Serial Port Interfacing with LabVIEW

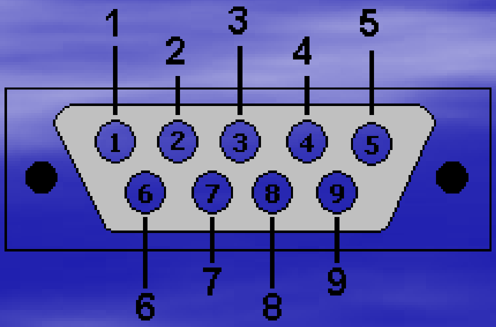
Umer Hassan

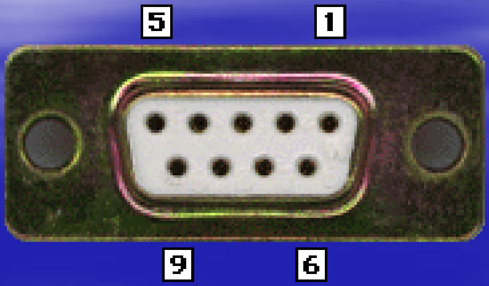
18 Feb. 2011

Details…

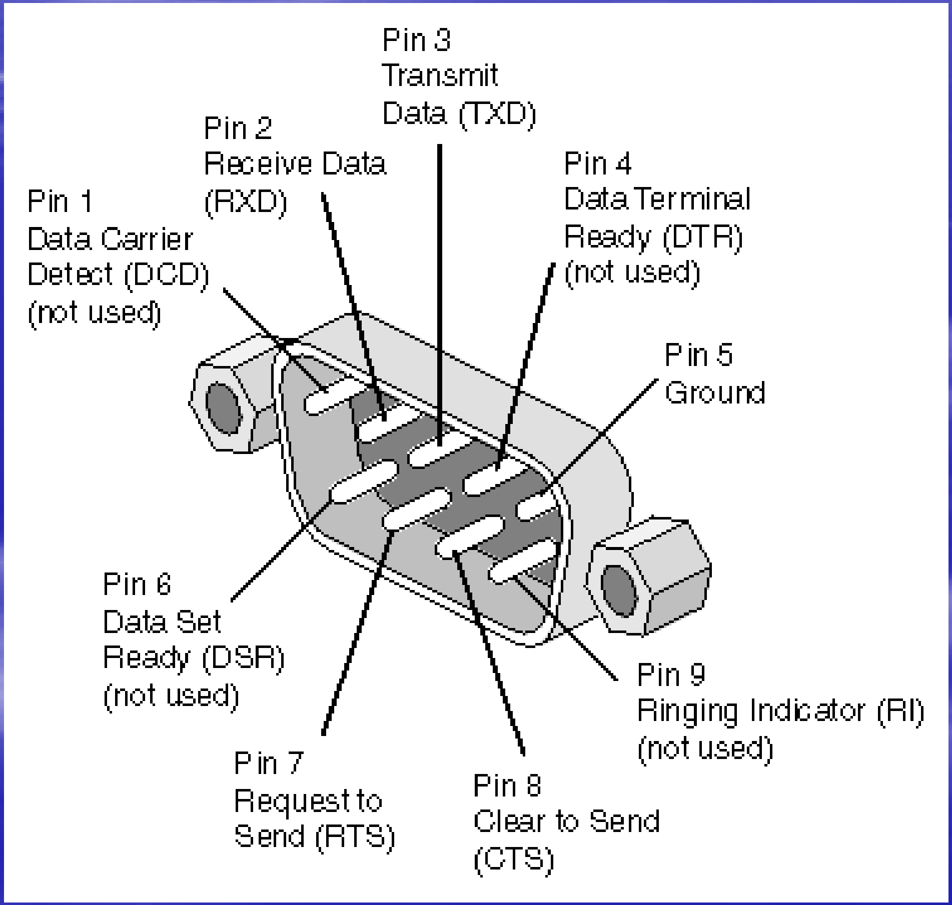
- Cable Connectors
- Hardware Specification
  1. Pin Identification
  2. Communications Specifications
- LabVIEW Interfacing

Cable Connectors DB-9 Connector





Pin-outs


Pin 1
Data Carrier Detect (DCD) (not used)
Pin 2
Receive Data (RXD)
Pin 3
Transmit Data (TXD)
Pin 4
Data Terminal Ready (DTR) (not used)
Pin 5
Ground
Pin 6
Data Set Ready (DSR) (not used)
Pin 7
Request to Send (RTS)
Pin 8
Clear to Send (CTS)
Pin 9
Ringing Indicator (RI) (not used)

- A Data Communications Equipment (DCE) can be interfaced to a Data Terminal Equipment (DTE) using a straight through serial cable. Typically, PC's are defined as a DTE and peripherals are defined as DCE. To interface a DCE with another DCE, or a DTE with another DTE, a null modem (or a Crossover) cable is required. The null modem simply swaps pins to convert a DCE to a DTE and vice-versa.

RS232 Serial Interface Standard

- A “Space” (logic 0) will be between 3 and 25 volts.
- A “ Mark” (logic 1) will be between -3 and -25 volts.
- The region between 3 & -3 volts is undefined.
- Maximum data rates may be up to 20 kbps.
- The reason to study RS-232C is that the serial part (Com port) found in PC’S uses this standard.

ASCII representation

- **American Standard Code for Information Interchange**

- http://www.asciitable.com/

Baud Rate

- The baud rate is simply “the rate of data transmission expressed in bits per second, kilo Bits per second or Mega bits per second etc”.

Data Bits

- The sender & receiver decide upon no of bits in one data frame such as 8 bits (1byte) etc.

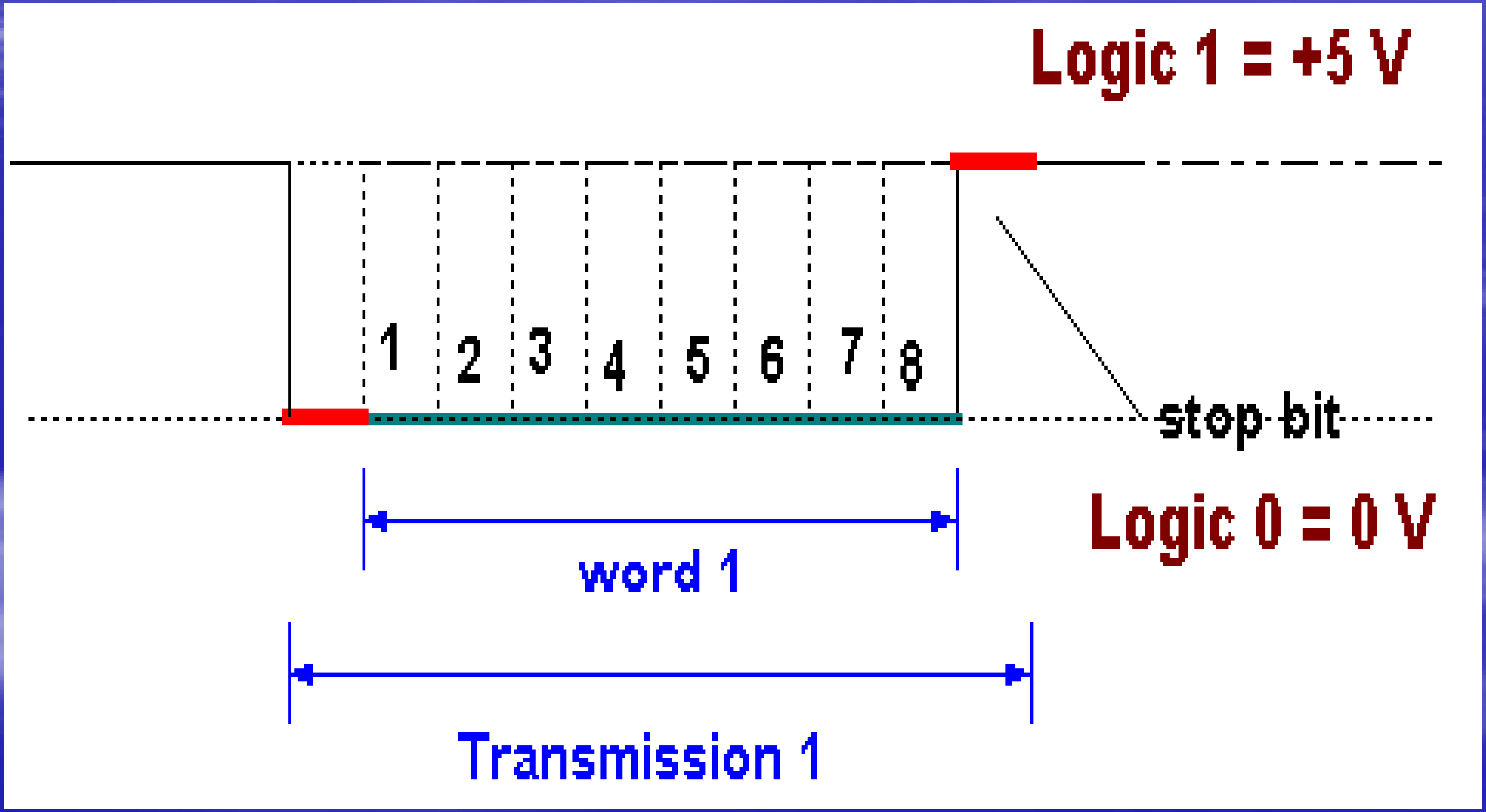
Start Bit

- Each frame is started with the *start bit*, which is always identified by the space line level. Because the line is in mark state when idle, the start bit is easily recognized by the receiver.

Stop Bit

- The stop bit identifying the end of a data frame can have different lengths. Actually, it is not a real bit but a minimum period of time the line must be idle (mark state) at the end of each word. On **PC**'s this period can have three lengths: the time equal to **1**, **1.5** or **2** bits. **1.5** bits is only used with data words of **5** bits length and **2** only for longer words. A stop bit length of **1** bit is possible for all data word sizes.

Data Frame



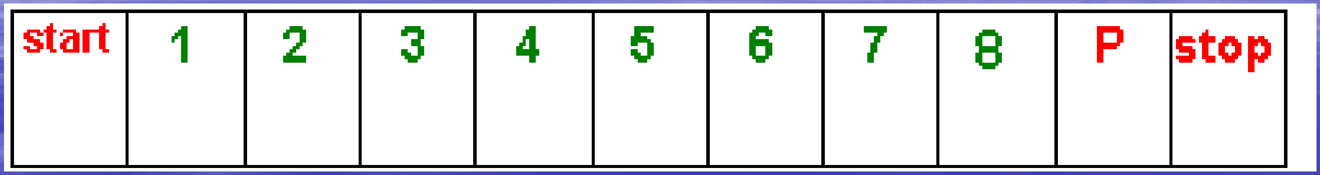
Parity Bit

- **Parity** Is of two types
- 1. Even parity
- 2. Odd parity
- Suppose your data word is 8 bit in length i.e. one byte.
- The sender, before transmitting byte, determines whether the no of bits in the byte to be sent are even. Suppose sender's wants to send
- 10011101
- In which there are 5 " 1's " and 3 " 0's ".
- To keep the no. of 1's even the sender adds an extra bit at the end of byte so that the total no. of 1's are 6 (an even no.). This extra bits is called parity bit. Since this bit keeps the no. of 1's even, so it is called even parity.

Failure of Parity

- What if you transmit above byte 10011101 with even parity being used and on receiver side you receive
- 10000101 1
- 2 bits inverted during transmission
- On receiver side the no. of 1’s including parity=4 (no error for even parity). But actually it is false.
- So parity does not ensure error detection in all cases.

Overview of total bits with Parity



start	1	2	3	4	5	6	7	8	P	stop

<u>Odd Parity</u>

Odd parity is added to keep the no. of 1's odd in transmission. For example for above case of data being

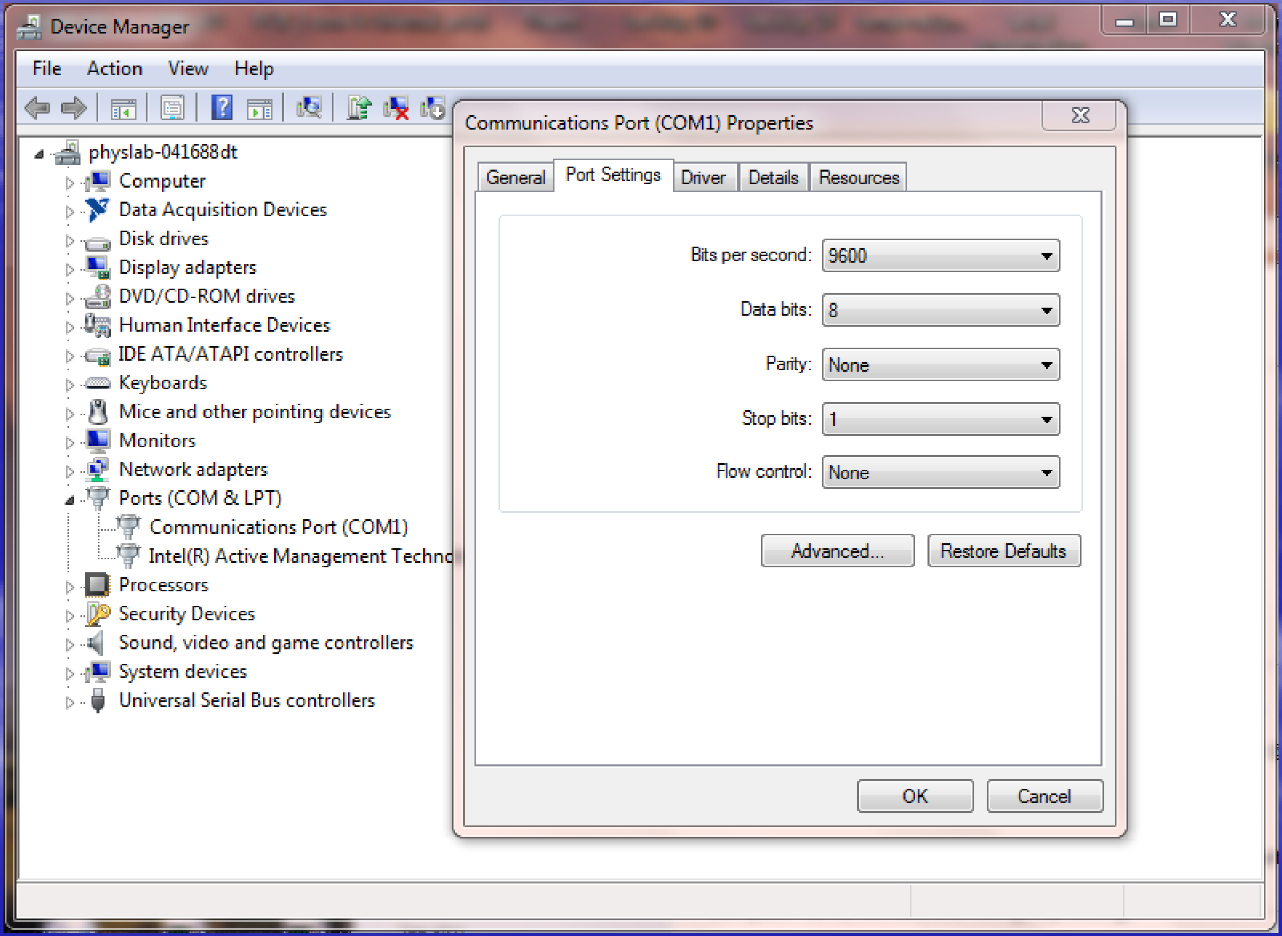
10011101

The odd parity will be added as 0 as no of 1's are already 5 (an odd no.).

Flow Control

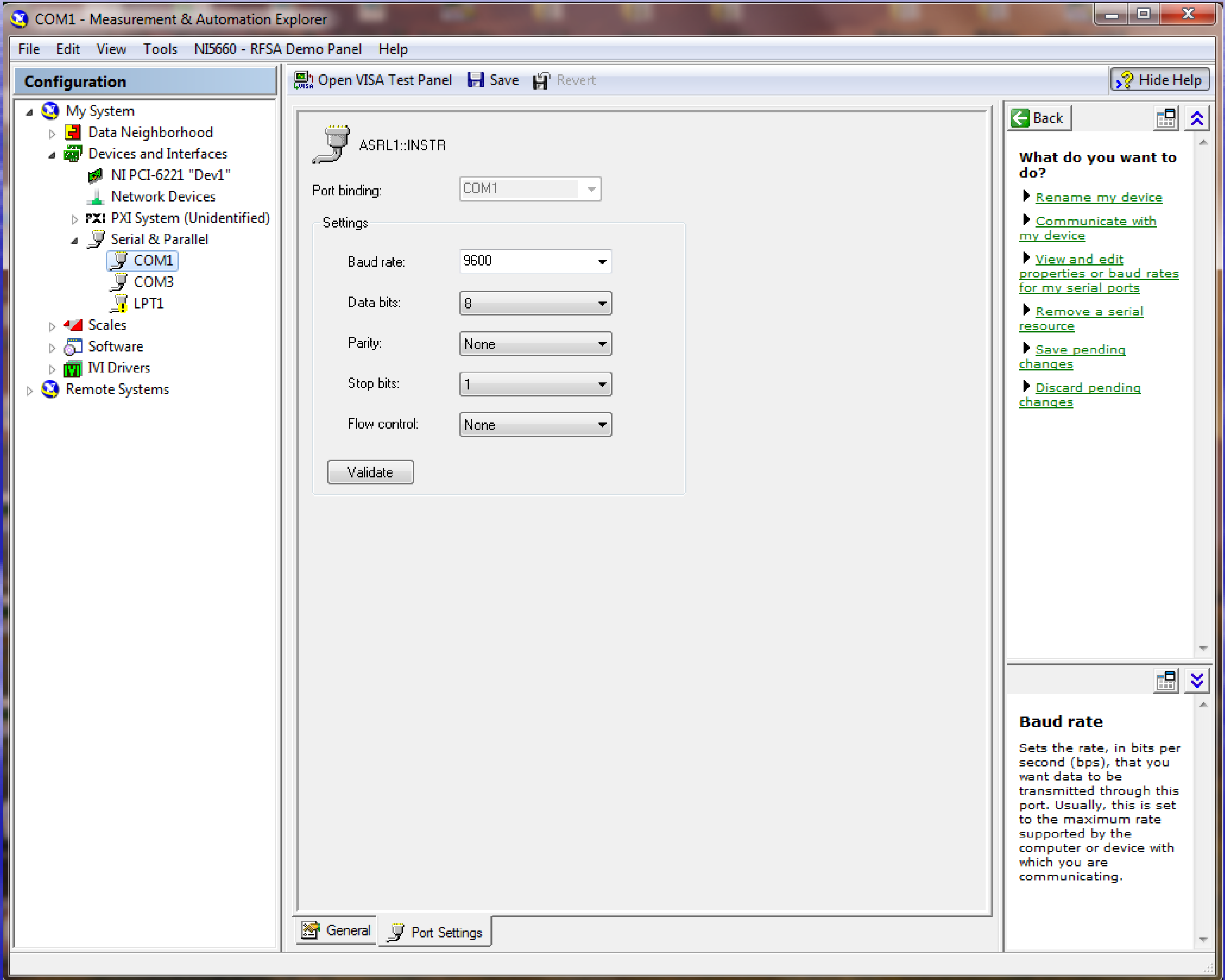
- So if our DTE to DCE speed is several times faster than our DCE to DCE speed the PC can send data to your modem at 115,200 BPS. Sooner or later data is going to get lost as buffers overflow, thus flow control is used. Flow control has two basic varieties, Hardware or Software.
- Software flow control, sometimes expressed as Xon/Xoff uses two characters Xon and Xoff. Xon is normally indicated by the ASCII 17 character where as the ASCII 19 character is used for Xoff. The modem will only have a small buffer so when the computer fills it up the modem sends a Xoff character to tell the computer to stop sending data. Once the modem has room for more data it then sends a Xon character and the computer sends more data. This type of flow control has the advantage that it doesn't require any more wires as the characters are sent via the TD/RD lines. However on slow links each character requires 10 bits which can slow communications down.
- Hardware flow control is also known as RTS/CTS flow control. It uses two wires in your serial cable rather than extra characters transmitted in your data lines. Thus hardware flow control will not slow down transmission times like Xon-Xoff does. When the computer wishes to send data it takes active the Request to Send line. If the modem has room for this data, then the modem will reply by taking active the Clear to Send line and the computer starts sending data. If the modem does not have the room then it will not send a Clear to Send.

Device Manager



Bits per second: 9600
Data bits: 8
Parity: None
Stop bits: 1
Flow control: None

Measurement & Automation explorer



ASRL1::INSTR

Port binding: COM1

Settings

Baud rate: 9600

Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

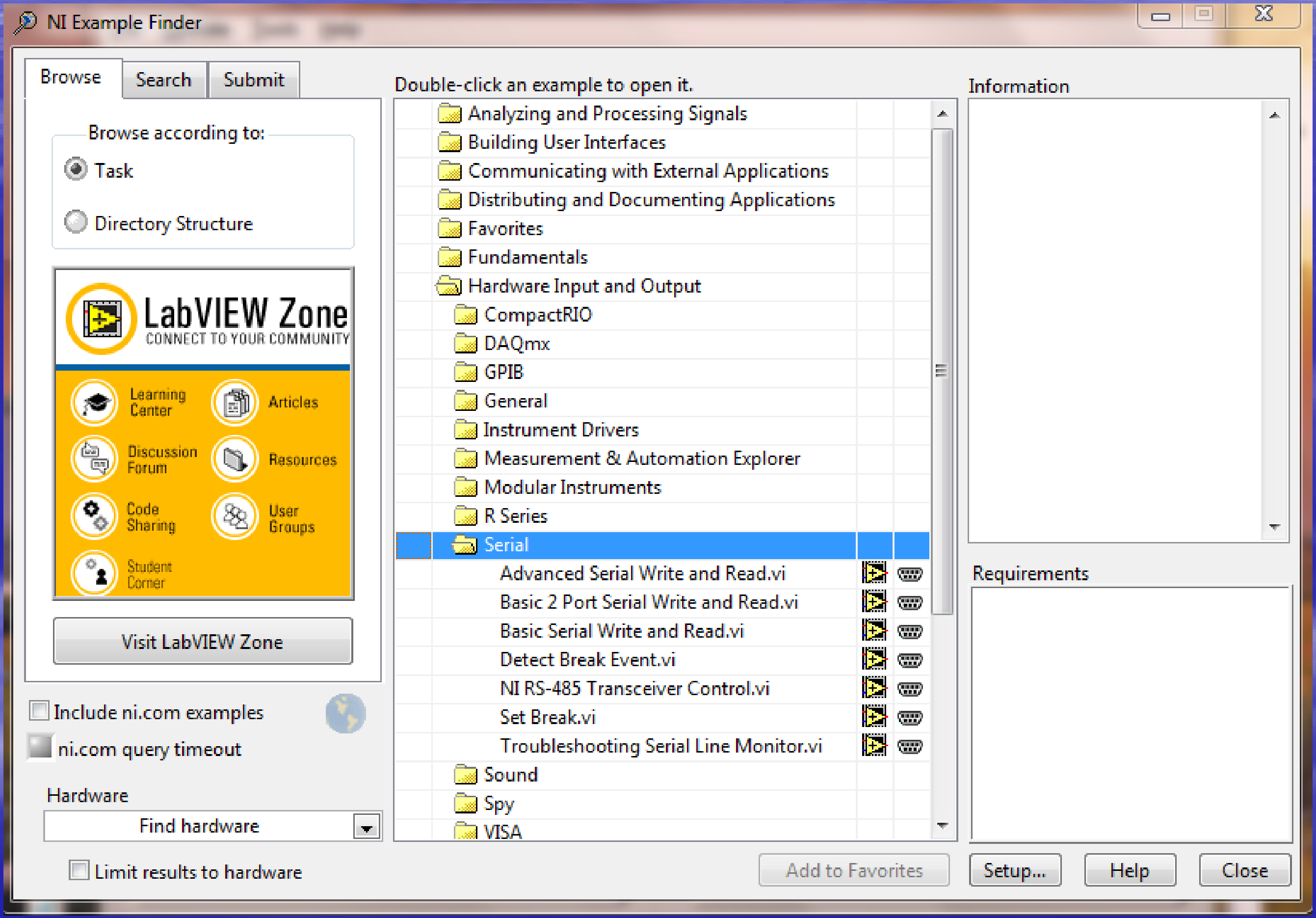
What do you want to do?

- Rename my device
- Communicate with my device
- View and edit properties or baud rates for my serial ports
- Remove a serial resource
- Save pending changes
- Discard pending changes

Baud rate

Sets the rate, in bits per second (bps), that you want data to be transmitted through this port. Usually, this is set to the maximum rate supported by the computer or device with which you are communicating.

LabVIEW



Browse according to:
[✓ Task]
Directory Structure

Include ni.com examples
ni.com query timeout

Hardware
Find hardware
Limit results to hardware

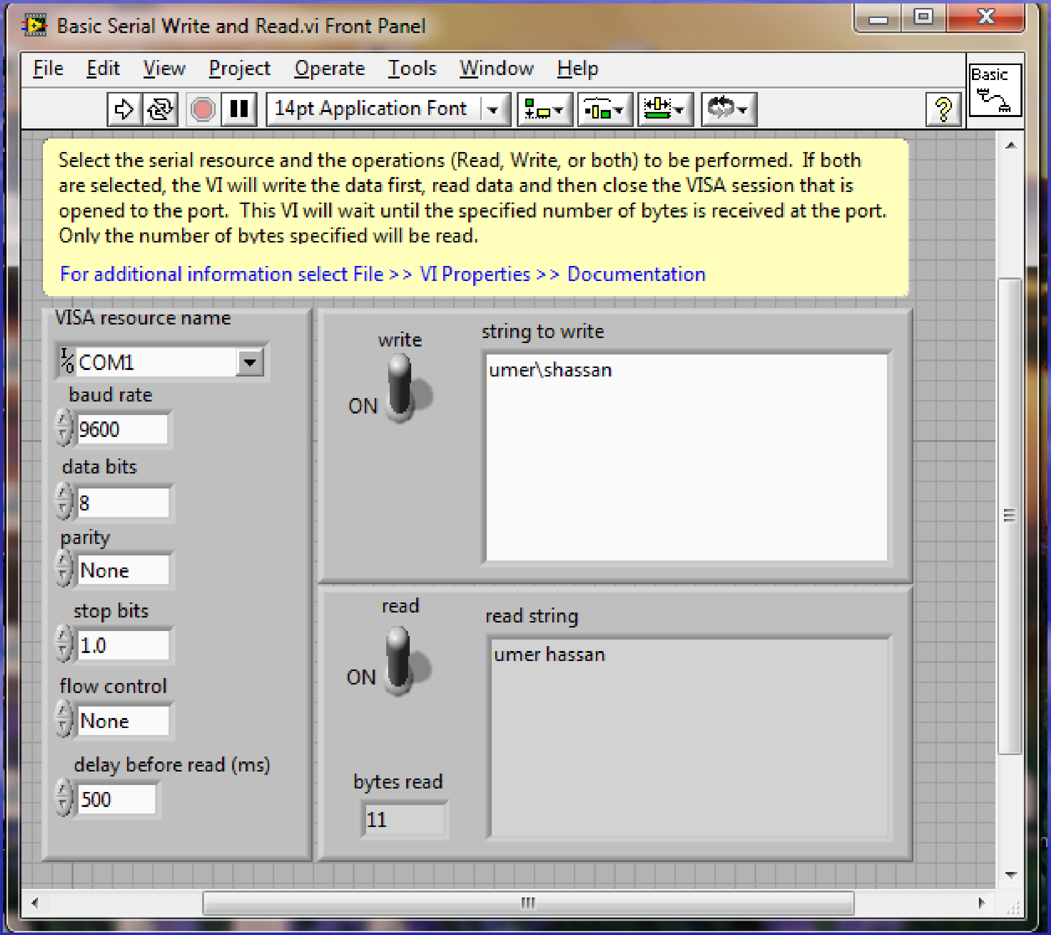
Double-click an example to open it.

- Analyzing and Processing Signals
- Building User Interfaces
- Communicating with External Applications
- Distributing and Documenting Applications
- Favorites
- Fundamentals
- Hardware Input and Output
  - CompactRIO
  - DAQmx
  - GPIB
  - General
  - Instrument Drivers
  - Measurement & Automation Explorer
  - Modular Instruments
  - R Series
  - Serial
    - Advanced Serial Write and Read.vi
    - Basic 2 Port Serial Write and Read.vi
    - Basic Serial Write and Read.vi
    - Detect Break Event.vi
    - NI RS-485 Transceiver Control.vi
    - Set Break.vi
    - Troubleshooting Serial Line Monitor.vi
  - Sound
  - Spy
  - VISA

Information

Requirements

Loopback test


Basic Serial Write and Read.vi Front Panel
File Edit View Project Operate Tools Window Help
14pt Application Font
Basic

Select the serial resource and the operations (Read, Write, or both) to be performed. If both are selected, the VI will write the data first, read data and then close the VISA session that is opened to the port. This VI will wait until the specified number of bytes is received at the port. Only the number of bytes specified will be read.

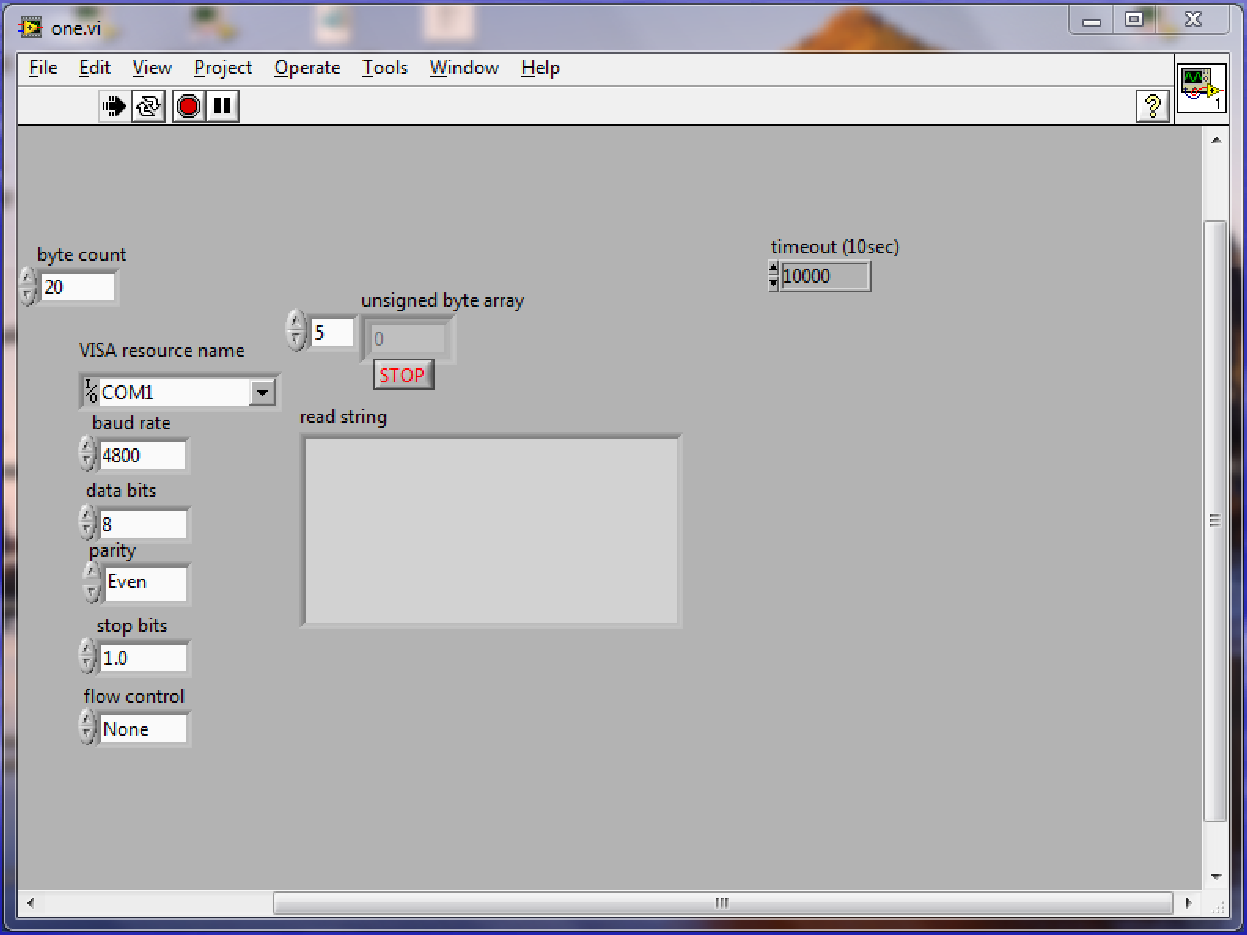
For additional information select File >> VI Properties >> Documentation

VISA resource name: COM1
baud rate: 9600
data bits: 8
parity: None
stop bits: 1.0
flow control: None
delay before read (ms): 500

write: ON
string to write: umer\shassan

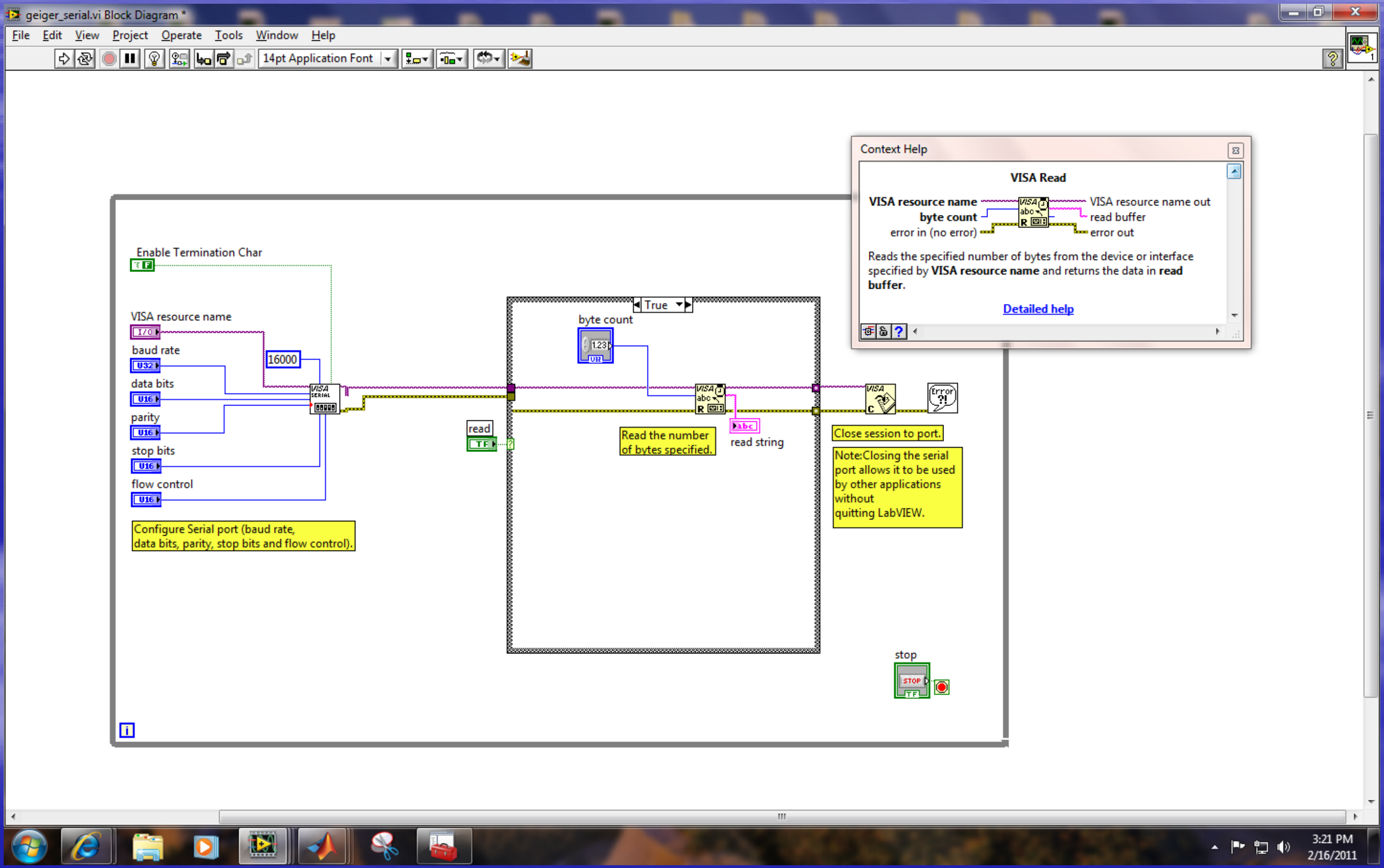
read: ON
read string: umer hassan
bytes read: 11

Geiger Counter VI



byte count: 20
VISA resource name: COM1
baud rate: 4800
data bits: 8
parity: Even
stop bits: 1.0
flow control: None
unsigned byte array: 5, 0
read string
timeout (10sec): 10000

Block Diagram



Configure Serial port (baud rate, data bits, parity, stop bits and flow control).

Read the number of bytes specified.

Close session to port.

Note:Closing the serial port allows it to be used by other applications without quitting LabVIEW.

Context Help

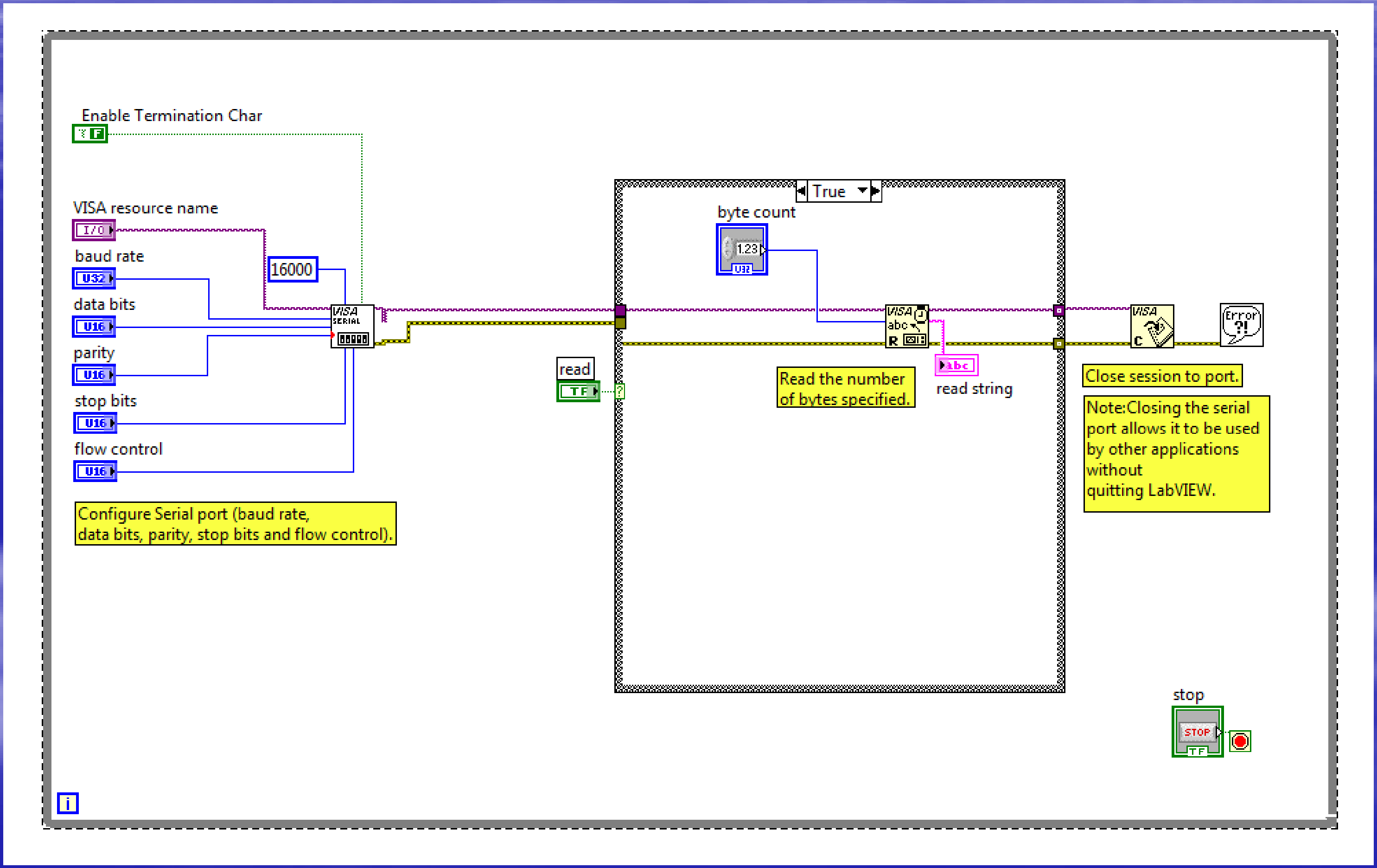
VISA Read

VISA resource name — VISA resource name out
byte count — read buffer
error in (no error) — error out

Reads the specified number of bytes from the device or interface specified by **VISA resource name** and returns the data in **read buffer**.

Detailed help

Continued…



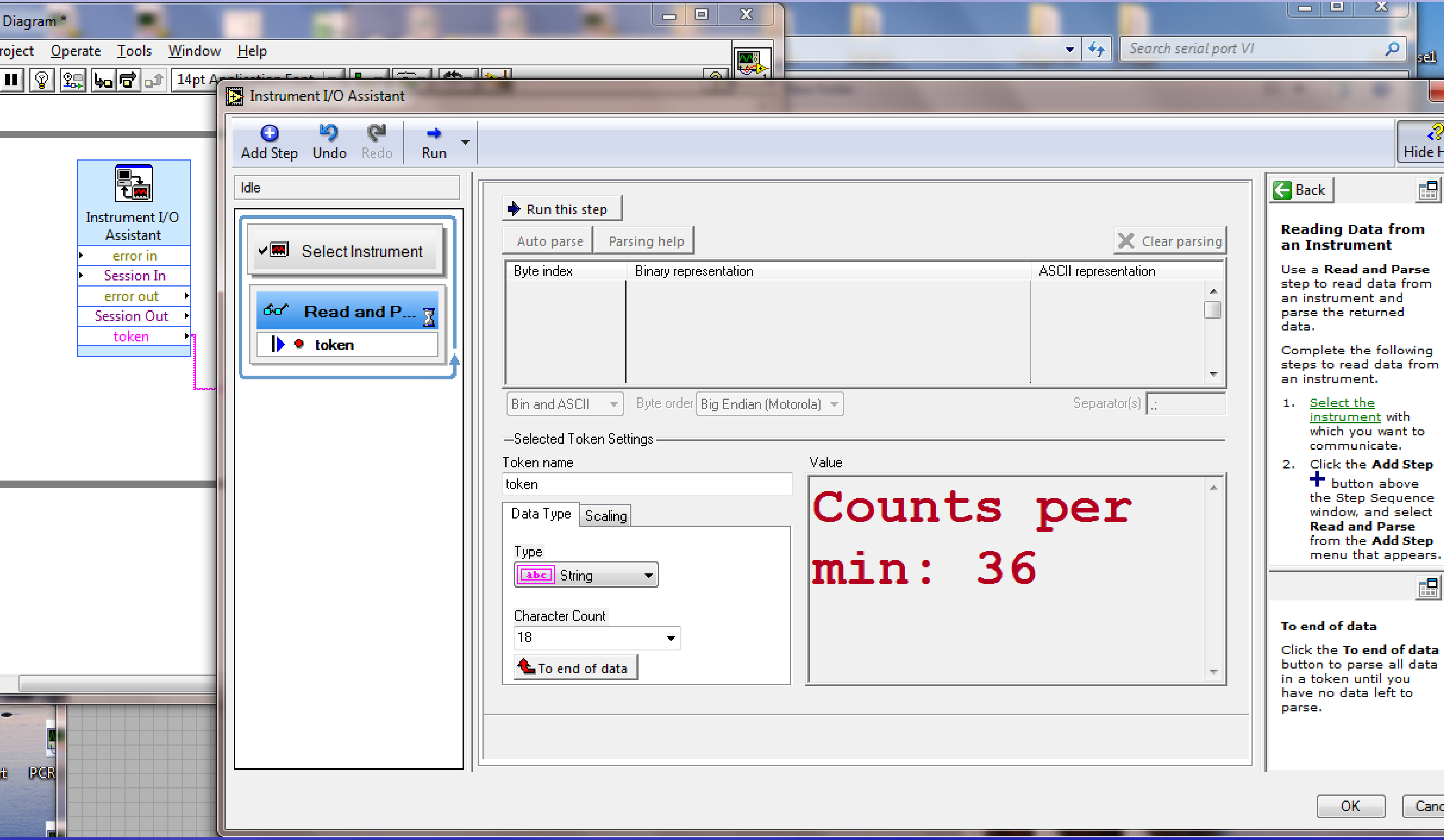
Configure Serial port (baud rate, data bits, parity, stop bits and flow control).

Read the number of bytes specified.

Close session to port.

Note:Closing the serial port allows it to be used by other applications without quitting LabVIEW.

Instrument I/O Assistant



Selected Token Settings

Token name: token

Data Type | Scaling

Type: String

Character Count: 18

To end of data

Value

Counts per min: 36

Reading Data from an Instrument

Use a **Read and Parse** step to read data from an instrument and parse the returned data.

Complete the following steps to read data from an instrument.

1. Select the instrument with which you want to communicate.
2. Click the **Add Step** button above the Step Sequence window, and select **Read and Parse** from the **Add Step** menu that appears.

To end of data

Click the **To end of data** button to parse all data in a token until you have no data left to parse.

Thanks