

RS232 port – Prosilica’s GE and GC series cameras

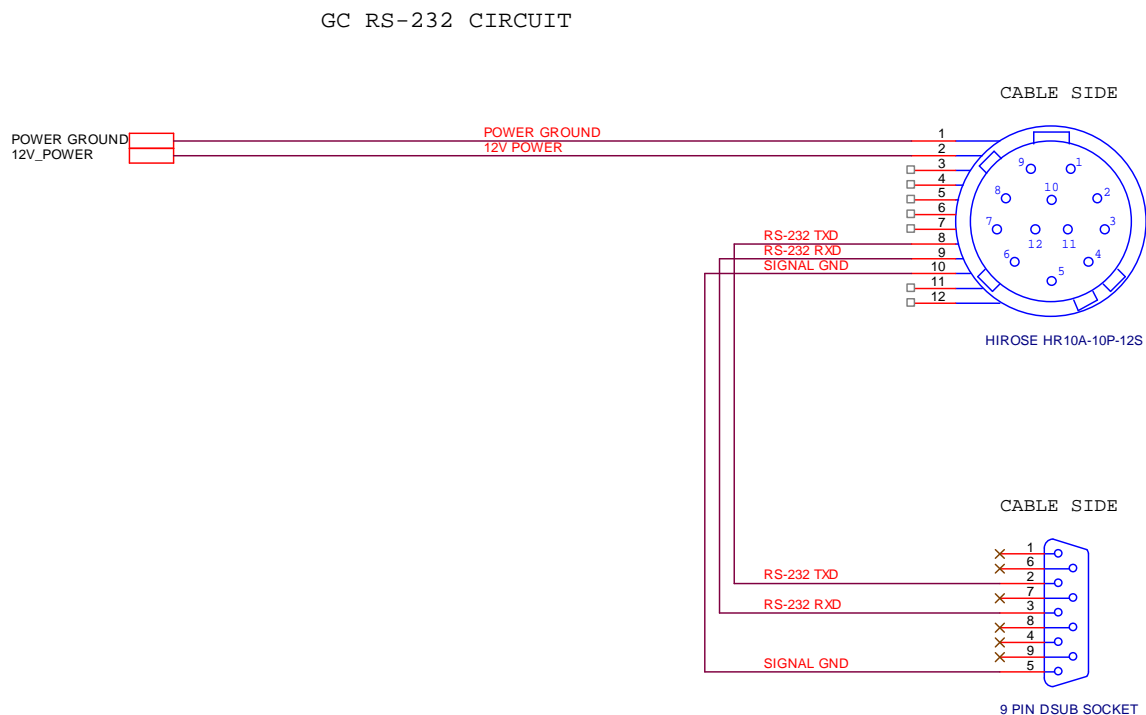
Prosilica’s GC and GE cameras can output and receive serial commands using an onboard RS232 port. Tx and Rx data lines are available via the Hirose connection on the back of the cameras. Enabling and controlling serial communication is performed by writing to serial IO registers on the camera through the Prosilica GigE SDK. See the `GigESDK/examples/siotest` source code for detailed information on controlling the RS232 port.

Testing the RS232 port

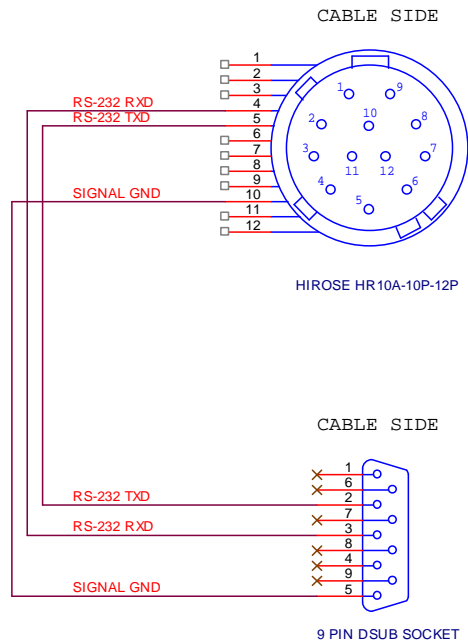
The `siotest` example code is a useful tool for testing your camera’s RS232 port.

Requirements:

- A computer with a working serial port.
- A C++ compiler to compile the `siotest` sample code.
- A terminal application which allows serial communication.
- Connect your Prosilica camera to the computer serial port according to the following diagram:



GE RS-232 CIRCUIT



Testing:

Plug in your Prosilica camera, ensure it is recognized by your GigE network and that the RS232 lines are connected to your computer serial port.

Open your terminal application, with baud rate 9600, and whatever COM your computer's serial port is on (typically COM1).

Compile the siotest example code, and run the resulting exe in a windows command prompt.

You should see a test pattern of integers appear in the terminal window. This verifies that the camera's RS232 Tx is working.

Type in the terminal window, you should see your typed input appear in the windows command prompt. This verifies that the camera's RS232 Rx is working.

```
C:\WINDOWS\system32\cmd.exe - siotest.exe
C:\Program Files\Prosilica\GigESDK\bin-pc>
C:\Program Files\Prosilica\GigESDK\bin-pc>siotest.exe
Serial IO Test. Press CTRL-C to end.

Searching for a camera... found.

SerialIoInquiry: 0x00000007
SerialModeInquiry: 0x07c703ff
SerialTxInquiry: 0x00000400
SerialRxInquiry: 0x00000400

Hello World???
```

Tera Term - COM1 VT

File Edit Setup Control Window Help

0000000548 0000000549 0000000550 00000000
0000000549 0000000550 0000000551 00000000
0000000550 0000000551 0000000552 00000000
0000000551 0000000552 0000000553 00000000
0000000552 0000000553 0000000554 00000000
0000000553 0000000554 0000000555 00000000
0000000554 0000000555 0000000556 00000000
0000000555 0000000556 0000000557 00000000
0000000556 0000000557 0000000558 00000000
0000000557 0000000558 0000000559 00000000
0000000558 0000000559 0000000560 00000000
0000000559 0000000560 0000000561 00000000
0000000560 0000000561 0000000562 00000000
0000000561 0000000562 0000000563 00000000
0000000562 0000000563 0000000564 00000000
0000000563 0000000564 0000000565 00000000
0000000564 0000000565 0000000566 00000000
0000000565 0000000566 0000000567 00000000
0000000566 0000000567 0000000568 00000000
0000000567 0000000568 0000000569 00000000
0000000568 0000000569 0000000570 00000000
0000000569 0000000570 0000000571 00000000
0000000570 0000000571 0000000572 00000000
0000000571 0000000572 0000000573 00000000
0000000572 0000000573 0000000574 0000000575
0000000573 0000000574 0000000575 0000000576
0000000574 0000000575 0000000576 0000000577
0000000575 0000000576 0000000577 0000000578

Tera Term: Serial port setup

Port: COM1 OK

Baud rate: 9600 Cancel

Data: 8 bit Help

Parity: none

Stop: 1 bit

Flow control: none

Transmit delay

0 msec/char 0 msec/line

Register Map

SerialIo Control

Address	Name	Bits	Description
16000h	SerialIoInquiry	[0]	[R] Serial IO transmitter is available.
		[1]	[R] Serial IO receiver is available.
		[2]	[R] Serial IO receiver has timestamp mode.
		[other]	[R] Reserved. All zeros.
16010h	SerialIoErrorStatus	[other]	[R] Reserved. All zeros.
16100h	SerialModeInquiry	[0]	[R] 300 baud supported.
		[1]	[R] 600 baud supported.
		[2]	[R] 1200 baud supported.
		[3]	[R] 2400 baud supported.
		[4]	[R] 4800 baud supported.
		[5]	[R] 9600 baud supported.
		[6]	[R] 19200 baud supported.
		[7]	[R] 38400 baud supported.
		[8]	[R] 57600 baud supported.
		[9]	[R] 115200 baud supported.
		[10]	[R] 230400 baud supported.
		[16]	[R] No parity supported.
		[17]	[R] Odd parity supported.
		[18]	[R] Even parity supported.
		[20]	[R] Character length 5 bits supported.
		[21]	[R] Character length 6 bits supported.
		[22]	[R] Character length 7 bits supported.
		[23]	[R] Character length 8 bits supported.
		[24]	[R] 1 stop bit supported.
		[25]	[R] 1.5 stop bits supported.
[26]	[R] 2 stop bits supported.		
[other]	[R] Reserved. All zeros.		

16104h	SerialMode	[7..0]	[RW] Baud rate: 0: 300 1: 600 2: 1200 3: 2400 4: 4800 5: 9600 6: 19200 7: 38400 8: 57600 9: 115200 10: 230400
		[9..8]	[RW] Parity: 0: None 1: Odd 2: Even
		[11..10]	[RW] Character length: 0: 5 bits 1: 6 bits 2: 7 bits 3: 8 bits
		[13..12]	[RW] Stop bits: 0: 1 stop bit 1: 1.5 stop bits 2: 2 stop bits
		[other]	[R] Reserved. All zeros.
16120h	SerialTxInquiry	[15..0]	[R] Transmitter buffer size, in bytes.
		[other]	[R] Reserved. All zeros.
16124h	SerialTxStatus	[0]	[R] Transmitter ready.
		[other]	[R] Reserved. All zeros.
16128h	SerialTxControl	[0]	[W] Transmitter reset when 1. Occurs immediately. [R] Always zero.
		[1]	[RW] Transmitter enable, when 1.
		[other]	[R] Reserved. All zeros.
1612Ch	SerialTxLength	[15..0]	[W] Transmit data length, in bytes. When this is written, the data in <i>SerialTxBuffer</i> is sent through the serial port. <i>SerialTxLength</i> may not be written if "Transmitter Ready" is zero. [R] Always zeros.
		[other]	[R] Reserved. All zeros.
16140h	SerialRxInquiry	[15..0]	[R] Receiver buffer size, in bytes.

		[other]	[R]	Reserved. All zeros.
16144h	SerialRxStatus	[0]	[R] [W]	Receive overrun. Write 1 to clear.
		[1]	[R] [W]	Receive framing error. Write 1 to clear.
		[2]	[R] [W]	Receive parity error. Write 1 to clear.
		[other]	[R]	Reserved. All zeros.
16148h	SerialRxControl	[0]	[W] [R]	Receiver reset when 1. Occurs immediately. Always zero.
		[1]	[RW]	Receiver enable, when 1.
		[2]	[RW]	Enable timestamp mode. See <i>SerialRxBuffer</i> for a description of timestamp mode. (Don't change this bit on the fly, otherwise some data will be timestamped and some will not.) Check <i>SerialIoInquiry</i> to see if this feature is available.
		[other]	[R]	Reserved. All zeros.
1614Ch	SerialRxLength	[15..0]	[R] [W]	Number of bytes in the receive buffer. Number of bytes read from receive buffer. The counter (see [R] above) is decremented by this amount.
16400h	SerialTxBuffer		[W]	Transmit buffer. Write your data into the buffer, then write your data length into <i>SerialTxLength</i> to begin transmission. Each serial word is stored as a byte, LSBit aligned. The bytes are packed into 32-bit registers; the MSByte of each register is the first serial-word transmitted. (When the data length is not a multiple of 4, trailing bytes are ignored.) ex. write 0x41424344 to output "ABCD" <i>SerialTxBuffer</i> may not be written if " <i>Transmitter Ready</i> " is zero.
16800h	SerialRxBuffer		[R]	Receive buffer. Read data from this buffer. Read <i>SerialRxLength</i> for the number of valid bytes in this receive buffer. After reading the data, you must write the length of your read to <i>SerialRxLength</i> . See <i>SerialTxBuffer</i> for data packing. Timestamp mode: each received byte is preceded by 8 timestamp bytes, MSB first. This timestamp is the frame timestamp. ex. 0x11223344AABBCCDD41 is "A" received at timestamp 0x11223344AABBCCDD