AN 037: Using the MIPI Virtual Channel Example Design

AN037-v1.3
November 2022
www.efinixinc.com

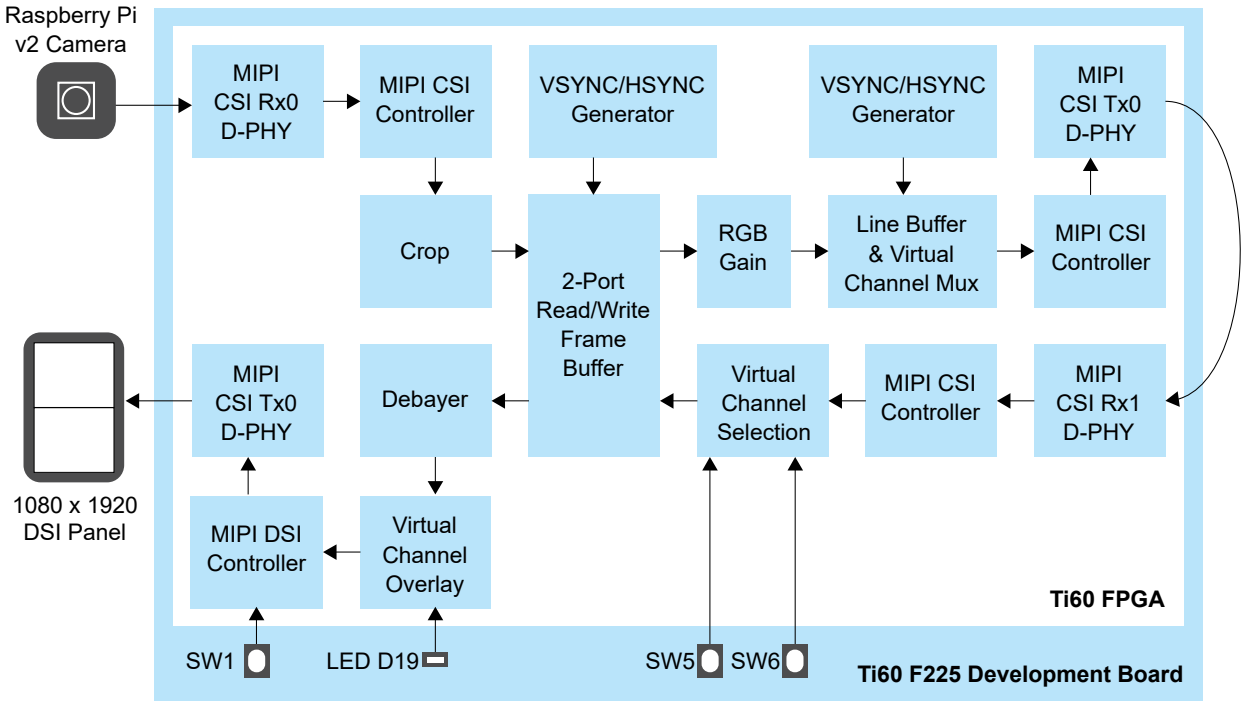
Contents

Introduction

Efinix provides a MIPI virtual channel example design that demonstrates how the Titanium Ti60 FPGA can receive video data from a camera through a MIPI CSI-2 interface, process it, send and receive the data through a MIPI D-PHY loopback, re-process the data for video display, and then display the video stream on an LCD panel.

A Raspberry Pi v2 camera captures video and sends it through the Titanium Ti60 F225 Development Board's RX interface. The interface processes the video (1920 x 1080 resolution) using two MIPI DSI lanes. The Ti60 FPGA manipulates the image data and creates 16 virtual channels of the input video, each with different RGB gain values. The Ti60 FPGA then sends the data out of the board and receives back the manipulated data through a loop-back connection using four MIPI CSI-2 lanes. You can select two videos out of the 16 virtual channels by using the SW5 and SW6 pushbuttons. The design adds virtual channel numbers to the selected video, and sends the output to the Mini-DSI panel through four MIPI CSI-2 channels. The Mini-DSI panel displays the two selected virtual channel videos in a 1080 x 1920 resolution.

Figure 1: Example Design Block Diagram



Required Hardware

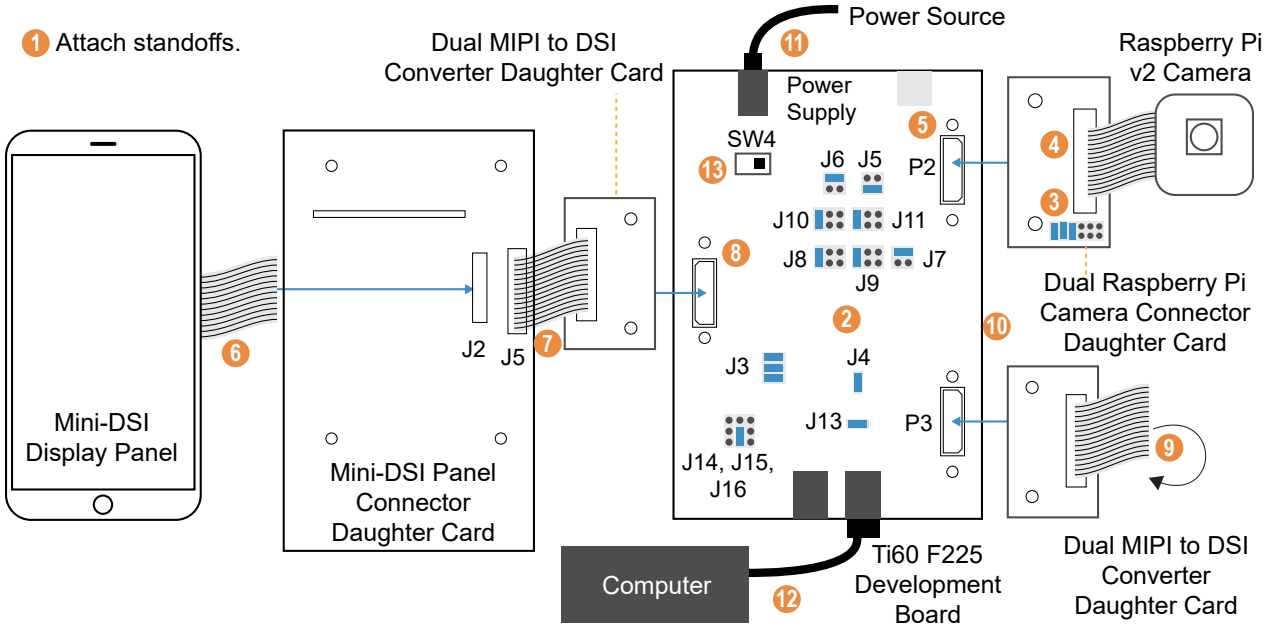
The example design uses the following hardware:

- Titanium Ti60 F225 Development Board
- 1 Mini-DSI Panel Connector Daughter Card
- 2 Dual MIPI to DSI Converter Daughter Cards
- 1 MIPI and LVDS Expansion Daughter Card
- 1 Dual Raspberry Pi Camera Connector Daughter Card
- 1 Mini-DSI panel
- 1 Raspberry Pi v2 camera module
- 1 15-pin flat cable
- 2 30-pin flat cables
- 1 USB type-C cable (required for programming the example design into the Titanium Ti60 F225 Development Board)
- Universal AC to DC power adapter

Set Up the Hardware

The following figure shows the hardware setup steps:

Figure 2: Hardware Setup



1. Attach standoffs to the board if you have not already done so.
2. On the Titanium Ti60 F225 Development Board, connect the following jumpers:

Header	Net Name	State	Short Pins
J3	VCC	0.95 V	1 and 2
J3	VCC	0.95 V	3 and 4
J3	VCC	0.95 V	5 and 6
J4	VCCAUX	1.8 V	1 and 2
J5	VCCIO33_TR	3.3 V	1 and 2
J6	VCCIO33_TL	1.8 V	3 and 4
J7	VCCIO33_BR	1.8 V	3 and 4
J8	VCCIO3A	1.2 V	5 and 6
J9	VCCIO3B	1.2 V	5 and 6
J10	VCCIO4A	1.2 V	5 and 6
J11	VCCIO4B	1.2 V	5 and 6
J13	SPI_ENA	Short	1 and 2
J14	FX3_PMODE0	Float	Unconnected
J15	FX3_PMODE1	Pull-up	1 and 2
J16	FX3_PMODE2	Float	Unconnected

3. On the Raspberry Pi Camera Connector Daughter Card, connect the following pins with jumpers: 1 - 2, 3 - 4, and 5 - 6.
4. Connect the Raspberry Pi v2 camera module to the daughter card using the 15-pin flat cable.
5. Connect the Raspberry Pi Camera Connector Daughter Card to the P2 header of the Titanium Ti60 F225 Development Board.
6. Connect the Mini-DSI Display panel to the Mini-DSI Panel Connector Daughter Card using the attached 48-pin flat cable.
7. Connect header J5 of Mini-DSI Panel Connector Daughter Card to connector J2 of Dual MIPI to DSI Converter Daughter Card using the 30-pin flat cable.
8. Connect the Dual MIPI to DSI Converter Daughter Card to the P1 header of the Titanium Ti60 F225 Development Board.
9. Connect header J2 to header J3 of the Dual MIPI to DSI Converter Daughter Card using the 30-pin flat cable.
10. Connect the Dual MIPI to DSI Converter Daughter Card to the P3 header of the Titanium Ti60 F225 Development Board.
11. Ensure the board power switch is turned off, then connect the 12 V power cable to the board connector and to a power source.
12. Connect the USB header J12 to USB port of your computer.
13. Turn on the board's power switch.

Note: The Mini-DSI Panel Connector Daughter Card includes a potentiometer, R14, which you can use to adjust the Mini-DSI display panel brightness. You can adjust the brightness by setting the R14 potentiometer while probing pin LEDA on the board. Efinix recommends that you limit the range from 16 V to 17 V.

Program the Titanium Ti60 F225 Development Board

The Titanium Ti60 F225 Development Board ships pre-loaded with a demonstration design. To use the MIPI virtual channel example design, you must program the design into the board.

1. Turn on the Titanium Ti60 F225 Development Board.
2. Download the example design file, **TI60F225-16VC-DEMO-v<version>.zip** from the Support Center.
3. Unzip the file into your working directory.
4. Open the project in the Efinity software and review it.
5. Use the Efinity® Programmer to download the bitstream file to your board. The example includes a bitstream file, **ti60f225_demo.hex**.

Learn more: Instructions on how to use the Efinity® software is available in the Support Center.

Run the Example Design

After the FPGA configures, the Mini-DSI panel displays channels 8 and 9 of the 16 available virtual video channels one on top of the other. Each channel has a unique RGB gain and a channel number overlay for reference. Use the following pushbuttons to change to a different video channel:

- *Pushbutton SW5*—Change video channel displayed on the lower half of the display
- *Pushbutton SW6*—Change video channel displayed on the upper half of the display

Important: To protect the Mini-DSI panel from LCD flickering and LCD image retention, press pushbutton SW1 to turn-off the Mini-DSI panel before turning off the Titanium Ti60 F225 Development Board.

Customizing the Example Design

You can customize the example design by changing the following register mapping files:

- *piv2_1080p_reg.mem*—Register mapping for the Raspberry Pi camera resolution and frame rate
- *Panel_1080p_reg.mem*—Register mapping that includes the MIPI commands and video timing registers for the Mini-DSI panel display

Raspberry Pi Camera Resolution and Frame Rate

To use a different Raspberry Pi camera resolution and frame rate, modify the **piv2_1080p_reg.mem** register map file. Each line represents a register or data.

- Line 1 is the MSB of the 8 bit register.
- Line 2 is the LSB of the 8 bit register.
- Line 3 is the 8-bit data.
- Line 4 is the MSB of the next register, and so on

For example:

```
30
00
01
30
05
01
30
06
00
```

Lines 1, 2, and 3 refer to register 3000 and data 01.

Lines 4, 5, and 6 refer to register 3005 and data 01.

Mini-DSI Panel Display Settings

The Mini-DSI panel display settings can be changed using the **Panel_1080p_reg.mem** file. The following table lists the default MIPI commands and video timing registers used in the example design.

Table 1: Design Example MIPI Commands and Video Timing

Register	Name	Description
0000030D40	Delay setting.	MSB 16 bits are "00", following 64 bits are delay which are number of AXI clocks.
00000186A0	Delay setting.	
2000001105	MIPI commands.	See MIPI Commands on page 8.
2000002905	MIPI commands.	
4000001590	Total H line wordcount in byte.	See Video Timing Register on page 11.
440000012C	Horizontal sync active (HSA) in byte.	
4800000258	Horizontal black porch (HBP) in byte.	
4C0000012C	Horizontal black porch (HFP) in byte.	
5000000003	Vertical sync active (VSA) in line.	
5400000005	Vertical black porch (VBP) in line.	
5800000006	Vertical black porch (VFP) in line.	
5C00000780	Vertical active (VACT) in line.	
180000000A	Send HS blanking packet and video stream enable.	

MIPI Commands

The following tables list the register definition for MIPI display command set (DCS) and generic commands.

Table 2: MIPI DCS and Generic Short Commands

Bits	Description
15:0	MIPI DCS or generic short command type: 0x02: Color mode off command 0x12: Color mode on command 0x22: Shutdown peripheral command 0x32: Turn on peripheral command 0x03: Generic short write, no parameters 0x13: Generic short write, 1 parameters 0x23: Generic short write, 2 parameters 0x04: Generic short read, no parameters 0x14: Generic short read, 1 parameters 0x24: Generic short read, 2 parameters 0x05: DCS short write, no parameters 0x15: DCS short write, 1 parameters 0x06: DCS read short LP/HS 0x37: Set Max Return packet size 0x29: Generic long write 0x39: DCS long write
31:16	• MIPI DCS: Command • Generic: Parameter 1 (MSB) • DCS and generic long command: Total number of byte including command and data
47:32	• MIPI DCS: Parameter • Generic: Parameter 2 (LSB)
63:48	0x00 (Reserved)
79:64	Command mode: 0x20: LP command 0x1C: HS command

Table 3: MIPI DCS and Generic Long Commands

Bits	Description
15:0	• MIPI DCS: DCS long command • Generic: data (MSB)
31:16	MIPI DCS or generic long command data
47:32	MIPI DCS or generic long command data (LSB)
63:48	0x00 (Reserved)
79:64	0x28: Low Power Long command 0x2C: High Speed Long command

Table 4: Short Command Examples

Example	Command	Description
LP DCS short write, no parameter	2000001105	0x20: LP command
		0x05: DCS short command without parameter
		0x11: MIPI DCS command
HS DCS short write, no parameter	1C00001105	0x1C: HS command
		0x05: DCS short command without parameter
		0x11: DCS command
LP DCS short write, 1 parameter	2000013515	0x20: LP command
		0x15: DCS short command with 1 parameter
		0x35: DCS command
		0x01: DCS parameter
HS DCS short write, 1 parameter	1C00013515	0x1C: HS command
		0x15: DCS short command with 1 parameter
		0x35: DCS command
		0x01: DCS parameter
LP Generic short write, 1 parameter	2000001113	0x20: LP command
		0x13: Generic short command with 1 parameter
		0x11: Generic data
HS Generic short write, 1 parameter	1C00001113	0x1C: HS command
		0x13: Generic short command with 1 parameter
		0x11: Generic data
LP Generic short write, 2 parameter	2000013523	0x20: LP command
		0x23: Generic short command with 2 parameter
		0x35 (MSB), 0x01 (LSB): Data
HS Generic short write, 2 parameter	1C00013523	0x1C: HS command
		0x23: Generic short command with 2 parameter
		0x35 (MSB), 0x01 (LSB): Data

Table 5: Long Command Examples

Example	Command	Description
LP DCS long write	2800980381 280008001C	0x28: Low Power command write, long data FIFO
		0x81: DCS command
		0x03 (MSB), 0x98, 0x00, 0x1C, 0x00, 0x08, 0x00 (LSB): Data
	2000000839	0x20: LP command
		0x39: DCS long command
		0x08: Total number of byte including command and data
HS DCS long write	2C00980381 2C0008001C	0x2C: High Speed command write, long data FIFO
		0x81: DCS command
		0x03 (MSB), 0x98, 0x00, 0x1C, 0x00, 0x08, 0x00 (LSB): Data
	1C00000839	0x1C: HS command
		0x39: DCS long command
		0x08: Total number of byte including command and data
LP Generic long write	2800980381 280008001C	0x28: Low Power command write, long data FIFO
		0x81 (MSB), 0x03, 0x98, 0x00, 0x1C, 0x00, 0x08, 0x00 (LSB): Data
	2000000829	0x20: LP command
		0x29: Generic long command
		0x08: Total number of byte of data
HS Generic long write	2C00980381 2C0008001C	0x2C: High Speed command write, long data FIFO
		0x81 (MSB), 0x03, 0x98, 0x00, 0x1C, 0x00, 0x08, 0x00 (LSB): Data
	1C00000829	0x1C: HS command
		0x29: Generic long command
		0x08: Total number of byte of data

Video Timing Register

The video timing register values can be obtained using the **Titanium MIPI Utility.xlsm** file available in the Support Center.

To obtain the video timing register, enter the video details and timing parameters in the **Non-burst sync event** portion. The **Titanium MIPI Utility.xlsm** calculates the video timing register settings and displays them in the Video Timing Register Setting section (green cells).

Table 6: Titanium MIPI Utility.xlsm Definitions

Porch File Item	Definition
HACT	Total number of pixel per line
VACT	Total number of line per frame
HSA	HSYNC pulse width
HBP	Horizontal back porch
HFP	Horizontal front porch
VSA	VSYNC pulse width
VBP	Vertical back porch
VFP	Vertical front porch
Pixel Clock	Video stream pixel clock frequency in MHz
MIPI Speed	DSI TX MIPI lane speed in Mbps
No. data lane	Number of MIPI data lane

The following waveforms show the MIPI video timing parameters.

Figure 3: Video Timing Waveform (Horizontal)

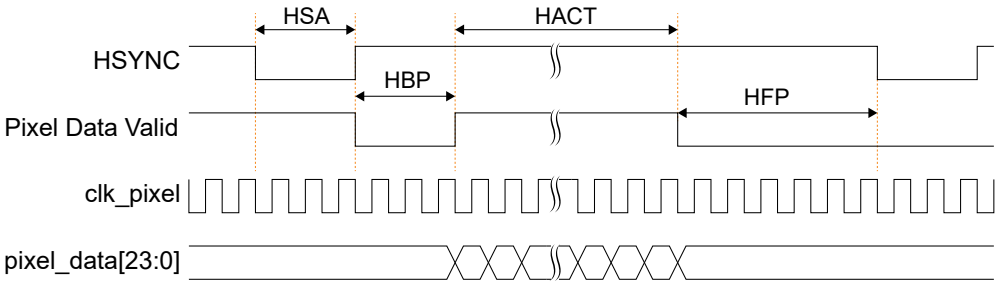
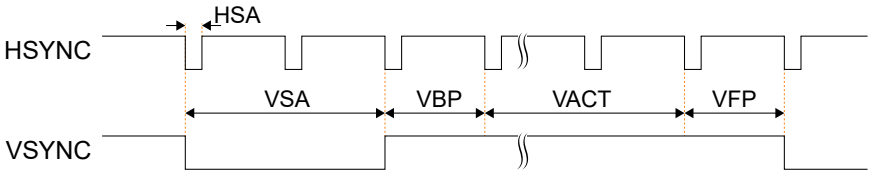


Figure 4: Video Timing Waveform (Vertical)



Revision History

Table 7: Revision History

Date	Version	Description
November 2022	1.3	Update voltage of power cable. (DOC-959)
February 2022	1.2	Update block diagram. (DOC-727)
December 2021	1.1	Updated **DSI_porch_cal.xls** to **Titanium MIPI Utility.xlsm**
November 2021	1.0	Initial release.