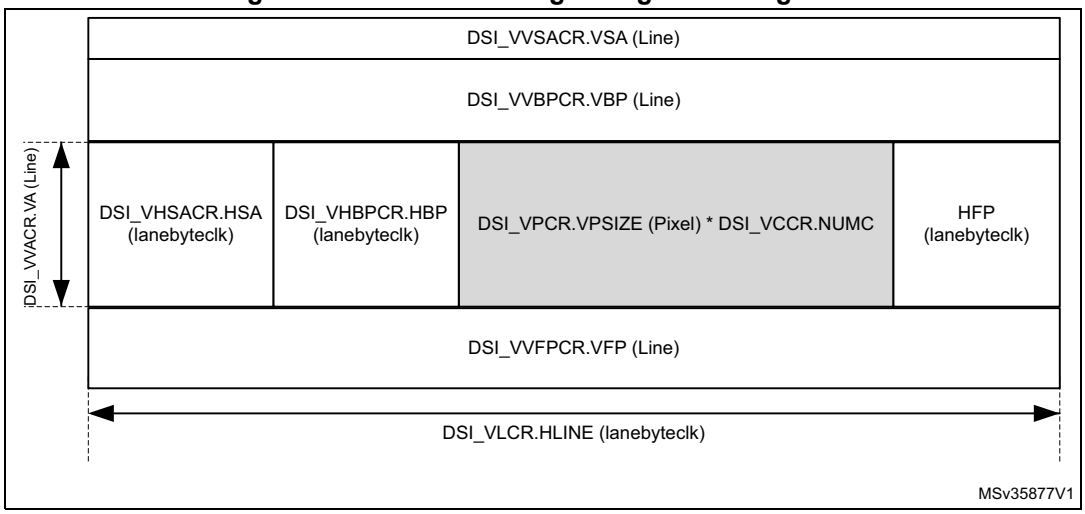
Table 26. DSI host video timing registers

Description	Register field
Frame horizontal timing	
Set the horizontal sync active (HSA) duration in lane byte clock cycles	DSI_VHSACR.HSA
Set the horizontal back porch (HBP) duration in lane byte clock cycles	DSI_VHBPCR.HBP
Set the horizontal line time (HSA+HBP+HACT+HFP) counted in lane byte clock cycles	DSI_VLCR.HLINE
Frame vertical timing	
Configure the vertical active (VSA) duration measured in number of horizontal lines	DSI_VVSACR.VSA
Configure the vertical back porch (VBP) duration measured in number of horizontal lines	DSI_VVBPCR.VBP
Configure the vertical front porch (VFP) duration measured in number of horizontal lines	DSI_VVFPCR.VFP
Configure the vertical active (VACT) duration measured in number of horizontal lines	DSI_VVACR.VA

The video timing in DSI host must have the same length as in the LTDC. The values in the LTDC are expressed in pixel_clock cycles and the values programmed in the DSI are expressed in lane byte clock cycles.

Video timing registers are described in *Figure 90*.

Figure 86. DSI video timing configuration registers



- Frame horizontal timing
  - Configure the horizontal sync duration (DSI_VHSACR.HSA) with the time taken by an LTDC horizontal sync active period measured in cycles of lane byte clock
  - Configure the horizontal back porch duration (DSI_VHBPCR.HBP) with the time taken by the LTDC horizontal back porch period measured in cycles of lane byte clock
  - Configure the horizontal line time (DSI_VLCR.HLINE) with the time taken by an LTDC video line measured in cycles of lane byte clock
- Frame vertical timing
  - Configure the vertical sync duration (DSI_VVSACR.VSA) with the number of lines existing in the LTDC vertical sync active period.
  - Configure the vertical back porch duration (DSI_VVBPCR.VBP) with the number of lines existing in the LTDC vertical back porch period.
  - Configure the vertical front porch duration (DSI_VVFPCR.VFP) with the number of lines existing in the LTDC vertical front porch period.
  - Configure the vertical active duration (DSI_VVACR.VA) with the number of lines existing in the LTDC vertical active period.

  Example configuration:

  DSI_VLCR.HLINE = (HSA + HBP + HACT + HFP) x (lane_byte_clk / pixel_clock) = (5 + 35 + 800 + 35) x (62.5 / 27.4) = 1995

  DSI_VHSACR.HSA = HSA x (lane_byte_clk/pixel_clock) = 5 x (62.5 / 27.4) = 11

  DSI_VHDPCR.HBP = HBP x (lane_byte_clk/pixel_clock) = 35 x (62.5 / 27.4) = 80

  Vertical line configuration:

  DSI_VVSACR.VSA = 2

  DSI_VVBPCR.VBP = 20

  DSI_VVFPCR.VFP = 20

  DSI_VVACR.VA = 480

DSI clock setting

The DSI clock depends on the used video mode:

- Burst mode:

  In burst mode, the DSI clock can be put at its maximum so that the link goes to low-power mode for longer periods.

  The maximum lane_byte_clock supported is 62.5 MHz.
- Nonburst mode:

  The DSI nonburst mode should be configured in a way that allows the DSI output pixel ratio to match with the LTDC interface input pixel ratio. This is discussed in the section below.

DSI video packet parameters

Table 27 shows the DSI host registers needed to configure the video packets during the video active region.

Table 27. DSI video packet parameters registers

Description	Register field
Configure the video packet size in pixels	DSI_VPCR.VPSIZE
Configure the number of chunks	DSI_VCCR.NUMC
Configure the null packet size in bytes	DSI_VNPCR.NPSIZE

The video packet parameters configuration depends on the video mode chosen:

- Burst mode

  In burst mode, the VPSIZE must be equal to an entire line length in pixels. So, the number of chunks can be set to 0 to transmit the video line in a single packet.

  After sending the packed pixel stream, the link goes to LP mode to save power. There is no need for null packet (NPSIZE = 0).

  Figure 87 shows a video line in burst mode.

Figure 87. Video line in burst mode

- Nonburst mode

  The DSI nonburst mode should be configured in a way that allows matching the DSI output pixel ratio with the LTDC interface input pixel ratio. This is achieved by dividing a pixel line (HACT region) into several chunks of pixels and optionally interleaving them with null packets.

  The following equations allow setting the DSI host transmission parameters in nonburst mode. Both equations allow balancing the time required to output the pixels on DSI (right side of equation) with time required to input pixels from LTDC (left side of equation).

  – When the null packets are enabled:
  Equation 1
  lanebyteclkperiod x NUMC (VPSIZE x bytes_per_pixel + 12 + NPSIZE) / number_of_lanes = pixels_per_line x LTDC_clock_period

  – When the null packets are disabled
  Equation 2
  lanebyteclkperiod x NUMC (VPSIZE x bytes_per_pixel + 6) / number_of_lanes = pixels_per_line x LTDC_clock_period

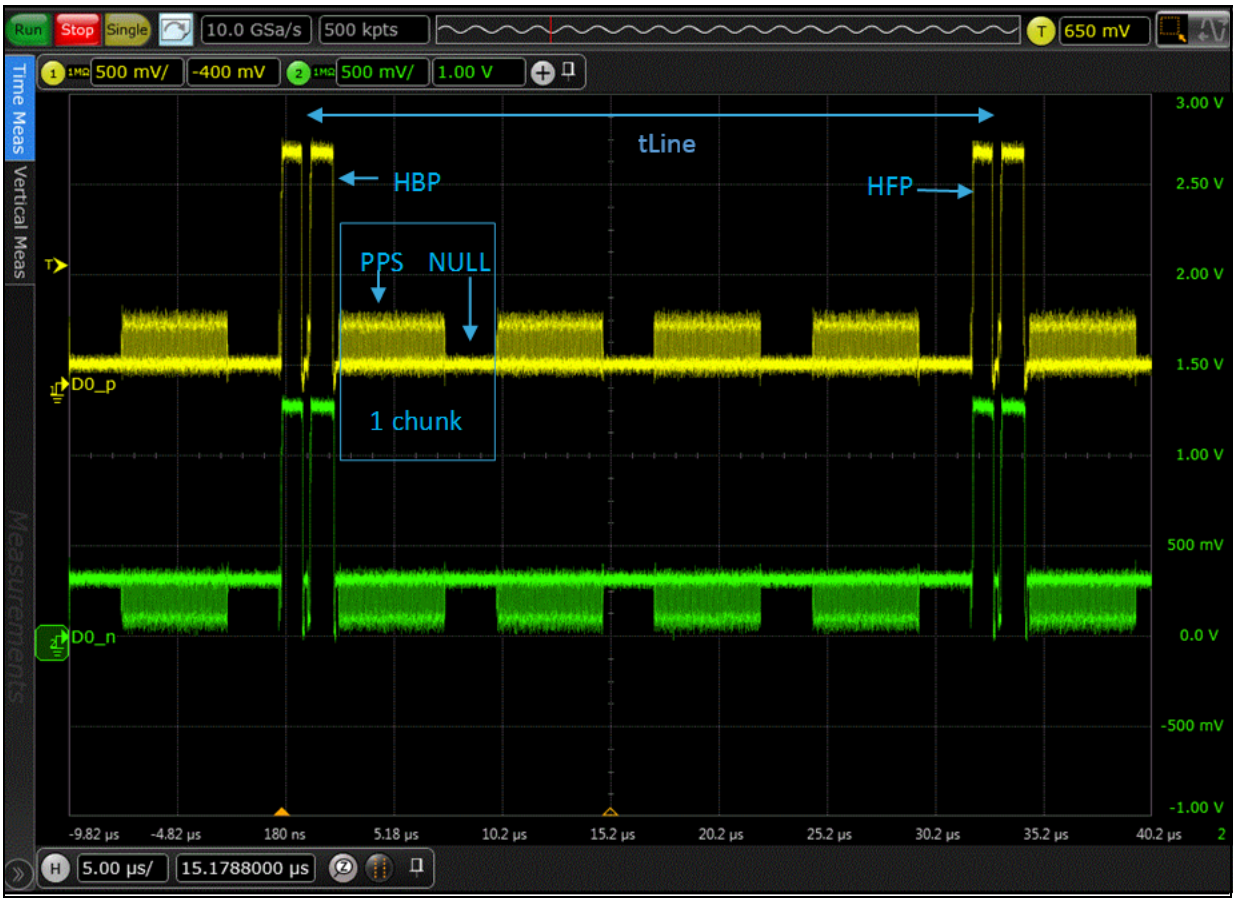
Note: VPSIZE = video packet size and NUMC = number of chunks.

Example of configuration with four chunks with null packets enabled:

- Pixel_clock is calculated using video timing parameters (see section *Section : LTDC settings on page 89*); pixel_clock = 27.429 MHz.
- In this case, the Lane_byte_clk is set at its maximum speed: 62.5 MHz.
- Pixels per packet and number of chunks are set to match the line width:
  Line width in pixels = VPSIZE x NUMC.
  VPSIZE depends on the FIFO size inside the display. In this example we set the VPSIZE to 200 pixels and the number of chunks to 4.
- Using equation 2 presented above, the DSI outputs pixels much faster than the pixels input from LTDC:
  DSI pixels output time = 4(200 x 3 + 6) / (2 x 62.5 MHz) = 19 us
  LTDC interface pixels input time = 800 / 27.4 MHz = 29 us
  This requires the use of null packets in order to balance the LTDC and the DSI throughput using equation 1 above.
- Null packet size calculation:
  NPSIZE is calculated using the equation 1 above.
  4 x (200 x 3 +12 +NPSIZE) / (62.5 x2) = 800 x 1 / 27.4
  NPSIZE=299 bytes.

Figure 87 shows a video active line with four chunks with null packets configuration:

Figure 88. Four chunks with null packets configuration



Example of configuration with four chunks and without null packets enabled

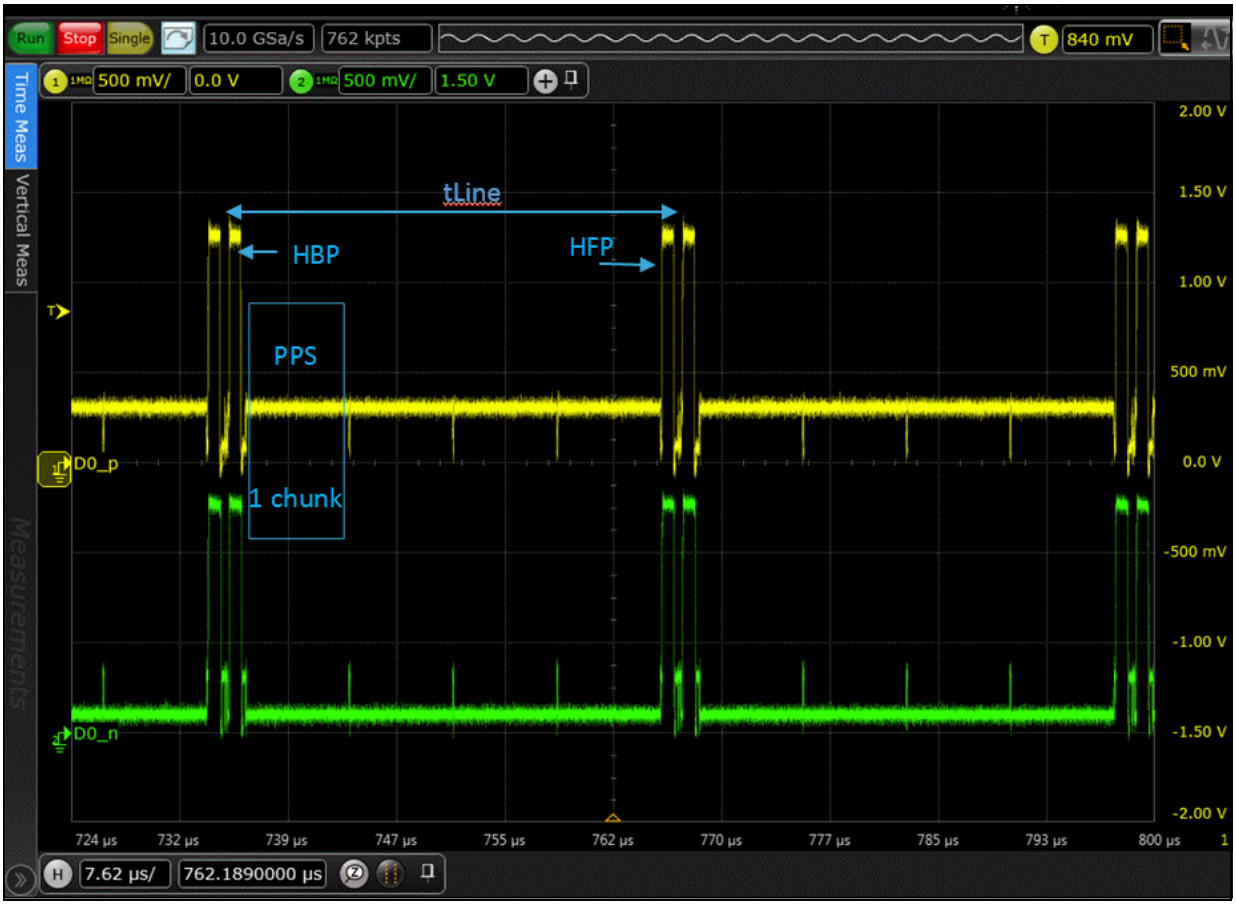
In order to remove the need for null packets, the DSI and LTDC throughput must be balanced using equation 2 above.

This example uses a four chunks configuration, and the lane_byte_clk is measured as follows:

– Lane_byte_clk = pixel_clock x NUMC x (VPSIZE x bytes_per_pixel + 6) / (pixels_per_line x num_lanes) = 41.24 MHz

Figure 89 shows an active video line with four chunks configuration without null packet, the DSI link is at 41.2 MHz.

Figure 89. Four chunks without null packets configuration



Note: The video packet size (VPSIZE) and consequently the number of chunks (NUMC) have to be set in concordance with the display's internal FIFO size. For example, if the internal FIFO only allows to accommodate 200 pixels, then the number of chunks must be greater than 4 to ensure that VPSIZE is equal to 200 pixels or less.

Command transmission in video mode

During video transmission, the DSI host allows sending commands through the APB generic interface during blanking regions. Commands may be sent in low-power or high-speed mode.

If the command is transmitted in HS mode, the DSI host automatically determines the area where each command can be sent and no programming or calculation is required.

For LP commands, the DSI host needs an input from the user to determine the appropriate area in which the command may be transmitted.

- Command transmission mode

Table 28 shows the register bit used to select between HS and LP transmission of commands in video mode.

Table 28. Command transmission mode register

Description	Register field
Configure if commands are to be transmitted in low-power	DSI_VMCR.LPCE

If DSI_VMCR.LPE = 1 then the commands are sent in LP mode, otherwise they are sent in HS mode.

Note: Some displays require initialization commands to be sent in LP mode. In that case, the LP command transmission must be enabled during the initialization phase.

- LP command packet size

When the LP command transmission is enabled, the user has to inform the DSI host of the maximum allowed packet size in bytes using the register fields defined in *Table 29*.

Table 29. LP command packet size registers

Description	Register field
Largest packet size	DSI_LPMCR LPSIZE
VACT largest packet size	DSI_LPMCR VLPSIZE

When the DSI host is configured to send the low-power (LP) commands during the HS video mode transmission(DSI_VMCR.LPCE=1), it is necessary to calculate the time available, in bytes, to transmit a command in LP mode during the horizontal front porch (HFP), the vertical sync active (VSA), the vertical back porch (VBP), and the vertical front porch (VFP) regions.

- LPSIZE: largest packet size out of VACT region

  This field is used for the transmission of commands in LP mode. It defines the size, in bytes, of the largest packet that can fit in a line during VSA, VBP and VFP regions.
- VLPSIZE: largest packet size in VACT region

  This field is used for the transmission of commands in LP mode. It defines the size, in bytes, of the largest packet that can fit in a line during the HFP of the VACT region.

Note: It is important to correctly configure LPSIZE and VLPSIZE. If LPSIZE and VLPSIZE are higher than actual available time, it may cause a video timing violation. In the other hand, if LPSIZE and VLPSIZE are much lower than the available timing, many commands are delayed to the last line of the frame.

Note: It is recommended to avoid sending lines on the last line of the frame. If large commands have to be sent, the user may disable the DSI video mode and send the command in DSI command mode, then reenable the video mode.

LPSIZE calculation

Figure 90 shows LPSIZE calculation for video nonburst mode with sync pulses.

Figure 90. LPSIZE for nonburst with sync pulses

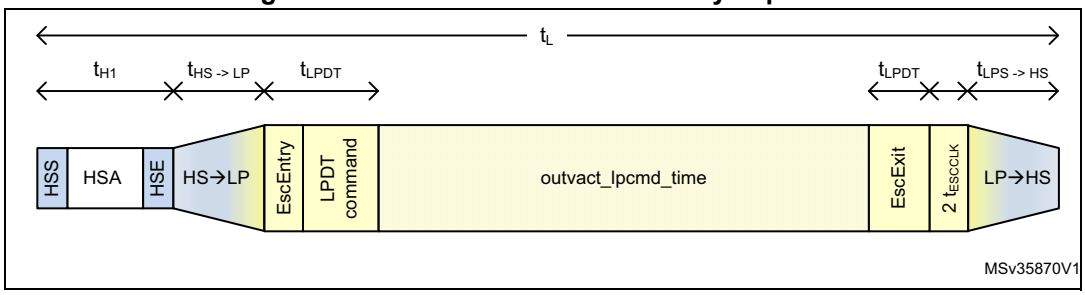
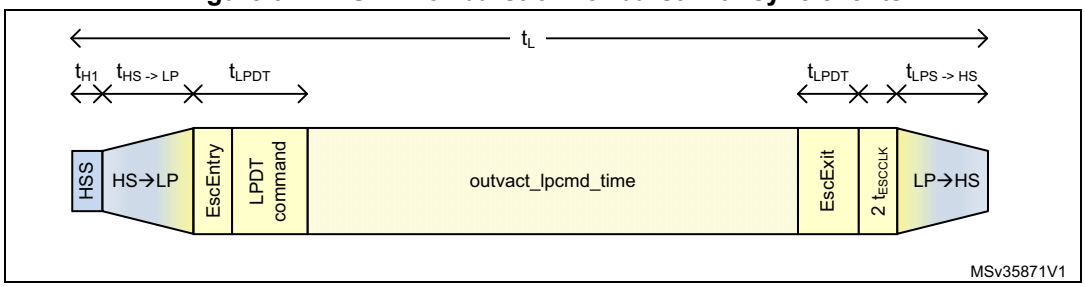


Figure 91 shows LPSIZE calculation for video burst mode and video nonburst mode with sync events.

Figure 91. LPSIZE for burst or nonburst with sync events



Where

t_L = line time = (HSA + HBP + HACT + HFP) / PCLK;

tH1

Nonburst mode with sync pulses:

t_{H1}= time of the HSA pulse = tHSA = HSA / PCLK

Burst mode or nonburst mode with sync events:

tH1= time to send the HSS packet = 4 bytes / (lane_byte_clk x Number_Lanes)

tHS->LP = time to enter the low-power mode;

tLP->HS = time to leave the low-power mode;

tLPDT = D-PHY timing related with escape mode escape, LPDT command, and escape exit. According to the D-PHY specification, this value is always 11 bits in LP (or 22 TX escape clock cycles);

tESCCLK = escape clock period= DSI_CCR.TXECKDIV /lane_byteclk

2 tESCCLK = delay imposed by the DSI host implementation.

Example calculation:

t_L = (5 + 35 + 800 + 35) / 27.429 MHz = 31.9 us

t_{H1}

Nonburst mode with sync pulses:

t_{H1}=tHSA =5/27,429 MHz=0,182us

Burst mode or nonburst mode with sync events:

tH1 = 4 / (62.5 x 2) = 0.032 us

tHS->LP = 291 ns (this is D-PHY specific timing)

tLP->HS = 264 ns (this is PHY specific timing)

tESCCLK = 4 / 62.5 = 0.064 ns

tLPDT = 22 x tESCCLK = 1.408

LPSIZE in burst and nonburst with sync events modes:

LPSIZE = (31.9 - (0.032 + 0.291 + 0.264 + 1.408 + 0.128)) / (2 x 8 x 0.064) = 29.07

LPSIZE in nonburst with sync pulses mode:

LPSIZE = (31.9 - (0.182 + 0.291 + 0.264 + 1.408 + 0.128)) / (2 x 8 x 0.064) = 28.93

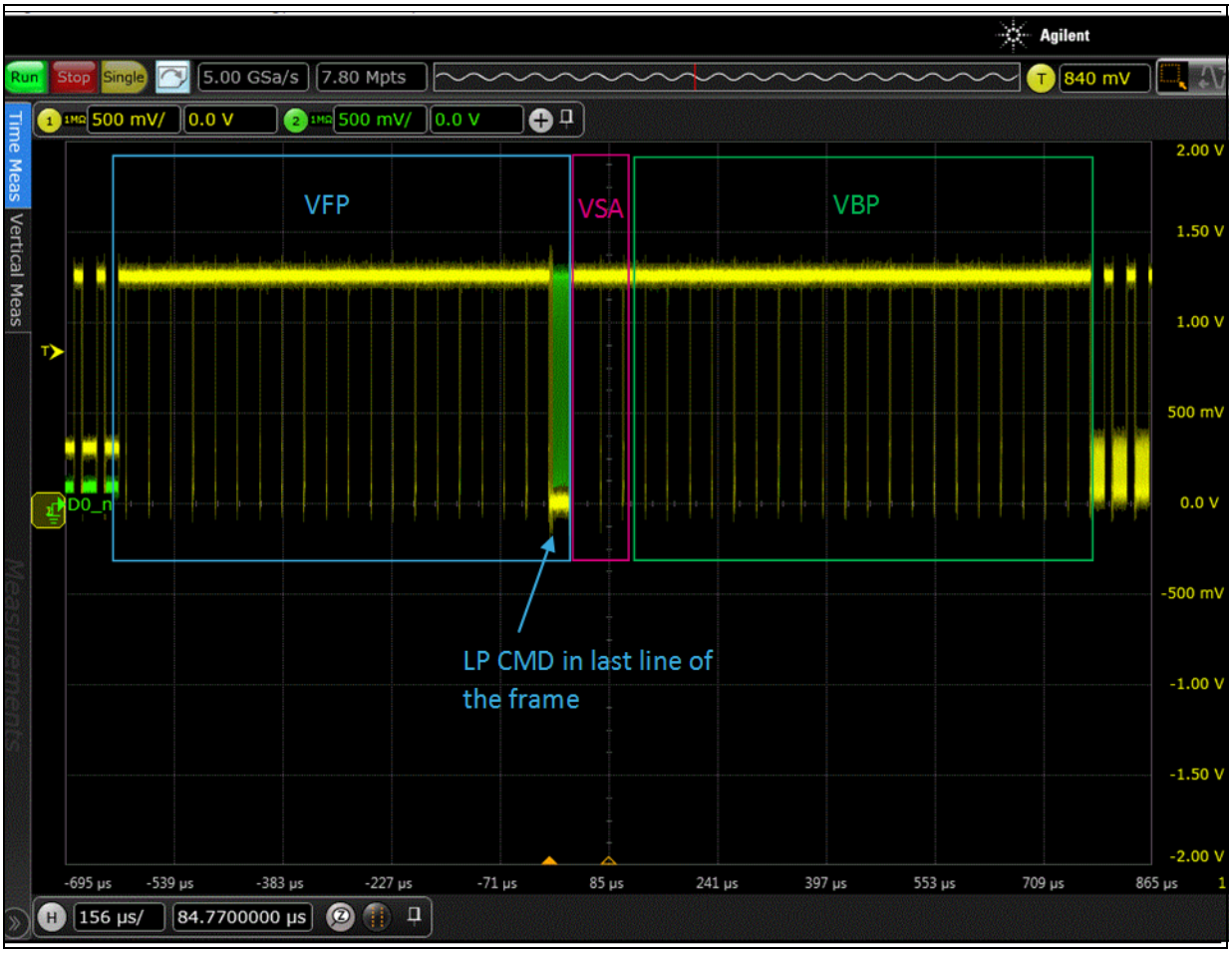
In order to have some margin, the LPSIZE to be programmed is 28 bytes in both cases.

Figure 92 shows a 28-bytes command sent in LP mode during a blanking region. The command fits into the blanking period without causing a time line violation.

Figure 92. 28 bytes LP CMD during VFP in burst mode

Figure 93 shows a 29-bytes command which does not fit into active or blanking regions. In this case, the DSI host postpones the command to the last line of the frame.

Figure 93. 29 bytes LP CMD delayed to last line



VLPSIZE calculation

Figure 94 shows VLPSIZE calculation for video nonburst mode with sync pulses.

Figure 94. VLPSIZE for nonburst with sync pulses

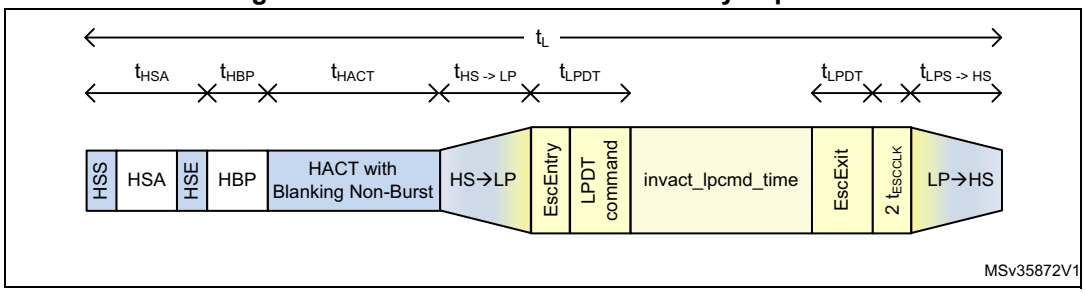


Figure 95 shows VLPSIZE calculation for video nonburst mode with sync events.

Figure 95. VLPSIZE for nonburst with sync events

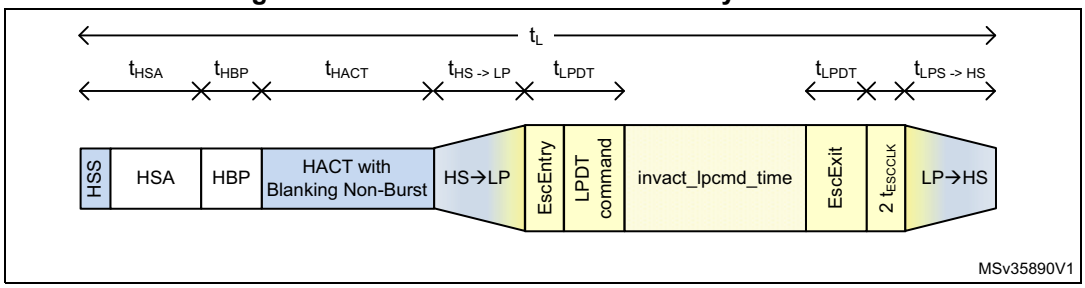
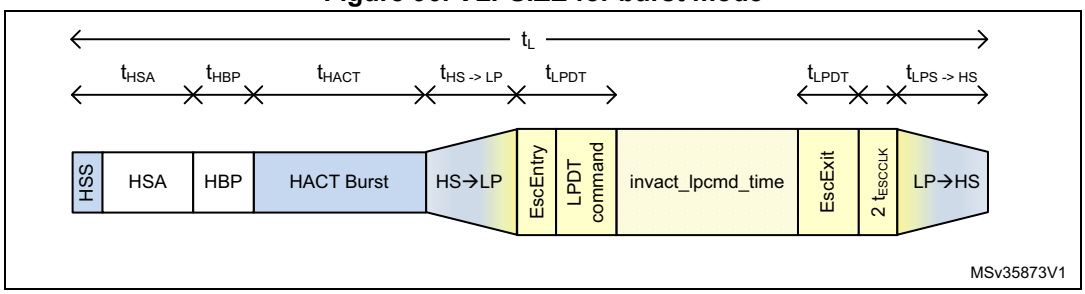


Figure 96 shows VLPSIZE calculation for video burst mode.

Figure 96. VLPSIZE for burst mode



Where

t_L = line time = (HSA + HBP + HACT + HFP) / PCLK;

tHSA = time of the HSA pulse = HSA / PCLK;

tHBP = time of horizontal back porch= HBP / PCLK;

tHACT = time of video active.

Nonburst mode: pixels_per_line / PCLK

Burst mode: the video active is time compressed and is calculated as

t_{HACT} = (VPSIZE x Bytes_per_Pixel) / (Number_Lanes x $t_{Lane\_byte\_clk}$);

tESCCLK = escape clock period= DSI_CCR.TXECKDIV / lane_byteclk

Example calculation:

tL = line time = (5 + 35 + 800 + 35) / 27.4 MHz = 31.9 us

tHSA = 5 / 27.4 MHz = 0.182 us

tHBP = 35 / 27.4 MHz = 1.27 us

tHACT

Nonburst mode:

t_{HACT} =800 / 27.4 MHz = 29.16 us

Burst mode:

t_{HACT} = (800 x 3) / (2 x 62.5 MHz) = 19.2 us

VLPSIZE in burst mode

VLPSIZE = (31.9 - (0.182 + 1.27 + 19.2 + 0.291 + 0.264 + 1.408 + 0.128)) / (2 x 8 x 0. 064) = 8.94

8-bytes long packets are allowed in burst mode VACT region.

VLPSIZE in nonburst mode

VLPSIZE = (31.9 - (0.182 + 1.27 + 29.16 + 0.291 + 0.264 + 1.408 + 0.128)) / (2 x 8 x 0.064) = -0.78

➔ No packets are allowed in VACT region.

Figure 97 shows an 8-bytes command sent in LP mode during a video active line. The command is sent without causing timing violation of the active line.

Figure 97. 8 bytes LP CMD during VACT in burst mode

Note: The user may take 10% margin on the allowed number of bytes (LPSIZE and VLPSIZE) to avoid video timing issues.

Frame acknowledge

In order to ensure that the frame has been correctly received by the display, the DSI host may ask for frame acknowledge.

Table 30 shows the register bit used to enable frame acknowledge in video mode.

Table 30. Frame acknowledge register

Description	Register field
Frame bus-turnaround acknowledge enable	DSI_VMCR.FBTAAE

During the last line of the frame, the host performs a BTA procedure. Then, the display takes control of the bus and sends an acknowledge trigger if all previous packets have been received with no errors. This means that the frame has been correctly received by the display. After that the display performs a BTA sequence to give back bus control to the DSI host. If the display has encountered errors from the previous packets it responds with an error report.

Note: It is mandatory to set DSI_PCR.BTAE to 1 in order to enable the bus turnaround (BTA) request.

Figure 98 and *Figure 99* shows an example of frame acknowledge trigger in video mode.

Figure 98. Frame acknowledge example

Figure 99. Zoom on frame acknowledge

5.2.2 Adapted command mode over LTDC interface

This section describes the DSI host settings related to the adapted command mode.

DSI command mode

The adapted command mode must be selected in both DSI host and DSI Wrapper (see *Table 31*).

Table 31. DSI command mode registers

Description	Register field
Selects the mode for the video transmission: video mode or adapted command mode	DSI_WCFGR.DSIM
Set the DSI host in either video or command mode	DSI_MCR.CMDM

Select the adapted command mode in the DSI Wrapper configuration register (DSI_WCFGR.DSIM=1).

Select the command mode in the DSI host mode configuration register (DSI_MCR.CMDM=1).

Stop wait time configuration

The register fields in *Table 32* configure the minimum wait period to request a HS transmission after the stop state.

Table 32. Stop wait time configuration register

Description	Register field
Configure the minimum wait period to request a high-speed transmission after the stop state.	DSI_PCONFR.SW_TIME

The SW_TIME is mandatory for the transmitter D-PHY to guarantee that all data lanes are in stop state before a new HS transmission is initiated.

A display may also require a specific time before receiving a new HS transmission. This must be checked in the display's datasheet.

The programmed value must be the maximum between host SW_Time and display SW_TIME.

Note: The minimum SW_Time for the DSI host is 10 lanebyteclk command mode.

Command size (CMDSIZE)

The user has to define the size in pixels of the DCS long write commands (WMS and WMC) used in adapted command mode to refresh the display's GRAM. The command size is set in register described in *Table 33*.

Table 33. Command size register

Description	Register field
This field configures the maximum allowed size for an LTDC write memory command in pixels	DSI_LCCR.CMDSIZE

The DSI host pixel FIFO size is 960 32-bit words. This implies that:

- In 24 bpp mode this field shall not exceed 1280 pixels.
- In 16 bpp mode this field shall not exceed 1920 pixels.

LTDC halt polarity

Select the VSYNC edge on which the LTDC is halted after frame refresh has finished. The VSYNC edge is set as described in *Table 34*

Table 34. LTDC halt polarity

Description	Register field
Configure the DSI Wrapper VSYNC polarity	DSI_WCFGR.VSPOL

LTDC can be halted either on falling or rising edge of VSYNC. The edge polarity must be consistent with the VSYNC polarity of the LTDC interface: if DSI_LPCR.VSP is active high, then LTDC must be halted on rising edge. If DSI_LPCR.VSP is active low, then LTDC must be halted on falling edge.

Tearing effect settings

The tearing effect in command mode is managed using the registers described in *Table 35*.

Table 35. Tearing effect setting registers

Description	Register field
TE source	DSI_WCFGR.TESRC
TE polarity	DSI_WCFGR.TEPOL
Tearing effect acknowledge request enable	DSI_CMCR.TEARE

- Tearing effect over link
  - TE source

    The tearing effect source (DSI_WCFGR.TESRC) of the DSI Wrapper configuration register, must be set to 0 for TE over link.
  - TE acknowledge request

    In the case of tearing effect reporting over link, the bit DSI_CMCR.TEARE of the DSI host command mode configuration register must be set.

Note: Note: The DSI_PCR.BTAE bit of the DSI host protocol configuration register must be set in order to allow the reverse communication for the TE reporting.

- Tearing effect over pin
  - TE source

    The tearing effect source (TESRC) must be set to 1 for TE over external pin.
  - TE polarity

    In the case of TE used over GPIO link, this must be programmed with respect to the TE polarity in the display. The DSI host supports both polarities (low and high).

Refresh mode

The DSI host supports two modes to start the display refresh operation. *Table 35* shows the register field used to choose the refresh mode.

Table 36. Refresh mode register

Description	Register field
Automatic Refresh	DSI_WCFGR.AR

The automatic refresh (AR) bit of the DSI Wrapper configuration register (DSI_WCFGR) is set if the display needs to be updated automatically each time that a tearing effect event is received.

- Automatic refresh: the DSI_WCR.LTDCEN is set automatically after receiving a TE event.
- Manual refresh: it is the software responsibility to refresh GRAM by setting DSI_WCR.LTDCEN bit after reception of TE event.

LTDC settings

In adapted command mode, the DSI host inputs the pixel stream from the LTDC. The LTDC pixel rate and video timing have a specific configuration when DSI operates in adapted command mode.

- LTDC pixel clock setting

The pixel clock frequency does not need to match the display pixel clock since the relies on its internal controller for timing information generation.

In adapted command, the LTDC pixel clock has to be selected to ensure following requirements:

  – Minimum pixel clock must be fast enough in order to ensure that GRAM refresh time is shorter than display internal refresh rate to avoid visual artifacts.
  – Maximum pixel clock must be consistent with system constraints to avoid FIFO under-run issues on LTDC side.

Refer to *Section 7: DSI host performance* for more details on the pixel clock minimum and maximum values in adapted command mode.

- LTDC video timing

Since the display does not rely on the host for timing information, it is possible to set all the vertical and horizontal blanking periods (HSA, HBP, HFP, VSA, VBP, VFP) to the minimum that is 1, but the user must set HACT and VACT correctly with line length and number of lines per frame respectively.

Command transmission mode

All the commands, depending on their type, can be transmitted or received either in high-speed or low-power. For each of them, a dedicated configuration bit is programmed in the DSI host command mode configuration register (DSI_CMCR) (see *Table 37*).

Table 37. Command transmission registers

Description	Register field
Maximum read packet size	DSI_CMCR.MRDPS
DCS long write transmission	DSI_CMCR.DLWTX
DCS short read zero parameter transmission	DSI_CMCR.DSR0TX
DCS short read one parameter transmission	DSI_CMCR.DSW1TX
DCS short write zero parameter transmission	DSI_CMCR.DSW0TX
Generic long write transmission	DSI_CMCR.GLWTX
Generic short read two parameters transmission	DSI_CMCR.GSR2TX
Generic short read one parameters transmission	DSI_CMCR. GSR1TX
Generic short read zero parameters transmission	DSI_CMCR. GSR0TX
Generic short write two parameters transmission	DSI_CMCR.GSW2TX
Generic short write one parameters transmission	DSI_CMCR.GSW1TX
Generic short write zero parameters transmission	DSI_CMCR.GSW0TX

Note: Some displays require initialization commands to be sent in LP mode. In this case, the commands used for display initialization must be configured in LP mode during the initialization phase.
After initialization phase, commands may be reconfigured to be sent in high-speed mode. This is important especially for DCS long write commands that are used to accommodate the WMS and WMC DCS commands used during the display refresh.

Acknowledge request

The DSI host may request an acknowledge after each sent command. This is enabled by setting DSI_ CMCR.ARE bit (see *Table 38*).

Table 38. Acknowledge request register

Description	Register field
Enables the acknowledge request after each packet transmission	DSI_ CMCR.ARE

When this feature is enabled, DSI host starts a BTA procedure after each command sent. The display takes control of the bus and responds with an acknowledge trigger or an error report in case of errors.

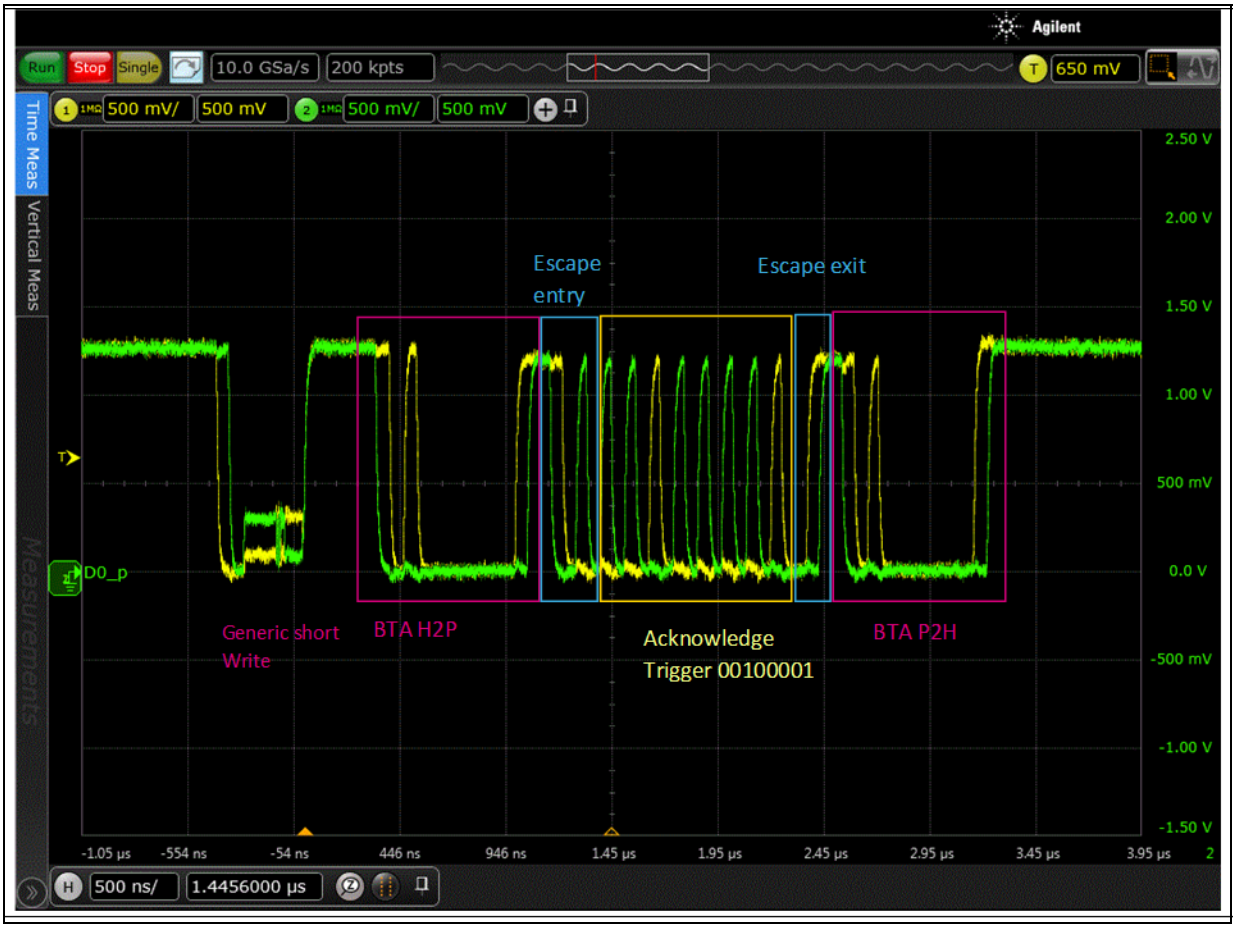
Then the display sends a BTA to give back bus control to the DSI host.

Note: This feature should be avoided in case of adapted command mode during refresh operation, since it adds a lot of overhead, slowing the display refresh time.

This feature can be useful in order to have a safer system and detect errors as soon as possible. So, its utilization depends on the application needs.

Figure 100 shows an example of write command with acknowledge request enabled.

Figure 100. Generic short write with acknowledge request enabled



6 STM32CubeMX configuration example

The STM32CubeMX tool can be used to configure the DSI host peripheral. This section shows basic configuration steps required to configure the DSI host in different operating modes: video burst mode, video nonburst mode with sync pulses and adapted command mode. This section also provides software code examples to configure both LTDC and DSI host in different operating modes. For more complete examples, the user may refer to the STM32Cube examples. The examples have been generated for the STM32F769I-Discovery board.

Since the DSI host uses the LTDC as video streamer, the LTDC configuration is mandatory. These examples show a very basic configuration of the LTDC. More information on LTDC configuration is available in application note AN4861.

6.1 DSI host video burst mode

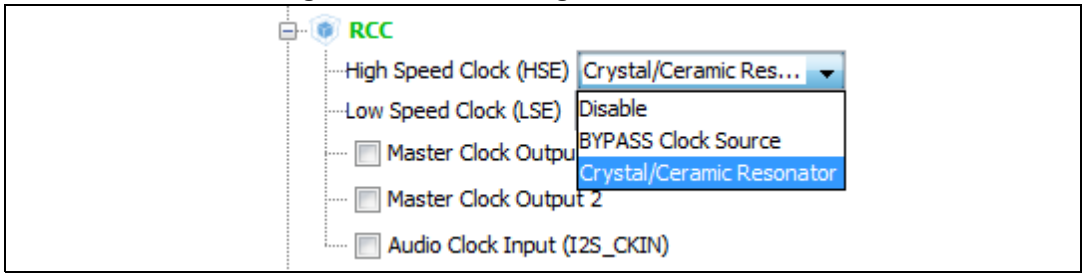
This section shows the different steps required to configure the DSI host in video burst mode. Only the mandatory settings are exposed.

6.1.1 Pinout configuration

Enable HSE in RCC

The DSI requires the use of the HSE (high-speed external) oscillator as clock input for DSI PLL. *Figure 101* shows how to configure the RCC (reset and clock control) to enable the HSE.

Figure 101. RCC configuration to use HSE

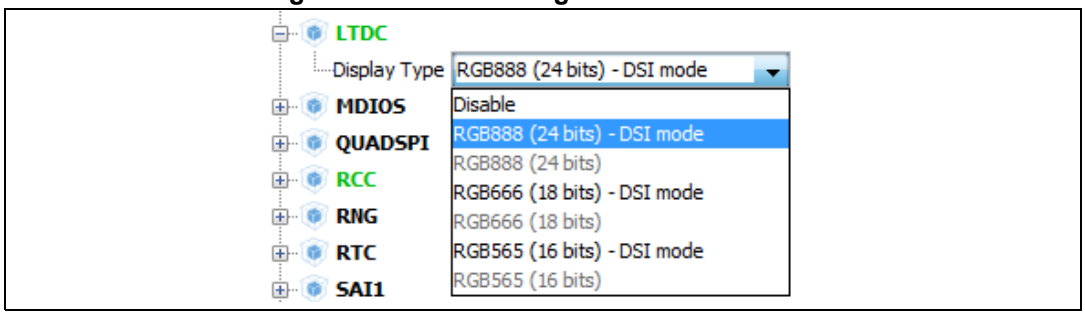


Enable LTDC in DSI mode

The LTDC must be enabled and set to DSI mode.

Note that the LTDC is used as video streamer to feed the DSI. So, in DSI mode, the LTDC output is connected on chip to the DSI host, which means that there is no pin configuration for LTDC. *Figure 102* shows the LTDC configuration.

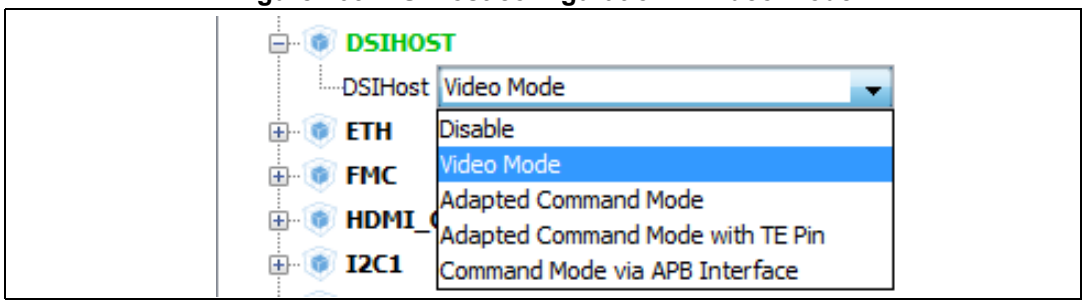
Figure 102. LTDC configuration in DSI mode



Enable DSI host in video mode

Select the video mode for the DSI host (see *Figure 103*).

Figure 103. DSI host configuration in video mode



Note: The DSI host uses dedicated pins, there is no alternate function configuration required.

6.1.2 Clock configuration

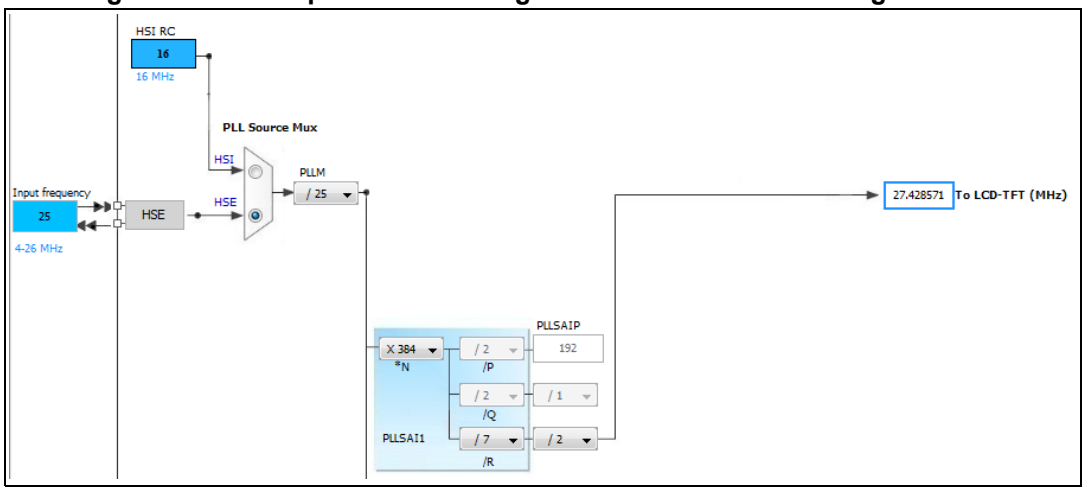
During the clock configuration phase, the DSI PLL (for DSI host and PHY) and PLLSAI (for LTDC) must be configured.

LTDC clock configuration

The LTDC pixel clock frequency is set according to the display requirements. Refer to the step *LTDC pixel clock setting* from *Section : LTDC settings on page 89* for pixel clock setting in video mode.

In this example, the pixel clock frequency is set to 27.4 MHz. *Figure 104* shows the PLLSAI1 settings needed to set the pixel clock to 27.4 MHz.

Figure 104. LTDC pixel clock configuration in video mode using PLLSAI1

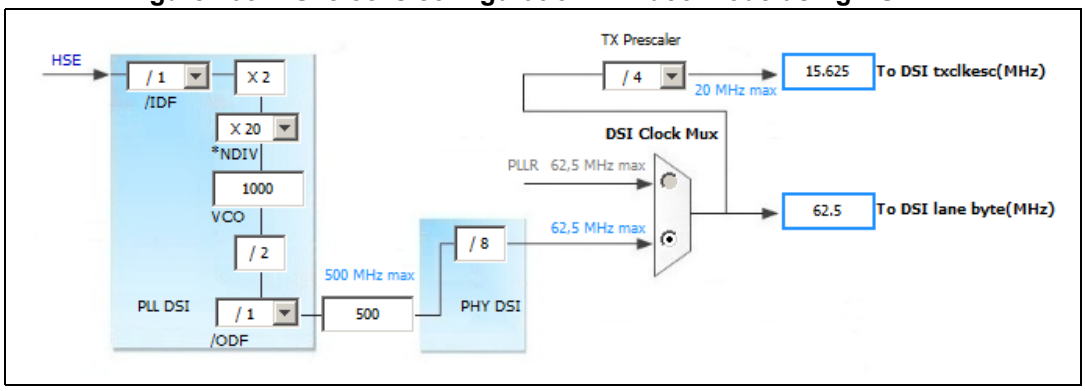


DSI clock configuration

This example shows the DSI PLL configuration to generate the different DSI clocks.

Figure 105 shows an example configuration of the DSI PLL. The link rate in this example is 500 Mbit/s per lane, the lane_byte_clock is set to 62.5 MHz.

Figure 105. DSI clocks configuration in video mode using DSI PLL



The escape mode clock prescaler value must be selected to generate the escape clock used during the TX LP transmission. This clock must not exceed 20 MHz.

Note: The TX prescaler must be set to a value higher than 2. A prescaler value 0 or 1 disables the generation of the TX escape clock.

6.1.3 LTDC and DSI configuration

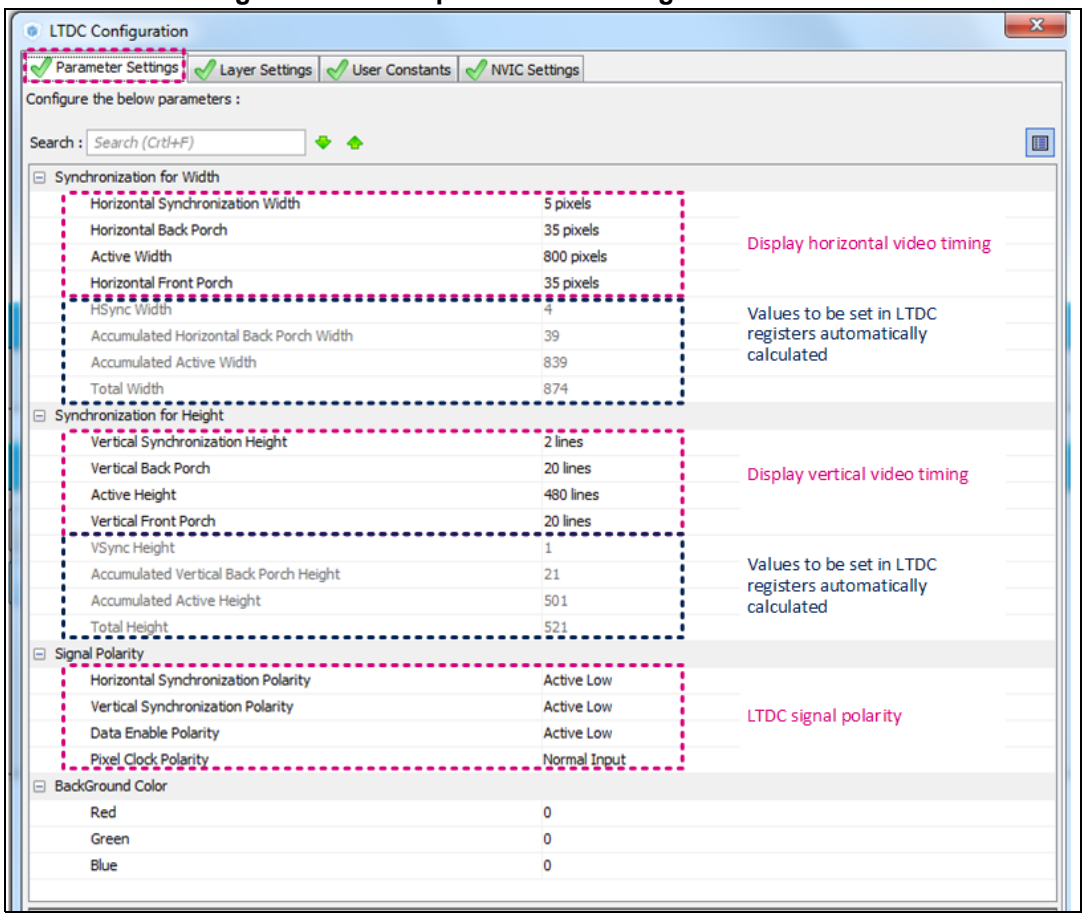
LTDC configuration

This section presents the LTDC settings needed to display an image using the DSI interface.

- LTDC parameter settings

Figure 105 shows the LTDC parameter settings.

Figure 106. LTDC parameters settings in video mode



Vertical and horizontal video timings: the user enters the display timing and STM32CubeMX automatically generates the corresponding values to be programmed inside the LTDC registers.

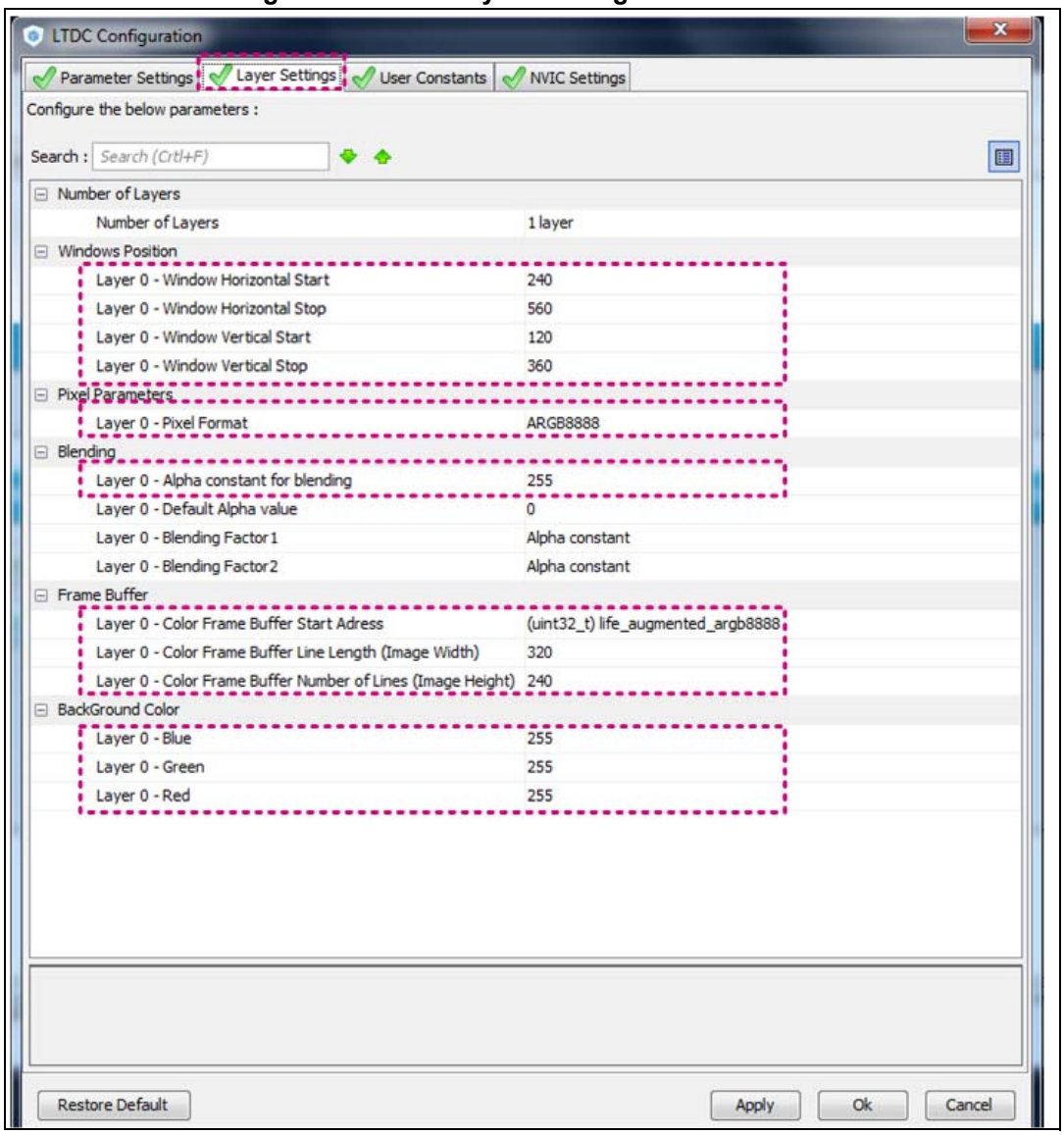
Video signal polarity: these signals are fed to the DSI host. The user has to enter polarity for LTDC and then STM32CubeMX ensures that the corresponding polarity is automatically set inside the DSI.

It is recommended to keep the default LTDC signal polarity (all signals active low).

- LTDC layer settings

Figure 107 shows LTDC layer settings. Only one layer is used in this example.

Figure 107. LTDC layers settings in video mode



Window position definition: this example uses an image with 320 pixels width and 240 lines height, and the rest of the screen is filled with a layer default color. The window position is chosen so that image is displayed in the center of the screen.

Pixel parameters setting: pixel parameters must be set according to the source image color format. ARGB8888 is the color format of the image used in this example.

Blending parameters setting: this example uses only one layer with 255 constant Alpha parameter for blending.

Frame buffer parameters:

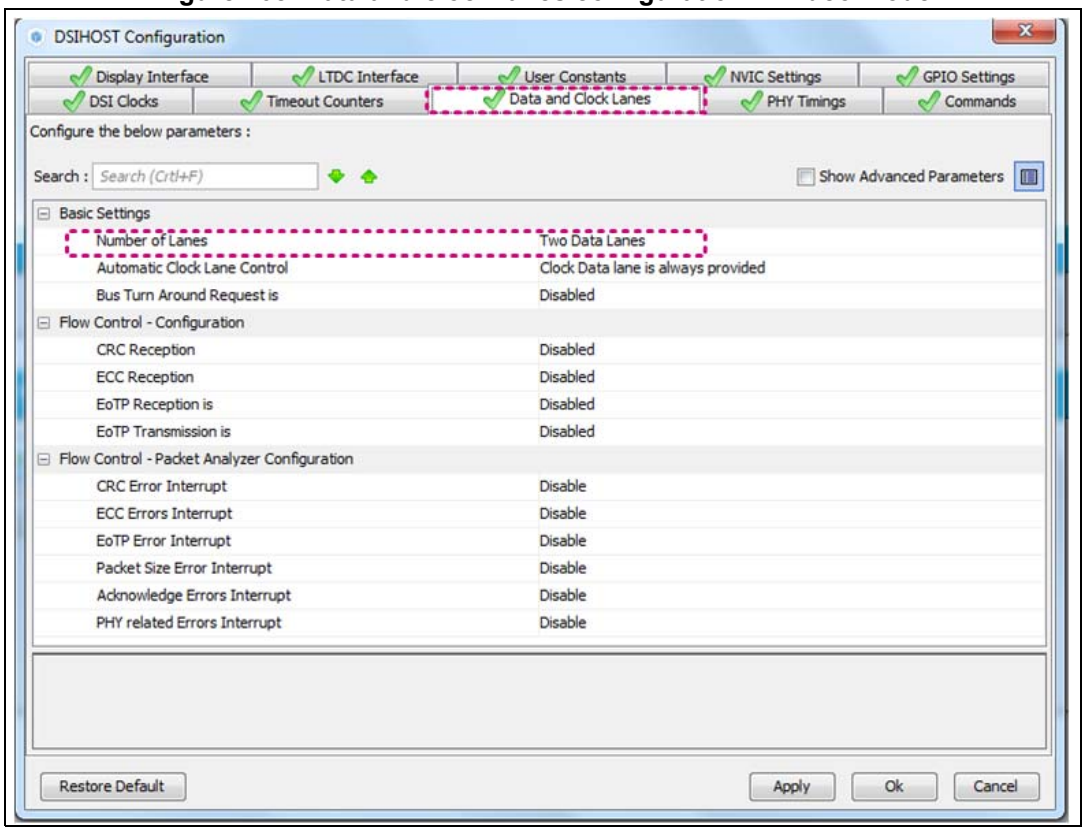
- The start address of the frame buffer is set to the image source start address. This example sets it to a pointer on the source image (life_augmented_argb8888).
- Set the frame buffer length and number of lines with image width and height respectively. A 320 x 240 image is used in this example.
- Layer default color setting: all the areas outside the displayed image are colored with the default color that is white in this example.

DSI host configuration

- Data and clock lanes configuration

Figure 108 shows the DSI data and clock lanes settings.

Figure 108. Data and clock lanes configuration in video mode



Lane number selection: two lanes are used in this example.

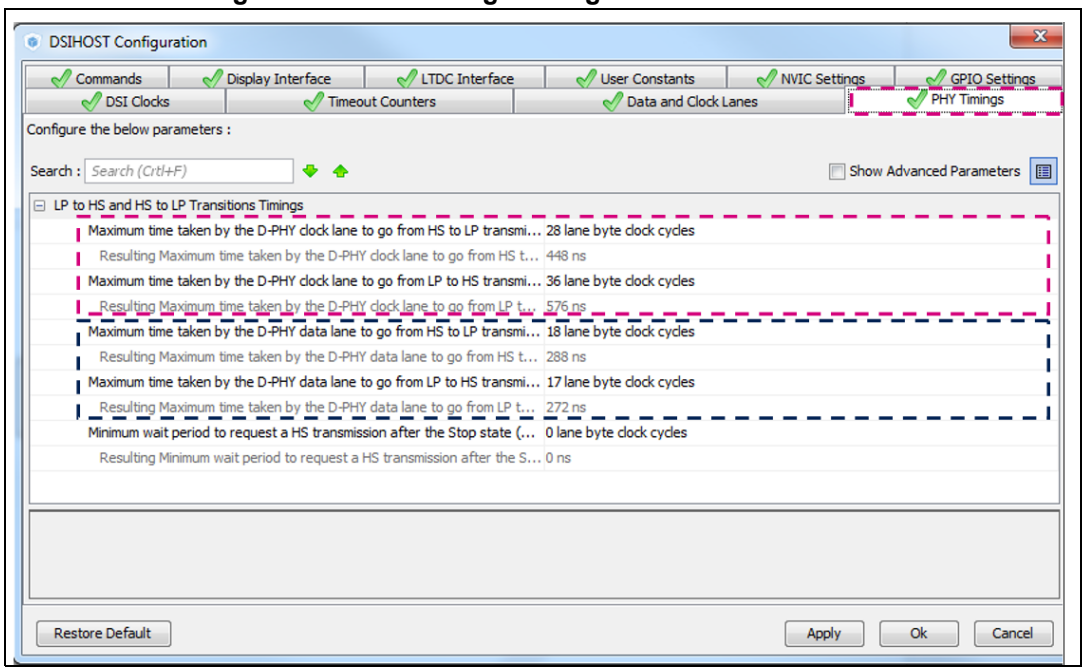
Clock lane control mode: the clock lane is always provided in this example.

BTA request: if reverse direction communication is needed (read request, frame acknowledge request, or other), the BTA request must be enabled. Not required in this example.

- PHY timings configuration

Figure 109 shows the DSI PHY timings configuration.

Figure 109. PHY timings configuration in video mode



– Data lane transition timing:

  Keep default LP to HS and HS to LP transition timings as shown above.

  The data lane transition timings configuration is mandatory.

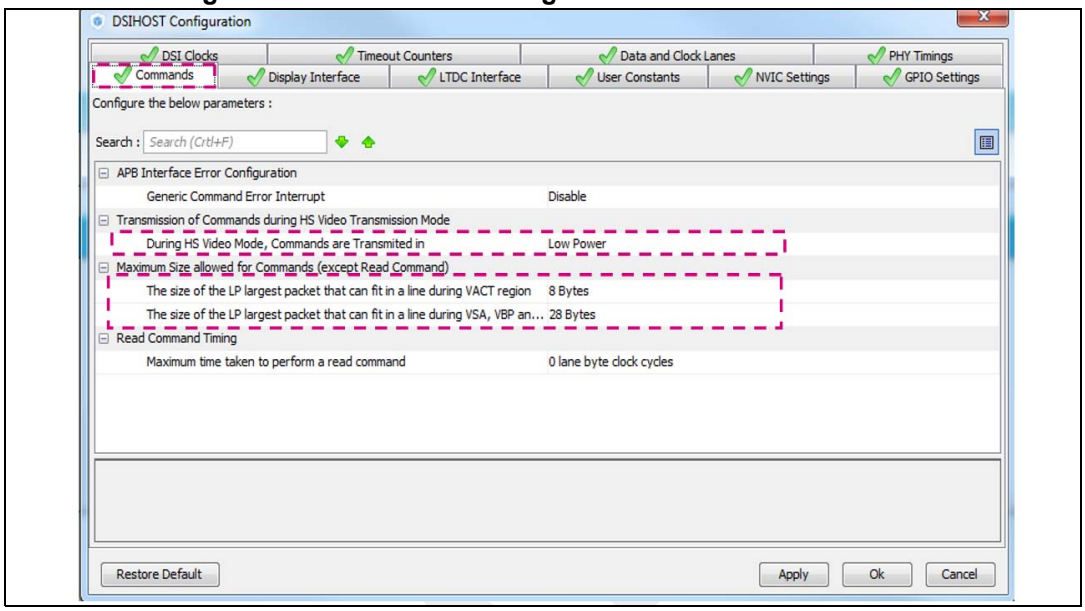
– Clock lane transition timing: these timings are required only if “automatic clock lane control” has been set to “clock lane is not provided when time allows”.

– Minimum wait period to request an HS transmission after the stop state (SW_TIME) is set to 0 in video mode.

- Commands transmission configuration.

Figure 110 shows commands transmission configuration.

Figure 110. Commands configuration in video burst mode



– Command transmission mode: enable transmission of commands in low-power mode. This is mandatory in this example since the display initialization commands must be sent in LP mode.

– Maximum LP command size: this has to be calculated as shown in *Section : DSI video packet parameters*.

  Set LP largest packet size in VACT region = 8 bytes

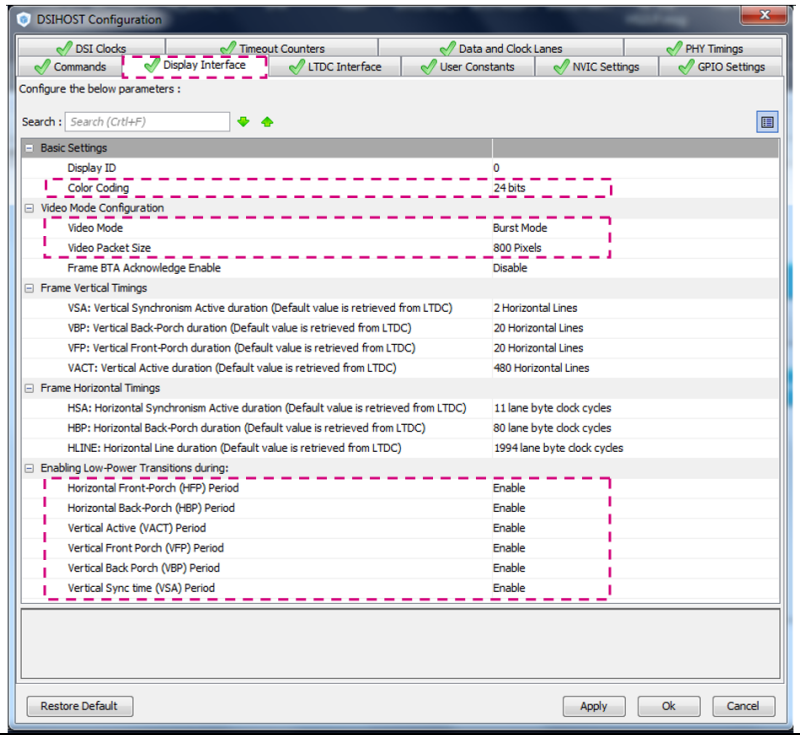
  Set LP largest packet size in vertical blanking regions = 28 bytes. Refer to *Section : Command transmission in video mode* for more details.

– No read commands needed in this example, so keep read command timing 0.

- Display interface configuration

Figure 111 shows DSI display interface configuration:

Figure 111. Display interface configuration in video burst mode



– Color coding selection: this allows choosing the color format of the DSI packets transmitted over the link. This setting is independent from the LTDC layer color format of the image source. The image source can have 565 color format and the DSI host set to 24-bit color format.
In this example, the pixel streams are sent using packed pixel streams in 24-bit format.

– Video mode configuration:

Select video burst mode.

Set video packet size: in burst mode video packet size is set to a full video line length in pixels.

Enable frame BTA acknowledge if needed. This is not used in this example.

– Frame vertical and horizontal timing are retrieved from the LTDC configuration. Values are automatically calculated by STM32CubeMX based on the LTDC video settings, lane byte clock and LTDC pixel clock.

– The LP transitions must be enabled for all regions. The DSI host automatically checks if a transition is possible based on the region's length and the PHY transition. The user can disable the LP transition in specific regions if needed but this is not recommended.

6.1.4 Generated code example for video burst mode

After finishing the configuration phase, STM32CubeMX offers the users the possibility to generate code using different toolchains. Parts of the code generated by STM32CubeMX are shown in this section and the code sections required to be added by the user for display reset and initialization in “USER CODE” sections are highlighted as well.

```
/* main function */
/* Enable I-Cache */
 SCB_EnableICache();
/* Enable D-Cache */
SCB_EnableDCache();
/* MCU configuration */
 /* Reset of all peripherals, initializes the flash interface and the
Systick. */
HAL_Init();
 /* Configure the system clock */
  SystemClock_Config();
  /* Initialize all configured peripherals */
        /*GPIO init for HSE*/
  MX_GPIO_Init();
      /*DSI host initialization, DSI is not started yet at this stage*/
  MX_DSIHOST_DSI_Init();
     /*LTDC initialization. LTDC is enabled at this stage.*/
  MX_LTDC_Init();
  /* USER CODE BEGIN 2 This is the only section to be modified by user*/
/* LCD Display hardware reset through XRES signal*/
BSP_LCD_Reset();
/*start the DSI host. This enables DSI host and Wrapper. This must be done
after the LTDC has been enabled*/
HAL_DSI_Start(&hdsi);
/*LCD Display initialization: This sends display init commands. Arguments
are color format and orientation. LCD display color format must be
consistent with DSI host color format*/
OTM8009A_Init(OTM8009A_FORMAT_RGB888, OTM8009A_ORIENTATION_LANDSCAPE);
/* USER CODE END 2*/

/*Global initialization*/
static void MX_DSIHOST_DSI_Init(void){
  hdsi.Instance = DSI;
  hdsi.Init.AutomaticClockLaneControl = DSI_AUTO_CLK_LANE_CTRL_DISABLE;
  hdsi.Init.TXEscapeCkdiv = 4;
  hdsi.Init.NumberOfLanes = DSI_TWO_DATA_LANES;
/*DSI PLL configuration for 500mbps per lane rate. Lane_byte_clock is at
62,5 Mhz*/
  PLLInit.PLLNDIV = 20;
  PLLInit.PLLIDF = DSI_PLL_IN_DIV1;
  PLLInit.PLLODF = DSI_PLL_OUT_DIV1;
```

```
  if (HAL_DSI_Init(&hdsi, &PLLInit) != HAL_OK)
  {
    Error_Handler();
  }

/*Video mode initialization*/
  VidCfg.VirtualChannelID = 0;
/*DSI host color format configuration. The image data is transmitted using
packed pixel stream with 24 bit format.*/
  VidCfg.ColorCoding = DSI_RGB888;
  VidCfg.LooselyPacked = DSI_LOOSELY_PACKED_DISABLE;
/*Video mode configuration*/
  VidCfg.Mode = DSI_VID_MODE_BURST;
/*Video packet configuration*/
  VidCfg.PacketSize = 800;
  VidCfg.NumberOfChunks = 0;
  VidCfg.NullPacketSize = 0;
/*Signal polarity configuration. Same polarity between LTDC and DSI except
for Data Enable which has opposite polarity */
  VidCfg.HSPolarity = DSI_HSYNC_ACTIVE_LOW;
  VidCfg.VSPolarity = DSI_VSYNC_ACTIVE_LOW;
  VidCfg.DEPolarity = DSI_DATA_ENABLE_ACTIVE_HIGH;
/*Video timing configuration. Retrieved from LTDC config*/
  VidCfg.HorizontalSyncActive = 11;
  VidCfg.HorizontalBackPorch = 80;
  VidCfg.HorizontalLine = 1994;
  VidCfg.VerticalSyncActive = 2;
  VidCfg.VerticalBackPorch = 20;
  VidCfg.VerticalFrontPorch = 20;
  VidCfg.VerticalActive = 480;
/*Command transmission mode and Max LP packet size*/
  VidCfg.LPCommandEnable = DSI_LP_COMMAND_ENABLE;
  VidCfg.LPLargestPacketSize = 29;
  VidCfg.LPVACTLargestPacketSize = 8;
/*LP transition configuration. It is recommended to enable LP transition in
all regions*/
  VidCfg.LPHorizontalFrontPorchEnable = DSI_LP_HFP_ENABLE;
  VidCfg.LPHorizontalBackPorchEnable = DSI_LP_HBP_ENABLE;
  VidCfg.LPVerticalActiveEnable = DSI_LP_VACT_ENABLE;
  VidCfg.LPVerticalFrontPorchEnable = DSI_LP_VFP_ENABLE;
  VidCfg.LPVerticalBackPorchEnable = DSI_LP_VBP_ENABLE;
  VidCfg.LPVerticalSyncActiveEnable = DSI_LP_VSYNC_ENABLE;
/*Flow control configuration*/
  VidCfg.FrameBTAAcknowledgeEnable = DSI_FBTAA_DISABLE;
  if (HAL_DSI_ConfigVideoMode(&hdsi, &VidCfg) != HAL_OK)
  {
```

```
    Error_Handler();
  }

/*LTDC initialization*/
  LTDC_LayerCfgTypeDef pLayerCfg;
  hltdc.Instance = LTDC;
/*LTDC signal polarity*/
  hltdc.Init.HSPolarity = LTDC_HSPOLARITY_AL;
  hltdc.Init.VSPolarity = LTDC_VSPOLARITY_AL;
  hltdc.Init.DEPolarity = LTDC_DEPOLARITY_AL;
  hltdc.Init.PCPolarity = LTDC_PCPOLARITY_IPC;
/*Video timing configuration according to display timing*/
  hltdc.Init.HorizontalSync = 4;
  hltdc.Init.VerticalSync = 1;
  hltdc.Init.AccumulatedHBP = 39;
  hltdc.Init.AccumulatedVBP = 21;
  hltdc.Init.AccumulatedActiveW = 839;
  hltdc.Init.AccumulatedActiveH = 501;
  hltdc.Init.TotalWidth = 874;
  hltdc.Init.TotalHeigh = 521;
  if (HAL_LTDC_Init(&hltdc) != HAL_OK)
  {
    Error_Handler();
  }

/*LTDC Layer initialization*/
/*Window position configuration*/
   pLayerCfg.WindowX0 = 240;
  pLayerCfg.WindowX1 = 560;
  pLayerCfg.WindowY0 = 120;
  pLayerCfg.WindowY1 = 360;
/*Input Pixel color format. This must be consistent with source image color
format*/
  pLayerCfg.PixelFormat = LTDC_PIXEL_FORMAT_ARGB8888;
/*Blending parameters*/
  pLayerCfg.Alpha = 255;
  pLayerCfg.Alpha0 = 0;
  pLayerCfg.BlendingFactor1 = LTDC_BLENDING_FACTOR1_CA;
  pLayerCfg.BlendingFactor2 = LTDC_BLENDING_FACTOR2_CA;
/*Frame Buffer parameters*/
  pLayerCfg.FBStartAdress = (uint32_t) life_augmented_argb8888;
  pLayerCfg.ImageWidth = 320;
  pLayerCfg.ImageHeight = 240;
/*Backgound color configuration. White background in this example*/
  pLayerCfg.Backcolor.Blue = 255;
```

```
pLayerCfg.Backcolor.Green = 255;
pLayerCfg.Backcolor.Red = 255;
if (HAL_LTDC_ConfigLayer(&hltdc, &pLayerCfg, 0) != HAL_OK)
{
  Error_Handler();
}
```

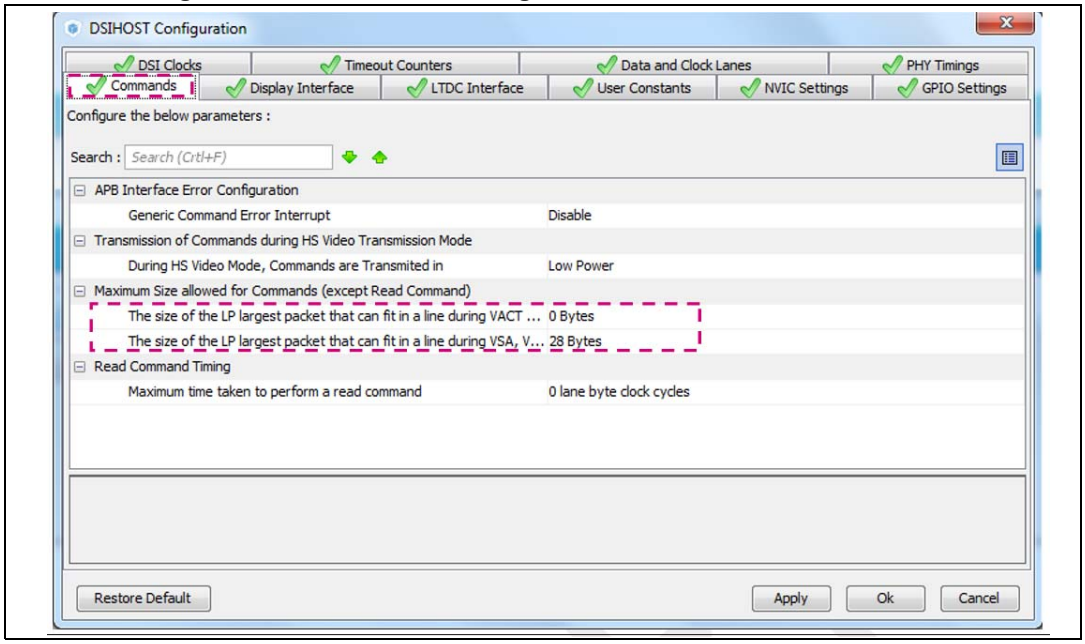
6.2 DSI host nonburst mode with sync pulses

Most of the configuration is similar to the burst mode configuration except for the commands configuration and the display interface configuration.

6.2.1 Commands configuration

Figure 112 shows the commands transmission configuration.

Figure 112. Commands configuration in video nonburst mode



In nonburst mode there is less margin during the VACT region for command transmission compared to the video burst mode.

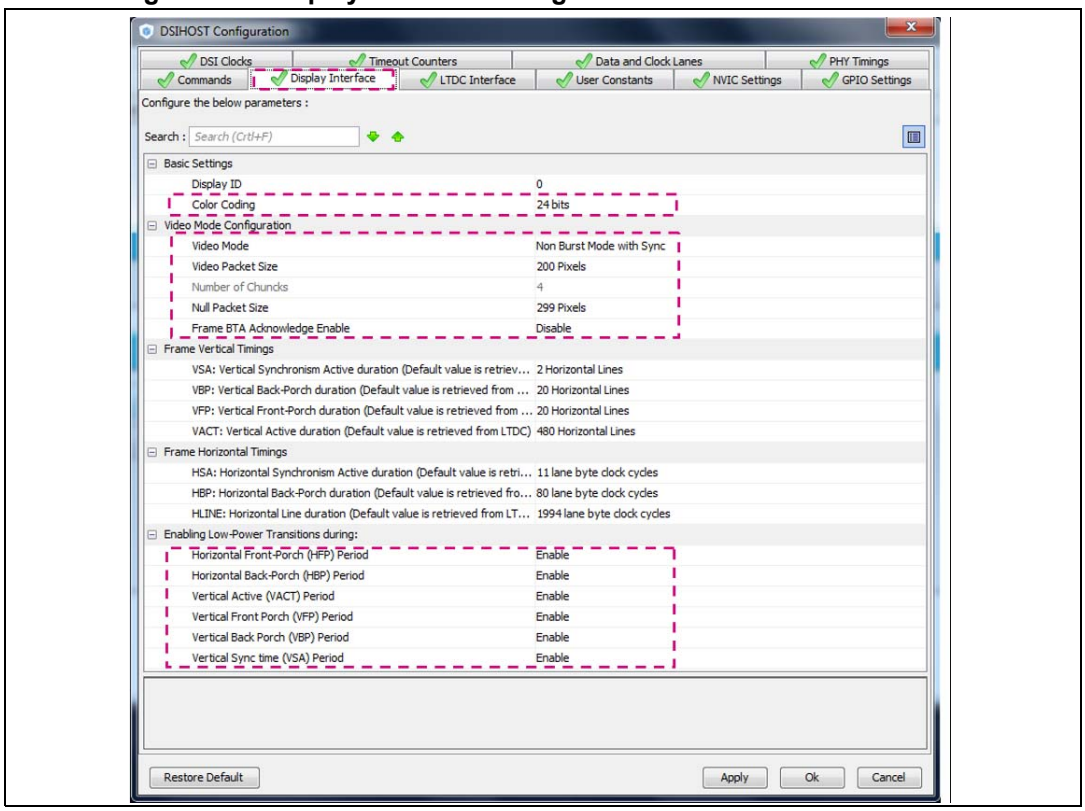
In this example no commands are allowed to be transmitted during this period, so LP largest packet during VACT must be set to 0.

Please refer to*Command transmission in video mode* for more details.

6.2.2 Display interface configuration

Figure 113 shows the DSI display interface configuration.

Figure 113. Display interface configuration in video nonburst mode



Video mode selection: select the nonburst with sync pulse in this example.

Video packet size: this has to be set according to available line buffer size inside the display. In this example the video packet size is set to 200 pixels.

Number of chunks is automatically calculated based on: active width configured in LTDC and packet size (number of chunks = active width / video packet size).

Null packet size: size of the null packet in bytes have to be calculated. 299 bytes are needed in this example. Please refer to *Section : DSI video packet parameters* for more details on calculation.

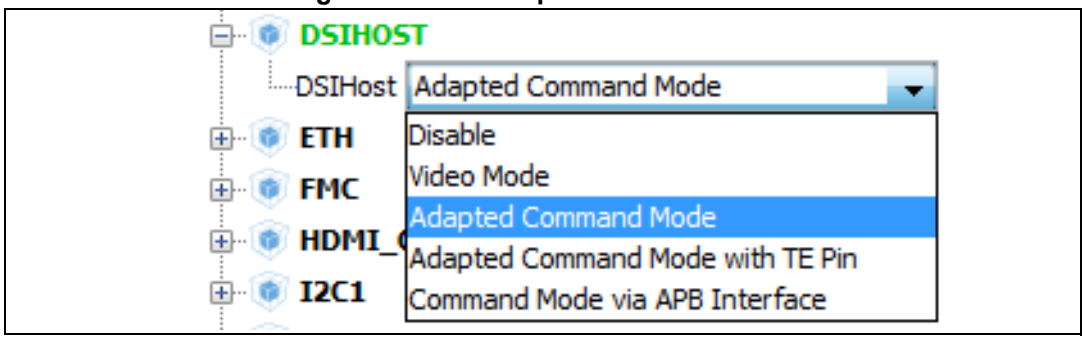
6.3 DSI host adapted command mode

6.3.1 Pin configuration

The RCC and LTDC pinout configurations are the same as in the video burst mode. Only the DSI host configuration must be changed and set to adapted command mode.

Select the DSI host adapted command mode in the pinout configuration section (see *Figure 114*).

Figure 114. DSI adapted mode selection



Note: If tearing effect reporting over the pin is required, "Adapted Command Mode with TE pin" must be selected. This configures the pin to be used for TE.

6.3.2 Clock configuration

The DSI clock configuration is done as in video mode. This example keeps the same lane_byte_clock frequency at 62.5 MHz.

LTDC clock configuration

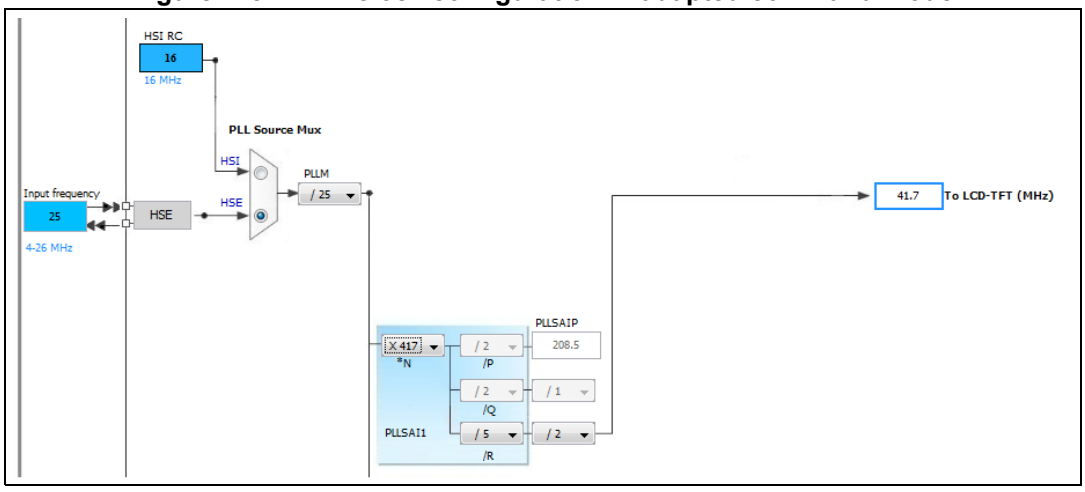
In adapted command mode, the LTDC pixel_clock is set to the maximum supported frequency in order to decrease refresh time.

Pixel clock = lane rate x number of lanes / bit_per_pixel

In 24-bit color mode with two data lanes at 500 Mbit/s each, the maximum supported LTDC pixel clock is 500 mbs x 2 / 24 = 41.7 MHz.

Figure 115 shows how to configure the PLLSAI to set the LTDC pixel clock at 41.7 MHz.

Figure 115. LTDC clock configuration in adapted command mode



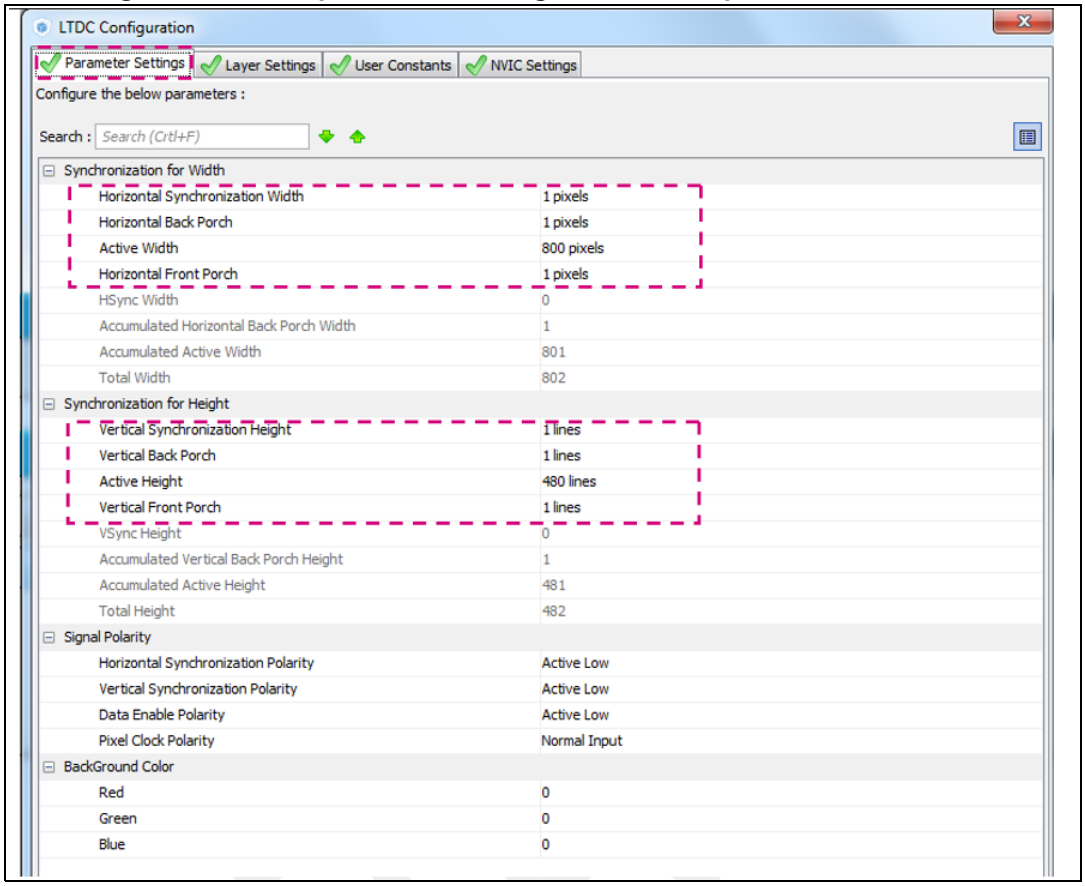
6.3.3 LTDC and DSI configuration

LTDC configuration

For the LTDC layer settings, refer to the video mode example (see *Section 6.1: DSI host video burst mode*).

The configuration of the LTDC parameters is the same as in video mode, except for video blanking timing (see *Figure 116*).

Figure 116. LTDC parameters configuration in adapted command mode



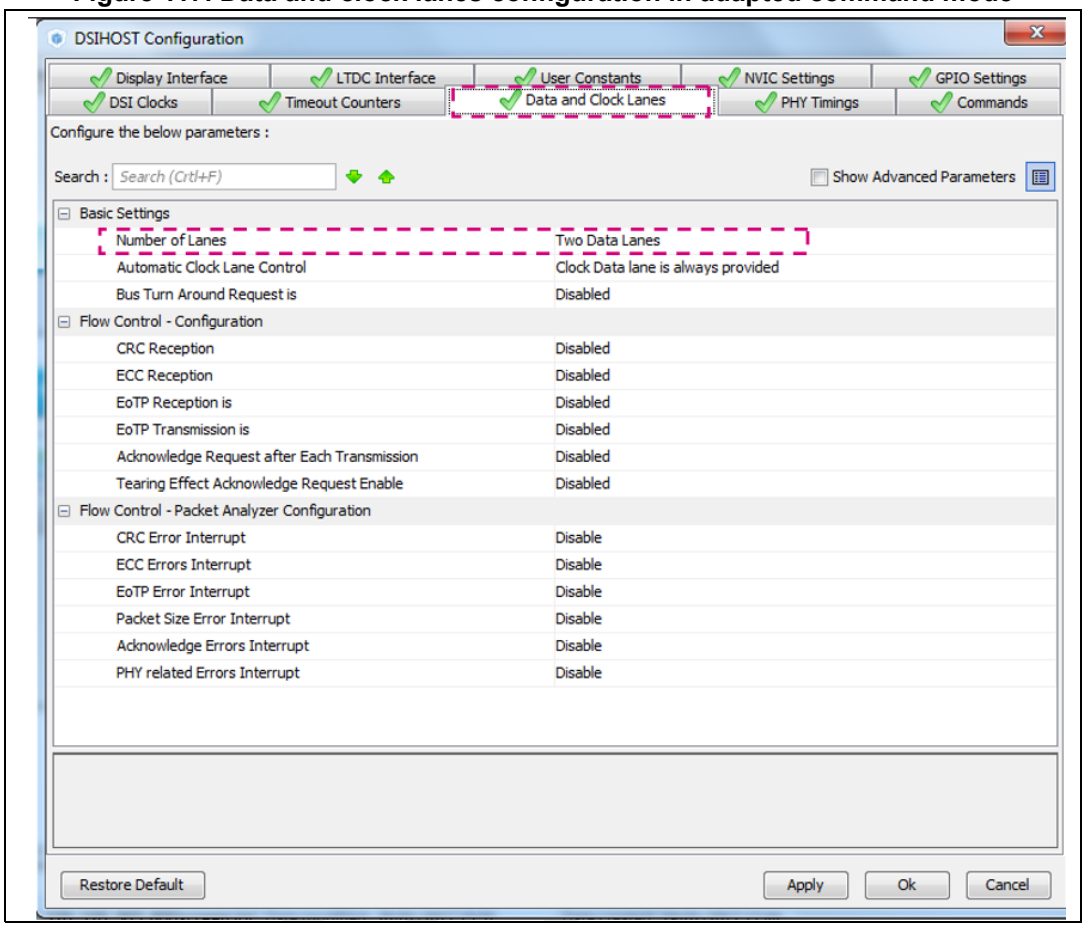
In an adapted command, the horizontal blanking timing can be set to the minimum that is one pixel clock and the vertical blanking timing may be set to the minimum that is one line. This is because the display relies on its internal display controller for video timing generation.

DSI host configuration

- Data and clock lanes configuration

Figure 117 shows the DSI data and clock lanes settings in adapted command mode.

Figure 117. Data and clock lanes configuration in adapted command mode



Number of data lanes selection: two data lanes are used in this example.

If tearing effect reporting over link is required, the user must enable:

– Bus turnaround request

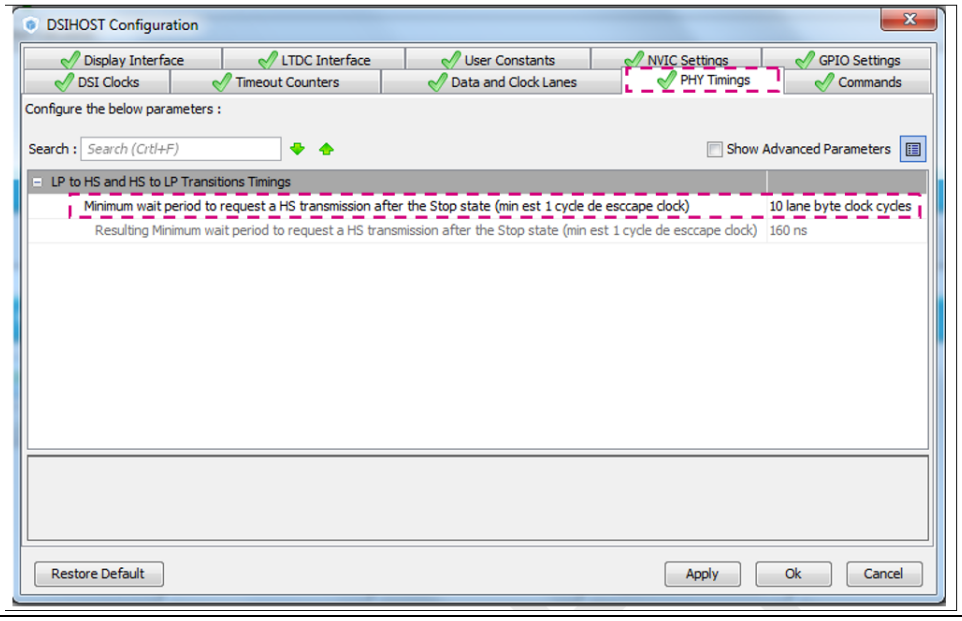
– Tearing effect acknowledge request

This example does not use the tearing effect reporting.

- PHY timings configuration

Figure 118 shows the DSI PHY timings configuration. The only parameter to be configured is the stop wait time.

Figure 118. PHY timing configuration in adapted command mode

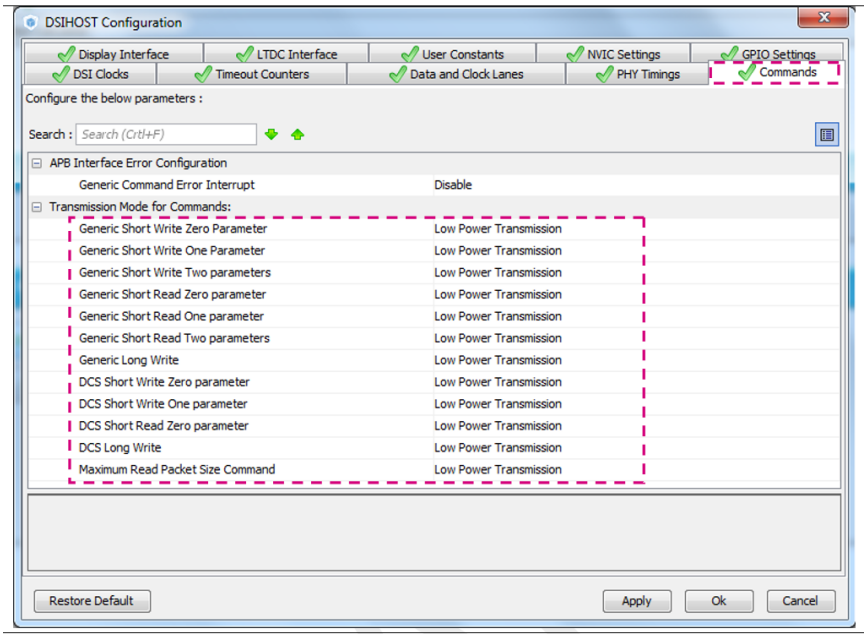


It is mandatory in command mode to set the minimum wait period to request an HS transmission after the stop state, which is the stop wait time (SW_Time). The required SW_Time for DSI host is 10 lane byte clock cycles. If the display requires more stop wait time, the user must program this field with display SW_TIME.

- Commands transmission configuration

Figure 119 shows the commands transmission configuration.

Figure 119. Command transmission configuration in adapted command mode



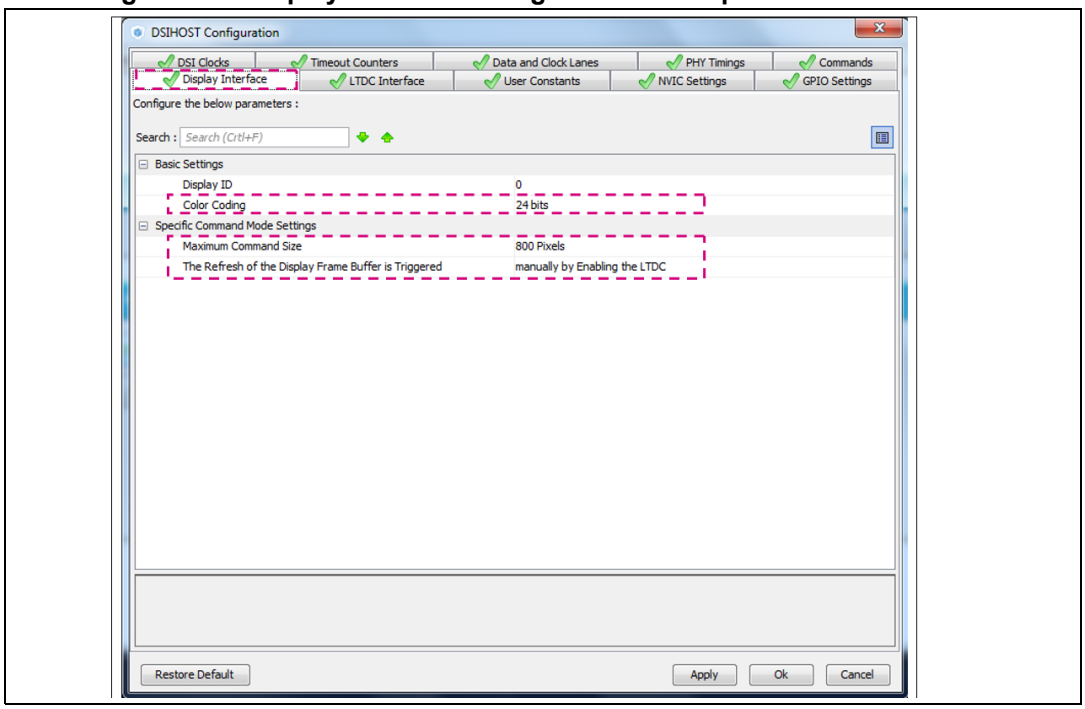
Each command type may be independently set to be sent either in LP or HS mode.

Some displays require the initialization commands to be sent in LP mode. In this case, the user must enable the command transmission in low-power mode. Then, after the display's initialization has been finished, the user has to enable the HS transmission again (especially for DCS long write commands that are used for frame refresh).

- Display interface configuration

Figure 120 shows the DSI display interface configuration.

Figure 120. Display interface configuration in adapted command mode



Color coding selection: 24-bits color mode is used in this example.

In command mode, the host must inform the display with the used color format. This is done by sending DCS short command set_pixel_format. This can be done during display initialization phase.

Maximum command size is set to 800 pixels in this example. Each line of the frame is encapsulated into one WMS or WMC DCS command.

Refresh mode selection: in this example, the refresh is done manually by enabling the LTDC.

6.3.4 Generated code example for adapted command mode

This section shows the code generated by STM32CubeMX for the adapted command mode configuration. Few sections related to display reset and initialization must be added by the user in the “USER CODE” sections.

```
/*main function*/
  /* Enable ICache */
 SCB_EnableICache();

 /* Enable DCACHE */
 SCB_EnableDCache();

 /* MCU configuration */
```

```
  /* Reset of all peripherals, initializes the flash interface and the
Systick. */
  HAL_Init();

  /* Configure the system clock */
  SystemClock_Config();

  /* Initialize all configured peripherals */
        /*GPIO init for HSE*/
  MX_GPIO_Init();
      /*DSI host initialization, DSI is not started yet at this stage*/
  MX_DSIHOST_DSI_Init();
          /*LTDC initialization. LTDC is enabled at this stage.*/
  MX_LTDC_Init();
  /* USER CODE BEGIN 2. This is the only section to be modified by user */
/* LCD Display hardware reset through XRES signal*/
BSP_LCD_Reset();
/*start the DSI host. This will enable DSI host and Wrapper. This must be
done after the LTDC has been enabled*/
  HAL_DSI_Start(&hdsi);
/*LCD Display initialization: This sends display init commands. Arguments
are color format and orientation. Color format must be consistent with DSI
host color format*/
  OTM8009A_Init(OTM8009A_FORMAT_RGB888, OTM8009A_ORIENTATION_LANDSCAPE);

/*Enable HS transmission of commands. This is in case display init has been
done in LP as required by some displays*/
  /* Stop the DSI and reconfigure the DCS Long Write in High Speed only,
since display refresh is done with DCS Long Write commands*/
  HAL_DSI_Stop(&hdsi_eval);
  /* Configure the command mode */
  dsiLPCmdInit.LPGenShortWriteNoP       = DSI_LP_GSW0P_ENABLE;
  dsiLPCmdInit.LPGenShortWriteOneP      = DSI_LP_GSW1P_ENABLE;
  dsiLPCmdInit.LPGenShortWriteTwoP      = DSI_LP_GSW2P_ENABLE;
  dsiLPCmdInit.LPGenShortReadNoP        = DSI_LP_GSR0P_ENABLE;
  dsiLPCmdInit.LPGenShortReadOneP       = DSI_LP_GSR1P_ENABLE;
  dsiLPCmdInit.LPGenShortReadTwoP       = DSI_LP_GSR2P_ENABLE;
  dsiLPCmdInit.LPGenLongWrite           = DSI_LP_GLW_ENABLE;
  dsiLPCmdInit.LPDcsShortWriteNoP       = DSI_LP_DSW0P_ENABLE;
  dsiLPCmdInit.LPDcsShortWriteOneP      = DSI_LP_DSW1P_ENABLE;
  dsiLPCmdInit.LPDcsShortReadNoP        = DSI_LP_DSR0P_ENABLE;
  dsiLPCmdInit.LPDcsLongWrite           = DSI_LP_DLW_DISABLE;
  dsiLPCmdInit.LPMaxReadPacket          = DSI_LP_MRDP_ENABLE;
  dsiLPCmdInit.AcknowledgeRequest       = DSI_ACKNOWLEDGE_DISABLE;
  /* Init the command mode */
  HAL_DSI_ConfigCommand(&hdsi_eval, &dsiLPCmdInit);
```

```
  HAL_DSI_Start(&hdsi_eval);

/* Manual refresh. This enables LTDCEN bit in the DSI Wrapper Control
register */
HAL_DSI_Refresh(&hdsi);
/* USER CODE END 2 */

/*Global initialization*/
  hdsi.Instance = DSI;
  hdsi.Init.AutomaticClockLaneControl = DSI_AUTO_CLK_LANE_CTRL_DISABLE;
  hdsi.Init.TXEscapeCkdiv = 4;
  hdsi.Init.NumberOfLanes = DSI_TWO_DATA_LANES;
/*DSI PLL configuration. This sets link rate at 500 mbps. Lane_byte_clock is
at 62,5 Mhz*/
  PLLInit.PLLNDIV = 20;
  PLLInit.PLLIDF = DSI_PLL_IN_DIV1;
  PLLInit.PLLODF = DSI_PLL_OUT_DIV1;
  if (HAL_DSI_Init(&hdsi, &PLLInit) != HAL_OK)
  {
    Error_Handler();
  }

/*Command transmission configuration*/
/*enable command transmission in LP mode. This is mandatory for some
displays during init phase. After display init LP transmission has to be
disabled for DCS Long Write commands*/
  LPCmd.LPGenShortWriteNoP = DSI_LP_GSW0P_ENABLE;
  LPCmd.LPGenShortWriteOneP = DSI_LP_GSW1P_ENABLE;
  LPCmd.LPGenShortWriteTwoP = DSI_LP_GSW2P_ENABLE;
  LPCmd.LPGenShortReadNoP = DSI_LP_GSR0P_ENABLE;
  LPCmd.LPGenShortReadOneP = DSI_LP_GSR1P_ENABLE;
  LPCmd.LPGenShortReadTwoP = DSI_LP_GSR2P_ENABLE;
  LPCmd.LPGenLongWrite = DSI_LP_GLW_ENABLE;
  LPCmd.LPDcsShortWriteNoP = DSI_LP_DSW0P_ENABLE;
  LPCmd.LPDcsShortWriteOneP = DSI_LP_DSW1P_ENABLE;
  LPCmd.LPDcsShortReadNoP = DSI_LP_DSR0P_ENABLE;
  LPCmd.LPDcsLongWrite = DSI_LP_DLW_ENABLE;
  LPCmd.LPMaxReadPacket = DSI_LP_MRDP_ENABLE;
  LPCmd.AcknowledgeRequest = DSI_ACKNOWLEDGE_DISABLE;
  if (HAL_DSI_ConfigCommand(&hdsi, &LPCmd) != HAL_OK)
  {
    Error_Handler();
  }

/*Adapted command configuration*/
```

```
  CmdCfg.VirtualChannelID = 0;
/*Select DSI host color format*/
  CmdCfg.ColorCoding = DSI_RGB888;
  CmdCfg.CommandSize = 800;
  CmdCfg.TearingEffectSource = DSI_TE_DSILINK;
  CmdCfg.TearingEffectPolarity = DSI_TE_RISING_EDGE;
/* DSI host Signal polarity. Same polarity between LTDC and DSI except for
Data Enable which has opposite polarity*/
  CmdCfg.HSPolarity = DSI_HSYNC_ACTIVE_LOW;
  CmdCfg.VSPolarity = DSI_VSYNC_ACTIVE_LOW;
  CmdCfg.DEPolarity = DSI_DATA_ENABLE_ACTIVE_HIGH;
  CmdCfg.VSYNCPol = DSI_VSYNC_FALLING;
/*Manual refresh is used*/
  CmdCfg.AutomaticRefresh = DSI_AR_DISABLE;
  CmdCfg.TEAcknowledgeRequest = DSI_TE_ACKNOWLEDGE_DISABLE;
  if (HAL_DSI_ConfigAdaptedCommandMode(&hdsi, &CmdCfg) != HAL_OK)
  {
    Error_Handler();
  }

/*LTDC configuration*/
  hltdc.Instance = LTDC;
  hltdc.Init.HSPolarity = LTDC_HSPOLARITY_AL;
  hltdc.Init.VSPolarity = LTDC_VSPOLARITY_AL;
  hltdc.Init.DEPolarity = LTDC_DEPOLARITY_AL;
  hltdc.Init.PCPolarity = LTDC_PCPOLARITY_IPC;
/*Video timing configuration. Vertical and horizontal blanking can be set
to 1 in adapted command mode.*/
  hltdc.Init.HorizontalSync = 0;
  hltdc.Init.VerticalSync = 0;
  hltdc.Init.AccumulatedHBP = 1;
  hltdc.Init.AccumulatedVBP = 1;
  hltdc.Init.AccumulatedActiveW = 801;
  hltdc.Init.AccumulatedActiveH = 481;
  hltdc.Init.TotalWidth = 802;
  hltdc.Init.TotalHeigh = 482;
/*Background color configuration*/
  hltdc.Init.Backcolor.Blue = 0;
  hltdc.Init.Backcolor.Green = 0;
  hltdc.Init.Backcolor.Red = 0;
  if (HAL_LTDC_Init(&hltdc) != HAL_OK)
  {
    Error_Handler();
  }
/*LTDC Layer configuration is same as video burst mode example.*/
```

7 DSI host performance

The DSI host performance is impacted by the physical limit of DSI link bandwidth and by system level constraints.

7.1 DSI link maximum bandwidth impact on LTDC pixel clock

The maximum DSI link speed depends on the maximum lane rate and the number of lanes.

Table 39 shows the maximum DSI link rate per product:

Table 39. Maximum DSI link rate per product

STM32	Number of lanes	Maximum lane rate	Maximum link rate
STM32F469xx and STM32F479xx	2	500 Mbit/s	1 Gbit/s
STM32F76xxx and STM32F77xxx	2	500 Mbit/s	1 Gbit/s
STM324Rxxx and STM32L4Sxxx	2	500 Mbit/s	1 Gbit/s
STM32H747/757	2	1 Gbit/s	2 Gbit/s
STM32MP157	2	1 Gbit/s	2 Gbit/s
STM32U599/5A9 STM32U595/5A5	2	500 Mbit/s	1 Gbit/s

There is a relationship between the equivalent pixel clock and the DSI host configuration.

Depending on the DSI color coding, the number of data lanes used and the speed of the data lane, it is possible to evaluate the equivalent pixel clock as follows:

$$\text{Pixel clock} = \frac{(\text{Lane\_rate} \times \text{number\_of\_lanes})}{\text{Bits\_per\_pixel}}$$

As an example, when using two data lanes at 500 Mbit/s for a total data rate of 1 Gbit/s:

- 16 bits per pixel coding:
  Maximum equivalent pixel clock is 1 Gb/s / 16 bpp = 62,5 MHz
- 24 bits per pixel coding:
  Maximum equivalent pixel clock is 1 Gb/s / 24 bpp = 41,66 MHz

For products supporting 1Gbit/s per lane rate for a total data rate of 2Gbit/s:

- 16 bits per pixel coding:
  Maximum equivalent pixel clock is 2 Gb/s / 16 bpp = 125 MHz
- 24 bits per pixel coding:
  Maximum equivalent pixel clock is 2 Gb/s / 24 bpp = 83.33 MHz

Table 40 shows the maximum pixel clock frequency depending on the color coding and the DSI link speed.

Table 40. Maximum pixel clock frequency depending on color coding and DSI link speed

	500 Mbit/s	1 Gbit/s	2 Gbit/s
24 bpp	20.83 MHz	41.66 MHz	83.33 MHz
16 bpp	31.25 MHz	62.5 MHz	125 MHz

7.2 System constraints impact on LTDC pixel clock

The maximum equivalent pixel clock can be impacted by system constraints. Depending on the application use case, system constraints may require to reduce the LTDC pixel clock.

Refer to AN4861 “LCD-TFT display controller (LTDC) in STM32 microcontrollers” for further details about maximum pixel clock supported in function of system constraints.

7.3 DSI link bandwidth estimation

This section aims to evaluate the required DSI link bandwidth in video and adapted command modes.

7.3.1 Video mode

In video mode the LTDC pixel clock and the minimum link BW (bandwidth) are imposed by the display timing.

The pixel clock can be calculated using following formula:

pixel clock = total width x total height x refresh rate

The minimum DSI link BW required to drive the display is calculated using this formula:

Min DSI link BW = pixel clock x color depth

Consider an example display with the following timings:

- HSA = 5, HBP = 35, HACT = 800, HFP = 35
- VSA = 2, VBP = 20, VACT = 480, VFP = 20
- Refresh rate =60 fps
- Color depth= 24 bpp
  - Pixel clock = 875 x 522 x 60 = 27.4 MHz
  - Minimum link BW = 657 Mbit/s

This is the minimum link BW required to drive the display.

For products that maximum lane rate equals to 500 Mbit/s, this display cannot be driven using only one lane.

This display may be driven by two lanes at a minimum lane rate of 328 Mbit/s each.

For products which maximum lane rate is 1Gbit/s, this display can be driven by one lane. It is thus necessary to check the maximum allowable lane rate on the display side.

Note: The link BW shown above gives just an estimate of the minimum required BW and number of lanes to be used. The video mode type (burst versus nonburst) and protocol overhead have an impact on the link BW to be chosen.

In burst mode the DSI link BW can be increased. This ensures that the DSI host sends the pixel data fast enough to go to LP mode for long periods.

In nonburst mode, the DSI link BW calculation must be fine-tuned to consider the protocol overhead as shown in equation 1 and 2 of section*Section : DSI video packet parameters on page 93*.

Note that in both cases, even if the DSI BW is increased, the LTDC pixel clock must keep the same value as the one calculated from the display video timing.

7.3.2 Adapted command mode

In adapted command mode, the maximum GRAM refresh time is imposed by the internal refresh rate of the display.

The maximum allowed refresh time must be less than 1/(display refresh rate) in order to avoid visual artifact and tearing.

The maximum allowed refresh time is given by the following formula:

Maximum refresh time = 1 / display refresh rate

For example, if the display's internal refresh rate is 60 Hz, the maximum allowed refresh time is 1/60 Hz = 16.6 ms.

The DSI link BW can be calculated using following formula:

DSI link BW = FB size/ refres time = HACT x VACT x color depth / refresh time

The minimum link BW is calculated using the maximum refresh time. It is the minimum BW that allows to avoid visual artifact on display side.

Min DSI link BW = HACT x VACT x color depth / maximum refresh time

For example for a 320 x 320 display with 16 bpp color depth and with 60 Hz display internal refresh rate:

FB size (KB) = 320 x 320 x 2/1024 = 200 Kbyte

Minimum link bandwidth = 320 x 320 x 16 / 0.016 = 102 Mbit/s

This bandwidth can be supported using only one lane at 102 Mbit/s.

The LTDC pixel clock can be calculated using below formula:

LTDC pixel clock = DSI link BW / color depth

In this example, the LTDC pixel clock = 102 / 16 = 6.3 MHz

If the system is using a single buffer to store the frame buffer, it is important to minimize the GRAM refresh time in order to provide sufficient time for graphic computation. This can be done by increasing the DSI link speed.

The GRAM refresh time is calculated as follows:

Refresh time = HACT x VACT x color depth / DSI link BW

If the link speed is increased to 500 Mbit/s, the required time to refresh the display GRAM is:

Refresh time = 320 x 320 x 16 / 500 ~ 3 ms.

This leaves 16.6ms - 3ms = 13.6ms for graphic computation.

Note: While increasing the DSI link speed, the pixel clock has also to be increased.

The LTDC pixel clock in this example is increased to 500 / 16 = 31.25 MHz, but user has to pay attention to the system constraints to avoid FIFO under-run issues on LTDC side.

The maximum link speed to guarantee the minimum refresh time has to be evaluated according to:

- Maximum DSI link physical limit (refer to *Section 7.1: DSI link maximum bandwidth impact on LTDC pixel clock*).
- System constraints impact on LTDC pixel clock (refer to *Section : There is a relationship between the equivalent pixel clock and the DSI host configuration.*).

8 DSI host application examples

This section shows some use case examples for display driving using the DSI host. The DSI host operating mode is chosen according to the display characteristics. The link bandwidth and frame buffer size requirements have been presented *Section 7: DSI host performance* .

8.1 Small display driving example

In this example a small display that has 320 x 320 pixels resolution with 16bpp color depth is driven.

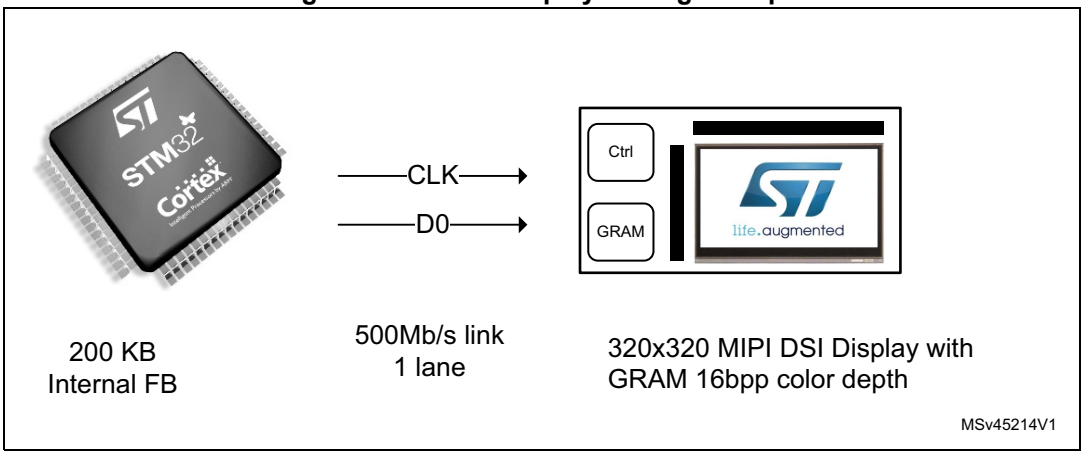
The display embeds an internal GRAM to store the frame buffer. So, the command mode can be used to drive the display.

The required frame buffer size is 200 Kbyte, which can fit into the internal SRAM.

In order to use a single buffer on MCU (or MPU) side, the lane rate is set to 500 Mbit/s to speed the refresh operation and allow more time for graphic computation.

Figure 121 shows an example configuration for small display driving in command mode.

Figure 121. Small display driving example



8.2 Large display driving example

In this example, an 800 x 480 display with 24 bpp color depth is driven.

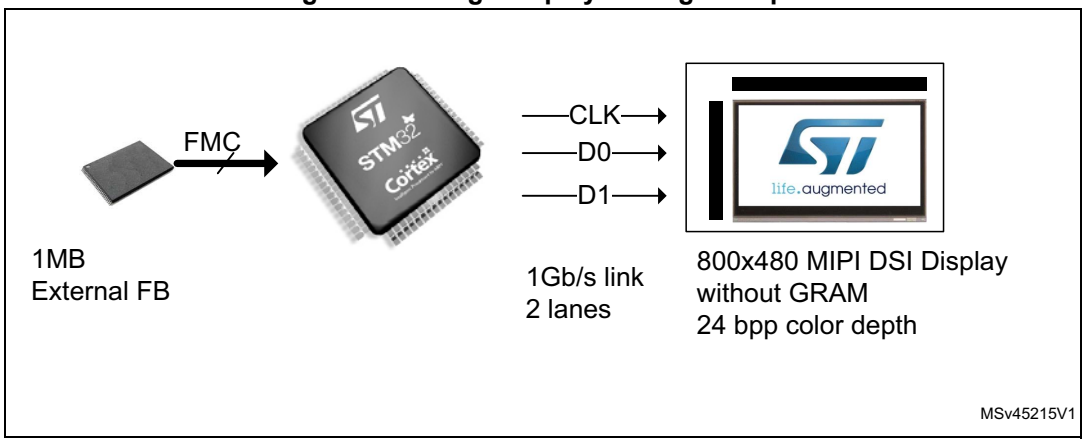
The display does not embed an internal GRAM, so the video mode must be used.

The required frame buffer size is about 1 Mbyte, so an external storage device must be used. An external SRAM or SDRAM may be driven by the FMC to store the frame buffer.

Two lanes must be used to ensure the required refresh rate on the display side.

Figure 122 shows an example configuration for large display driving in video mode.

Figure 122. Large display driving example



9 Supported devices

The STM32 DSI host supports all the operating modes defined in the MIPI DSI specification.

It supports command mode and all the video mode variants (burst, nonburst with sync pulses and nonburst with sync events).

It supports up to 1 Gbit/s or 2 Gbit/s link speed depending on the product.

It can be interfaced with any DSI-compliant display.

10 Conclusion

The new STM32 MCUs and MPUs with DSI host represent the most effective way to connect to a display thanks to the low pin-count and low-power features of the DSI protocol.

This application note presented the STM32 DSI host capabilities and features that allow to interface with a wide range of displays.

The STM32 DSI host scalable architecture and various operating modes that offer flexibility to the customer had been presented.

The DSI host adapted command mode is the most appropriate way to interface with a display thanks to its one-shot refresh capability and its low-power consumption, but it comes at a higher cost on the display side that requires a full frame buffer GRAM.

For displays that do not embed a controller and a GRAM, the best option is the video nonburst mode since it is the most power efficient among video modes. But it requires to store a video line on the display side.

For low cost displays that do not include GRAM and line buffer, the STM32 DSI host supports the nonburst mode to interface with this kind of displays.

A step by step configuration using STM32CubeMX has been presented and sample codes have been provided to allow users to start quickly an application’s development.

11 Revision history

Table 41. Document revision history

Date	Revision	Changes
10-Feb-2017	1	Initial release.
17-Oct-2017	2	Updated: – Document's title – Cover page*: Related documents* – *Section : Video control signal polarity on page 82* – *Example of configuration with four chunks with null packets enabled: on page 95* – *Section : LTDC configuration on page 113* – *Figure 121: Small display driving example* Added: – *Section : LTDC halt polarity* Deleted: – *Table 1: Applicable products*
19-Dec-2022	3	Updated: – Document's title – Section*: Related documents* – *Figure 77: DSI clock scheme* – *Table 3: STM32 microcontrollers featuring DSI host* – *Table 39: Maximum DSI link rate per product* Added: – *Table 1: Applicable products* – *DSI link maximum bandwidth impact on LTDC pixel clock*
01-Mar-2023	4	Replaced RM0476 by RM0456 in *Section : Related documents* Updated *Table 8: DCS command list*