PLEORA TECHNOLOGIES INC.

iPORT™ NTx-NBT50 Embedded Video Interface

User Guide

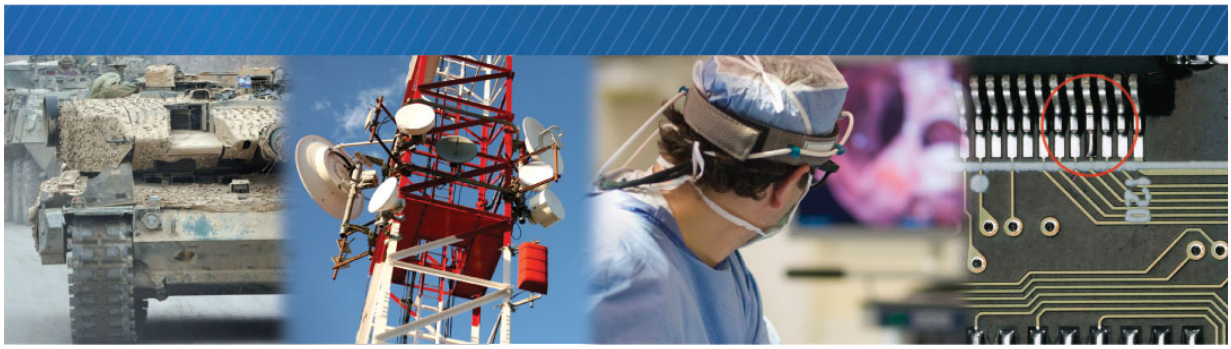
Pleora Technologies

Document Number

EX001-030-0007, Version 5.2 6/10/21

Table of Contents

Chapter 1



About this Guide

This chapter describes the purpose and scope of this guide, and provides a list of complementary guides.

The following topics are covered in this chapter:

What this Guide Provides

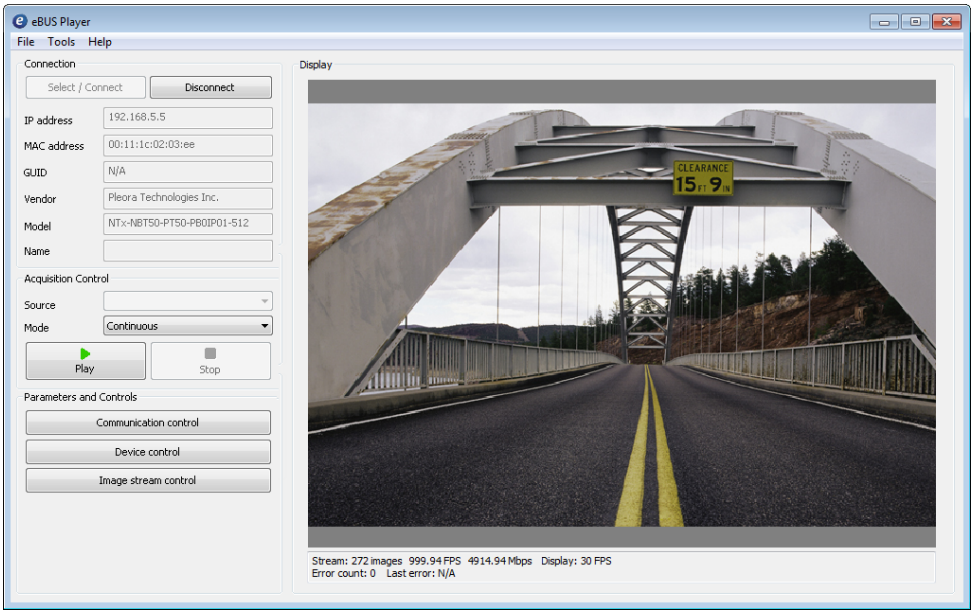
This guide provides you with all of the information you need to connect the iPORT™ NTx-NBT50 Embedded Video Interface to your sensor and related electronics to create a camera or other imaging device. In this guide you will find a product overview, connector details, and mechanical drawings, along with instructions for installing the Pleora eBUS™ SDK, connecting the device, and performing general configuration tasks to properly display video.

The last chapter of this guide provides Technical Support contact information for Pleora Technologies.

Start Streaming Video

If you want to quickly start streaming video, you can jump to:

- "Confirming Image Streaming" on page 66
- "Configuring the Embedded Video Interface's Image Settings" on page 69



Related Documents

The *iPORT NTx-NBT50 Embedded Video Interface User Guide* is complemented by the following Pleora Technologies documents, which are available on the Pleora Technologies Support Center (https://supportcenter.pleora.com):

- *eBUS Player Quick Start Guide* and *eBUS Player User Guide,* available for Windows, Linux, and macOS
- *eBUS SDK API Quick Start Guides*, available for C++, .NET, Linux, and macOS
- *eBUS SDK API Help Files*
- *iPORT Advanced Features User Guide*
- *Serial Bridge Knowledge Base Articles*
- *Configuring Your Computer and Network Adapters for Best Performance Knowledge Base Article*

The following guide is available from your Pleora Support representative, for customers who have purchased the Pleora AutoGen XML generation tool and Firmware Reference Design:

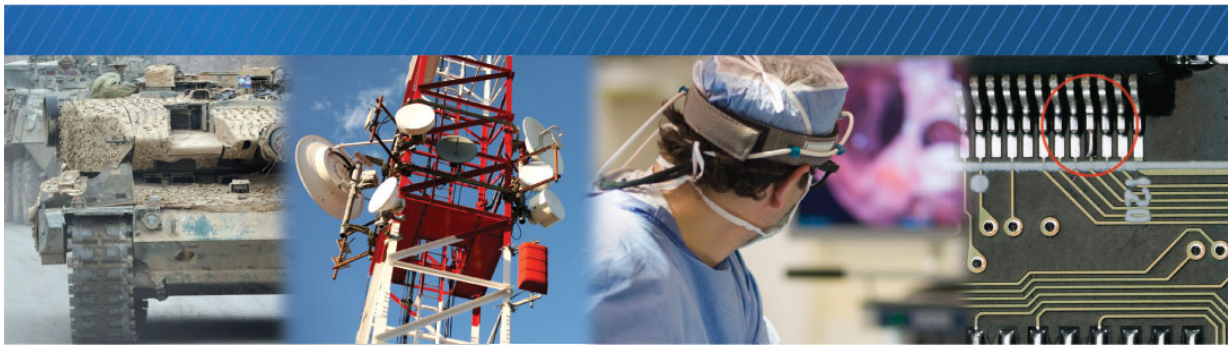
- *GenICam Integration Guide for Altera-Based Products*

Further Reading

Although not required in order to successfully use the embedded video interface, you can find details about industry-related standards and naming conventions in the following documents:

- *GigE Vision Standard, version 2.0* available from the Automated Imaging Association (AIA) at www.visiononline.org
- *GenICam Standard Features Naming Convention* available from the European Machine Vision Association (EMVA) at www.emva.org
- *Pixel Format Naming Convention*, available from the EMVA at www.emva.org
- *Camera Link Standard Specification*, available from the AIA at www.visiononline.org

Chapter 2



About the iPORT NTx-NBT50 Embedded Video Interface

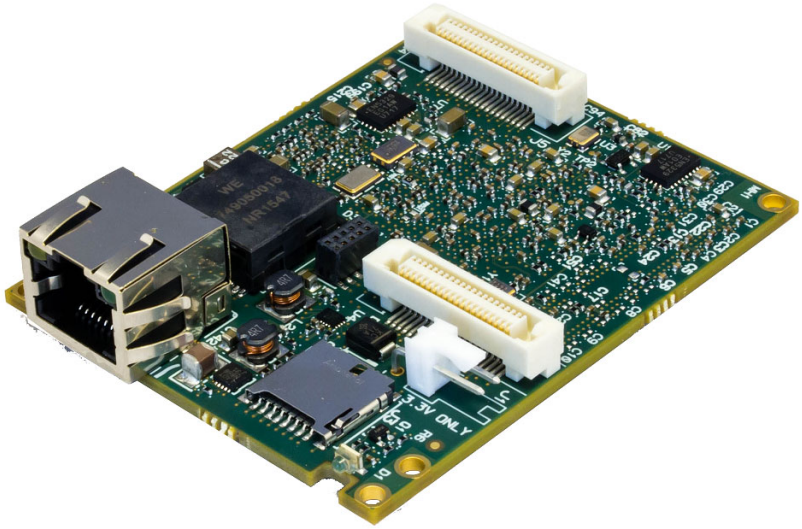
This chapter describes the embedded video interface, including the product models and key features.

The following topics are covered in this chapter:

The iPORT NTx-NBT50 Embedded Video Interface

Pleora's iPORT™ NTx-NBT50 Embedded Video Interface hardware helps manufacturers shorten time-to-market, reduce risk, and lower costs by providing a straightforward way to integrate high-bandwidth GigE Vision® 2.0 video connectivity over NBASE-T into imaging devices and systems.

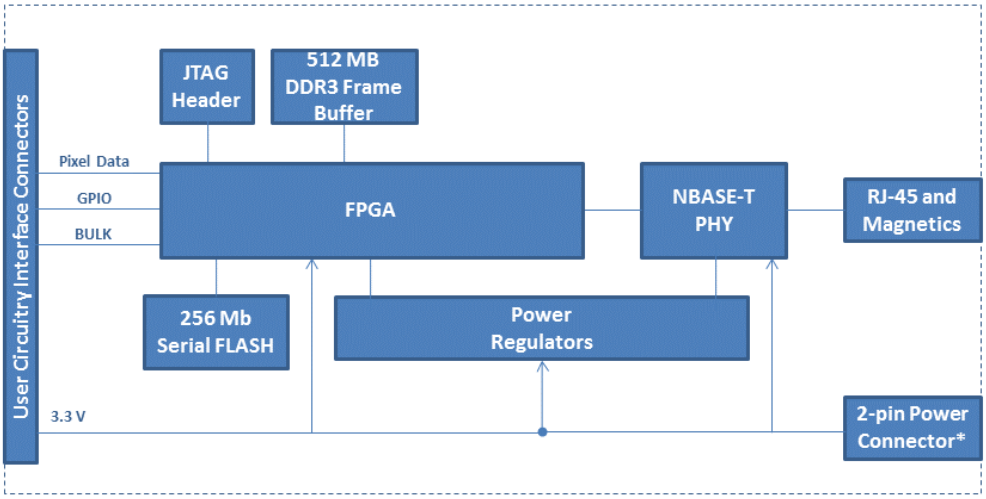
The NTx-NBT Embedded Video Interface supports 5 Gbps transmission of uncompressed images over a standard Cat 5e Ethernet cable for distances up to 100 meters. The video interface complies with the GigE Vision 2.0 and GenICam™ standards, ensuring interoperability in multi-vendor networked or point-to-point digital video systems. The compact hardware solution is easily embedded into flat panel X-ray detectors, imaging systems, and cameras. The product supports the IEEE 1588 Precision Time Protocol to synchronize image capture functions and other system elements, enabling the exact triggering of image acquisition. Integrated image management allows users to track and retrieve images that are transmitted during a particular acquisition session.



Pleora's iPORT NTx-NBT Embedded Video Interface is supported by:

- A Development Kit to help speed time-to-market by enabling the rapid design of prototypes and proof-of-concept demonstrations, often without requiring hardware development;
- eBUS SDK, a feature-rich application development toolkit for manufacturers to rebrand and distribute with their end-products;
- The AutoGen XML generation tool and a firmware reference design, which make it fast and easy for manufacturers to create a user-friendly GenICam interface for their products.

Figure 1: iPORT NTx-NBT50 Embedded Video Interface Block Diagram



*The 2-pin power connector is for use with the power supply that is included with the NTx-NBT50 Development Kit. There is no filter or protection on this connector.

eBUS SDK

As an element of Pleora's video connectivity solutions, the iPORT NTx-NBT50 Embedded Video Interface is offered with the eBUS™ SDK, a feature-rich toolkit that provides the building blocks needed to quickly and easily design high-performance video applications that consume minimal CPU resources.

NTx-NBT50 Embedded Video Interface Models

The iPORT NTx-NBT50 Embedded Video Interface is available in two models and is equipped with the parts listed in the following table.

Table 1: iPORT NTx-NBT50 Embedded Video Interface Models

Order code	Model	Quantity
900-8003	**iPORT NTx-NBT50 OEM Board**	
	NTx-NBT50 OEM board with horizontal RJ-45 jack	1

Order code	Model	Quantity
900-8006	**iPORT NTx-NBT50 Development Kit**	
	NTx-NBT50 OEM board with horizontal RJ-45 jack, mounted to a thermal baseplate	1
	NBASE-T Ethernet desktop NIC	1
	Cat 5e Ethernet cable	1
	Power supply	1
	eBUS SDK USB stick	1

Feature Set

This section provides information about the NTx-NBT50 Embedded Video Interface features.

Table 2: NTx-NBT50 Embedded Video Interface General Features

Embedded video interface general features	
Number of channels	1
Scan modes	Area Scan (Progressive) and Line Scan
Pixel depth (bits)	8, 10, 12, 14, 16, and 24 bits
Serial LVDS clock	37.5 MHz to 100 MHz
Taps per data channel	Up to 4 for internal 48-bit pixel depth parallel video bus
Image width (pixels)	• Minimum: 4[a] • Default: 640 • Maximum: 16 376 • Increment: 4[a]
Image height (pixels)	• Minimum: 1 • Default: 480 • Maximum: 16 383 • Increment: 1
Windowing/region of interest	Yes
Tap geometry	1X_1Y, 1X2_1Y, 1X4_1Y, 1X, 1X2, 1X4

a. Image width increment of 8 when in Extended Chunk Mode.

Table 3: NTx-NBT50 Inputs/Outputs on the User Circuitry Interface

Inputs/outputs on user circuitry interface	
Video input	Serialized LVDS†
GPIO inputs	4 x 3.3 V LVTTL/2.5 V LVCMOS**
GPIO outputs	3 x 3.3 V LVTTL**
Serial (bulk)*	1 x 2.5 V LVCMOS ** 2 x 3.3 V LVTTL **
Camera control outputs	4 x 3.3 V LVTTL**

† For detailed signal specifications, see the characteristics of the sensor interface that are provided in "Sensor Interface" on page 41.

* The bulk interface ports support the UART (Universal Asynchronous Receiver/Transmitter), USRT (Universal Synchronous Receiver/Transmitter), I2C (Inter-Integrated Circuit), Serial Peripheral Interface

(SPI), and High-Speed Download (HSD) protocols. For information about which protocols are supported on each bulk interface, see "Bulk Interfaces and Supported Protocols" on page 28.

** For detailed signal specifications, see the characteristics of the inputs and outputs on the user circuitry connector that are provided in "40-Pin User Circuitry Interface Connector Pinouts" on page 17.

Table 4: NTx-NBT50 Embedded Video Interface Hardware

Hardware	
User circuitry interface (including internal power interface)	Two 40-pin Hirose connectors: FX6-40S-08SV2(93)
External power interface	2-pin 0.10" header
NBASE-T interface	RJ-45
NBASE-T PHY	Marvell 88X3310
Image buffer	512 MB
Persistent memory	256 Mb Serial FLASH

Table 5: NTx-NBT50 Embedded Video Interface Physical Characteristics

Physical characteristics	
Size (L x W x D)	72.5 mm x 56.0 mm x 17.6 mm (approximate, including RJ-45 jack)
Weight	32.5 grams
IC operating temperature range	Commercial[a,b]
Storage temperature	-40°C to 85°C
Power supply	3.3V +/- 3% (3.20V to 3.40V)
Typical power consumption, 30m CAT5e, 4.7Gbps	6 W
MTBF at 40° C	1,059,389 hours

a. Case and junction temperature limits vary by IC device. See "Ambient and Junction Temperatures" on page 25.

b. Consult your local Pleora Sales representative for information about industrial temperature grade devices.

Selected GenICam Features

The iPORT NTx-NBT50 Embedded Video Interface supports the seven features mandated by the GigE Vision standard along with additional features, some of which are listed in the following table. The full list of features can be seen in the Device Control dialog box of Pleora's eBUS Player application.

Table 6: Selected GenICam Features

Feature	Description
Width	Specifies the width of the image (in pixels).
Height	Specifies the height of the image (in pixels).
OffsetX	Specifies the horizontal image offset (in pixels).
OffsetY	Specifies the vertical image offset (in pixels).
PixelFormat	Specifies the format of the pixel provided by the device. Available pixel formats are: • Monochrome pixel formats, 8 to 16 bits • Bayer pixel formats, 8 to 16 bits • RGB and BGR, 24 bits
DeviceTapGeometry	Describes the geometrical properties characterizing the taps of a Camera Link camera, as seen from the frame grabber or acquisition card. This device tap geometry feature is defined in the GenICam SFNC. Available tap geometries are: • Geometry_1X_1Y • Geometry_1X2_1Y • Geometry_1X4_1Y • Geometry_1X • Geometry_1X2 • Geometry_1X4
DeviceScanType	Specifies the sensor scan type, such as Area Scan or Line Scan.
SensorDigitizationTaps	Specifies the number of digitized samples output simultaneously by the camera, 1, 2, or 4 taps.

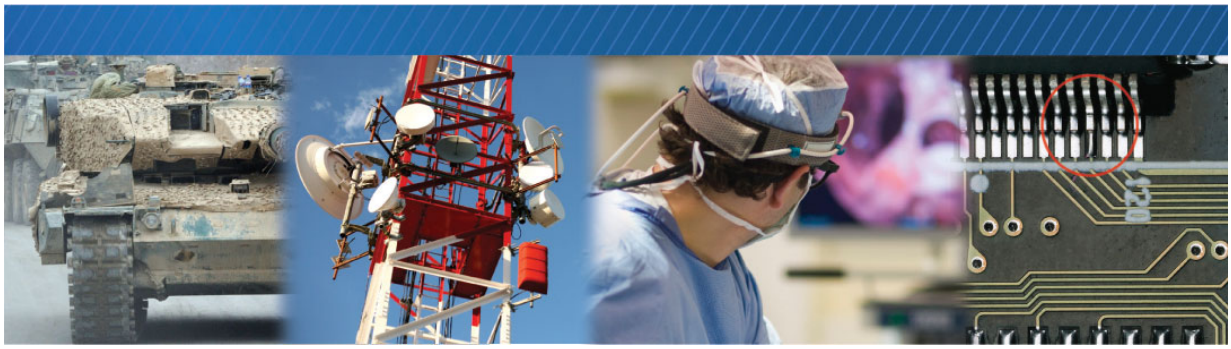
NTx-NBT50 Embedded Video Interface Pixel Formats

The pixel formats available on the NTx-NBT50 Embedded Video Interface are listed in the following table.

Table 7: NTx-NBT50 Embedded Video Interface Pixel Formats

Taps	Pixel format			
1, 2, 4 taps	Mono8 (Default)	Mono8s		
1, 2, 4 taps	Mono10	Mono10Packed		
1, 2, 4 taps	Mono12	Mono12Packed		
1 tap	Mono14			
1, 2 taps	Mono16			
1, 2, 4 taps	BayerGR8	BayerRG8	BayerGB8	BayerBG8
1, 2, 4 taps	BayerGR10	BayerRG10	BayerGB10	BayerBG10
1, 2, 4 taps	BayerGR12	BayerRG12	BayerGB12	BayerBG12
1, 2, 4 taps	BayerGR10Packed	BayerRG10Packed	BayerGB10Packed	BayerBG10Packed
1, 2, 4 taps	BayerGR12Packed	BayerRG12Packed	BayerGB12Packed	BayerBG12Packed
1, 2 taps	BayerGR16	BayerRG16	BayerGB16	BayerBG16
1, 2 taps	RGB8			
1, 2 taps	BGR8			

Chapter 3



NTx-NBT50 Embedded Video Interface External Connections

This chapter describes the NTx-NBT50 Embedded Video Interface connections.

The following topics are covered in this chapter:

NTx-NBT50 Embedded Video Interface Connector Locations

The following figure and table describe the NTx-NBT50 Embedded Video Interface connectors.

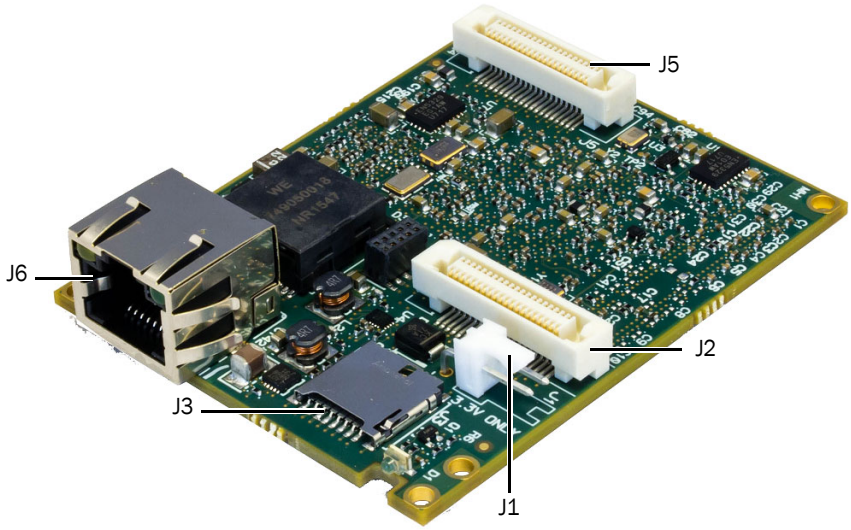
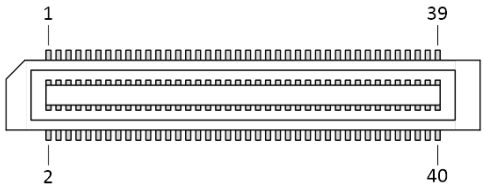


Table 8: NTx-NBT50 Embedded Video Interface Connector Descriptions

ID	Type	Description
J2, J5	40-pin user circuitry interface	Interfaces directly to the imaging device, such as an X-ray flat panel detector or camera.
J1	2-pin power connector	Receives 3.3 V of filtered DC input. For more information, see "Power Connector" on page 22.
J6	RJ-45 Ethernet connector	Interfaces the embedded video interface to Ethernet networks, as specified in IEEE 802.3. The Ethernet interface can operate at 1000, 2500, or 5000 Mbps, and supports Internet Protocol Version 4 (IPv4).
J3	Micro SD card connector	Reserved for future use

40-Pin User Circuitry Interface Connector Pinouts

The tables in this section provide the pinouts for the NTx-NBT50 Embedded Video Interface 40-pin user circuitry connectors, J5 and J2.



The following table lists the characteristics for the single-ended I/Os on the user circuitry interface connectors (J5 and J2).

Table 9: 2.5 V and 3.3 V I/Os on the NTx-NBT50 Embedded Video Interface

2.5 V I/Os	
VIL, minimum, absolute	-0.5 V
VIL, minimum	0 V
VIL, maximum	0.7 V
VIH, minimum	1.6 V
VIH, maximum	3.6 V
VIH, maximum, absolute	3.8 V
VOL, minimum	0 V
VOL, maximum	0.4 V
VOH, minimum	2.0 V
VOH, maximum	2.625 V

3.3 V I/Os	
VIL, minimum, absolute	-0.5 V
VIL, minimum	0 V
VIL, maximum	0.8 V
VIH, minimum	1.7 V
VIH, maximum	3.6 V
VIH, maximum, absolute	3.8 V
VOL, minimum	0 V
VOL, maximum	0.2 V
VOH, minimum	2.9 V
VOH, maximum	3.465 V

The following table lists the pinouts of the J5 40-pin user circuitry interface connector.

Table 10: J5 User Circuitry Interface Connector Pinouts

Pin	Signal Name	Type	Direction	Description
1	GND	GND	—	Ground
2	GND	GND	—	Ground
3	LVDS0_CLK_P	LVDS	IN	Data clock positive polarity
4	LVDS0_DATA0_P	LVDS	IN	Data 0 positive polarity
5	LVDS0_CLK_N	LVDS	IN	Data clock negative polarity
6	LVDS0_DATA0_N	LVDS	IN	Data 0 negative polarity
7	GND	GND	—	Ground
8	GND	GND	—	Ground
9	LVDS0_DATA1_P	LVDS	IN	Data 1 positive polarity
10	LVDS0_DATA2_P	LVDS	IN	Data 2 positive polarity
11	LVDS0_DATA1_N	LVDS	IN	Data 1 negative polarity
12	LVDS0_DATA2_N	LVDS	IN	Data 2 negative polarity
13	+3.3V	PWR	—	Power input[a]
14	+3.3V	PWR	—	Power input[a]
15	LVDS0_DATA3_P	LVDS	IN	Data 3 positive polarity
16	LVDS0_DATA4_P	LVDS	IN	Data 4 positive polarity
17	LVDS0_DATA3_N	LVDS	IN	Data 3 negative polarity
18	LVDS0_DATA4_N	LVDS	IN	Data 4 negative polarity
19	BULK_CS0	3.3 V	OUT	Bulk interface 0 SPI chip select and HSD chip select
20	BULK_RX0	3.3 V	IN, OUT	Bulk interface 0 UART and USRT input, SPI MISO signal, and HSD SOUT1
21	BULK_TX0	3.3 V	IN, OUT	Bulk interface 0 UART and USRT output, I2C input/open drain output, SPI MOSI signal, and HSD SOUT0
22	BULK_CLK0	3.3 V	IN, OUT	Bulk interface 0 USRT, I2C, SPI output clock, and HSD output clock
23	LVDS0_DATA5_P	LVDS	IN	Data 5 positive polarity
24	LVDS0_DATA6_P	LVDS	IN	Data 6 positive polarity
25	LVDS0_DATA5_N	LVDS	IN	Data 5 negative polarity
26	LVDS0_DATA6_N	LVDS	IN	Data 6 negative polarity
27	+3.3V	PWR	—	Power input[a]
28	+3.3V	PWR	—	Power input[a]

Table 10: J5 User Circuitry Interface Connector Pinouts (Continued)

Pin	Signal Name	Type	Direction	Description
29	Reserved	—	—	Do not connect (FPGA output, always logical '0')
30	Reserved	—	—	Do not connect (FPGA output, always logical '0')
31	Reserved	—	—	Do not connect (FPGA output, always logical '0')
32	Reserved	—	—	Do not connect (FPGA output, always logical '0')
33	GND	GND	—	Ground
34	GND	GND	—	Ground
35	Reserved	—	—	Do not connect (FPGA output, always logical '0')
36	Not used	NC	—	Not connected
37	Reserved	—	OUT	Do not connect (FPGA output, always logical '0')
38	PWR_ON_RSTN	3.3 V	IN, OUT, OC	Power on Reset[b]
39	GND	GND	—	Ground
40	GND	GND	—	Ground

a. 3.3 V input, 0.5 A maximum current rating per pin. Maximum current required at 3.3 V from the power regulator is 2.2 A (7.2 W). There is no filter or protection for this pin.

b. **PWR_ON_RSTN** is a bidirectional open collector pin with a 10 KOhm resistor to +3.3 V on the FPGA board. This signal is high when power on the NTx-NBT50 Embedded Video Interface is at the appropriate levels. You can do any of the following:

- Leave the pin set to N.C.
- Connect the signal to the open-collector/open-drain power ready signal of the user circuitry. When you drive the signal low, it holds the embedded video interface in reset.
- Use the signal to start the configuration of user devices, such as FPGAs or CPUs.
- Use the signal to initiate a reset of the embedded video interface FPGA. A minimum low pulse width of 2 us is required.

The following table lists the pinouts of the J2 40-pin user circuitry interface connector.

Table 11: J2 User Circuitry Interface Connector Pinouts

Pin	Signal Name	Type	Direction	Description
1	GND	GND	—	Ground
2	GND	GND	—	Ground
3	Not used	NC	—	Not connected
4	Reserved	—	—	Do not connect (FPGA output, always logical '0')
5	Reserved	—	—	Do not connect (FPGA output, always logical '0')
6	Reserved	—	—	Do not connect (FPGA output, always logical '0')
7	GND	GND	—	Ground
8	GND	GND	—	Ground
9	Reserved	—	—	Do not connect (FPGA output, always logical '0')
10	Reserved	—	—	Do not connect (FPGA output, always logical '0')
11	Reserved	—	—	Do not connect (FPGA output, always logical '0')
12	Reserved	—	—	Do not connect (FPGA output, always logical '0')
13	+3.3V	PWR	—	Power input[a]
14	+3.3V	PWR	—	Power input[a]
15	Reserved	—	—	Do not connect (FPGA output, always logical '0')
16	BULK_RX2	3.3 V	IN	Bulk interface 2 UART input[b]
17	BULK_TX2	3.3 V	OUT	Bulk interface 2 UART output
18	PB0_CTRL_OUT0	3.3 V	OUT	Connected to the PLC, signal Pb0CC0
19	BULK_CLK1	2.5 V	IN, OUT	Bulk interface 1 USRT, I2C, and SPI clock
20	BULK_RX1	2.5 V	IN	Bulk interface 1 UART and USRT input, and SPI MISO signal[b]
21	BULK_TX1	2.5 V	IN, OUT	Bulk interface 1 UART and USRT output, I2C input/open drain output, and SPI MOSI
22	FPGA_GPIO_IN2	3.3 V	IN	Connected to the PLC, signal GpioIn2[b]
23	PB0_CTRL_OUT1	3.3 V	OUT	Connected to the PLC, signal Pb0CC1
24	PB0_CTRL_OUT2	3.3 V	OUT	Connected to the PLC, signal Pb0CC2
25	PB0_CTRL_OUT3	3.3 V	OUT	Connected to the PLC, signal Pb0CC3
26	FPGA_GPIO_OUT2	3.3 V	OUT	Connected to the PLC, signal GpioOut2. Bulk interface 0 HSD SOUT3.
27	+3.3V	PWR	—	Power input[a]
28	+3.3V	PWR	—	Power input[a]
29	FPGA_GPIO_IN3	3.3 V	IN	Connected to the PLC, signal GpioIn3[b]

Table 11: J2 User Circuitry Interface Connector Pinouts (Continued)

Pin	Signal Name	Type	Direction	Description
30	FPGA_GPIO_OUT1	3.3 V	OUT	Connected to the PLC, signal GpioOut1
31	FPGA_GPIO_IN0	3.3 V	IN	Connected to the PLC, signal GpioIn0[b]
32	FPGA_GPIO_OUT0	3.3 V	OUT	Connected to the PLC, signal GpioOut0. Bulk interface 0 HSD SOUT2.
33	GND	GND	—	Ground
34	GND	GND	—	Ground
35	FPGA_GPIO_IN1	3.3 V	IN	Connected to the PLC, signal GpioIn1[b]
36	BULK_CS1	2.5 V	IN, OUT	Bulk interface 1 SPI chip select
37	Not used	NC	—	Not connected
38	Not used	NC	—	Not connected
39	GND	GND	—	Ground
40	GND	GND	—	Ground

a. 3.3 V input, 0.5 A maximum current rating per pin. Maximum current required at 3.3 V from the power regulator is 2.2 A (7.2 W). There is no filter or protection for this pin.

b. If you do not use these pins, we recommend that you tie them to GND instead of leaving them unconnected.

Power Connector

The 2-pin power connector is only for use with the power supply that is included with the NTx-NBT50 Development Kit. There is no filter or protection on this connector.

The development kit is intended for the design of prototypes and proof-of-concept demonstrations. The development kit can be used with the 2-pin power connector and the included power supply.

When you are using the OEM board in your system, power should be supplied through the user circuitry connector, not the 2-pin connector.

Do not power the embedded video interface using the 2-pin power connector and user circuitry connector at the same time.

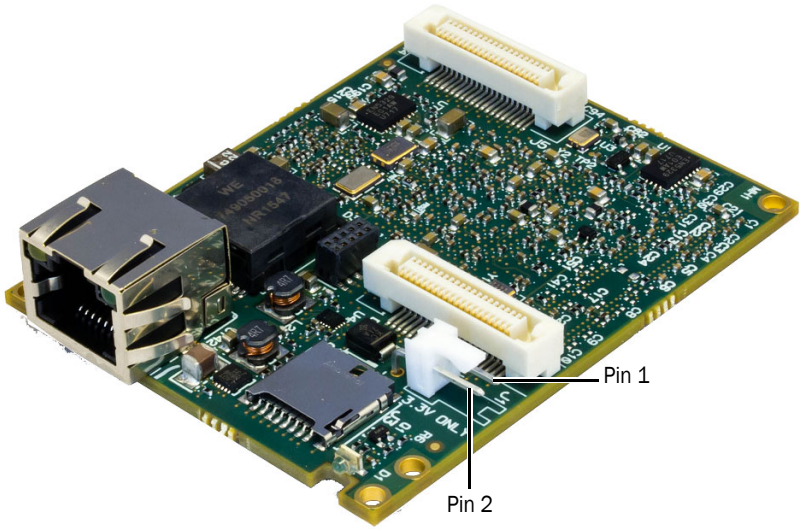


Table 12: NTx-NBT Embedded Video Interface Power Connector Details

Pin	Name	Type	Description
1	+3.3V	Power	Power input
2	GND	Ground	Ground

Status LEDs

The NTx-NBT50 Embedded Video Interface has status LEDs that indicate the operating status of the network connection, the power, and the firmware, as described in the following image and table.

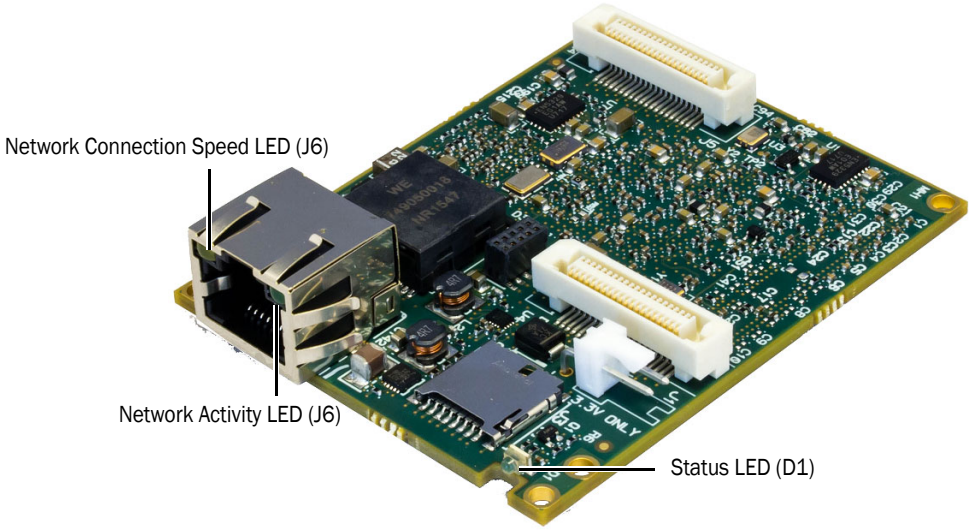
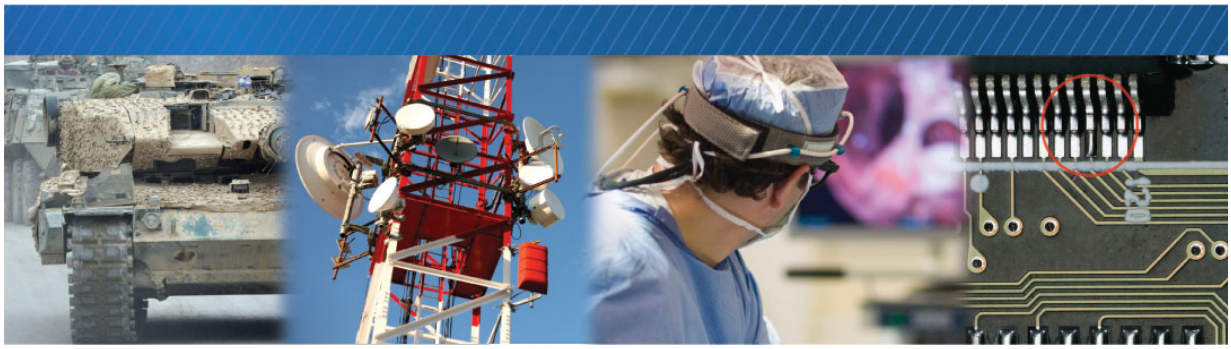


Table 13: NTx-NBT50 Embedded Video Interface Status LED Descriptions

LED	ID	Description
Status	D1	**Yellow and green off:** Power is not supplied or no firmware load was programmed **Yellow solid on:** The backup firmware load is running **Green solid on:** The main firmware load is running
Network Connection Speed	J6	**Off:** 1 Gbps/1000BASE-T connection **Yellow solid on:** 2.5 Gbps/2.5GBASE-T connection **Yellow blinking:** 5Gbps/5GBASE-T connection
Network Activity	J6	**Green off:** No Ethernet connection **Green solid on:** Ethernet link **Green blinking:** Data is being transmitted or received

Chapter 4



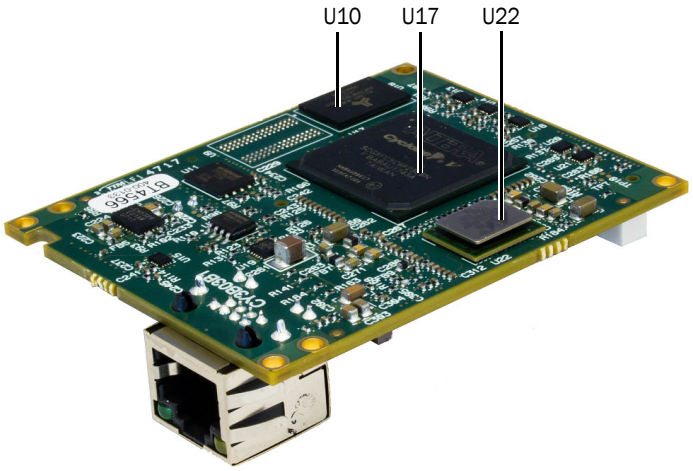
Ambient and Junction Temperatures

This chapter provides you with the information you need to ensure the optimal operating temperature for your NTx-NBT50 Embedded Video Interface.

The NTx-NBT50 Embedded Video Interface is designed to be used with a thermal solution or heat sink; otherwise, it will malfunction and can be damaged.

The following figure shows the components that consume the largest amount of power on the NTx-NBT50 Embedded Video Interface, and that will therefore be most affected by high temperatures.

Figure 2: NTx-NBT50 Embedded Video Interface Underside



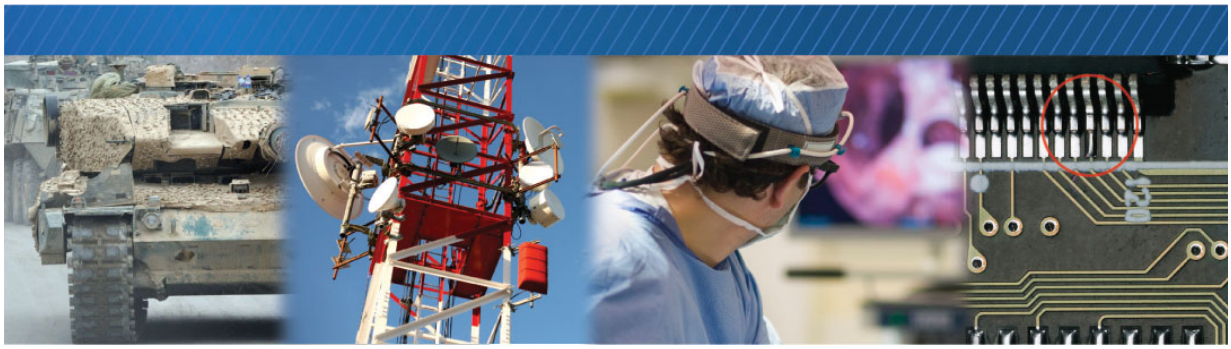
The following table provides thermal guidelines, including ambient and junction temperatures.

Table 14: NTx-NBT50 Embedded Video Interface Thermal Guidelines

Reference designator	Component and manufacturer part number	Manufacturer rating for component on standard Pleora product*
U22	Marvell PHY Part number: 88X3310-A1-BUS4C000	**Ambient:** 0 °C to +70 °C **Junction:** 0 °C to +105 °C **Case:** Not specified **Junction-to-ambient thermal resistance Θ_{JA}:** • **Still air**: 19.56 (°C/W) • **1 m/sec:** 16.90 (°C/W) • **2 m/sec**: 15.42 (°C/W) **Junction-to-case thermal resistance Θ_{JC}, still air:** 0.66 (°C/W) **Junction-to-board thermal resistance Θ_{JB}, still air:** 8.18 (°C/W) **Power consumption:** ~ 2500 mW @ 5Gb
U10	Alliance DDR3 Part number: AS4C256M16D3LA-12BCN	**Ambient:** Not specified **Junction:** Not specified **Case:** 0 °C to +95 °C **Junction-to-case thermal resistance Θ_{JC}:** Not specified **Junction-to-ambient thermal resistance Θ_{JA}:** Not specified **Power consumption:** ~ 350 mW per device
U17	Altera FPGA Part number: 5CGXFC5C6F23C6N	**Ambient:** Not specified **Junction:** 0 ° to +85 °C **Case:** Not specified **Junction-to-ambient thermal resistance Θ_{JA}:** • **Still air:** 17.2 (°C/W) • **100 ft./min:** 15.4 (°C/W) • **200 ft./min:** 13.5 (°C/W) • **400 ft./min:** 12 (°C/W) **Junction-to-case thermal resistance Θ_{JC}:** 3.2 (°C/W) **Junction-to-board thermal resistance Θ_{JB}:** 6.8 (°C/W) **Power consumption:** ~ 2200 mW

* Θ_{JC} = (Tj - Ta)/Ptop, where Ptop = Power dissipation from the top of the package. Θ_{JA} = (Tc - Ta)/P, where P = Total power dissipation.

Chapter 5



Bulk Interfaces

This chapter describes the bulk interfaces available on the NTx-NBT50 Embedded Video Interface, and the supported protocols.

The following topics are covered in this chapter:

Bulk Interfaces and Supported Protocols

The NTx-NBT50 Embedded Video Interface has three bulk interface ports available for serial communication.

Each bulk interface port supports the standard UART (Universal Asynchronous Receiver/Transmitter). Two of the bulk interface ports (Bulk0 and Bulk1) also support USRT (Universal Synchronous Receiver/Transmitter), I2C (Inter-Integrated Circuit) protocols. Serial Peripheral Interface (SPI) protocol. One bulk interface port (Bulk0) supports High-Speed Download protocol (HSD).

The bulk interface ports are available on the 40-pin user circuitry connectors, as outlined in the following table.

Table 15: NTx-NBT50 Embedded Video Interface Bulk Interface and Connector Pinouts

Pin	Bulk signal name	Bulk mode				
		UART	USRT	I2C	SPI	HSD
J5 40-pin user circuitry connector						
19	BULK_CS0				SS	SS
20	BULK_RX0	RXD	RXD		MISO	SOUT1
21	BULK_TX0	TXD	TXD	SDA	MOSI	SOUT0
22	BULK_CLK0		SCK	SCL	SCLK	SCLK
J2 40-pin user circuitry connector						
16	BULK_RX2	RXD				
17	BULK_TX2	TXD				
19	BULK_CLK1		SCK	SCL	SCLK	
20	BULK_RX1	RXD	RXD		MISO	
21	BULK_TX1	TXD	TXD	SDA	MOSI	
26	FPGA_GPIO_OUT2					SOUT3 (Quad HSD only)*
32	FPGA_GPIO_OUT0					SOUT2 (Quad HSD only)*
36	BULK_CS1				SS	

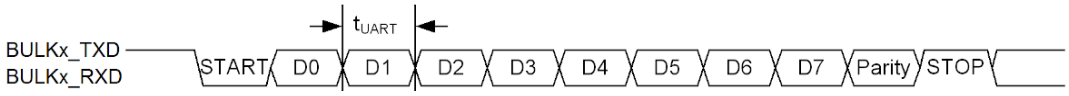
* When QHSD is selected in the **BulkMode** list, these two FPGA_GPIO_OUT signals become Quad HSD SOUT signals.

The bulk interfaces on the NTx-NBT50 Embedded Video Interface are 3.3 V LVTTL/2.5 V LVCMOS. For detailed signal specifications, see the characteristics of the inputs and outputs on the user circuitry connector that are provided in "40-Pin User Circuitry Interface Connector Pinouts" on page 17.

UART Timing

The UART interface supports:

- 8-bit data transfer
- 1 start bit
- Programmable stop bit(s): 1 or 2
- Parity: Even, odd, or none
- Baud rates:
  - Predefined rates: 9600, 14 400, 19 200, 28 800, 38 400, 57 600, 115 200, 230 400, 460 800, 921 600
  - Programmable
- Loop back mode from downstream to upstream



A number of preset baud rates can be used. If you require a baud rate that is not covered by the presets, you can specify your own baud rate. To specify your own baud rate:

1. In the **Device Control** dialog box, under **Port Communication**, choose **Programmable** in the **BulkBaudRate** list.
2. In the **BulkBaudRateFactor** field, enter a baud rate between 1 and 511.

The embedded video interface calculates the baud rate using the following equation:
(66.66MHz * 1000000)/(BulkBaudRateFactor * 16)

Table 16: UART Baud Rates

Baud rate (BR) [bps]	Notes
9,600	Preset 0 (default)
14,400	Preset 1
19,200	Preset 2
28,800	Preset 3
38,400	Preset 4
57,600	Preset 5
115,200	Preset 6
230,400	Preset 7
460,800	Preset 8
921,600	Preset 9
Minimum: 4,166,667 Maximum: 8,154	Programmable baud rate

The following table lists the A.C. operating characteristics of the UART interface.

Table 17: A.C Operating Characteristics of the UART Interface

Parameter	Symbol	Minimum	Maximum	Units
Data period	t_{UART}	0.240	122.64	µs
Baud rate	BR	8,154	4,166,667	bps

USRT Timing

The USRT (Universal Synchronous Receiver/Transmitter) serial interface resembles the UART interface, but adds a clock signal to enable synchronous communication.

The following table lists the supported USRT clock frequencies and periods.

Table 18: Supported USRT Clock Frequencies and Periods

Bulk system clock divider	Clock period, tSCK (ns)	Clock frequency (MHz)*
By 2	60	16.667
By 4	120	8.333
By 8	240	4.167
By 16	480	2.083
By 32	960	1.042
By 64	1920	0.521
By 128	3840	0.260
By 256	7680	0.130

* To obtain the exact frequency, divide the 33.333 MHz clock speed by one of: 2, 4, 8, 16, 32, 64, 128, or 256.

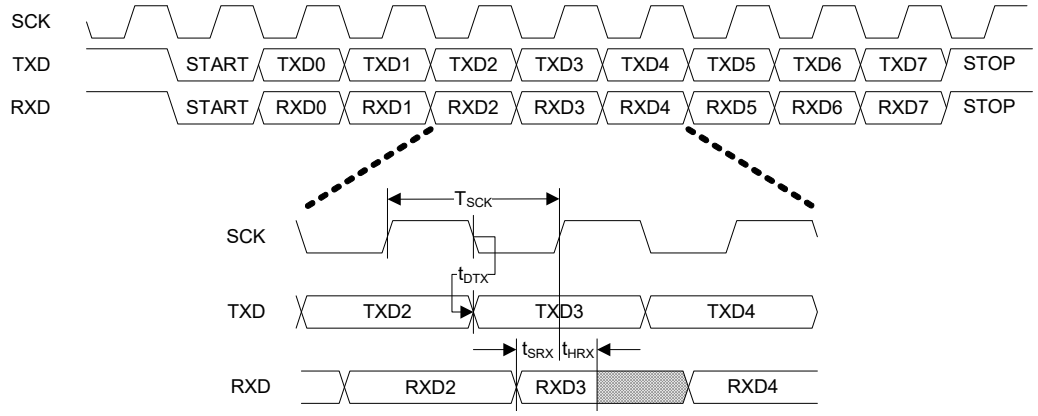


Table 19: USRT Delays

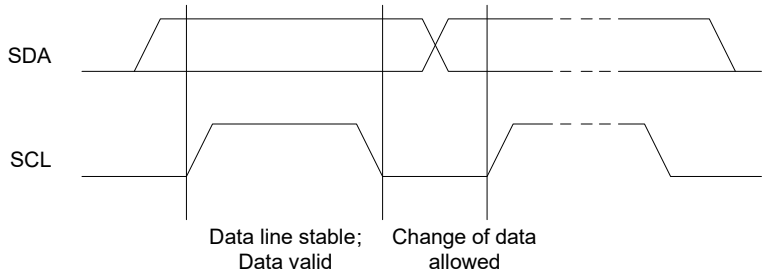
Delay	Minimum	Maximum
SCK to TXD delay t_{DTx}	-12 ns	12 ns
RXD setup time t_{SRx}	22 ns	N/A
RXD hold time t_{HRx}	0 ns	N/A

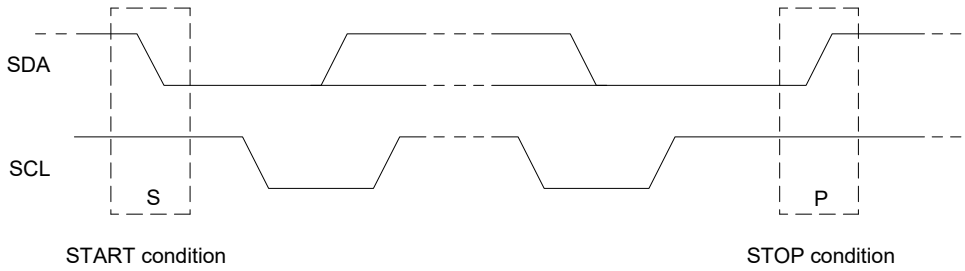
I2C Transmission Speeds

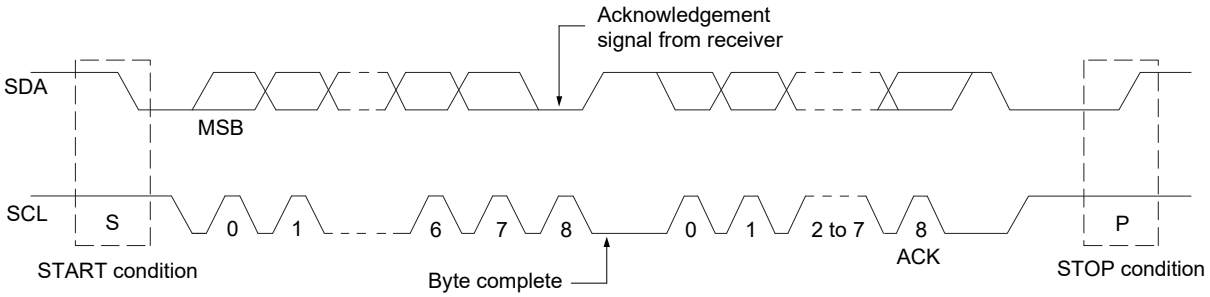
An I2C master mode is available that can be used to communicate with I2C slave devices. The I2C interface is a two-wire, bi-directional serial bus with a serial clock line (SCL) and a serial data line (SDA). Note that all devices connected to these signals must have open drain or open collector outputs. Both lines must be pulled up to VCC by external resistors.

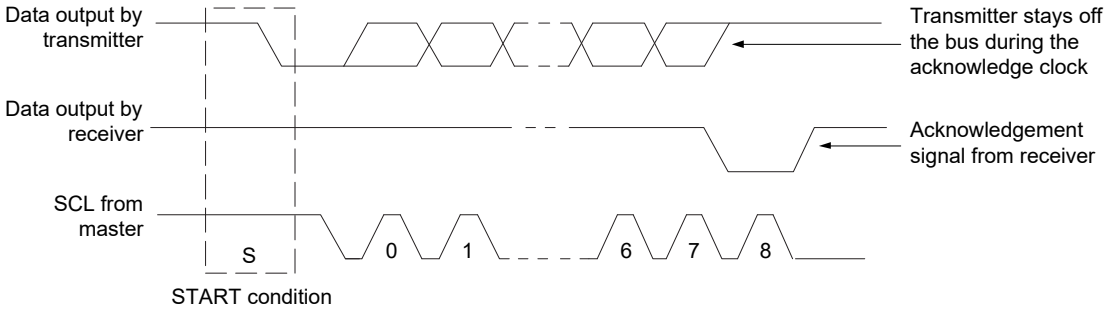
The embedded video interface is compatible with the Philips I2C standard and supports the following transmission speeds: Normal (100 kbit/s) and Fast (400 kbit/s).

Data is transferred synchronously to SCL on the SDA line on a byte-by-byte basis. Each data byte is 8 bits long. There is an SCL clock pulse for each data bit with the most significant bit (MSB) being transmitted first. An acknowledge bit follows each transferred byte. Each bit is sampled during the high period of SCL; therefore the SDA line may be changed only during the low period of SCL and must be held stable during the high period of SCL.






Acknowledgement
signal from receiver
SDA
MSB
SCL
S
0
1
6
7
8
0
1
2 to 7
8
ACK
P
START condition
Byte complete
STOP condition


Data output by
transmitter
Transmitter stays off
the bus during the
acknowledge clock
Data output by
receiver
Acknowledgement
signal from receiver
SCL from
master
S
0
1
6
7
8
START condition

SPI Signals

The NTx-NBT50 Embedded Video Interface has a SPI master mode that is used to provide full duplex, synchronous serial communication using four wires. The master initiates the transaction by asserting slave select (SS). The master also drives a serial clock (SCLK) that provides a synchronous clock source to a slave device. The master transmits data on the Master Out Slave In (MOSI) line and receives data on the Master In Slave Out (MISO) line.

The SPI interface has the following characteristics:

- Master mode supports Motorola SPI protocol
- Programmable transfer rate using the **BulkSystemClockDivider** feature. For more information, see "GenICam Interface for Serial Communication Configuration" on page 38.
  - Maximum value: 33.33 Mbps
  - Minimum value: 0.260 Mbps
- Serial clock with programmable phase and polarity
- SPI word length: 8, 10, 12, 14, and 16 bits
- Bit transmission: Most significant bit (msb) first

The following operating modes are supported:

Table 20: Supported SPI Operating Modes

SPI mode	Clock polarity (CPOL/CKP)	Clock phase (CPHA)
0	0	0
1	0	1
2	1	0
3	1	1

SPI Timing

The following table lists the supported SPI clock frequencies and periods.

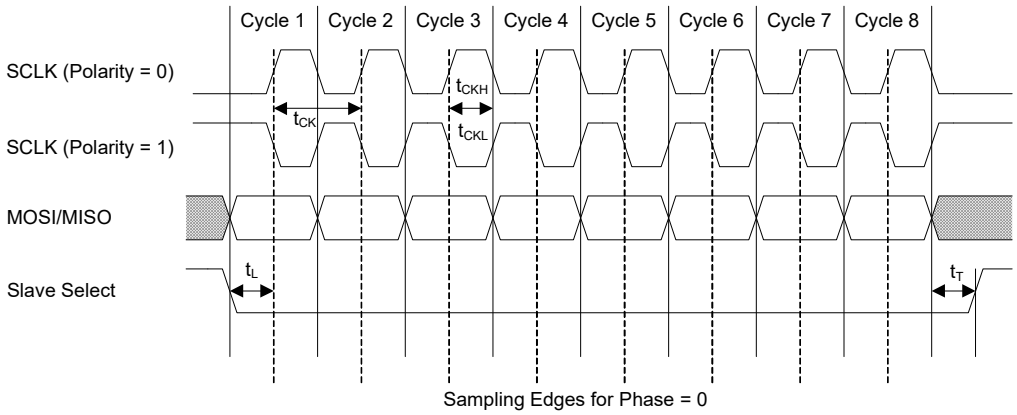
Table 21: Supported SPI Clock Frequencies and Periods

Bulk system clock divider	Clock period, tSCK (ns)	Clock frequency (MHz)*
By 2	30	33.333
By 4	60	16.667
By 8	120	8.333
By 16	240	4.167
By 32	480	2.083
By 64	960	1.042
By 128	1920	0.521
By 256	3840	0.260

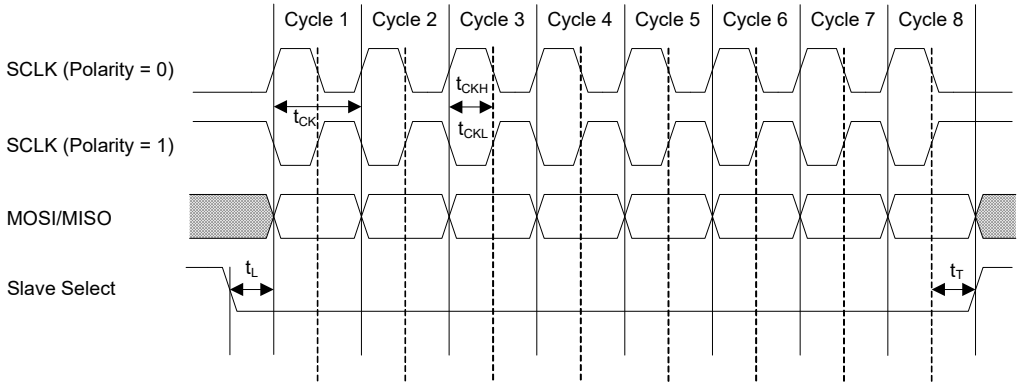
*To obtain the exact frequency, divide the 66.66 MHz clock speed by one of: 2, 4, 8, 16, 32, 64, 128, or 256.

The SPI interface has four modes of operation based on two parameters: clock polarity and clock phase. The master and slave must use the same mode to communicate properly.

BulkSPIClockPolarity controls an active high or low clock. **BulkSPIClockPhase** controls how data should be launched or captured.



Sampling Edges for Phase = 0



Sampling Edges for Phase = 1

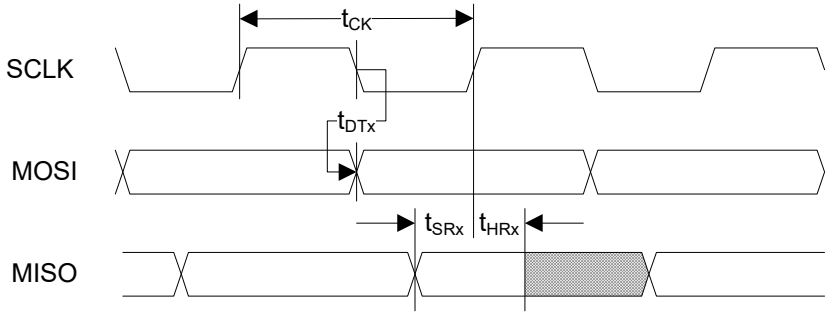


Table 22: SPI Delays

Delay	Minimum	Maximum
SPI transmit clock period t_{CK}	30 ns	3840 ns
SPI transmit clock high pulse t_{CKH}	15 ns	1920 ns
SPI transmit clock low pulse t_{CKL}	15 ns	1920 ns
Leading time from slave select assertion to first clock edge t_L	1/2 t_{CK}	1/2 t_{CK}
Trailing time from last clock edge to slave select deassertion t_T	1/2 t_{CK}	1/2 t_{CK}
SCLK to MOSI delay t_{DTx}	-12 ns	12 ns
MISO setup time t_{SRx}	22 ns	N/A
MISO hold time t_{HRx}	0 ns	N/A

High Speed Download Signals

The NTx-NBT50 Embedded Video Interface has a High Speed Download (HSD) mode that is used to provide high-speed output-only synchronous serial communication using four (DualHSD) or six (QuadHSD) wires. The master initiates the transaction by asserting slave select (SS). The master also drives a serial clock (SCLK) that provides a synchronous clock source to a slave device. The master transmits data on the SOUT lines.

The HSD interface has the following characteristics:

- Data transfer based on DualSPI and QuadSPI protocols, output-only
- Programmable transfer rate using the **BulkSystemClockDivider** feature. For more information, see "GenICam Interface for Serial Communication Configuration" on page 38.
- Maximum value: 66.66 Mbps (DualHSD), 133.32 Mbps (QuadHSD)*
- Minimum value: 0.520 Mbps (DualHSD), 1.04 Mbps (QuadHSD)
- Serial clock with programmable phase and polarity
- HSD word length: 8 bits
- Bit transmission: Most significant bit (msb) first

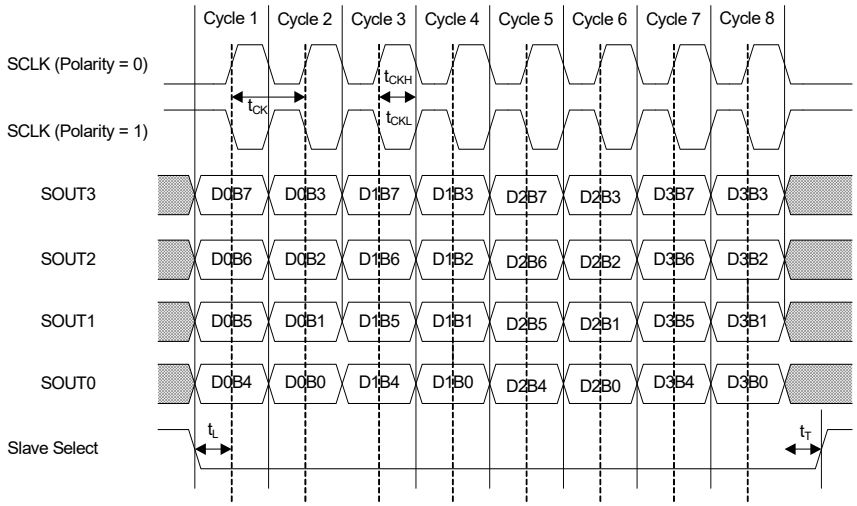
*The maximum HSD speed that can be achieved is 30 Mbps in version 5.0.0 of the eBUS SDK. An upcoming eBUS SDK release will increase the maximum throughput.

High Speed Download Timing

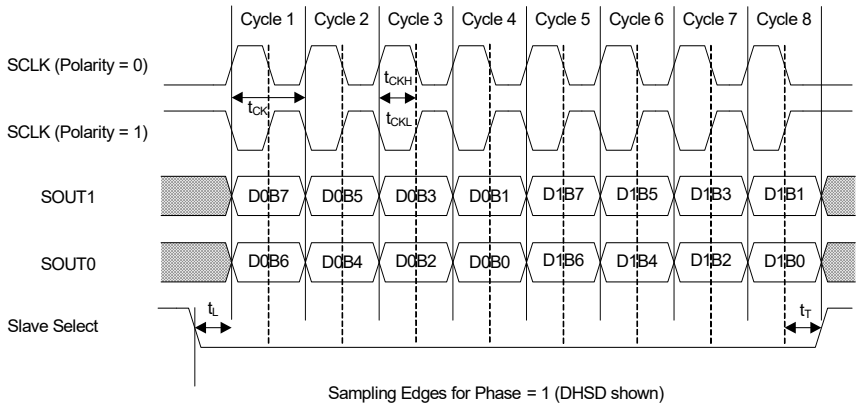
The clock frequencies and periods for the HSD clock are the same as the SPI clock frequencies. See Table 21: "Supported SPI Clock Frequencies and Periods" on page 34.

The DualHSD (DHSD) bit rate is equal to two times the selected clock frequency, and the QuadHSD (QHSD) bit rate is equal to four times the selected clock frequency.

The HSD interface has four modes of operation based on two parameters: clock polarity and clock phase. The master and slave must use the same mode to communicate properly.



Sampling Edges for Phase = 0 (QHSD shown)



Sampling Edges for Phase = 1 (DHSD shown)

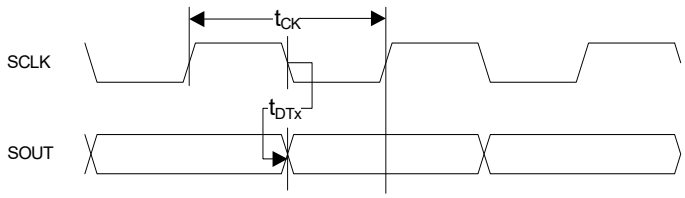


Table 23: HSD Delays

Delay	Minimum	Maximum
HSD transmit clock period t_{CK}	30 ns	3840 ns
HSD transmit clock high pulse t_{CKH}	15 ns	1920 ns
HSD transmit clock low pulse t_{CKL}	15 ns	1920 ns
Leading time from slave select assertion to first clock edge t_L	1/2 t_{CK}	1/2t_{CK}
Trailing time from last clock edge to slave select deassertion t_T	1/2 t_{CK}	1/2t_{CK}
SCLK to SOUTn delay t_{DTx}	-12 ns	12 ns

GenICam Interface for Serial Communication Configuration

The following GenICam features are available for serial communication configuration.

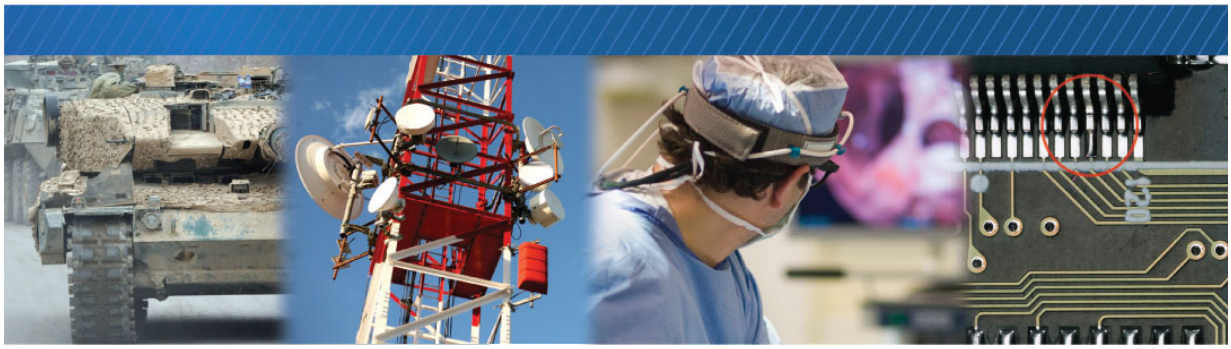
Table 24: GenICam Features Available for Serial Communication

Feature	Description
BulkSelector	Selects Bulk0, Bulk1, or Bulk2 for configuration.
BulkMode	UART/USRT/I2C/SPI/HSD protocol. Please note the following: • UART is available on all bulks • USRT is available when BulkSelector = Bulk0 or Bulk1 • I2C is available when BulkSelector = Bulk0 or Bullk1 • SPI is available when BulkSelector = Bulk0 or Bulk1 • DHSD is available when BulkSelector = Bulk0 • QHSD is available when BulkSelector = Bulk0
BulkSystemClockDivider	Defines the frequency of the USRT, SPI, DHSD, or QHSD output clock. The actual frequency produced is equal to the system clock frequency divided by the factor set by this feature. Available dividers are 2, 4, 8, 16, 32, 64, 128, and 256.
BulkOutputClockFrequency	Represents the frequency of the USRT, SPI, DHSD, or QHSD* output clock controlled by the BulkSystemClockDivider. The frequency is calculated using the following equation: • USRT: $\frac{33.33\ \text{MHz}}{\text{BulkSystemClockDivider}}$ • SPI, DHSD, and QHSD: $\frac{66.66\ \text{MHz}}{\text{BulkSystemClockDivider}}$

Table 24: GenICam Features Available for Serial Communication (Continued)

Feature	Description
BulkBaudRate	Selects a predefined baud rate or programmable option for the selected UART.
BulkBaudRateFactor	Programs a user defined baud rate for the selected UART.
BulkBaudRateValue	Displays the programmed baud rate for the selected UART.
BulkLoopback	Receives serial data sent from a host PC application to the video interface and loops it back to the host PC application. Available for UART or USRT only.
BulkNumOfStopBits	Selects a stop bit option (either 1 or 2). Available for UART or USRT only.
BulkParity	Selects a parity option (None, Even, or Odd). Available for UART or USRT only.
BulkUpstreamFifoWatermark	Sets the level of upstream FIFO at which a GigE Vision event is generated. This feature controls the number of bytes that can be accumulated in the bulk interface upstream FIFO before the embedded video interface delivers them to the host using an event type packet.
BulkSoftReset	Resets the bulk SPI, DHSD, and QHSD interface to the default settings.
BulkSPIClockPolarity	Selects the polarity of the SPI, DHSD, and QHSD clock.
BulkSPIClockPhase	Selects the phase of the SPI, DHSD, and QHSD clock.
BulkSPIWordSize	Controls the word size for a SPI transfer.

Chapter 6



Sensor Interface

The embedded video interface receives serialized pixel data and control signals from the sensor over the Low Voltage Differential Signaling (LVDS) interface, which it then deserializes.

This chapter describes the LVDS interface, including the expected pixel ordering in the sensor and LVDS timing requirements. It also describes how the deserializer is implemented in the embedded video interface.

At the end of this chapter you will find information about the pixel bus signals and timing values.

The following topics are covered in this chapter:

Low Voltage Differential Signaling (LVDS) Signals

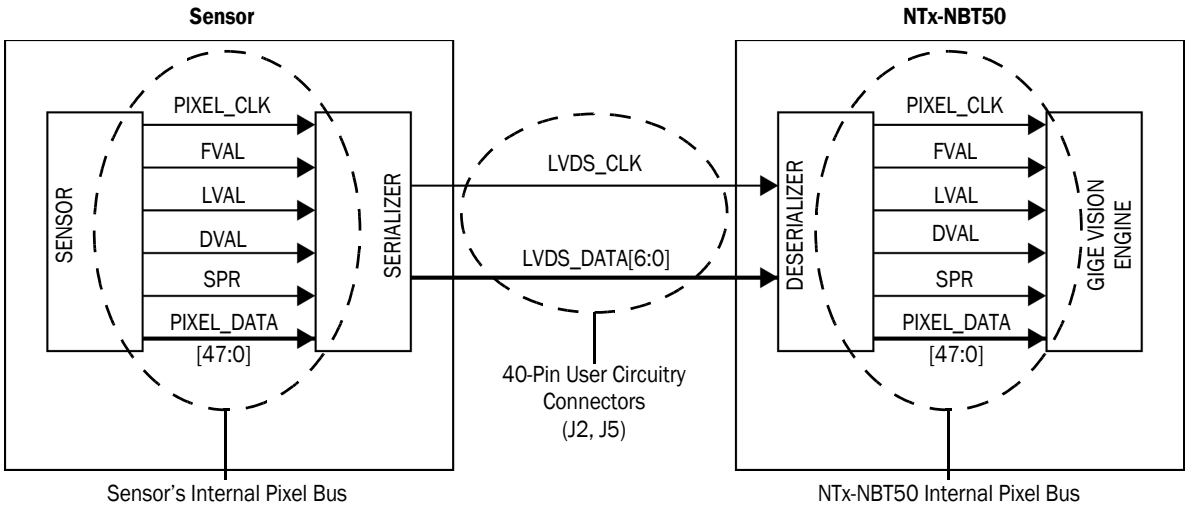
The NTx-NBT50 Embedded Video Interface pixel bus has an 8-lane LVDS source synchronous interface. This interface includes:

- 7 LVDS pairs that are used for pixel data and control signals
- 1 LVDS pair that is used for clocking

The LVDS pairs are serialized and deserialized using an 8:1 ratio. The internal pixel bus before serialization (and after deserialization) is 56 bits wide. It includes:

- 48 bits for pixel data
- 4 bits for control data (frame synchronization)
- 4 unused bits (reserved for future use, not shown)

Figure 3: Serialization and Deserialization of Sensor Data



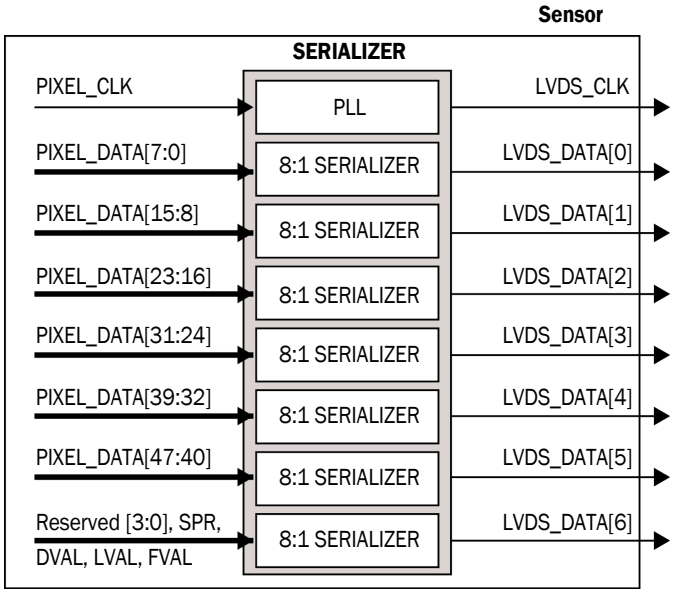
Note that the maximum parallel **PIXEL_CLK** frequency is 100 MHz, resulting in 800 Mbps LVDS data and a 100 MHz LVDS clock. As a result, the maximum pixel data throughput is 4.8 Gbps (48 bits x 100 MHz).

The pinouts for the two user circuitry interfaces are provided in "40-Pin User Circuitry Interface Connector Pinouts" on page 17.

LVDS Pixel Data Ordering

Figure 3 on page 42 shows the parallel pixel bus coming from the sensor interface to the serializer to transmit pixel data across the LVDS interface. It is important that the sensor's serializer insert the pixel bits and control signals in the order shown below to ensure that they are reconstructed correctly in the NTx-NBT50 Embedded Video Interface.

Figure 4: Pixel Ordering in the Sensor



The following timing diagram shows the bit mapping for the serialized data on the LVDS differential signals. The **LVDS_DATA[0]** to **LVDS_DATA[5]** pairs are used to transport pixel data, while the **LVDS_DATA[6]** pair is used for frame synchronization.

Figure 5: LVDS Timing

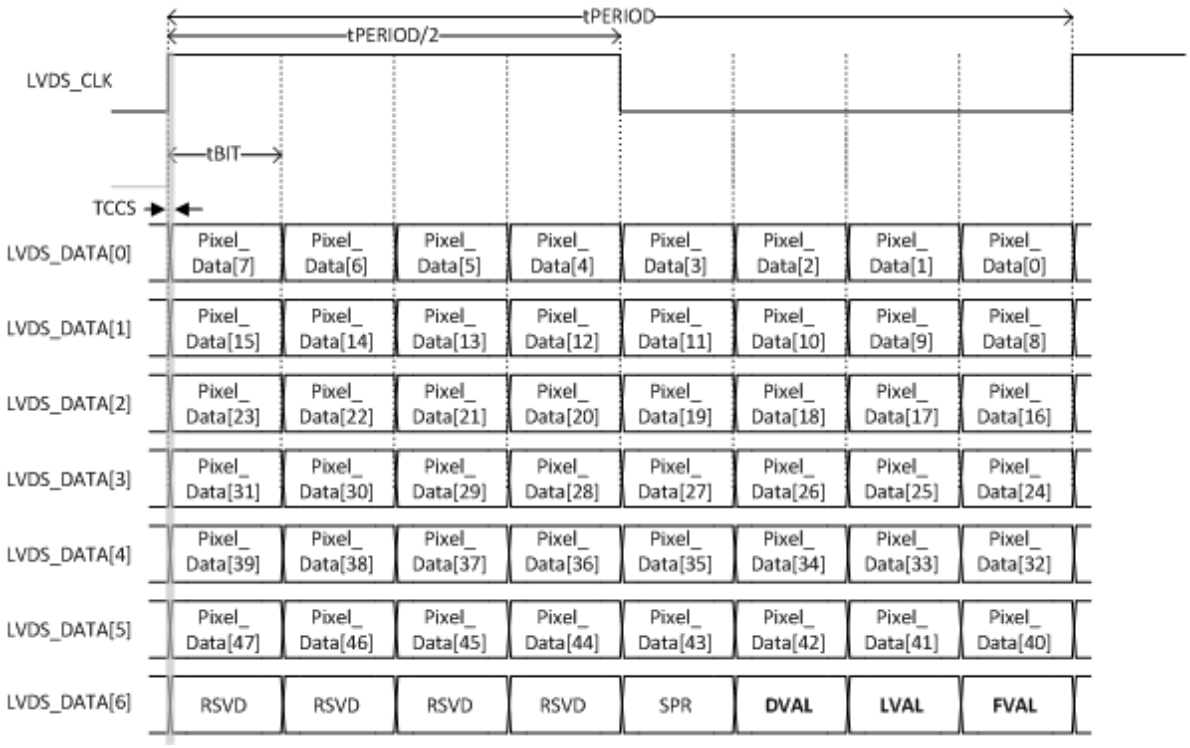


Table 25: LVDS Timing Information

Parameter	Minimum	Maximum
LVDS_CLK frequency	37.5 MHz	100 MHz
LVDS_CLK duty cycle	50%	
tPERIOD	26.7 ns	10 ns
tBIT	3.33 ns	1.25 ns
Transmitter Channel-to-Channel Skew (TCCS)	–	400 ps*

* This value represents the maximum allowed skew before user connectors (such as boards, cables, and other equipment).

LVDS Input Standard Specifications

The NTx-NBT50 Embedded Video Interface implements an LVDS deserializer inside the Cyclone V FPGA. The LVDS signals that are received from the sensor must respect the Cyclone V differential I/O standard, as specified in the following table.

Table 26: Differential I/O Standard Specifications

	V_{ID}(mV)			$V_{ICM(DC)}$(V)			V_{OD}(V)			V_{OCM}(V)		
	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max
PCML	Transmitter, receiver, and input reference clock pins of high-speed transceivers use the PCML I/O standard.											
2.5 V LVDS*	100	V_{CM} = 1.25 V	–	0.05	$D_{MAX} \leq$ 700 Mbps	1.80	0.247	–	0.6	1.125	1.25	1.375
				1.05	D_{MAX} > 700 Mbps	1.55						

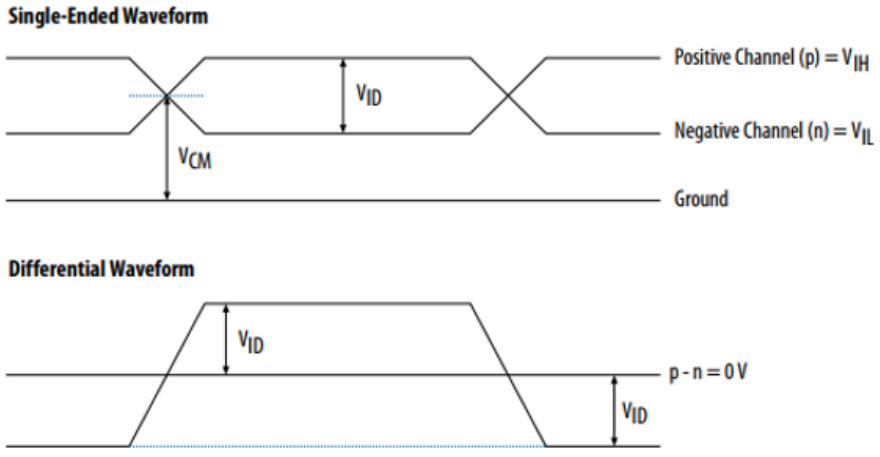
*For optimized LVDS receiver performance, the receiver voltage input range must be within 1.0 V to 1.6 V (for data rates above 700 Mbps) and 0.00 V to 1.85 V (for data rates below 700 Mbps).

Acronyms:

- VID. Input differential voltage swing. The difference in voltage between the positive and complementary conductors of a differential transmission at the receiver.
- VICM. Input common mode voltage. The common mode of the differential signal at the receiver.
- VCM(DC). DC common mode input voltage.

LVDS Differential I/O Standards

Figure 6: LVDS Single-Ended and Differential Waveforms



*Reference: Cyclone V Device Datasheet, CV-51002 (*https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/cyclone-v/cv_51002.pdf*)*

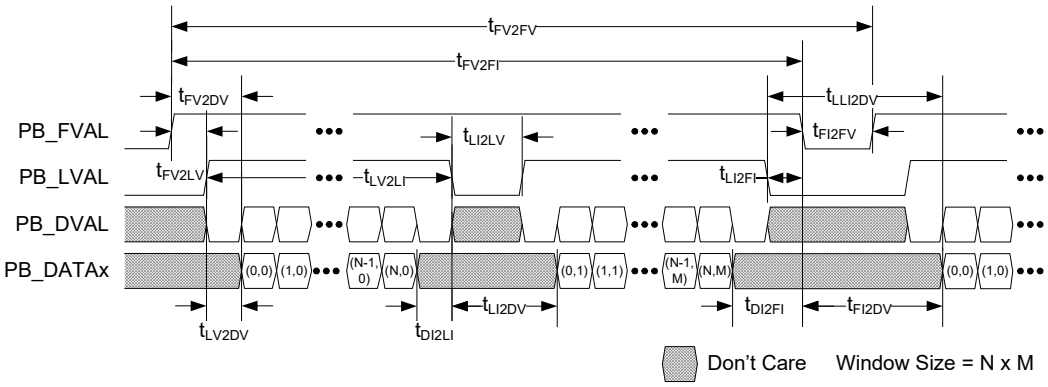
Limitations of the LVDS Clock

The LVDS clock (frequency) from the sensor should not be changed dynamically while the system is in operation. This could result in a reconfiguration of the internal deserializer and the loss of image data.

Embedded Video Interface Pixel Bus Timing

The output of the camera must match the format of the embedded video interface. You should select a case for your application and then refer to "Timing Values for All Cases" on page 49.

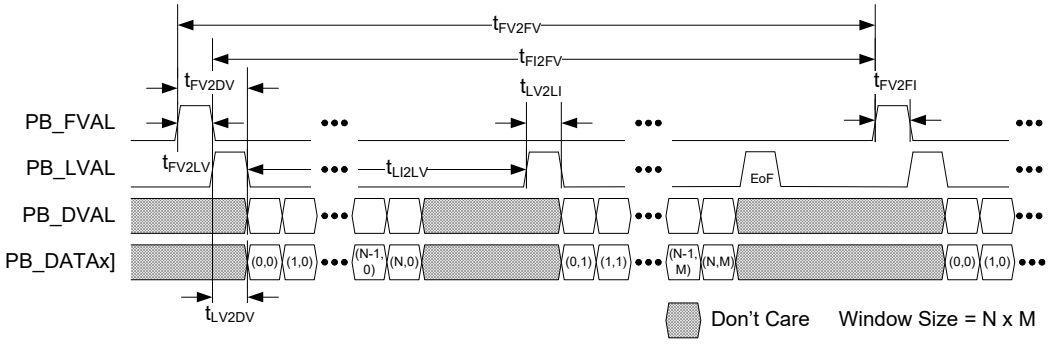
Case 1: FVAL and LVAL are Level-Sensitive



Case 2: FVAL and LVAL are Edge-Sensitive

In this case, FVAL and LVAL are edge-sensitive.

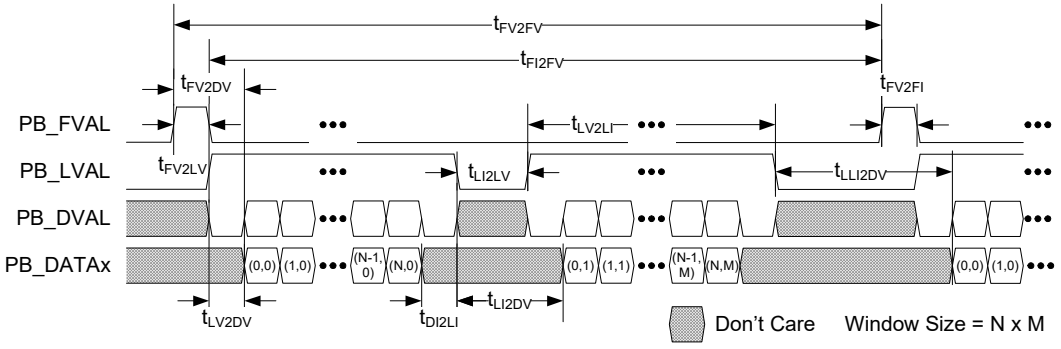
- **Start of frame/line is signaled by:** A rising (or falling) edge on FVAL, which signals the start of a *frame*. A rising (or falling) edge on LVAL, which signals the start of a *line*.
- **End of frame is signaled by:**
  - The next FVAL valid edge (rising edge when rising-edge sensitive or falling edge when falling-edge sensitive) occurs.
  - Or, when all of the pixels have been acquired (as set in the image height and width settings) **AND** an end-of-frame (EOF) occurs.
    Note: EOF occurs at LVAL rising edge (when rising-edge sensitive) or LVAL falling edge (when falling-edge sensitive). This is an additional LVAL edge, in addition to the configured/expected number of lines. See the EOF indicator in the illustration below.
- **Line Missing status and Partial Line Missing errors:** Partial Line Missing indicates lines are ending early (the next LVAL valid edge occurs before all of the pixels have been acquired). Full Line Missing indicates that the frame is ending early (the next FVAL edge occurs before all of the lines have been acquired).



Case 3: FVAL is Edge-Sensitive and LVAL is Level-Sensitive

In this case, FVAL is edge-sensitive and LVAL is level-sensitive.

- **Start of frame/line is signaled by:** A rising (or falling) edge on FVAL, which signals start of *frame*. The line is valid when LVAL is active (high or low depending on settings).
- **End of frame is signaled by:**
  - The next FVAL valid edge (rising edge when rising-edge sensitive or falling edge when falling-edge sensitive) occurs.
  - Or when all of the lines have been acquired (as set in the image height settings) **AND** the last LVAL with valid data is de-asserted (low when high level sensitive or high when low level sensitive).
- **Line Missing status or a Partial Line Missing error generated:**
  - Full Line Missing indicates that the frame is ending early (the next FVAL edge occurs before all of the lines have been acquired). Partial Line Missing indicates that lines are ending early (in this case, LVAL is de-asserted before all pixels in a line are captured).



Timing Values for All Cases

The TCP (PB_CLK period) timing values listed in the following table are minimum values only.

Table 27: TCP Timing Values for All Cases

From	To	Symbol	Case 1 (level) (t_{cp})	Case 2 (edge) (t_{cp})	Case 3 (both) (t_{cp})
FVAL valid	LVAL valid [a]	t_{FV2LV}	0 [b]	0	1
FVAL valid	Data valid [a,c,d]	t_{FV2DV}	0 [b]	16 [f]	1
LVAL valid	Data valid [a,c,d]	t_{LV2DV}	0	1	0
LVAL valid	LVAL invalid [a]	t_{LV2LI}	1	1	1
LVAL invalid	LVAL valid [a]	t_{LI2LV}	1	1	1
Data invalid	LVAL invalid [a,c,d]	t_{DI2LI}	0	N/A	0
LVAL invalid	FVAL invalid [a]	t_{LI2FI}	0 [e]	N/A	N/A
Data invalid	FVAL invalid [a,c,d]	t_{DI2FI}	0 [e]	N/A	N/A
FVAL invalid	FVAL valid [a]	t_{FI2FV}	1	1	1
FVAL invalid	Data valid [a,c,d]	t_{FI2DV}	1	N/A	N/A
Last LVAL invalid	Data valid	t_{LLI2DV}	16 [f]	N/A	16 [f]
FVAL valid	FVAL invalid	t_{FV2FI}	16 [f]	1	1
FVAL valid	FVAL valid	t_{2FV2FV}	17 [f]	17 [f]	17 [f]

a. The valid state of FVAL and LVAL is high when they are set as level-high sensitive or rising-edge sensitive. Their valid state is low when they are set as level-low sensitive or falling-edge sensitive.

b. If LVAL is valid before FVAL becomes valid, the grabber drops the full line.

c. Data valid is defined by FVAL valid (note a), LVAL valid (note a), and DVAL valid (note d).

d. The valid state of DVAL is high when it is set as level-high sensitive, and low when set as level-low sensitive. DVAL is always valid in the grabber when the **PixelBusDataValidEnabled** feature is off.

e. If FVAL becomes invalid and LVAL is still valid, the line is truncated.

f. This is a worst-case value. Subtract 3 cycles if the pixel type is 8-bit, 1-tap. Subtract 1 cycle for all other pixel types except 10/12- bit, 2-tap, unpacked, and RGB unpacked. Subtract up to 7 cycles if the image size is a multiple of 32 bytes.

Pixel Bus Bit Map

The following table shows how the pixel bus signals are ordered, based on the available pixel formats.

Mono/Bayer/RGB/BGR Color Filter

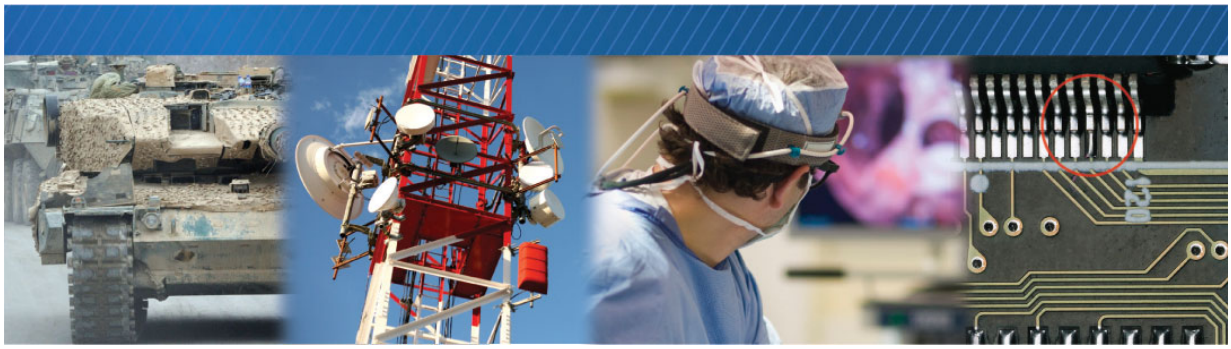
Table 28: Mono/Bayer/RGB/BGR Color Filter

	Mono8/ Bayer8		Mono10/ Bayer10		Mono12/ Bayer12		Mono14		Mono16 / Bayer16		RGB8			BGR8		
	Tap	Bit	Tap	Bit	Tap	Bit	Tap	Bit	Tap	Bit	Tap	Comp.	Bit	Tap	Comp.	Bit
PB0_DATA00	0	0	0	0	0	0	0	0	0	0	0	R	0	0	B	0
PB0_DATA01	0	1	0	1	0	1	0	1	0	1	0	R	1	0	B	1
PB0_DATA02	0	2	0	2	0	2	0	2	0	2	0	R	2	0	B	2
PB0_DATA03	0	3	0	3	0	3	0	3	0	3	0	R	3	0	B	3
PB0_DATA04	0	4	0	4	0	4	0	4	0	4	0	R	4	0	B	4
PB0_DATA05	0	5	0	5	0	5	0	5	0	5	0	R	5	0	B	5
PB0_DATA06	0	6	0	6	0	6	0	6	0	6	0	R	6	0	B	6
PB0_DATA07	0	7	0	7	0	7	0	7	0	7	0	R	7	0	B	7
PB0_DATA08	1	0	0	8	0	8	0	8	0	8	0	G	0	0	G	0
PB0_DATA09	1	1	0	9	0	9	0	9	0	9	0	G	1	0	G	1
PB0_DATA10	1	2	nc	nc	0	10	0	10	0	10	0	G	2	0	G	2
PB0_DATA11	1	3	nc	nc	0	11	0	11	0	11	0	G	3	0	G	3
PB0_DATA12	1	4	1	8	1	8	0	12	0	12	0	G	4	0	G	4
PB0_DATA13	1	5	1	9	1	9	0	13	0	13	0	G	5	0	G	5
PB0_DATA14	1	6	nc	nc	1	10	nc	nc	0	14	0	G	6	0	G	6
PB0_DATA15	1	7	nc	nc	1	11	nc	nc	0	15	0	G	7	0	G	7
PB0_DATA16	2	0	1	0	1	0	nc	nc	1	0	0	B	0	0	R	0
PB0_DATA17	2	1	1	1	1	1	nc	nc	1	1	0	B	1	0	R	1
PB0_DATA18	2	2	1	2	1	2	nc	nc	1	2	0	B	2	0	R	2
PB0_DATA19	2	3	1	3	1	3	nc	nc	1	3	0	B	3	0	R	3
PB0_DATA20	2	4	1	4	1	4	nc	nc	1	4	0	B	4	0	R	4
PB0_DATA21	2	5	1	5	1	5	nc	nc	1	5	0	B	5	0	R	5
PB0_DATA22	2	6	1	6	1	6	nc	nc	1	6	0	B	6	0	R	6
PB0_DATA23	2	7	1	7	1	7	nc	nc	1	7	0	B	7	0	R	7
PB0_DATA24	3	0	3	0	3	0	nc	nc	1	8	1	R	0	1	B	0
PB0_DATA25	3	1	3	1	3	1	nc	nc	1	9	1	R	1	1	B	1

Table 28: Mono/Bayer/RGB/BGR Color Filter (Continued)

	Mono8/ Bayer8		Mono10/ Bayer10		Mono12/ Bayer12		Mono14		Mono16 / Bayer16		RGB8			BGR8		
	Tap	Bit	Tap	Bit	Tap	Bit	Tap	Bit	Tap	Bit	Tap	Comp.	Bit	Tap	Comp.	Bit
PB0_DATA26	3	2	3	2	3	2	nc	nc	1	10	1	R	2	1	B	2
PB0_DATA27	3	3	3	3	3	3	nc	nc	1	11	1	R	3	1	B	3
PB0_DATA28	3	4	3	4	3	4	nc	nc	1	12	1	R	4	1	B	4
PB0_DATA29	3	5	3	5	3	5	nc	nc	1	13	1	R	5	1	B	5
PB0_DATA30	3	6	3	6	3	6	nc	nc	1	14	1	R	6	1	B	6
PB0_DATA31	3	7	3	7	3	7	nc	nc	1	15	1	R	7	1	B	7
PB0_DATA32	nc	nc	2	0	2	0	nc	nc	nc	nc	1	G	0	1	G	0
PB0_DATA33	nc	nc	2	1	2	1	nc	nc	nc	nc	1	G	1	1	G	1
PB0_DATA34	nc	nc	2	2	2	2	nc	nc	nc	nc	1	G	2	1	G	2
PB0_DATA35	nc	nc	2	3	2	3	nc	nc	nc	nc	1	G	3	1	G	3
PB0_DATA36	nc	nc	2	4	2	4	nc	nc	nc	nc	1	G	4	1	G	4
PB0_DATA37	nc	nc	2	5	2	5	nc	nc	nc	nc	1	G	5	1	G	5
PB0_DATA38	nc	nc	2	6	2	6	nc	nc	nc	nc	1	G	6	1	G	6
PB0_DATA39	nc	nc	2	7	2	7	nc	nc	nc	nc	1	G	7	1	G	7
PB0_DATA40	nc	nc	2	8	2	8	nc	nc	nc	nc	1	B	0	1	R	0
PB0_DATA41	nc	nc	2	9	2	9	nc	nc	nc	nc	1	B	1	1	R	1
PB0_DATA42	nc	nc	nc	nc	2	10	nc	nc	nc	nc	1	B	2	1	R	2
PB0_DATA43	nc	nc	nc	nc	2	11	nc	nc	nc	nc	1	B	3	1	R	3
PB0_DATA44	nc	nc	3	8	3	8	nc	nc	nc	nc	1	B	4	1	R	4
PB0_DATA45	nc	nc	3	9	3	9	nc	nc	nc	nc	1	B	5	1	R	5
PB0_DATA46	nc	nc	nc	nc	3	10	nc	nc	nc	nc	1	B	6	1	R	6
PB0_DATA47	nc	nc	nc	nc	3	11	nc	nc	nc	nc	1	B	7	1	R	7

Chapter 7



Signal Handling

The embedded video interface includes a programmable logic controller (PLC) that lets you control external machines and react to inputs. By controlling your system using the PLC, you can make functional changes, adjust timing, or add features without having to add new hardware.

PLC Programming Signals

For an introduction to the PLC and for detailed information about how PLC signals are handled, see the *iPORT Advanced Features User Guide*, available on the Pleora Support Center at www.pleora.com.

The following table lists the PLC input and output programming signals that are specific to the embedded video interface, and indicates the pins on which they are available.

Table 29: PLC Signal Usage

Signal name	PLC equation usage	Associated pin
Pb0Fval	Input	Pin 11 on the J2 40-pin user circuitry connector, PB0_FVAL
Pb0Lval	Input	Pin 10 on the J2 40-pin user circuitry connector, PB0_LVAL
Pb0Dval	Input	Pin 12 on the J2 40-pin user circuitry connector, PB0_DVAL
Pb0Spare	Input	Pin 9 on the J2 40-pin user circuitry connector, PB0_MVAL[a]
GpioIn0	Input	Pin 31 on the J2 40-pin user circuitry connector, FPGA_GPIO_IN0
GpioIn1	Input	Pin 35 on the J2 40-pin user circuitry connector, FPGA_GPIO_IN1
GpioIn2	Input	Pin 22 on the J2 40-pin user circuitry connector, FPGA_GPIO_IN2
GpioIn3	Input	Pin 29 on the J2 40-pin user circuitry connector, FPGA_GPIO_IN3
BufferWM0	Input	No associated pin
Grb0AcqActive	Input	No associated pin

Table 29: PLC Signal Usage (Continued)

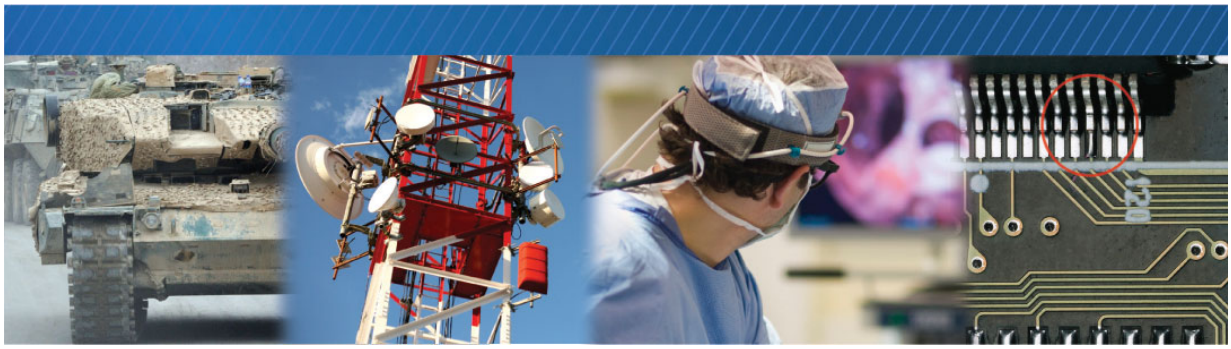
Signal name	PLC equation usage	Associated pin
PlcCtrl0	Input	No associated pin
PlcCtrl1	Input	No associated pin
PlcCtrl2	Input	No associated pin
PlcCtrl3	Input	No associated pin
Pb0CC0	Input, output	Pin 18 on the J2 40-pin user circuitry connector, PB0_CTRL_OUT0
Pb0CC1	Input, output	Pin 23 on the J2 40-pin user circuitry connector, PB0_CTRL_OUT1
Pb0CC2	Input, output	Pin 24 on the J2 40-pin user circuitry connector, PB0_CTRL_OUT2
Pb0CC3	Input, output	Pin 25 on the J2 40-pin user circuitry connector, PB0_CTRL_OUT3
GpioOut0	Input, output	Pin 32 on the J2 40-pin user circuitry connector, FPGA_GPIO_OUT0
GpioOut1	Input, output	Pin 30 on the J2 40-pin user circuitry connector, FPGA_GPIO_OUT1
GpioOut2	Input, output	Pin 26 on the J2 40-pin user circuitry connector, FPGA_GPIO_OUT2
PlcFval0	Input, output	No associated pin
PlcLval0	Input, output	No associated pin
PlcMval0	Input, output	No associated pin
PlcTrig0	Input, output	No associated pin
PlcTimestampCtrl	Input, output	No associated pin
Timer0Trig	Input, output	No associated pin
Timer0Out	Input	No associated pin
Timer1Trig	Input, output	No associated pin
Timer1Out	Input	No associated pin
Counter0Reset	Input, output	No associated pin
Counter0Inc	Input, output	No associated pin
Counter0Dec	Input, output	No associated pin
Counter0Eq	Input	No associated pin
Counter0Gt	Input	No associated pin
Counter1Reset	Input, output	No associated pin
Counter1Inc	Input, output	No associated pin
Counter1Dec	Input, output	No associated pin
Counter1Eq	Input	No associated pin
Counter1Gt	Input	No associated pin
Rescaler0In	Input, output	No associated pin
Rescaler0Out	Input	No associated pin

Table 29: PLC Signal Usage (Continued)

Signal name	PLC equation usage	Associated pin
Delayer0In	Input, output	No associated pin
Delayer0Out	Input	No associated pin
Event0	Input, output	No associated pin
Event1	Input, output	No associated pin
Event2	Input, output	No associated pin
Event3	Input, output	No associated pin
ActionTrig0	Input	No associated pin
ActionTrig1	Input	No associated pin

a. When chunk data is in use, this signal is Metadata Valid (MVAL). For more information about chunk data and MVAL, see the *iPORT Advanced Features User Guide*, available on the Pleora Support Center at www.pleora.com.

Chapter 8



Installing the eBUS SDK

This chapter describes how to install the eBUS SDK, and also provides information about installing the required driver.

Before you can configure and control your embedded video interface, you must install the eBUS SDK.

The following topics are covered in this chapter:

Installing the eBUS SDK

You can install the Pleora Technologies eBUS SDK on your computer to configure and control your embedded video interface.

The eBUS SDK contains an extensive library of sample applications, with source code, to create working applications for device configuration and control, image and data acquisition, and image display and diagnostics.

It is possible for you to configure the embedded video interface and GigE Vision compliant video sources using other GenICam compliant software; however, this guide provides you with the instructions you need to use the Pleora eBUS Player application.

Installing the Driver

The eBUS SDK includes a GigE Vision driver that enhances existing general-purpose drivers shipped with NICs, increases image acquisition throughput and performance, decreases latency and jitter, and minimizes CPU utilization.

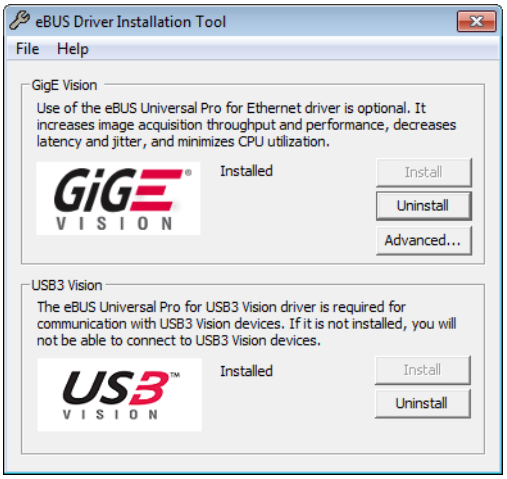
The USB3 Vision driver, which is available during the installation of the eBUS SDK, is for use with USB3 Vision compliant devices, such as the Pleora NTx-U3 Embedded Video Interface.

The drivers are selected for installation by default during the eBUS SDK installation process. If you choose not to install the drivers (or want to uninstall either driver), you can use the eBUS Driver Installation Tool.

To use the eBUS Driver Installation Tool

1. Click **Start** > **All Programs** > **Pleora Technologies Inc** > **eBUS SDK** > **eBUS Driver Installation Tool**.
2. Under **GigE Vision**, click **Install** or **Uninstall**.

   After a moment, the driver status changes. If you are installing a driver, the driver is installed across all network adapters on your computer.

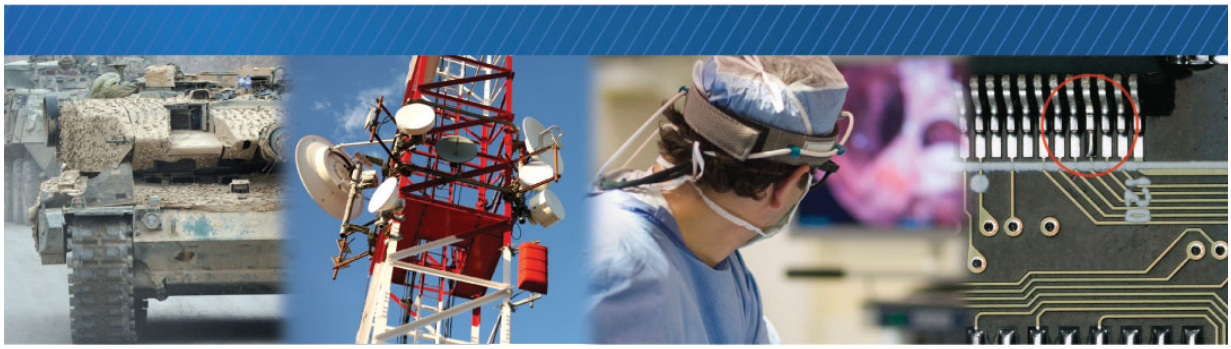


3. Close the eBUS Driver Installation Tool.

   You may be required to restart your computer.

To see the versions of the installed drivers, click **Help** > **About**.

Chapter 9



Configuring Your Computer's NIC for use with the NTx-NBT50 Embedded Video Interface

This chapter explains how to configure your computer's NIC for communication with the NTx-NBT50 Embedded Video Interface.

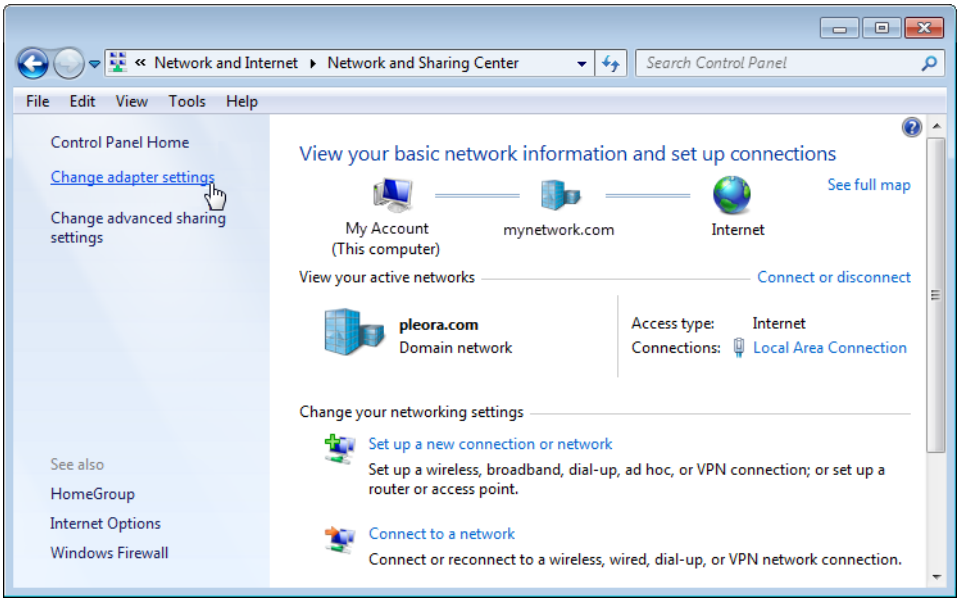
For optimal performance, we recommend that you enable jumbo packets (also known as jumbo frames) and set the receive descriptors to the maximum available value.

The instructions in this section are based on the Windows 7 operating system. The steps may vary depending on your computer's operating system.

To configure the NIC for optimal performance

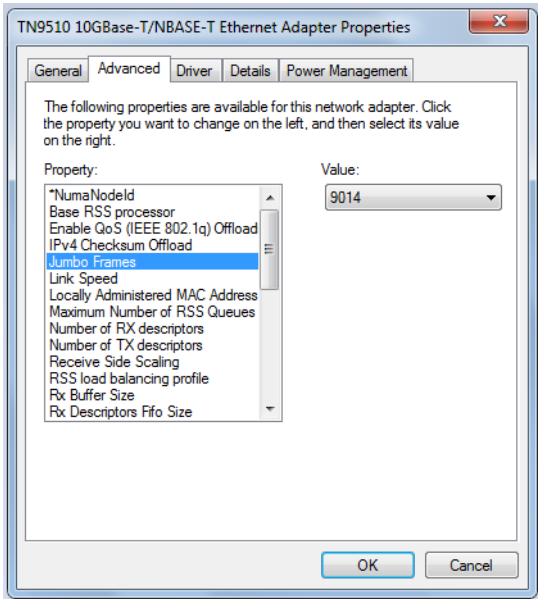
1. In the Windows Control Panel, click **Network and Internet**.

2. Click **Network and Sharing Center**.
3. In the left-hand panel, click **Change adapter settings**.



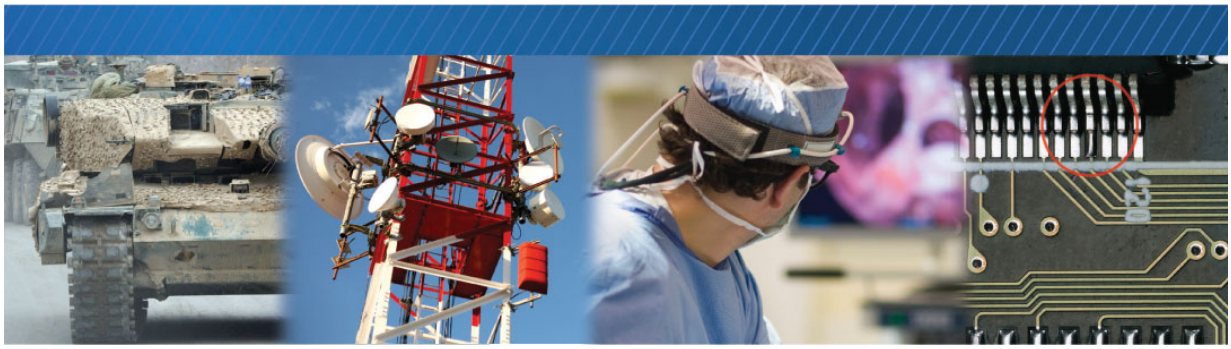
4. Configure the NIC for jumbo packets (often referred to as jumbo frames) and set the NIC's **Receive Buffers** (**Receive Descriptors**) to the maximum available value. Using jumbo packets allows you to increase system performance. However, you must ensure your NIC and GigE switch (if applicable) support jumbo packets.

   To complete this task, right-click the NIC and click **Properties**. Then, click **Configure**. The exact configuration procedure, as well as the jumbo packet size limit, depends on the NIC.



5. Close the open dialog boxes to apply the changes and close the Control Panel.

Chapter 10



Connecting to the Embedded Video Interface and Configuring General Settings

After you have set up the physical connections to the embedded video interface, you can start the eBUS Player application to configure image settings to ensure images are received and displayed properly. You can also configure the buffer options to reduce the likelihood of lost packets.

eBUS Player is documented in more detail in the *eBUS Player User Guide*. The *iPORT NTx-NBT50 Embedded Video Interface User Guide* only provides you with high-level eBUS Player instructions to help you set up your device and to start streaming video. The *eBUS Player User Guide* is available on the Pleora Support Center (https://supportcenter.pleora.com/).

The following topics are covered in this chapter:

Confirming Image Streaming

The embedded video interface can communicate with your computer using either a direct connection or by connecting to a GigE or NBASE-T switch.

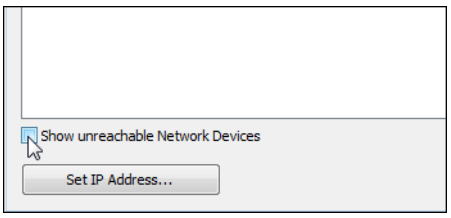
To connect the cables and apply power

- Connect the embedded video interface to the RJ-45 Ethernet connector on your computer's NIC, a GigE switch, or an NBASE-T switch. Then, apply power.

To start eBUS Player and connect to a device

1. Start eBUS Player from the Windows **Start** menu.
2. Click **Select/Connect**.

   If the NTx-NBT50 Embedded Video Interface does not appear in the list, click the **Show unreachable Network Devices** check box to show all devices.



3. In the **Device Selection** dialog box, click the embedded video interface.

If the IP address is not compatible with your NIC, a warning ( ) appears in the **Device Selection** dialog box. Provide the device with an IP address, as outlined in "Providing the NTx-NBT50 Embedded Video Interface with an IP Address" on page 68.

4. Click **OK**.

   eBUS Player is now connected to the embedded video interface.

To confirm image streaming

1. Click **Play** to stream live images or the test pattern.

   For information about using the test pattern, see "To turn the test pattern on or off" on page 69.
2. After you confirm that images are streaming, click **Stop**.

If images do not stream, see the tips provided in "System Troubleshooting" on page 91.

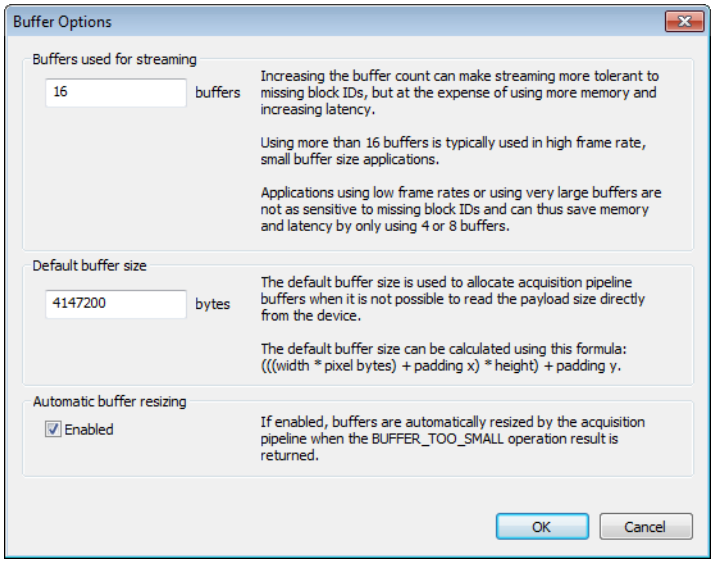
Configuring the Buffers

You can increase the buffer count using eBUS Player to make streaming more robust. A high number of buffers are needed in high frame rate applications, while a small number of buffers are needed for lower frame rates. Latency increases as the number of buffers increases.

To configure the buffers

1. Start eBUS Player and connect to the embedded video interface.
   For more information, see "To start eBUS Player and connect to a device" on page 66.
2. Click **Tools** > **Buffer Options**.
3. Click the buffer option that suits your requirements.
4. Click **OK**.

Default size for streaming is 16 buffers.



Providing the NTx-NBT50 Embedded Video Interface with an IP Address

The NTx-NBT50 Embedded Video Interface requires an IP address to communicate on a video network. This address must be on the same subnet as the computer that is performing the configuration and receiving the image stream.

To provide the NTx-NBT50 Embedded Video Interface with an IP address

1. Start eBUS Player.
2. Click **Select/Connect**.
3. Click the NTx-NBT50 Embedded Video Interface.
4. Click **Set IP Address**.
5. Provide the embedded video interface with a compatible IP address and subnet mask. You can optionally provide a default gateway.

   Note that this information is temporary and is reset when you power down the device. To set an IP address that is used permanently, see the next procedure in this guide.

If you are using a unicast network configuration, the management entity/data receiver and the embedded video interface must be on the same subnet. The unicast network configuration is outlined in "Unicast Network Configuration" on page 84.

6. Click **OK** to close the **Set IP Address** dialog box.
7. Click **OK** to close the **Device Selection** dialog box and connect to the device.

Configuring an Automatic/Persistent IP Address

The Device Control dialog box allows you to configure a persistent IP address for the NTx-NBT50 Embedded Video Interface. Alternatively, the embedded video interface can be configured to automatically obtain an IP address using Dynamic Host Configuration Protocol (DHCP) or Link Local Addressing (LLA). The embedded video interface uses its persistent IP address first, but if this option is set to **False**, it can be configured to attempt to obtain an address from a DHCP server. If this fails, it will use LLA to find an available IP address. LLA cannot be disabled and is always set to **True**.

The device can use the persistent IP address each time it is powered up as long as the IP address is valid and there are no IP address conflicts

To configure a persistent IP address

1. Start eBUS Player and connect to the NTx-NBT50 Embedded Video Interface.

   For more information, see "To start eBUS Player and connect to a device" on page 66.
2. Under **Parameters and Controls**, click **Device control**.
3. Under **TransportLayerControl**, set the **GevCurrentIPConfigurationPersistentIP** feature to **True**.

4. Set the **GevPersistentIPAddress** feature to a valid IP address in the **GevPersistentIPAddress** field.
5. Set the **GevPersistentSubnetMask** feature to a valid subnet mask address.
6. Optionally, enter a valid default gateway in the **GevPersistentDefaultGateway** field.
7. Close the **Device Control** dialog box.
8. Power cycle the embedded video interface.

To automatically configure an IP address

1. Start eBUS Player and connect to the NTx-NBT50 Embedded Video Interface.
   For more information, see "To start eBUS Player and connect to a device" on page 66.
2. Under **Parameters and Controls**, click **Device control**.
3. Under **TransportLayerControl**, set the **GevCurrentIPConfigurationPersistentIP** feature to **False**.
4. Set the **GevCurrentIPConfigurationLLA** and/or **GevCurrentIPConfigurationDHCP** values to **True**, depending on the type of automatic addressing you require.
5. Close the **Device Control** dialog box.
6. Power cycle the embedded video interface.

Configuring the Embedded Video Interface's Image Settings

You can configure the embedded video interface's image settings, which provide the embedded video interface with information about the image coming from the camera. These settings allow the images to appear correctly.

The image settings are located under **ImageFormatControl** in the **Device Control** dialog box.

Changes that you make to the embedded video interface are not persisted across power cycles, unless you use the **UserSetSave** feature. For information about saving settings to the embedded video interface's flash memory, see the *eBUS Player User Guide*, available on the Pleora Support Center.

To turn the test pattern on or off

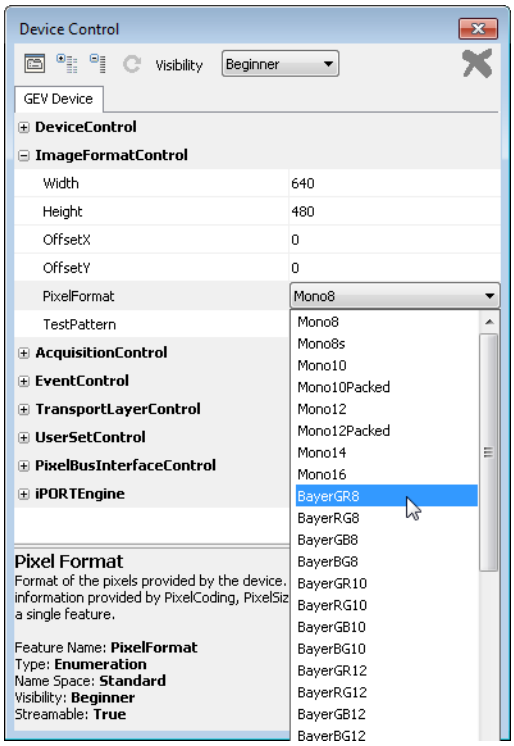
1. Start eBUS Player and connect to the embedded video interface.
   For more information, see "To start eBUS Player and connect to a device" on page 66.
2. Under **Parameters and Controls**, click **Device control**.
3. Under **ImageFormatControl**, click a test pattern option in the list.
4. Close the **Device Control** dialog box.

To reduce the amount of bandwidth that is used when transmitting the test pattern, increase the OffsetX and OffsetY values. Increasing OffsetX reduces the bandwidth by a factor of two. Increasing both OffsetX and OffsetY reduces the bandwidth by a factor of 4. An OffsetX and OffsetY of 0 will require the most bandwidth.

To change the pixel format

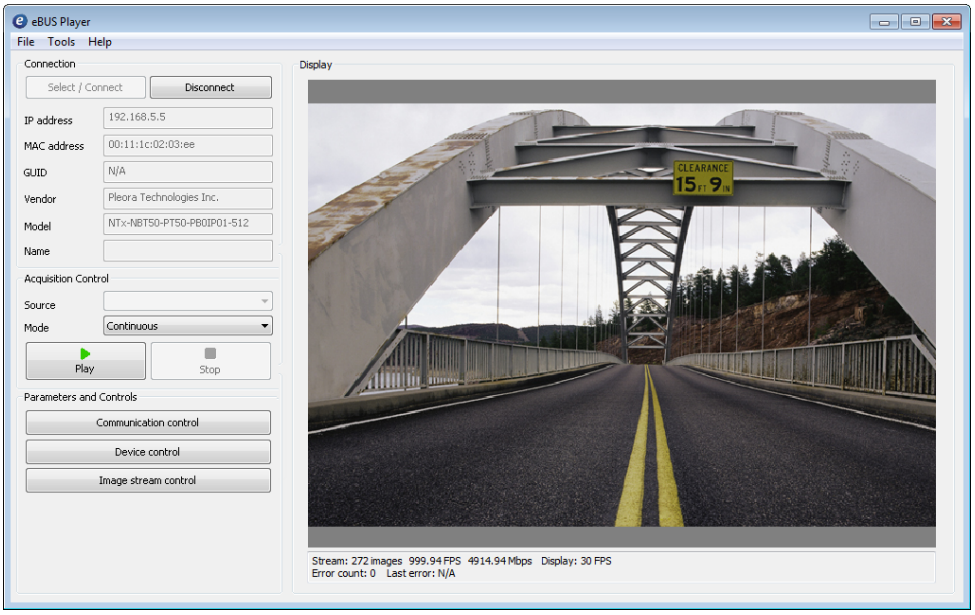
1. Start eBUS Player and connect to the embedded video interface.

   For more information, see "To start eBUS Player and connect to a device" on page 66.
2. If images are streaming, click the **Stop** button.
3. Under **Parameters and Controls**, click **Device control**.
4. Under **ImageFormatControl**, choose the pixel format that matches your camera's configuration.



5. Close the **Device Control** dialog box.

6. Click **Play** to see the changes.



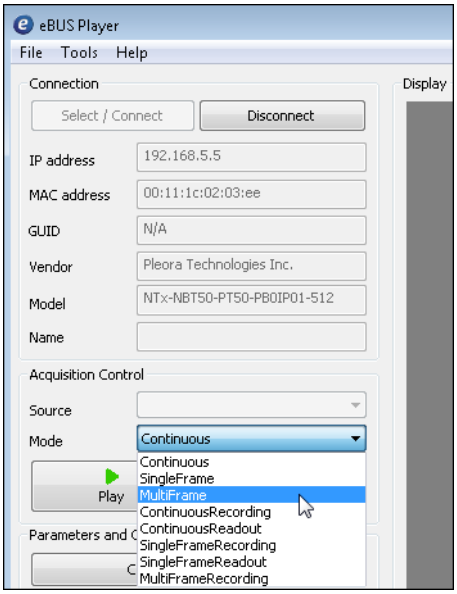
To configure the image width and height

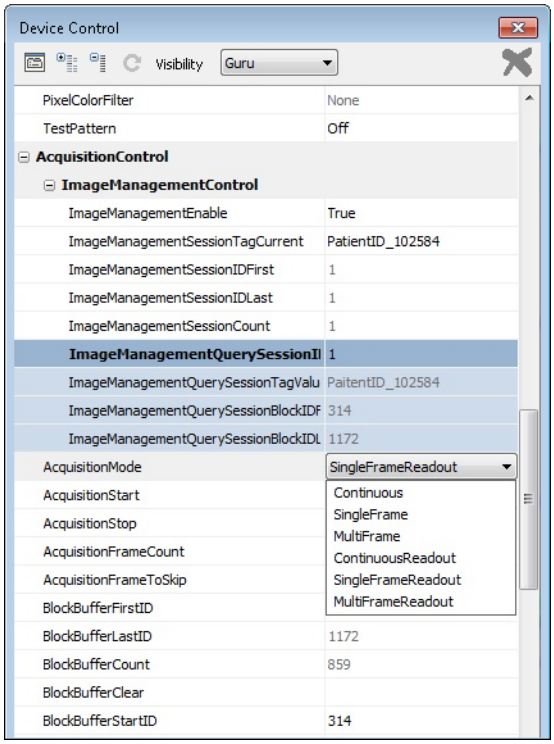
1. Start eBUS Player and connect to the embedded video interface.

   For more information, see "To start eBUS Player and connect to a device" on page 66.
2. If images are streaming, click the **Stop** button.
3. Under **Parameters and Controls**, click **Device control**.
4. Under **ImageFormatControl**, change the **Width** and **Height** to suit your camera.
5. Close the **Device Control** dialog box.

Specifying How Images are Acquired

Continuous, **SingleFrame**, and **MultiFrame** modes are usually standard for embedded video interfaces. Acquisition starts when the **Play** button is pressed (the **AcquisitionStart** command is executed).





Continuous Mode

This mode allows you to acquire images continuously and is the default mode for most embedded video interfaces.

Multiframe Mode

This mode allows you to acquire a fixed number of images. To configure the number of images, set the embedded video interface's **AcquisitionControl** > **AcquisitionFrameCount** feature.

You can set the **AcquisitionControl** > **AcquisitionFrameCount** feature in the **Device Control** dialog box.

SingleFrame Mode

This mode allows you to acquire one image at a time.

Recording and Retrieving Images in the Onboard Memory

In addition to the standard acquisition modes listed in the previous section, the embedded video interface also has recording and readout modes.

When using the image management feature, the readout modes operate in a slightly different manner. This section describes how these modes operate when **AcquisitionControl** > **ImageManagementEnable** feature is set to **False**. For information about using the readout modes when image management is enabled, see "Managing and Retrieving Images Related to an Acquisition Session" on page 76.

The **recording** acquisition modes allow you to capture images from a camera and store them in the embedded video interface's onboard memory.

The **readout** acquisition modes allow images to be retrieved from the device's memory at a slower rate, ensuring images are not lost.

These modes are helpful when you are working with a camera that transmits images at a rate that exceeds the connection between the embedded video interface Pleora embedded video interface and the computer, resulting in dropped images. By using the recording and readout modes in this example, you can capture and stream images from the camera without losing any images (as long as there is space in the onboard memory).

The recording acquisition modes support back-to-back recording, which allows you to click the **Stop** and **Play** buttons multiple consecutive times without clearing the onboard memory.

Acquisition starts when the **Play** button is pressed (the **AcquisitionStart** command is executed) and when one of the recording modes is selected.

Images can be stored in the embedded video interface's onboard memory as long as there is space or until there are 4096 images in memory. For information about calculating how many images you can store, see "Calculating How Many Images Can be Stored in Onboard Memory" on page 75.

Continuous Recording Mode

With this mode, images are acquired continuously and are stored in the device's onboard memory until the memory is full (or 4096 images are stored in onboard memory). When this limit is reached, the embedded video interface stops acquiring new images from the camera.

We recommend that you observe **AcquisitionControl** > **BlockBufferCount** in eBUS Player's **Device Control** dialog box (**Expert** or **Guru** visibility level is required). When the value for this feature stops increasing, the memory is full. For information about the actions that clear the images from onboard memory, see "Understanding When Images are Removed from the Onboard Memory" on page 75.

Continuous Readout Mode

With this mode, images are continuously read (and removed) from the device's onboard memory. The readout begins at the first image in memory. To see the number of images stored in onboard memory, see **AcquisitionControl** > **BlockBufferCount** in eBUS Player's **Device Control** dialog box (**Expert** or **Guru** visibility level is required).

Readout continues until the **Stop** button is pressed (**AcquisitionStop** command is executed) or until the last image has been sent by the device (**BlockBufferCount** will be **0**).

Multi Frame Recording Mode

With this mode, a fixed number of images are stored in the device's onboard memory. To configure the number of images, set the **AcquisitionControl** > **AcquisitionFrameCount** feature in eBUS Player's **Device Control** dialog box. Images can be read out from memory using **Continuous Readout mode**.

A maximum of 255 images can be acquired at one time in Multi Frame Recording mode.

To determine how many images can be stored in memory, see "Calculating How Many Images Can be Stored in Onboard Memory" on page 75.

If **AcquisitionControl** > **AcquisitionFrameCount** is set to a value that exceeds the amount of available memory, the embedded video interface stops acquiring new images when the onboard memory is full (or 4096 images are stored in onboard memory).

BlockBufferCount shows the number of images currently in memory. In **Multi Frame Recording mode**, this number is cumulative: If the memory is empty and you acquire an image, **BlockBufferCount** will match the **AcquisitionFrameCount**. If you stop and restart recording, **BlockBufferCount** will increment (to a maximum of 4096 images, depending on the image size) and will no longer match the **AcquisitionFrameCount**.

Single Frame Recording Mode

With this mode, a single image is saved in the embedded video interface's onboard memory after each **AcquisitionStart** command.

Single Frame Readout Mode

With this mode, only a single image is read out from the device's memory at a slower rate, ensuring images are not lost. This mode is helpful when you are working with a camera that transmits images at a rate that exceeds the connection between the embedded video interface and the computer, resulting in dropped images.

Understanding When Images are Removed from the Onboard Memory

When the **ImageManagementEnable** feature is set to **True**, image acquisition operates in a slightly different manner, with regard to when images are removed. For more information, see "Managing and Retrieving Images Related to an Acquisition Session" on page 76.

The following actions remove the images from the embedded video interface's onboard memory:

- Streaming images from the onboard memory using one of the readout acquisition modes (**ContinuousReadout** or **SingleFrameReadout**).
- Power cycling the device, which clears all images from the onboard memory.
- Making any of the following **AcquisitionMode** changes and then clicking the **Play** button (**AcquisitionStart** command):

First you acquire images with...	And then you change the Acquisition mode to...
ContinuousRecording, MultiFrameRecording, or SingleFrameRecording	Continuous, MultiFrame, or SingleFrame
SingleFrameReadout or ContinuousReadout	SingleFrame, MultiFrame, or Continuous
SingleFrameReadout or ContinuousReadout	ContinuousRecording, MultiFrameRecording, or SingleFrameRecording

Calculating How Many Images Can be Stored in Onboard Memory

First, take note of the **PayloadSize**, which appears under **TransportLayerControl** in eBUS Player's **Device Control** dialog box. **Expert** or **Guru** visibility level is required to access this feature.

The **PayloadSize** is automatically calculated by the device based on the selected image settings, which include **Width**, **Height**, **OffsetX**, **OffsetY**, **PixelSize**, any chunk data, as well as any padding that has to be added to the image payload.

For example, for a device configured to use Mono8 with images that are 1920 x 1080, the **PayloadSize** is equal to 2 073 600 bytes per image or about 1.978 MB (2 073 600 / 1 048 576).

After determining **PayloadSize**, you can use the following equation to determine the number of images that can be saved in onboard memory:

Available onboard memory MB / PayloadSize MB = Number of images that can be saved

Using our example, the equation is:

503.744 MB / 1.978 MB = 254 images

Managing and Retrieving Images Related to an Acquisition Session

In some systems you may be interested in keeping track of images that were transmitted during a particular acquisition session, in case you need to retransmit them to a receiving computer. This can be useful in systems where you want images to be available in the event that the receiving computer experiences a malfunction or a loss of power or when the embedded video interface experiences a loss of power.

The image management feature lets you specify a unique session tag before you start acquiring images, which indicates that the images belong to the same session. It also provides contextual information that will help you identify the session so you can retrieve it from the embedded video interface's memory.

As you continue to start new sessions (which requires that you specify a new unique session tag to mark the beginning of a new session), a directory structure forms within the onboard memory and you will have a selection of acquisition sessions that you can retrieve.

Note that the sessions and associated images are stored in the device's onboard memory and are not copied to non-volatile storage. A loss of power on the embedded video interface will result in the loss of the directory and all images.

To keep track of images captured during a session, complete the following tasks, which are described in detail later in this section:

- Enable image management.
- Specify an alphanumeric session tag that will be used to identify the session. All of the images with this tag are considered to be a part of the same session. When you begin a different session, you can use a different session tag to start a new session.
- Choose an acquisition mode (Continuous, SingleFrame, or MultiFrame) and start image acquisition.
- After streaming starts, the group of images are tagged with an automatically generated session ID number (which is associated with the user-defined session tag).

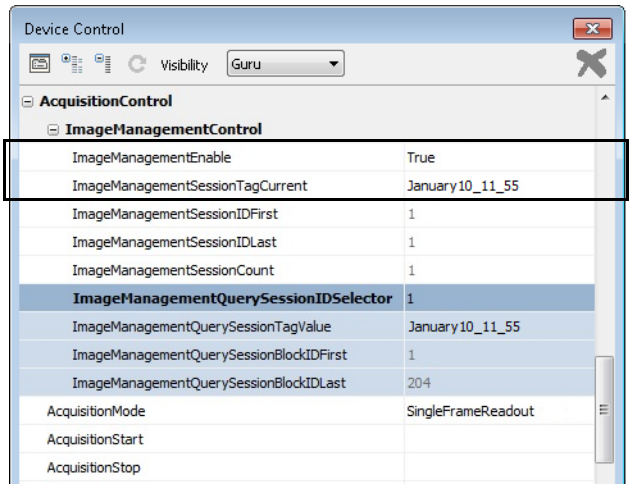
  Note: When there is no more space to store images or there are 4096 images in memory, the oldest images will start being overwritten with new ones.
- Retrieve the images by selecting a readout acquisition mode and starting a stream. You can retrieve the first or last image in the session, or an image with a particular block ID.

When image management is enabled, image acquisition operates in a slightly different manner than described previously in this guide, with regard to when images are deleted:

- Reading out images from memory does not cause the images to be deleted. Images can be read out of memory as many times as required.
- Changing from one acquisition mode to another does not cause the images to be deleted.
- Images remain in memory until the memory is explicitly cleared (by executing BlockBufferClear) or the embedded video interface is restarted or power cycled. When the memory is full, the oldest images are overwritten with new images.

To enable image management and specify a session name

1. Start eBUS Player and click **Select/Connect**.
2. Click the device in the **Available Devices** list.
3. Click **OK** in the bottom right corner.
4. Under **Parameters and Controls** click **Device control**.
5. Under **AcquisitionControl** > **Image Management Control**, set the **ImageManagementEnable** feature to **True**.
6. Beside the **ImageManagementSessionTagCurrent** feature, enter a session name.



7. Close the **Device Control** dialog box.

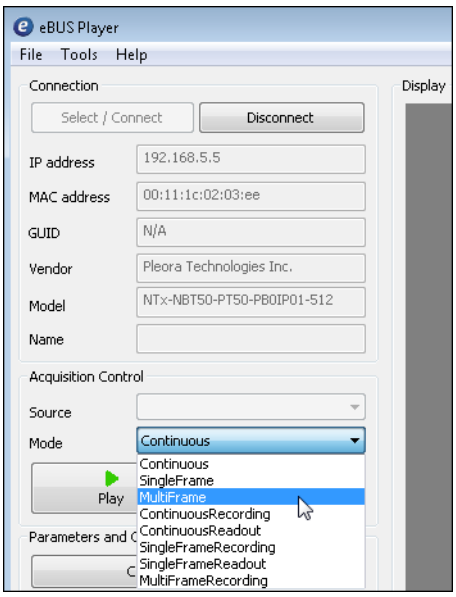
Images that were captured before **ImageManagementEnable** was set to **True** are not available for retransmission.

To capture images for the acquisition session

1. Ensure you have enabled image management and specified a session name to identify your session, as explained in the previous procedure.

2. On the main screen of eBUS Player, select one of the following acquisition modes in the **Mode** list: **Continuous**, **SingleFrame**, or **MultiFrame**.

   Note: The acquisition modes are described in "Specifying How Images are Acquired" on page 72.

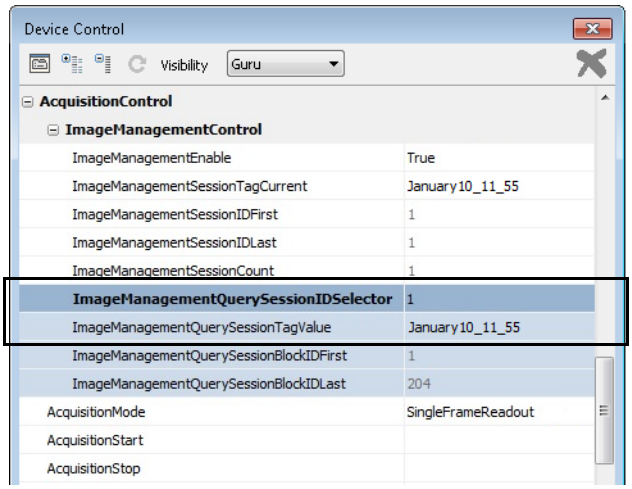


3. Click **Play** to acquire images.
4. When the session is complete, click **Stop**.

To retrieve images from an acquisition session

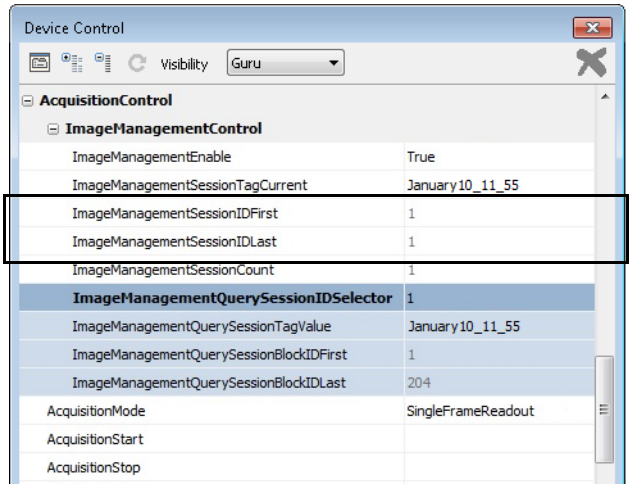
1. Under **Parameters and Controls** click **Device control**.
2. Under **Acquisition Control** > **Image Management Control**, verify that **ImageManagementQuerySessionTagValue** shows the session that you are interested in.

   If you want to choose a different session, change the **ImageManagementQuerySessionIDSelector** value.

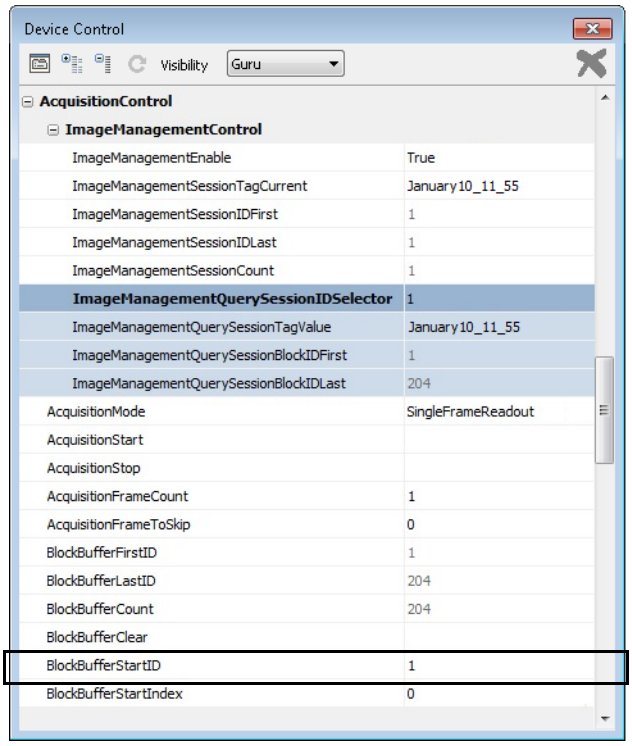


3. Take note of the values in the following fields, depending on whether you want to retrieve the first or last image from the session:
   - **ImageManagementQuerySessionBlockIDFirst**. Reports the Block ID of the first block for the selected session.
   - **ImageManagementQuerySessionBlockIDLast**. Reports the Block ID of the last block for the selected session.

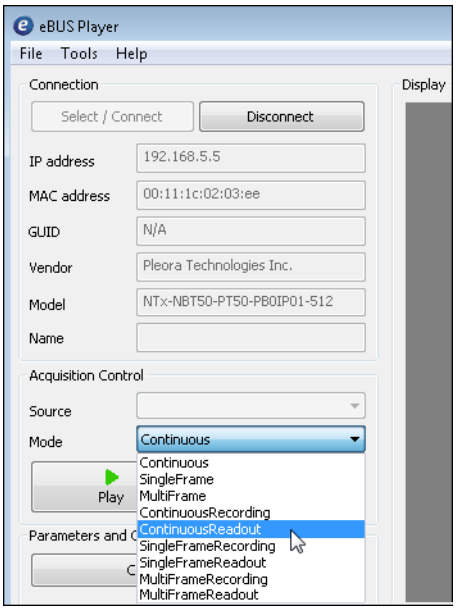
   Note that these values are only updated after you stop acquisition.



4. Set **BlockBufferStartID** to the value from **ImageManagementQuerySessionBlockIDFirst** or **ImageManagementQuerySessionBlockIDLast**, or a particular block ID within this range.



5. On the main screen of eBUS Player, select **ContinuousReadout, SingleFrameReadout, or MultiFrameReadout** in the **Mode** list.



6. Click **Play** to retrieve (readout) the images. When the session is complete, click **Stop**.

   Note: You can retrieve the images as many times as you want. If you are using **Single Frame Readout** mode, re-enter the block ID beside the **BlockBufferStartID** feature and click **Play**.

To clear the embedded or video interface memory

- Under **ImageAcquisition**, click **BlockBufferClear**.

Resetting or power cycling the embedded video interface will also remove any recorded images from memory.

Reference: Image Management Features

Table 30: Image Management Features

Name	Description, notes
BlockBufferStartID	Sets the starting Block ID in the Video Interface memory for acquisition readout. The range of Block ID is bounded by the **BlockBufferFirstID** and **BlockBufferLastID**.
BlockBufferStartIndex	Sets the starting index of block(s) in the Video Interface memory for acquisition readout. An index of 0 refers to the oldest block in the Video Interface memory.
BlockBufferFirstID	Reports the block ID of the first block in the Video Interface memory. Returns 0 when the memory is empty.

Table 30: Image Management Features (Continued)

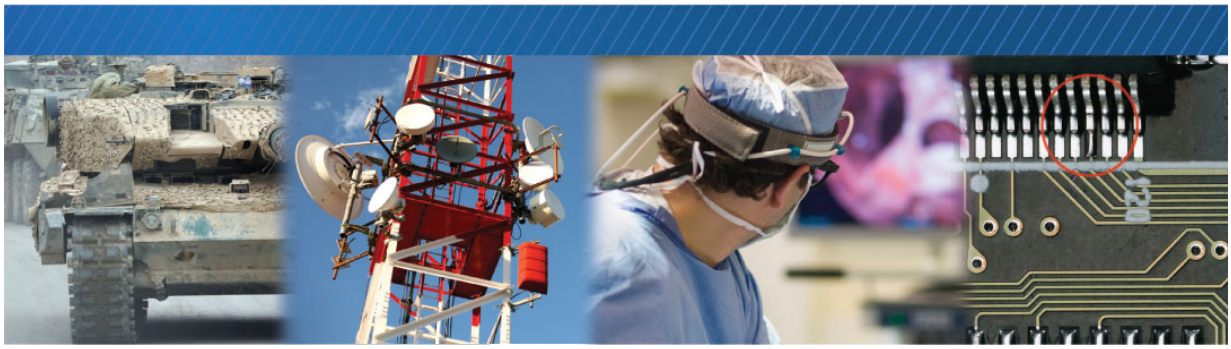
Name	Description, notes
BlockBufferLastID	Reports the block ID of the last complete block in the Video Interface memory. Returns 0 when the memory is empty.
BlockBufferClear	Resets the acquisition path which includes the logic handling of the acquisition, the streaming and all Block ID pointers. Available when **ImageManagementEnable** = **True**.
ImageManagementEnable	Enables the Image Management feature. Setting the value to **False** will reset all Image Management features to the default settings.
ImageManagementSessionTagCurrent	User-defined Session Tag for the current acquisition session.
ImageManagementSessionIDFirst	Reports the Session ID of the first acquisition session in the Video Interface memory.
ImageManagementSessionIDLast	Reports the Session ID of the last acquisition session in the Video Interface memory.
ImageManagementSessionCount	Reports the number of acquisition sessions available in the Video Interface memory.
ImageManagementQuerySessionIDSelector	Selects the Session ID to query information about an acquisition session. The Session ID range is specified by the **ImageManagementSessionIDFirst** and **ImageManagementSessionIDLast**.
ImageManagementQuerySessionTagValue	Reports the Session Tag associated with the selected Session ID in the **ImageManagementQuerySessionIDSelector**.
ImageManagementQuerySessionBlockIDFirst	Reports the Block ID of the first block in the selected Session ID in the **ImageManagementQuerySessionIDSelector**.
ImageManagementQuerySessionBlockIDLast	Reports the Block ID of the last block in the selected Session ID in the **ImageManagementQuerySessionIDSelector**.

Implementing the eBUS SDK

You can create your own image acquisition software for the embedded video interface. Consult the following guides for information about creating custom image acquisition software:

- *eBUS SDK API Quick Start Guides*, available for C++, .NET, Linux, and macOS
- *eBUS SDK API Help Files*

Chapter 11



Network Configurations for the NTx-NBT50 Embedded Video Interface

After you have connected to the NTx-NBT50 Embedded Video Interface and provided it with a unique IP address on your network, you can configure the embedded video interface for either unicast or multicast.

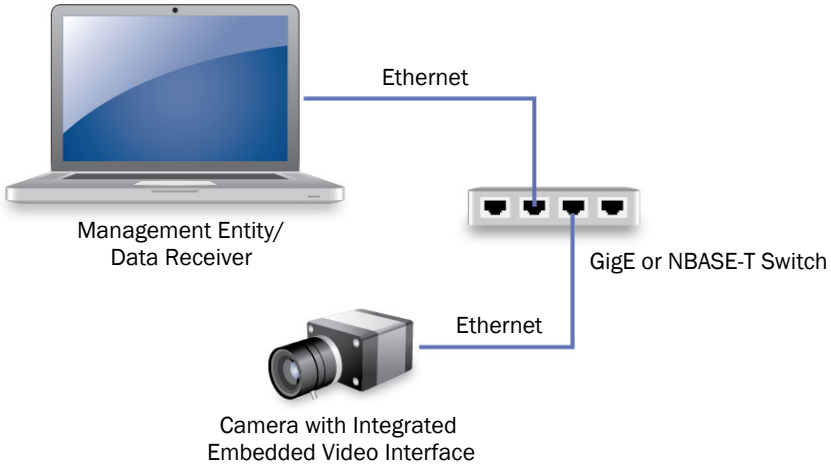
The following topics are covered in this chapter:

Unicast Network Configuration

In a unicast configuration, an NTx-NBT50 Embedded Video Interface is connected to a GigE or NBASE-T switch that sends a stream of images over Ethernet to the computer. Alternatively, the embedded video interface can be connected directly to the computer.

The computer is configured as both a data receiver and controller, and serves as a management entity for the embedded video interface.

Figure 7: Unicast Network Configuration



Required Items — Unicast Network Configuration

You require the following components to set up a unicast network configuration:

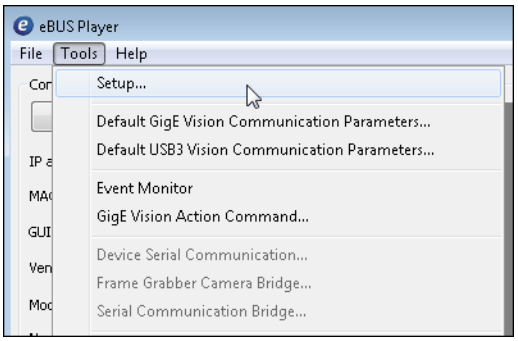
- Imaging device with integrated NTx-NBT50 Embedded Video Interface and cables
- CAT5e or CAT6 Ethernet cable (quantity: 1)
- NBASE-T compatible GigE or NBASE-T switch and an additional CAT5e or CAT6 Ethernet cable (optional)
- Desktop computer or laptop with eBUS SDK, version 5.0 (or later) installed, and NBASE-T Ethernet ports, which are required for optimal performance

Embedded Video Interface Configuration — Unicast Network Configuration

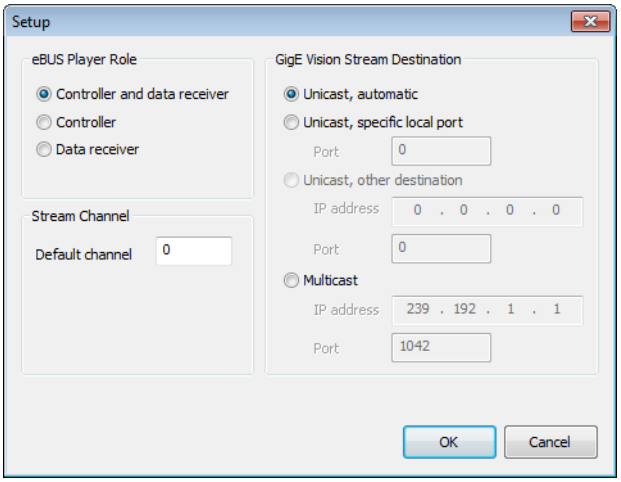
After you have connected and applied power to the hardware components, use eBUS Player to configure the NTx-NBT50 Embedded Video Interface for a unicast network configuration.

To configure the embedded video interface for a unicast network configuration

1. Start **eBUS Player**.
2. Click **Tools > Setup**.



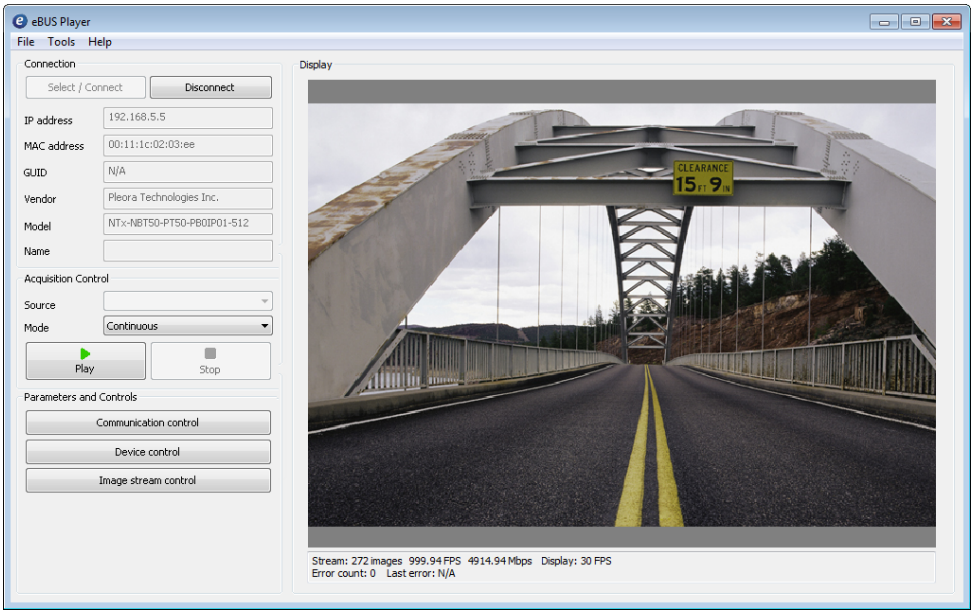
3. Under **eBUS Player Role**, click **Controller and data receiver**.



4. Under **GigE Vision Stream Destination**, click **Unicast, automatic**.

5. Click **OK**.
6. Connect to the NTx-NBT50 Embedded Video Interface.

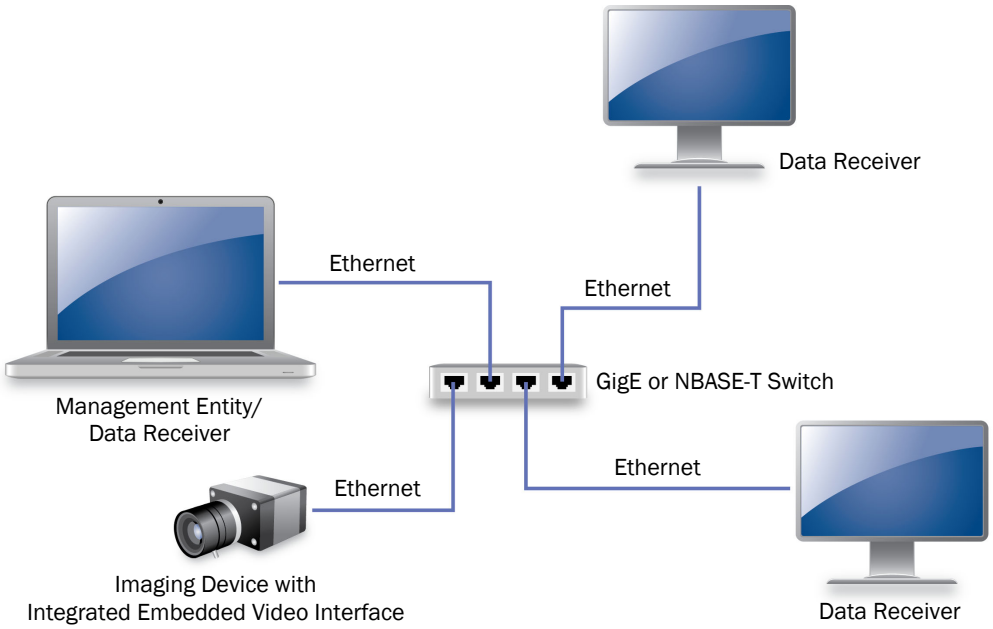
   For more information, see "To start eBUS Player and connect to a device" on page 66.
7. Click **Play** to view a live image stream.



Multicast Network Configuration

In a multicast network configuration, the NTx-NBT50 Embedded Video Interface is connected to a GigE or NBASE-T switch, and sends a stream of images over Ethernet simultaneously to multiple receiving computers.

Figure 8: Multicast Network Configuration



Required Items — Multicast Network Configuration

You require the following components to set up a multicast network configuration:

- Imaging device with integrated NTx-NBT50 Embedded Video Interface and cables
- CAT5e or CAT6 Ethernet cables
- GigE or NBASE-T switch
- 1 desktop or laptop computer* to manage the embedded video interface and receive the multicast image stream
- 1 or more desktop or laptop computers* to receive the multicast image stream

*Ensure the computers are running eBUS SDK, version 5.0 (or later)

Connecting the Devices

This section lists the physical connections that must be made.

To connect the device

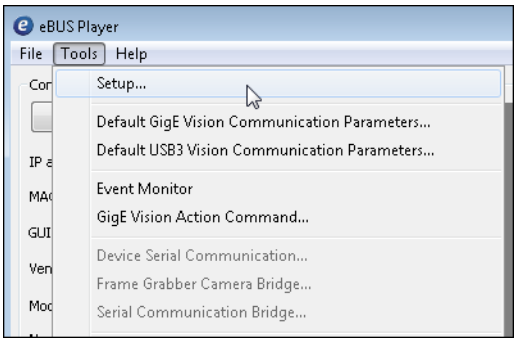
1. Using CAT5e/CAT6 cables, connect the computers to available ports on the GigE or NBASE-T switch.
2. Using a CAT5e/CAT6 cable, connect the iPORT NTx-NBT50 Embedded Video Interface to an available port on the GigE or NBASE-T switch.
3. Apply power to the devices.

Configuring the Devices for a Multicast Network Configuration

After you have connected and applied power to the hardware components, use eBUS Player to configure the iPORT NTx-NBT50 Embedded Video Interface and receiving computers for multicast configuration.

To configure a receiving computer to receive the multicast image stream

1. On the computer that you want to receive the multicast image stream, start **eBUS Player**.
2. Click **Tools** > **Setup**.



3. Under **eBUS Player Role**, click **Data receiver**.
4. Under **GigE Vision Stream Destination**, click **Multicast** and enter the **IP address** and **Port**. For example, 239.192.1.1.
5. Click **OK**.

   At this moment, the **Display** section of eBUS Player will be inactive because the image stream has not yet been multicast out from the embedded video interface.

   Note: Repeat this procedure for other computers that you want to receive the multicast image stream.

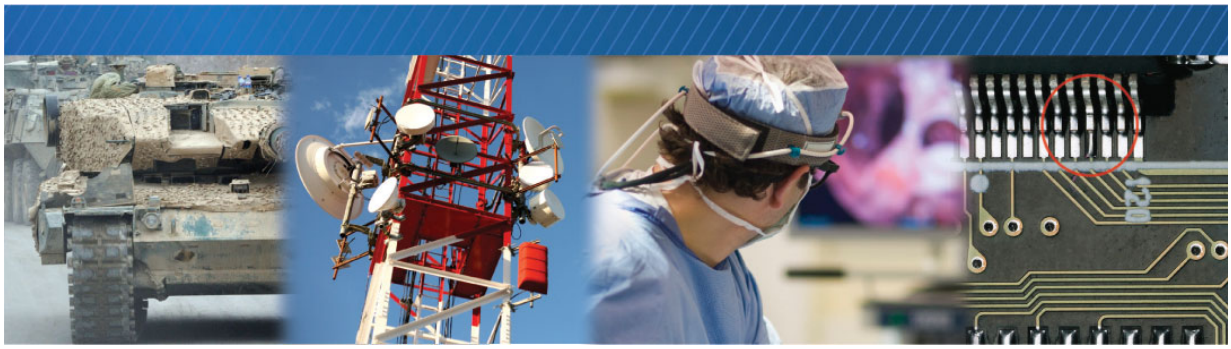
To configure the embedded video interface to multicast the image stream to receiving computers

1. On the computer that you are using to manage the embedded video interface, start **eBUS Player**.
2. Click **Tools > Setup**.
3. Under **eBUS Player Role**, click **Controller and data receiver**.
4. Under **GigE Vision Stream Destination**, click **Multicast** and enter the **IP address** and **Port**. The IP address and port must be identical to that configured for the receiving computer(s) in step 4 of "To configure a receiving computer to receive the multicast image stream" on page 88.
5. Click **OK**.
6. Connect to the embedded video interface.

   For more information, see "To start eBUS Player and connect to a device" on page 66.
7. Click **Play** to receive the image stream.

   The image stream will be multicast out from the embedded video interface and will appear automatically in eBUS Player on the receiving computers.

Chapter 12



System Troubleshooting

This chapter provides you with troubleshooting tips and recommended solutions for issues that can occur during configuration, setup, and operation of the NTx-NBT50 Embedded Video Interface. It also shows you how to switch between the backup and main firmware loads.

Not all scenarios and solutions are listed here. You can refer to the Pleora Technologies Support Center at www.pleora.com for additional support and assistance. Details for creating a customer account are available on the Pleora Technologies Support Center.

Refer to the product release notes that are available on the Pleora Technologies Support Center for known issues and other product features.

Troubleshooting Tips

The scenarios and known issues listed in this chapter are those that you might encounter during the setup and operation of your embedded video interface. Not all possible scenarios and errors are presented. The symptoms, possible causes, and resolutions depend upon your particular network, setup, and operation.

If you perform the resolution for your issue and the issue is not corrected, we recommend you review the other resolutions listed in this table. Some symptoms may be interrelated.

Table 31: Troubleshooting Tips

Symptom	Possible cause	Resolution
SDK cannot detect or connect to the embedded video interface	Power not supplied to the embedded video interface, or inadequate power supplied	Both the detection and connection to the embedded video interface will fail if adequate power is not supplied to the device. Re-try the connection to the device with eBUS Player. Verify that the status LED (D1) is on. For information about the status LED, see "Status LEDs" on page 23.
	Device is not connected to the network	Verify that the network activity LED and network connection speed LED (J6) are active. If these LEDs are illuminated, check the LEDs on your network switch to ensure the switch is functioning properly. If the problem continues, connect the embedded video interface directly to the computer to verify its operation. For information about the network LEDs, "Status LEDs" on page 23.
	The embedded video interface and computer are not on the same subnet	Images might not appear in eBUS Player if the embedded video interface and the computer running eBUS Player are not on the same subnet. Ensure that these devices are on the same subnet. In addition, ensure that these devices are connected using valid gateway and subnet mask information. You can view the embedded video interface IP address information in the **Available Devices** list in eBUS Player. A red icon appears beside the device if there is an invalid IP configuration.

Table 31: Troubleshooting Tips (Continued)

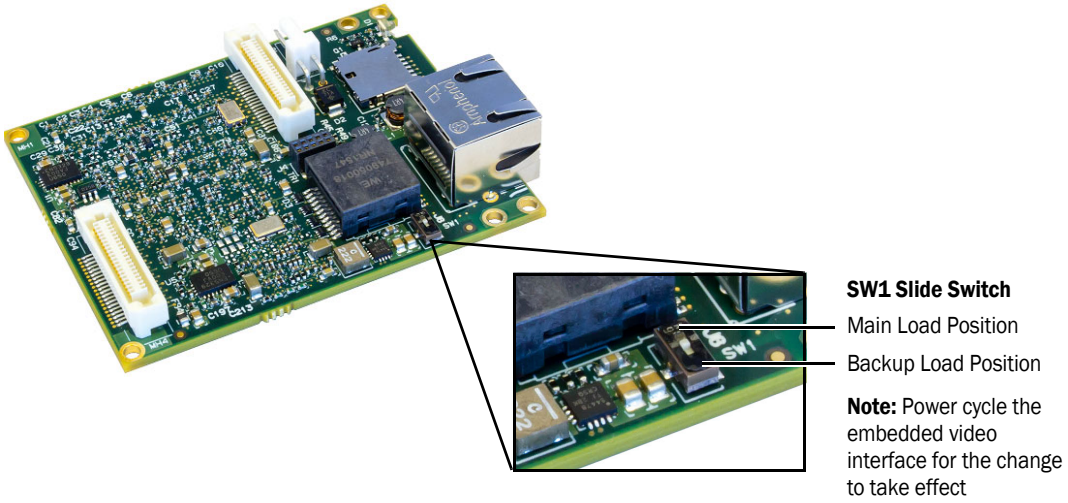
Symptom	Possible cause	Resolution
SDK is able to connect, but no images appear in eBUS Player.	In a multicast configuration, the embedded video interface may not be configured correctly	Images might not appear on the display if you have not configured the embedded video interface for a multicast network configuration. The embedded video interface and all multicast receivers must have identical values for both the **GevSCDA** and **GevSCPHostPort** features in the **TransportLayerControl** section. For more information, see "Multicast Network Configuration" on page 87.
	In a multicast configuration, your computer's firewall may be blocking eBUS Player	Ensure that eBUS Player is allowed to communicate through the firewall.
	Anti-virus software or firewalls blocking transmission	Images might not appear in eBUS Player because of anti-virus software or firewalls on your network. Disable all virus scanning software and firewalls, and re-attempt a connection to the embedded video interface with eBUS Player.

Changing to the Backup Firmware Load

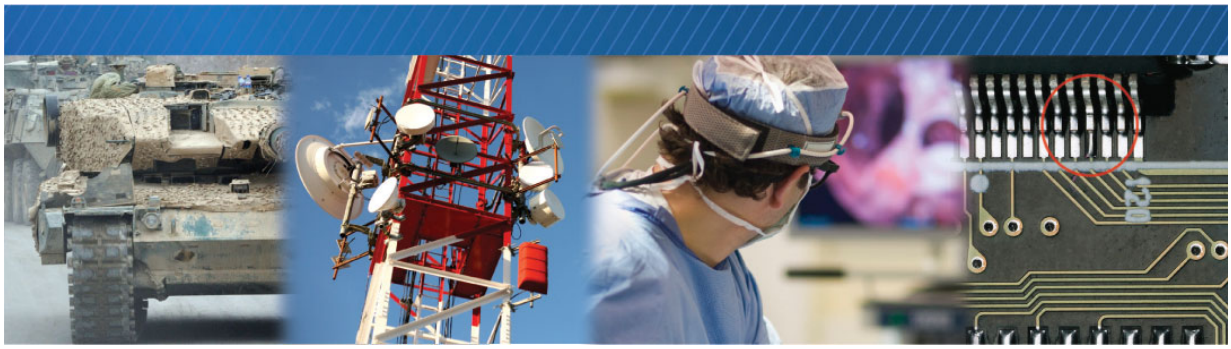
In the event that the main firmware load fails to load, the embedded video interface will start up using the backup firmware load when it is restarted or power cycled.

In the rare event that the backup load is not used automatically (as indicated by the fact that eBUS Player will not be able to detect the embedded video interface), you can use the slide switch to change to the backup load.

After the embedded video interface starts up using the backup load, you can apply a firmware update to the embedded video interface to recover the main load. For more information see the *Updating Pleora Firmware* knowledge base article on the Pleora Support Center (https://supportcenter.pleora.com).



Chapter 13



Reference: Mechanical Drawings and Material List

This chapter provides mechanical drawings and also provides a list of connectors and cables, with corresponding manufacturer details.

Three-dimensional (3-D) mechanical drawings are available at the Pleora Technologies Support Center.

The following topics are covered in this chapter:

NTx-NBT50 Embedded Video Interface Mechanical Drawings

The mechanical drawings in this section provide the NTx-NBT50 Embedded Video Interface's dimensions, features, and attributes. All dimensions are in millimeters.

Figure 9: NTx-NBT50 Embedded Video Interface Component Height — Top View

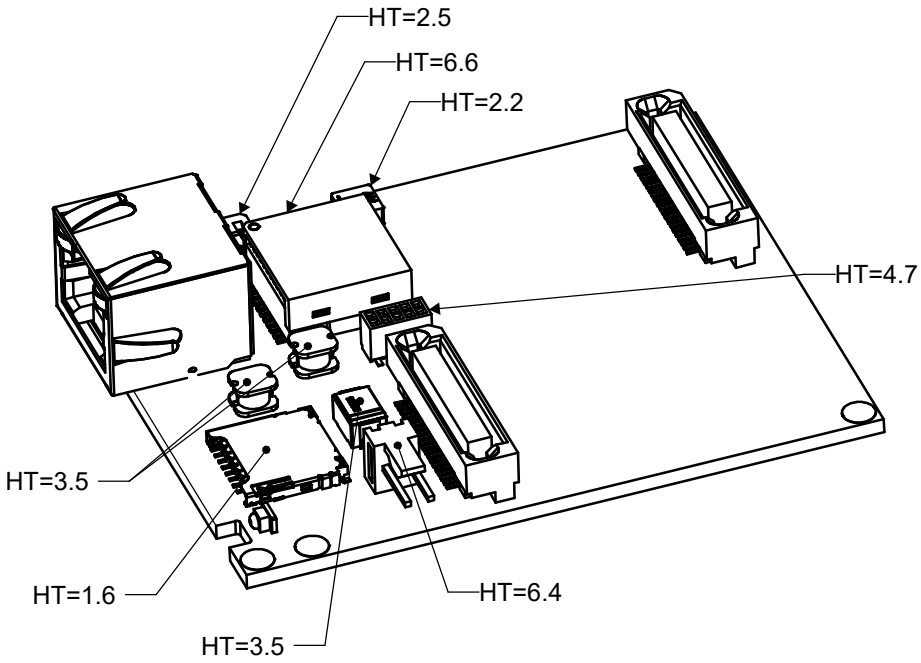


Figure 10: NTx-NBT50 Embedded Video Interface Component Height — Bottom View

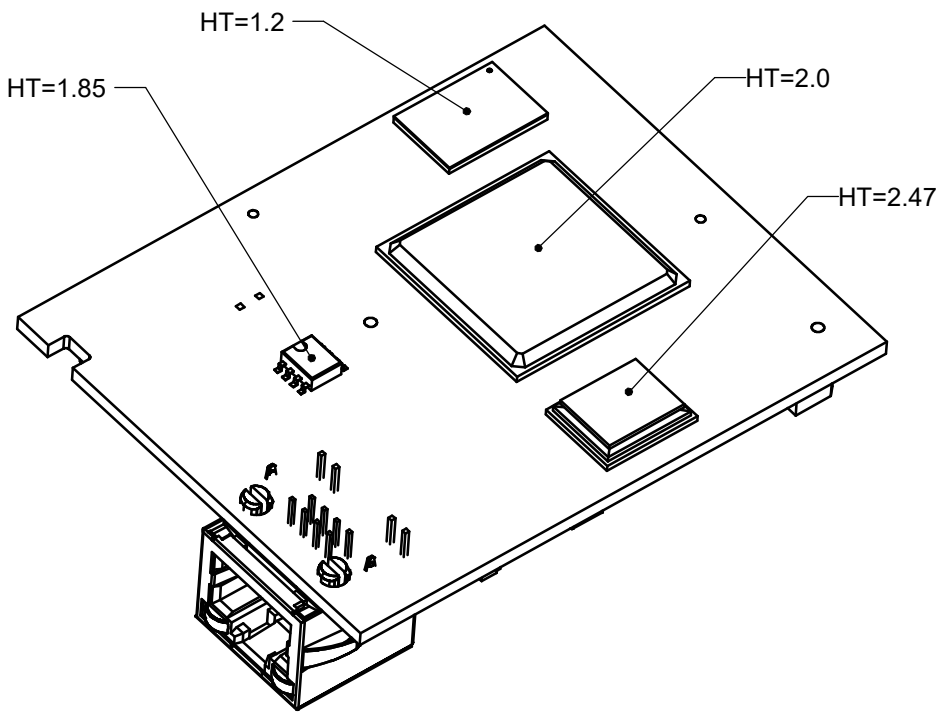


Figure 11: NTx-NBT50 Embedded Video Interface — Bottom View

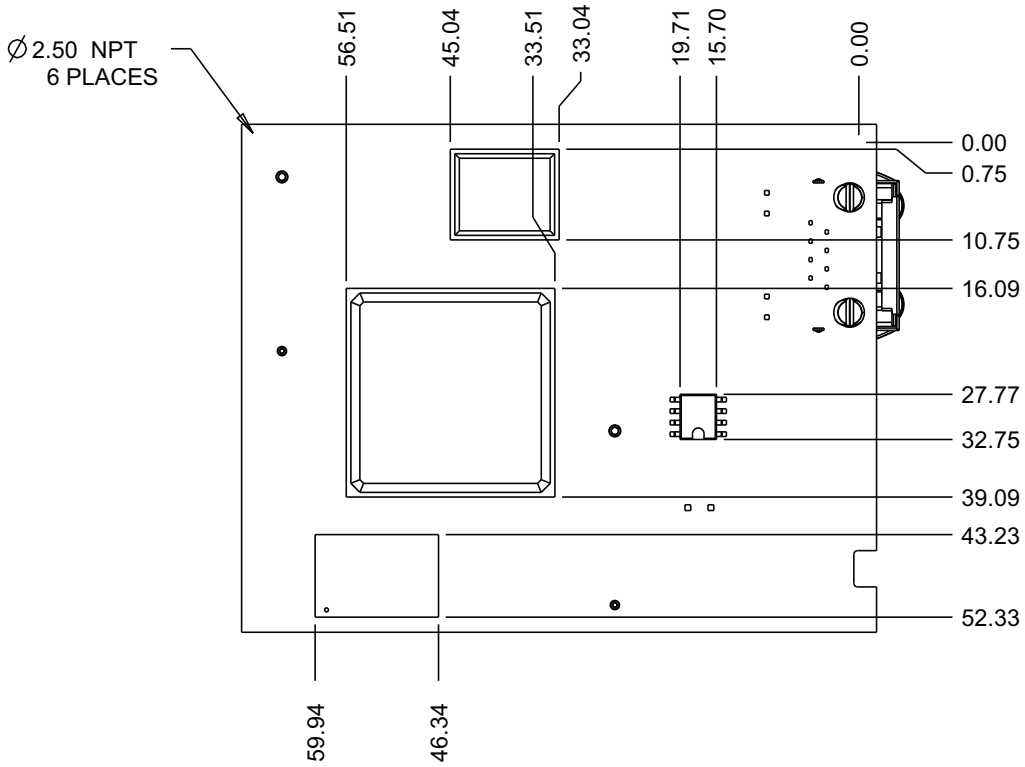


Figure 12: NTx-NBT50 Embedded Video Interface — Top View

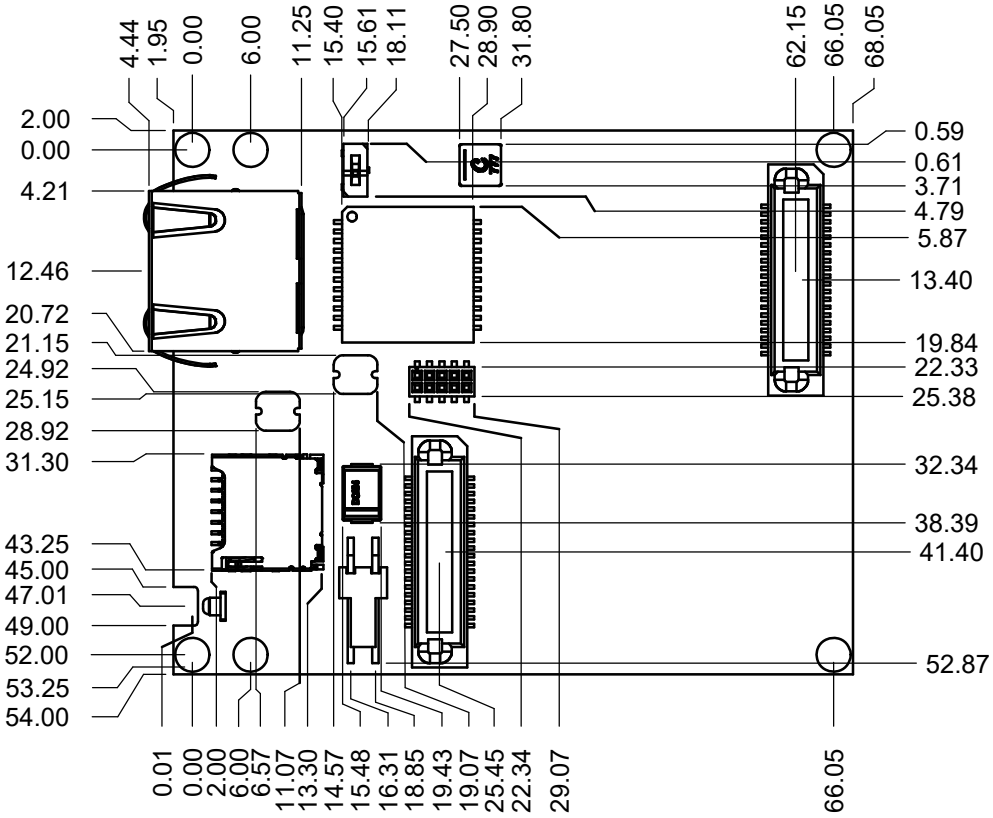


Figure 13: NTx-NBT50 Embedded Video Interface — Side Views

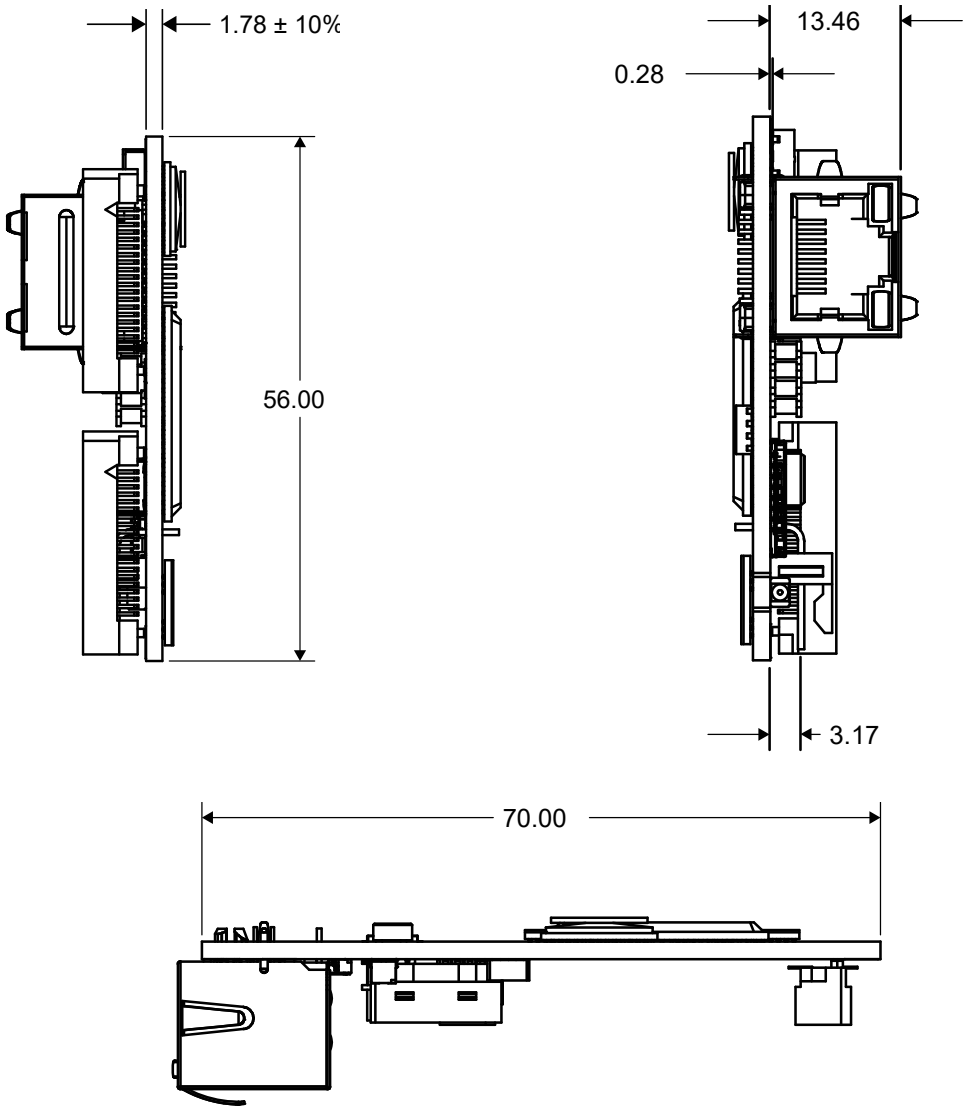
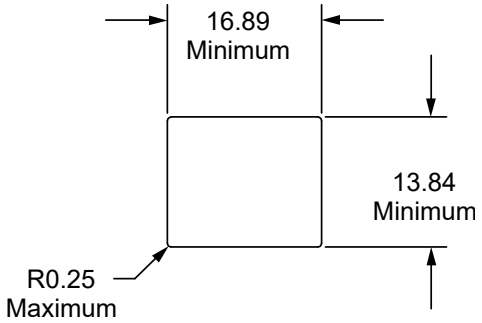


Figure 14: Recommended Ethernet Jack Panel Opening



Material List

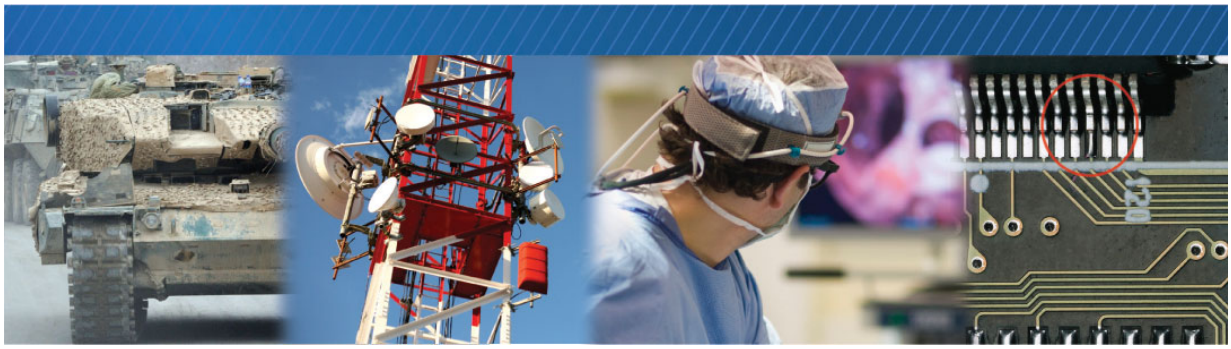
The connector summaries for the NTx-NBT50 Embedded Video Interface are listed in the following table.

Table 32: Connector Summary

ID	Description	Manufacturer part number	Manufacturer
J2, J5	40-pin user circuitry connector	FX6-40S-0.8SV2(93)	Hirose
J1	2-pin power connector	22-05-3021	Molex
J6	RJ-45 Ethernet connector	XM9B-18844-11	Omron Electronics

Source manufacturer, description, and identification may vary and are subject to change for each connector.

Chapter 14



Reference: Mean Time Between Failures (MTBF) Data

The following table provides MTBF data.

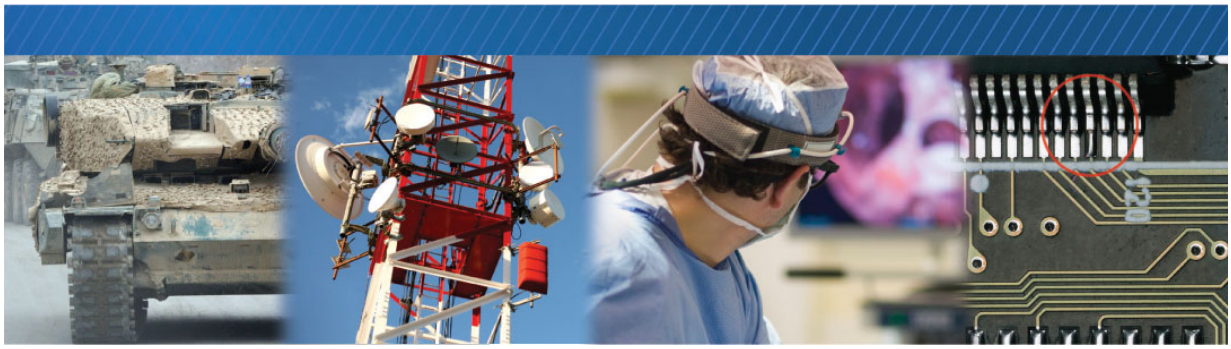
Table 33: MTBF Data

Model	MTBF @ 40°C
NTx-NBT50 Embedded Video Interface	1,059,389 hours

Assumptions:

1. The calculation is performed using the *RelCalc for Windows V5.1-TELC3* software, which implements Telcordia SR-332 (Issue 3) failure rate models.
2. The operating internal chassis temperature is 40°C. The calculation assumes the temperature across the boards is relatively constant.
3. The Telcordia environment is GB.
4. Each part's operating current/voltage/power stress is 50%.
5. The typical operating power value (as specified in the component's datasheet) is used for each IC and semiconductor.
6. The calculation uses the 90% UCL (Upper Confidence Level) Telcordia Issue 3 model.
7. Each part's Telcordia Quality Level is I.

Chapter 15



Technical Support

On the Pleora Support Center, you can:

- Download the latest software and firmware.
- Log a support issue.
- View documentation for current and past releases.
- Browse for solutions to problems other customers have encountered.
- Read knowledge base articles for information about common tasks.

To visit the Pleora Support Center

- Go to supportcenter.pleora.com.

  Most material is available without logging in to a Support Center account. To access software and firmware downloads, in addition to other content, log in to the Support Center. If you do not have an account, click **Request Account**.

  Accounts are usually validated within one business day.