AN12940

Use Case of RT1170 LCD Display System based on MIPI DSI

Rev. 0 — 08/2020

Application Note

Contents

1 Introduction

This application note describes how to use the MIPI DSI Host Controller and LCDIFv2 Controller to drive a DSI-compliant LCD panel on i.MX RT1170.

The Mobile Industry Processor Interface (MIPI) Display Serial Interface (DSI) controller is a flexible, high-performance digital core that provides a serial interface that allows communication with MIPI DSI-compliant peripherals.

Liquid Crystal Display Interface version 2 (LCDIFv2) is a display controller on i.MX RT1170. This block is a system master that fetches graphics stored in memory and display them on a TFT LCD panel.

The demo code used as an example in this document is the `sd_jpeg` project in the released SDK. The hardware environment is MIMXRT1170-EVK board.

2 MIPI DSI host controller

The MIPI DSI is a versatile, high-speed interface for LCD displays in smartphones, automotive and other platforms.

MIPI DSI controller of i.MX RT1170 implements all protocol functions defined in MIPI DSI specification and provides an interface that allows communication between MCUs and MIPI DSI-compliant LCDs.

MIPI DSI D-PHY of i.MX RT1170 is a high-frequency and low-power physical layer supporting the MIPI Alliance standard for D-PHY and provides physical implement for DSI.

2.1 DSI host controller core

Figure 1 shows the DSI layer definitions of DSI specification. The MIPI DSI controller of the i.MX RT1170 implements all three DSI layers (Lane Management, Low Level Protocol, Application).

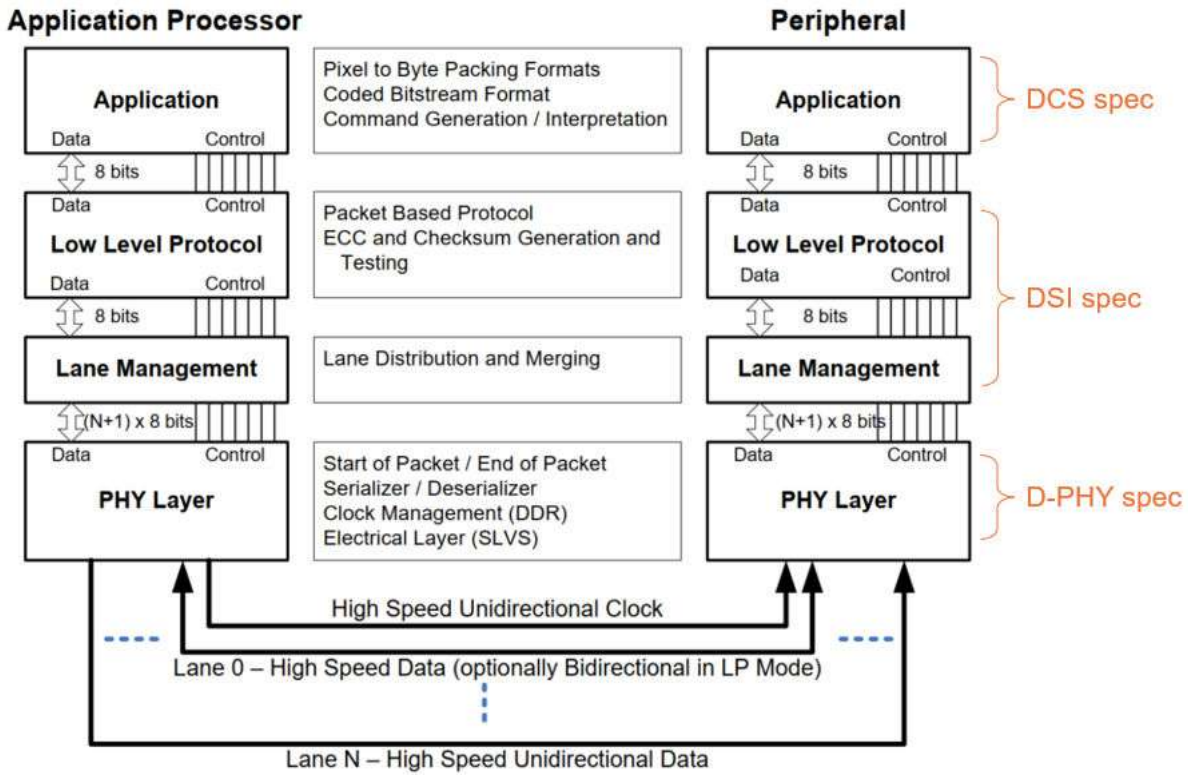

Figure 1. DSI layers

MIPI DSI uses differential signals to transmit clock and data between DSI Host and display module, it includes one clock lane and 1-4 data lanes. For DSI controller in i.MX RT1170, it can support one clock lane and up to two data lanes. Compared with the parallel interface, DSI greatly reduces the number of data and signal line, which saves hardware resources.

DSI-compliant LCD support either of two basic modes of operation: command mode and video mode. Which mode is used depends on the architecture and capabilities of the LCD.

Command mode refers to operation in which transactions primarily take the form of sending commands and data to a display module. The display module may include local registers and a compressed or an uncompressed frame buffer.

Video mode refers to operation in which transfers from the host processor to the peripheral take the form of a real-time pixel stream. Video information should only be transmitted using High-Speed mode. Figure 2 shows examples for command and video modes.

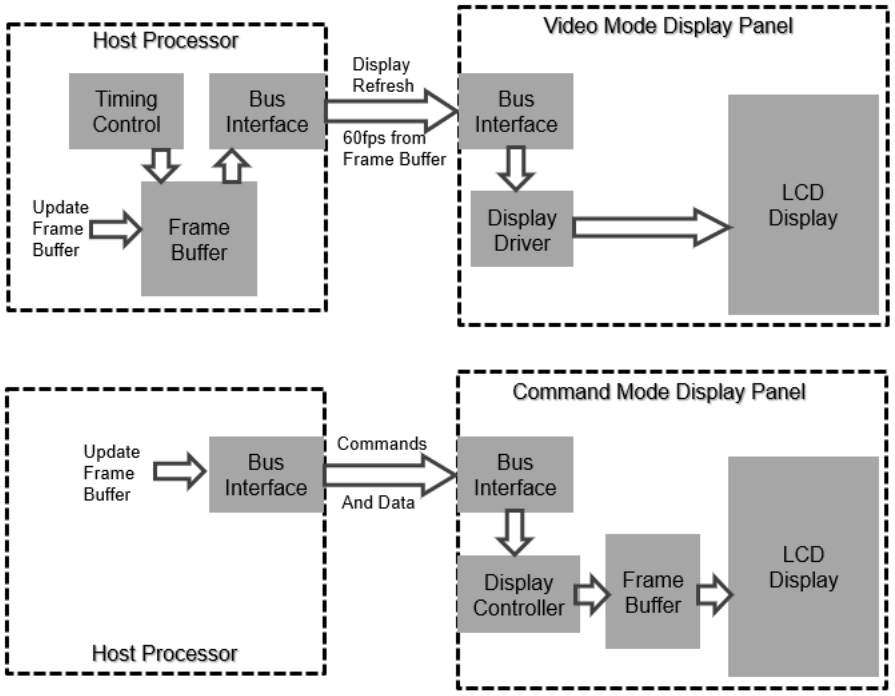

Figure 2. Examples for command and video modes

In general, MIPI DSI controller provides the following key features:

- Supports for command and video modes.
- Scalable data lane support, 1 to 2 Data Lanes.
- Supports high-speed and low-power operation.
- Flexible packet based user interface:
  - APB interface option (status and control).
  - Display Pixel Interface Core (DPI-2) option.

Figure 3 shows the DSI host controller core structure.

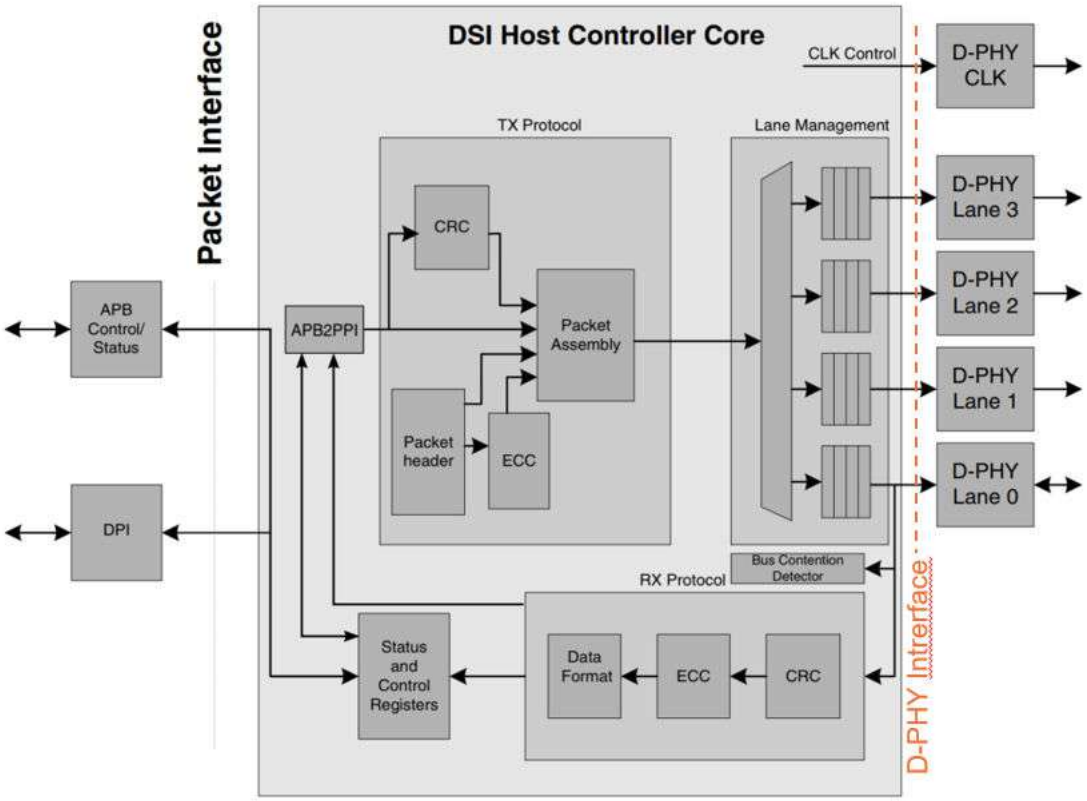

Figure 3. DSI host controller core block diagram

D-PHY interface connects directly to MIPI PPI compliant D-PHYs, which will be introduced in details in DSI D-PHY.

DSI host controller core sends and receives DSI commands and data via packet interface. There are two packet-based interfaces: APB interface and Display Pixel (DPI-2) interface.

2.2 APB interface

The APB Interface is used for the transmission of DCS and generic command mode packets. These packets are built using the APB register access. User application can have access to DSI Host Controller Core Packet Interface through `DSI_HOST_APB_PKT_IF` registers.

2.3 DPI-2 interface host bridge core

DPI-2 interface host bridge core provides a DPI-2 compliant interface to the DSI host controller core and handles converting the DPI-2 interface into a packet format that is compatible with the DSI host controller core's packet interface.

Figure 4 shows a block diagram of the DPI-2 interface host bridge core. The DPI-2 Core accepts a MIPI compliant DPI-2 set of signals (VSYNC, HSYNC, pixel data), creates DSI packets for the video timing events and video data and transmits those packets via the DSI host controller's packet interface.

DPI-2 interface host bridge core provides the following features:

- Support for 16-, 18- and 24- bit Pixel data and all alignment configuration.
- Support for 16, 18, 24, and 18 bit loosely packed DSI data types.
- Comes already integrated with the DSI host controller.

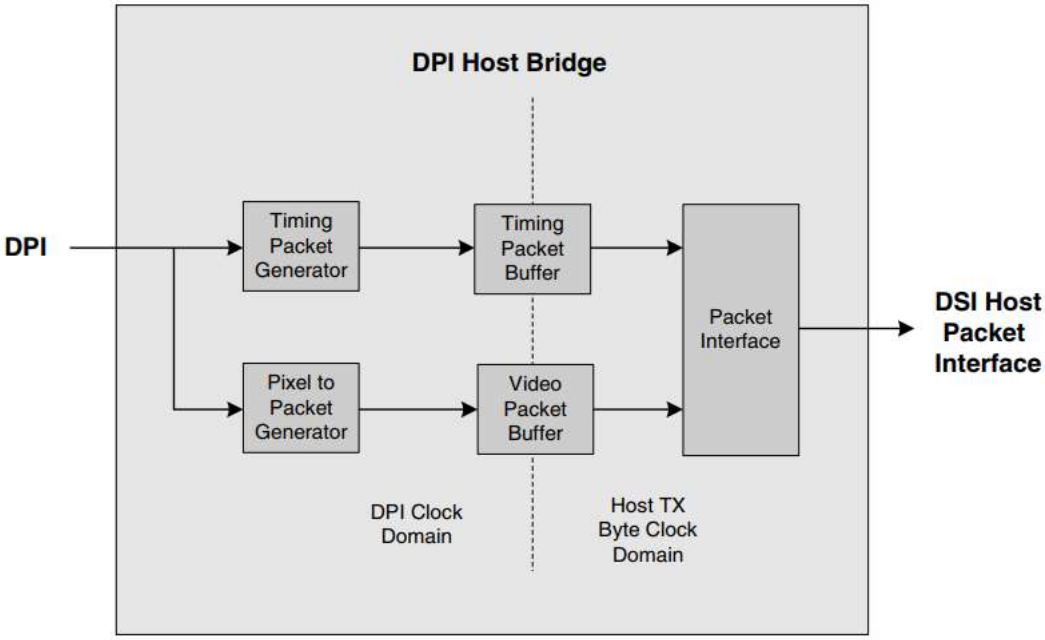

Figure 4. DPI-2 host bridge core block

DPI-2 interface is also called RGB interface. It uses VSYNC, HSYNC, DOTCLK and ENABLE signals to transmit data to LCD panel.

2.4 DSI D-PHY

Figure 5 shows the block diagram of the MIPI DSI D-PHY. It consists of one clock lane and up to two data lanes. Each lane contains two wires.

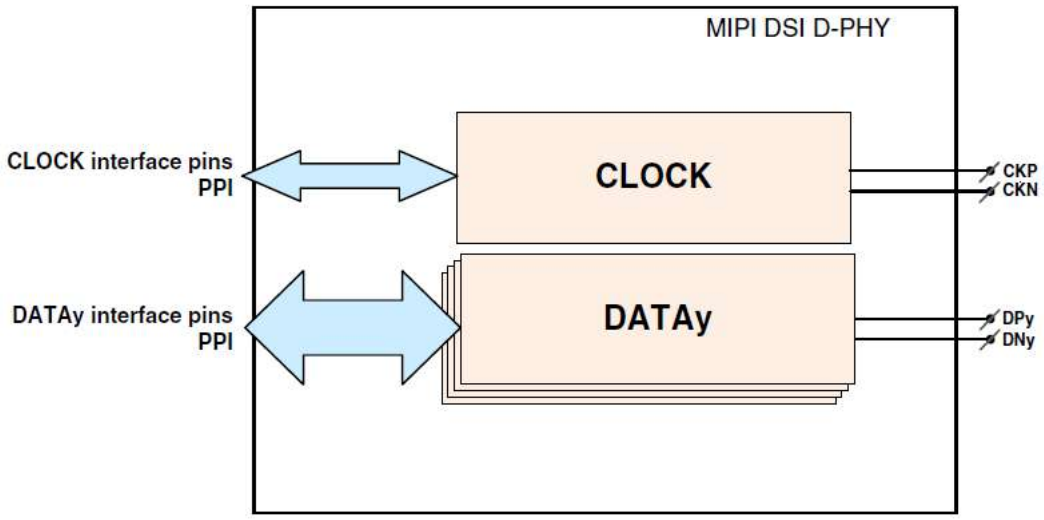

Figure 5. MIPI D-PHY block diagram

D-PHY supports two transmission modes: high-speed mode and low-power mode.

High-speed mode is used for fast-data traffic. High speed data communication appears in bursts with an arbitrary number of payload data bytes. In high-speed mode each Lane is terminated on both sides and driven by a low-swing, differential signal. The range of data rate in high-speed mode is 80 Mbps to 1.5 Gbps per lane.

Low-power mode is used for control purposes. Optionally, Low-power escape mode can be used for low speed asynchronous data communication. In low-power mode all wires are operated single-ended and non-terminated.

Table 1 details the interface signals for MIPI DSI D-PHY.

Table 1. DSI D-PHY signals

Signal	Type	Comments
TxByteClkHS	Clock	High-speed mode transmit byte clock. It is recommended that all transmitting data lane modules share on `TxByteClkHS` signal.
TxClkEsc	Clock	Low-power escape mode transmit clock. The period of this clock determines the symbol time for low-power mode signals. The frequency range of `TxClkEsc` is 12 to 20 MHz.
RxClkEsc	Clock	Low-power escape mode clock for RX. The maximum frequency of `RxClkEsc` should be 60 MHz.

Figure 6 shows the MIPI DSI D-PHY PLL block diagram. MIPI DSI D-PHY PLL is a high performance PLL based frequency synthesizer that incorporates a lock detector, independent output divider, and supports power down modes. It is used for generating high-speed mode transmit byte clock. The D-PHY PLL input clock is `ref_clk` and it ranges from 24 to 200 MHz. The input divider has to be programmed such that the frequency after the input divider ranges from 24 till 30 MHz. The VCO maximum output frequency is 1.5 GHz. The PLL output clock multiplies the `ref_clk` by *CM/(CN × CO)*. High-Speed mode transmit byte clock is equal to *PLL output clock/8*.

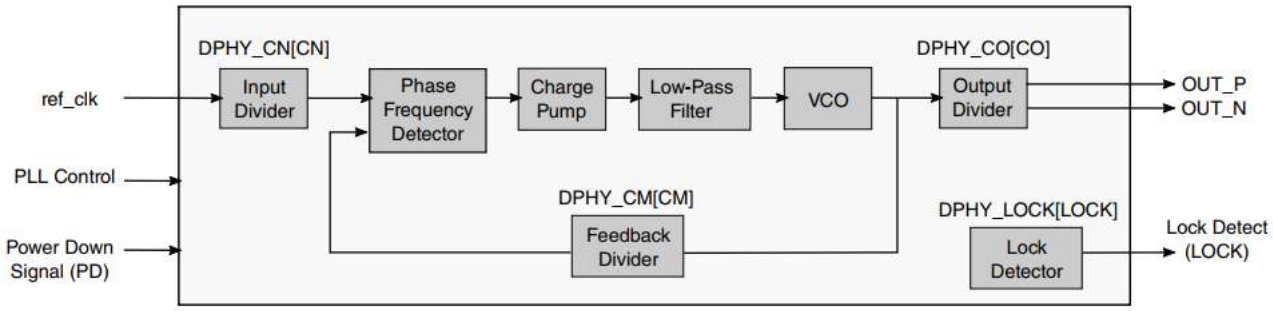

Figure 6. MIPI DSI D-PHY PLL block

2.5 DSI host clock

The DSI host controller requires the following clocks:

- Reference clock (`ref_clk`) – D-PHY PLL input clock.
- Byte clock (`clk_byte`) – generated by the DPHY and is 1/8th the data rate of the DPHY Data Lane High Speed data rate.
- Low-power Escape mode clock (`clk_tx_esc`, `clk_rx_esc`) - for internal LPDT state machines and low-power transmissions.
- Pixel clock (`dpi_pclk`) - used for DPI-2 host interface.

Figure 7 shows the brief clock block for DSI host controller.

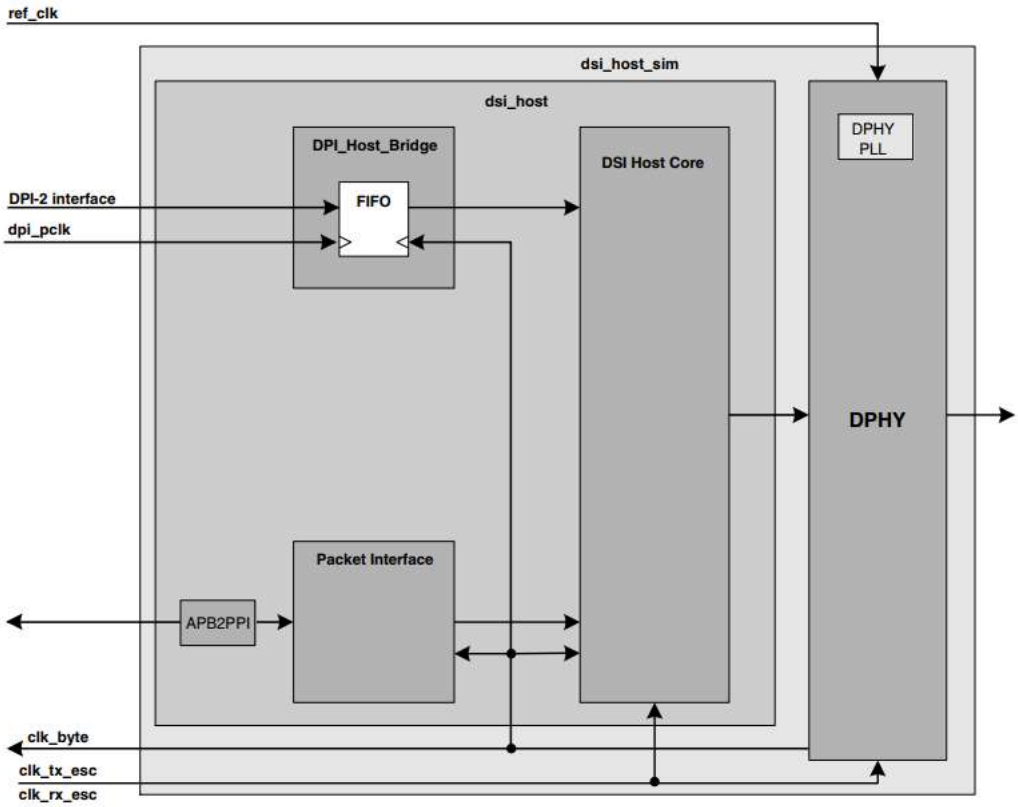

Figure 7. DSI host controller clock

2.6 LCD display system

This section introduces a basic embedded LCD display system which uses video mode. This system consists of an MCU, a framebuffer, a display controller, a video Mux controller, an MIPI DSI host controller and an LCD glass.

- The MCU computes the image/GUI to be displayed in the framebuffer. The more often the framebuffer is updated based on the computing capability, the more fluent the display is.
- The framebuffer is a memory space used to store pixel data of the image/GUI. This memory space is usually called as the framebuffer. The required size of the framebuffer depends on the LCD resolution and color depth of the display. Use double framebuffers to avoid breaking the current displaying data.
- The display controller continuously refreshes the LCD panel and transfers the framebuffer content to the display glass several times per second (for example, 60 Hz). The display controller can be embedded either in the display module or in the MCU. The i.MX RT1170 includes two display controllers named eLCDIF and LCDIFv2.
- The video Mux controller decides using which display controller to control which interface.
- The MIPI DSI host controller provides a high-speed serial interface that allows communication between MCUs and MIPI DSI-compliant LCDs.
- The display glass is driven by the display controller and displays the image/GUI. A display glass is characterized by:
  - Display size

    The display size is defined by the number of pixels of the display with horizontal (pixels number) multiplying vertical (lines number).
  - Color depth

The color depth defines the number of colors in which a pixel can be drawn. It is represented in bits per pixel (bpp). For a color depth of 24 bpp (which can also be represented by RGB888), a pixel can be represented in 16777216 colors.

— Refresh rate (in Hz)

The refresh rate is the number of times that the display panel is refreshed every second. The common refresh rate is 60 Hz, and a lower refresh rate may result in bad visual effects.

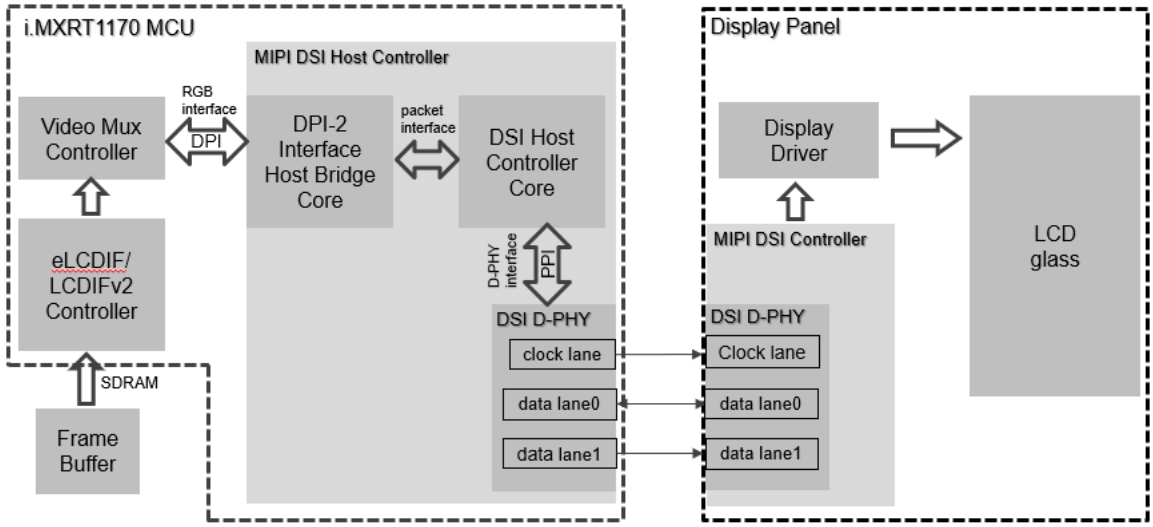

Figure 8. Basic MIPI DSI-compliant LCD system

Figure 8 shows the basic MIPI DSI-compliant LCD system.

For LCD display system, the framebuffer is the key point. It influences the performance of the system and visual effects.

- Memory space allocated for the framebuffer is usually shared with other system devices (CPU core, DMA, network, etc.).
- The data in framebuffer is organized as an array of bits, bytes, half-words, or words, depending on the selected color depth and color bit organization.
- Buffer size is computed using *columns × rows × size of (color depth)*.

  For example:

  720×1280 display @24bpp (RGB888) = 720 columns×1280 rows×3 bytes/pixel = 2764800 bytes

For the display compatibility, the most important step is to determine the framebuffer memory size and its location. The required framebuffer size is doubled for double framebuffer configuration. It is common to use a double buffer configuration where one buffer is used to store the current image while the other to prepare the next image.

3 LCDIFv2 controller

i.MX RT1170 includes the enhanced Liquid Crystal Display Interface (eLCDIF) and the LCDIF Interface version 2 (LCDIFv2). They are both display controllers used to fetch graphics stored in memory and display them on an LCD panel.

This chapter mainly introduces LCDIFv2 controller. LCDIFv2 includes following features:

- Support for RGB interface only.
- The display layers can support up to maximum eight layers of alpha blending.
- Support for one parallel camera interface input and typical data formats of CSI-2: 16 bpp (YUV422 8-bit), 24 bpp (RGB888), 18 bpp (RGB666), 16 bpp (RGB565), 15 bpp (RGB555), 12 bpp (RGB444).

3.1 RGB interface

RGB interface is the mode used in moving picture displays. It includes VSYNC, HSYNC, DOTCLK and ENABLE signals.

LCD (TFT) display with RGB interface can drive three segments (one pixel) per clock with variable electric field strength. The RGB interface provides following features:

- Supports many colors.
- Always supports one pixel per clock (three segments of Red, Green, and Blue).
- Contains the color levels depending on the number of data lines on the LCD panel and number of LCD controller data output signals. It may be 24 lines -24 bits per pixel (bpp), or 18 bpp, 16 bpp.
- Supports the parallel data interface. One clock requires 24 bits (or other formats) for one pixel.

For LCD RGB interface, on every pixel clock edge (raising or falling) and within the LCD active area, the controller fetches one pixel data from its FIFO, converts it to the pixel panel format (coded in the RGB888 or other formats), puts it in the signal generator unit, and drives out to the RGB interface. The pixel data is then displayed on the screen.

In order to drive correct picture on LCD, some timing parameters should be considered for a size of LCD display. The parameters should be programmed to match the display specifications.

Figure 9 illustrates the programmable timings and resolutions. Table 2 shows an example of timing parameters for a 1280 × 800 LCD display.

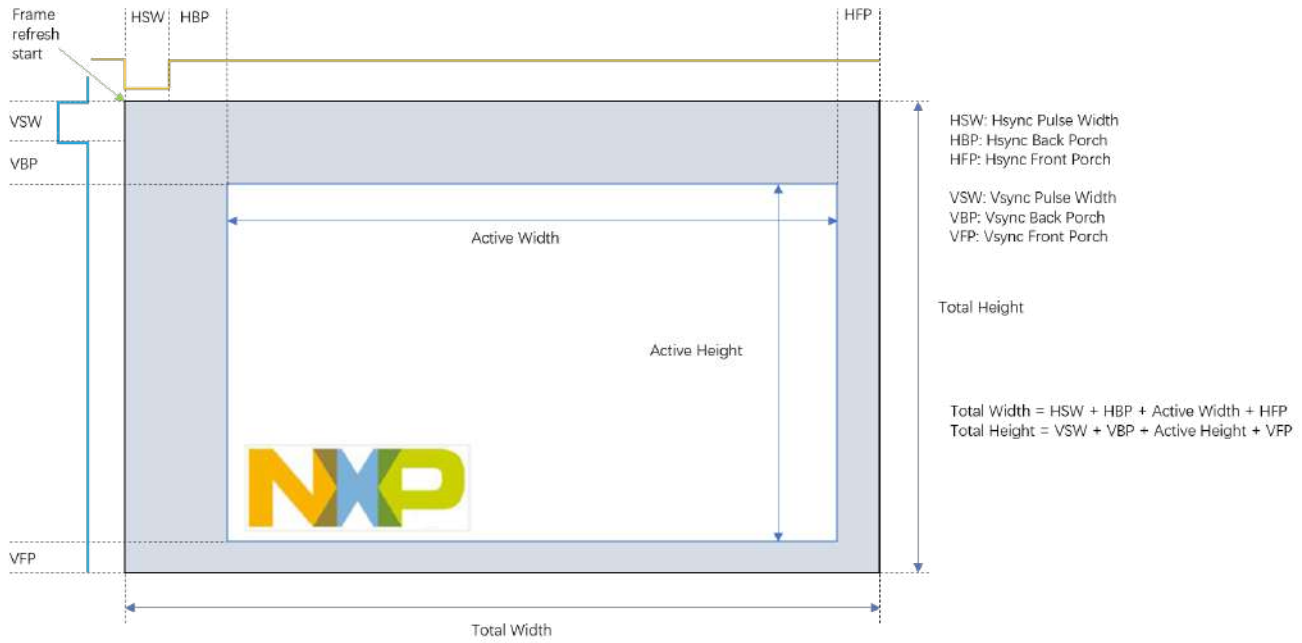

Figure 9. Programmable timings and resolution

Table 2. Timing parameters for a 1280 × 800 LCD display

Parameter		Symbol	Min.	Type	Max.	Unit
DCLK frequency@ Frame rate = 60 Hz		DCLK	68.9	71.1	73.4	MHz
Hsync	Period time	th	1340	1440	1470	DCLK
	Display period	thd	—	1280	—	DCLK

Table continues on the next page...

Table 2. Timing parameters for a 1280 × 800 LCD display (continued)

Parameter		Symbol	Min.	Type	Max.	Unit
	Back porch	thb	—	80	—	DCLK
	Front porch	thfp	—	70	—	DCLK
	Pulse width	thpw	—	10	—	DCLK
Vsync	Period time	tv	815	823	833	H
	Display period	tvd	—	800	—	H
	Back porch	tvb	—	10	—	H
	Front porch	tvfp	—	10	—	H
	Pulse width	tvpw	—	3	—	H

The RGB mode writes data at high speed to the LCD, and the display operation is synchronized with the VSYNC, HSYNC, ENABLE and DOTCLK signals. Figure 10 shows the process of timing parameters for a full frame.

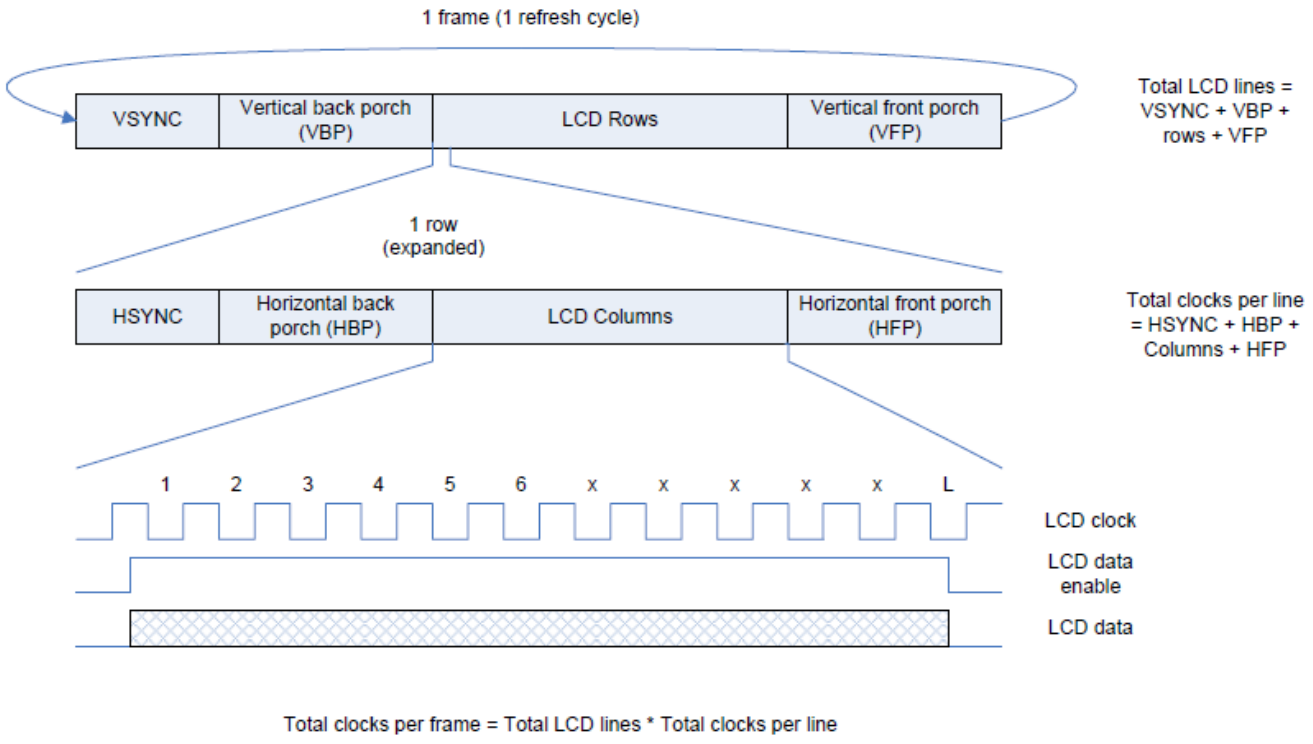

Figure 10. Frame and line timing parameters

3.2 LCDIFv2 signals

To drive LCD panels, the LCDIFv2 controller provides related signals. Table 3 shows the detailed signal descriptions. LCDIFv2 provides these signals through DPI-2 interface onto MIPI DSI host controller.

Table 3. Detailed signal description

Signal	Direction	Description
DCLK	OUT	Pixel clock used to drive the display panel.
VSYNC	OUT	Vertical sync signal, indicating the beginning of a new frame.
HSYNC	OUT	Horizontal sync signal, indicating the beginning of a new line.
ENABLE	OUT	Data Enable. Indicates when there is valid pixel data.
TXDATA[23:0]	OUT	Red, green and blue data output.

It is usual that display panel interface includes other signals that are not part of the LCDIFv2 signals described in Table 3. These additional signals are required for a display module to be fully functional. The LCDIFv2 controller can drive only signals described in Table 3. The signals that are not part of the LCDIFv2 may be managed using GPIOs and other peripherals need specific circuits. The display panels usually embed a backlight unit which requires an additional backlight control circuit and a GPIO. Some display panels need I^2C or SPI for touch panel.

3.3 LCDIFv2 clock

There are four clocks input to the LCDIFv2:

- `apb_clk, b_clk` – the same clock from SOC which are used for the register and bus access.
- `pix_clk` - should be configured according to TFT panel spec for supported display resolution.
- `pdi_clk` - be the same frequency as `pix_clk` for the `PASS_THROUGH` mode.

There is no frequency restriction between `pix_clk` and `b_clk` in the LCDIFv2, but for better performance for multiple layers data fetch, higher `b_clk` can provide more available bandwidth from system memory to avoid FIFO underrun.

3.4 Video Mux controller

i.MX RT1170 support various LCD/Camera interfaces and display controllers, so video mux controller is used to provide mux control between interfaces and controllers.

Figure 11 shows the video mux block. For example, LCDIFv2 can be selected to control MIPI DSI.

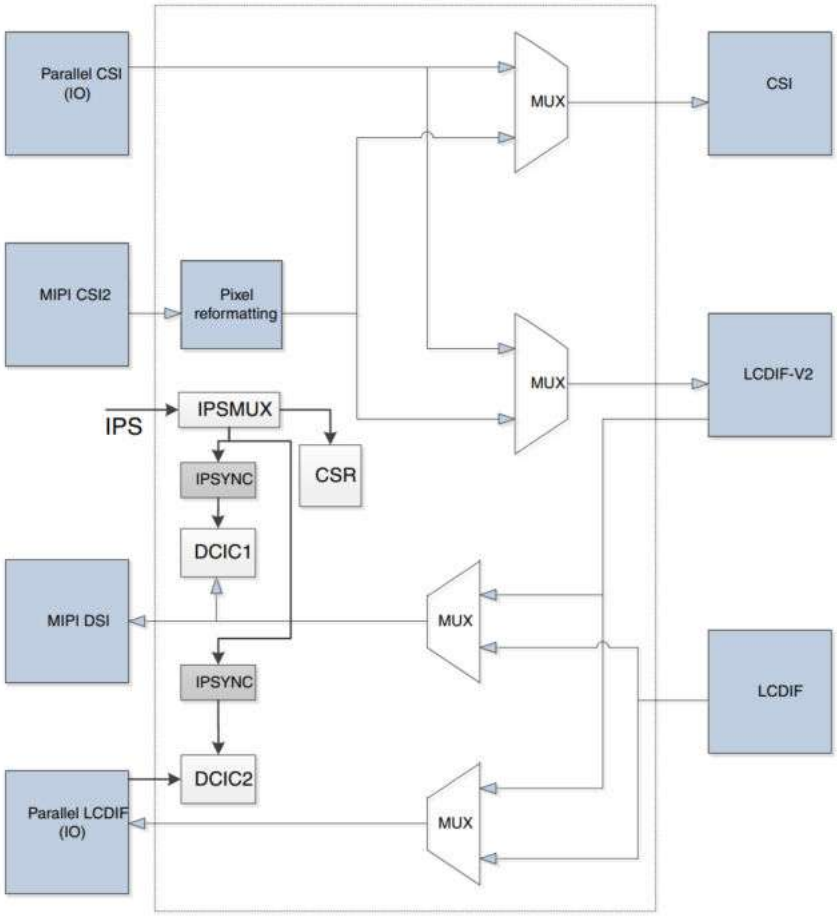

Figure 11. Video Mux block

4 Run the demo

There is a project named `sd_jpeg` in the i.MX RT1170 SDK package. The project can demonstrate the LCDIFv2 controller and MIPI DSI host controller functionalities. Please download the SDK package, based on your developing environment, from NXP official website.

4.1 sd_jpeg demo

The `sd_jpeg` demo can be found in the *boards\evkmimxrt1170\jpeg_examples\sd_jpeg* directory for i.MX RT1170 SDK. The demo application reads the JPEG pictures from the SD card, decodes them, and shows them in the LCD panel one by one.

Before running the demo, please perform one change to modify the stack and heap sizes for bigger framebuffer of 720 × 1280 LCD in IAR options. Figure 12 shows example of detail change for stack and heap sizes, please configure according to actual needs.

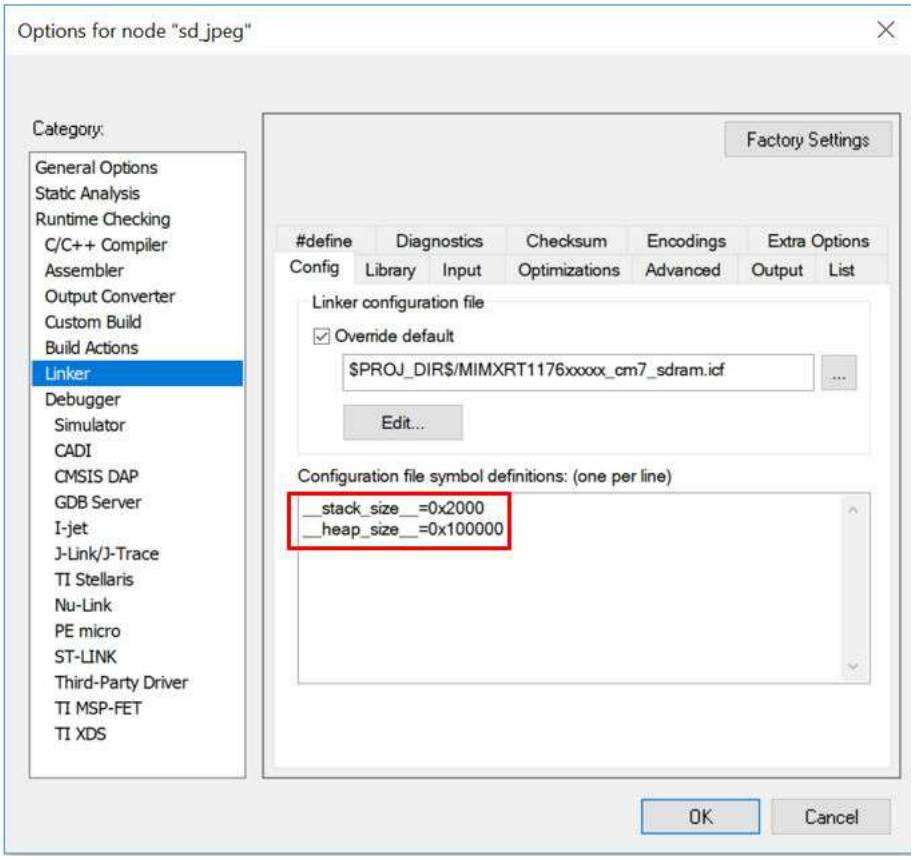

Figure 12. IAR options change

Build, download, and run the demo on i.MX RT1170-EVK to drive a 720 × 1280 LCD panel. The LCD is connected to the board via MIPI DSI interface. Then you can see pictures shown one by one on the panel, as shown in Figure 13 .

Figure 13. i.MX RT1170-EVK board with 720 × 1280 LCD panel

In order to use LCDIFv2 controller and MIPI DSI controller to drive a DSI-complaint LCD panel, user application should perform as following steps:

- Configure power, reset, and backlight GPIO signals for LCD.
- Configure the LCD timing parameters and signal polarity.

  For example, `sd_jpeg` demo defines LCD resolution and timing parameters as below:

```
#define DEMO_PANEL_WIDTH (720)
#define DEMO_PANEL_HEIGHT (1280)
#define DEMO_HSW 8
#define DEMO_HFP 32
#define DEMO_HBP 32
#define DEMO_VSW 2
#define DEMO_VFP 16
#define DEMO_VBP 14
#define DEMO_POL_FLAGS   \
(kLCDIFV2_DataEnableActiveHigh | kLCDIFV2_VsyncActiveLow | kLCDIFV2_HsyncActiveLow |
   kLCDIFV2_DriveDataOnRisingClkEdge)
```

- Configure Video Mux register to make mux control between display controller and interface.

For example, `sd_jpeg` demo defines LCDIFv2 output to MIPI DSI:

```
CLOCK_EnableClock(kCLOCK_Video_Mux);
VIDEO_MUX->VID_MUX_CTRL.SET = VIDEO_MUX_VID_MUX_CTRL_MIPI_DSI_SEL_MASK;
```

- Calculate and configure the pixel clock, and RxClkEsc, TxClkEsc, DPHY reference clocks for MIPI DSI. The pixel clock is equal to *(height + VSW + VFP + VBP) * (width + HSW + HFP + HBP) * refresh rate*.

  Figure 14 shows the clocks used in `sd_jpeg` demo.

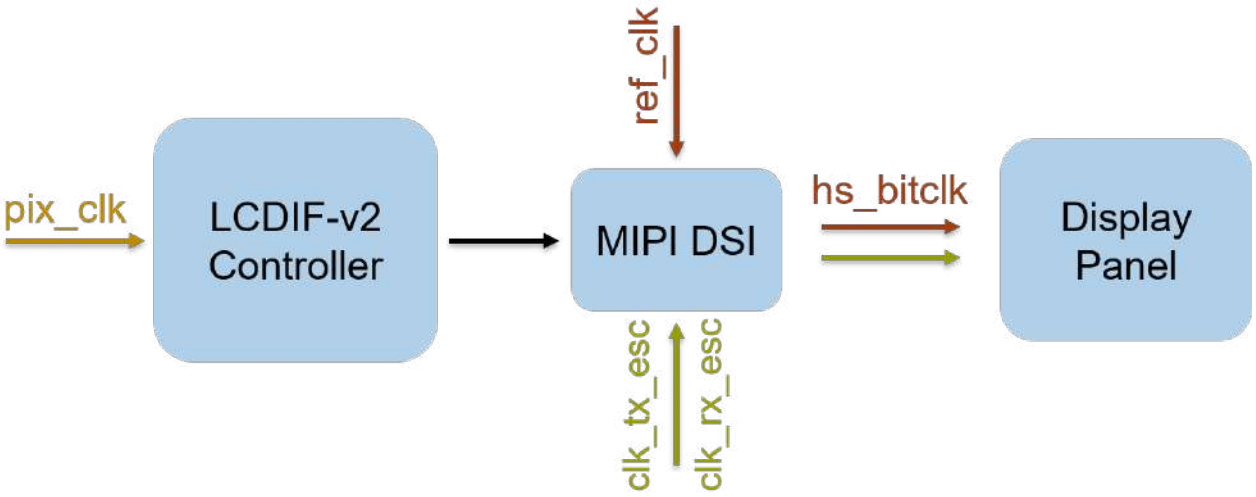

Figure 14. Clocks for demo

Detailed clock configurations of `sd_jpeg` demo for 720 × 1280 LCD are as below:

```
/* pix_clk = 528MHz / 9 = 58MHz. */
/* clk_rx_esc = 528MHz / 11 = 48MHz. */
/* clk_tx_esc = 528MHz / 11 / 3 = 16MHz. */
/* ref_clk = 24MHz / 1= 24MHz. */
```

- Calculate and configure the DPHY high-speed bit clock, *hs_bitclk = TxByteClkHS × 8*. `hs_bitclk` is generated by DPHY PLL using `ref_clk` and must be fast enough to send out the pixels to LCD panel, it should be larger than:

  *(pix_clk * bit per output pixel) / number of MIPI data lane*

- Configure DSI module, DPHY module and DPI-2 interface.
- Configure TFT driver for LCD panel, for example, LCD panel used in this demo includes a TFT driver called RM68200.

4.2 LCD refresh rate

The refresh rate is the number of times that the display panel is refreshed every second. The common refresh rate is 60 Hz, and a lower refresh rate may result in bad visual effects. This section uses `sd_jpeg` demo to test the largest refresh rate.

The refresh rate of LCD panel is calculated by the following formula:

refresh rate = pix_clk/(height + VSW + VFP + VBP) × (width + HSW + HFP + HBP)

In order to test the largest frame rate, continue to increase the value of `pix_clk` until the LCD panel does not display the normal image.

The test result shows that when `pix_clk` is increased to 528 M/6=88 Mhz, the LCD panel cannot display the normal image. At this point, for 720 × 1280 LCD panel, the frame rate is about 84 Hz.

As a result, the largest frame rate of the 720 × 1280 LCD panel is about 84 Hz.

5 References

- i.MX RT1170 Processor Reference Manual (Rev. E, 12/2019)
- *i.MX RT eLCDIF RGB Mode Use Case* (document AN12302)

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 08/2020

Document identifier: AN12940

arm