

Instrument Control

TOPICS

Instrument Control Overview

GPIB Communication and Configuration (IEEE 488)

Instrument I/O Assistant

Virtual Instrument Software Architecture (VISA)

Instrument Drivers

Serial Port Communication

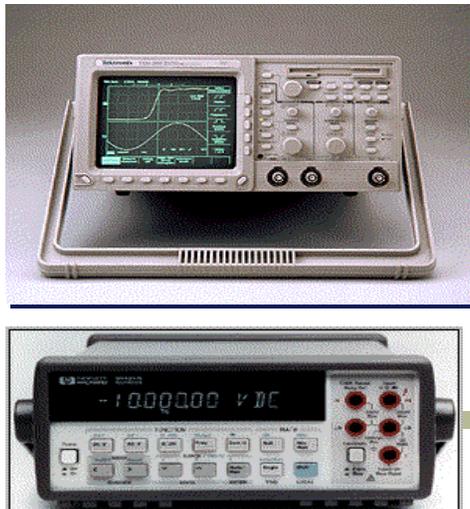
Waveform Transfers

Instrument Control Overview

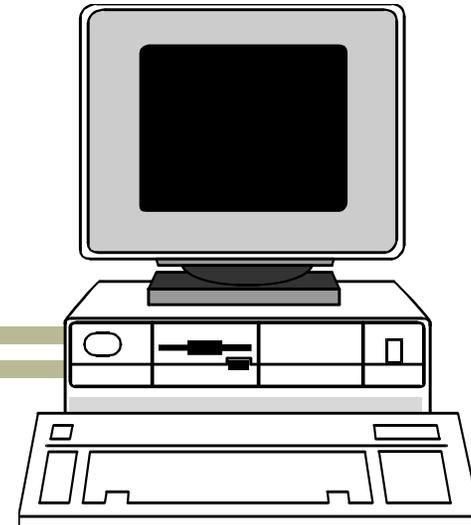
Control any instrument if you know the following:

- Type of connector on the instrument
- Electrical properties involved
- Software drivers available
- Type of cables needed
- Communication protocols used

Instruments

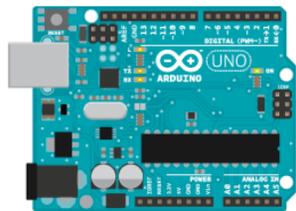


Computer

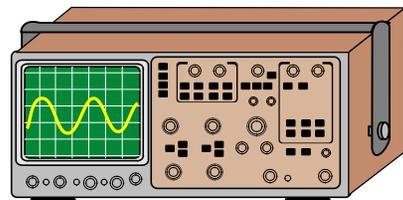


Serial Communication

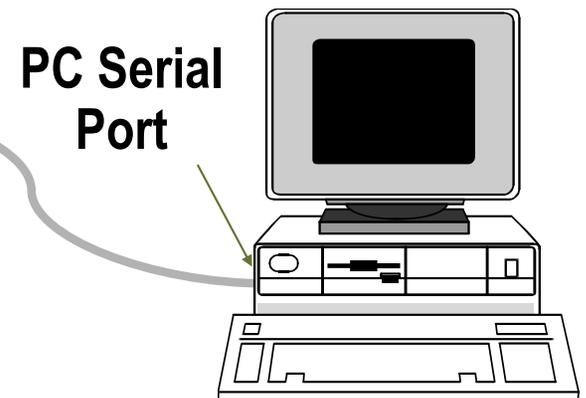
- Popular means of communication between computer and peripheral device
- Data sent one bit at a time across the cable
- Used for low transfer rates or long distances
- Only a cable is needed since most computers have at least one available serial port



RS-232 Instrument



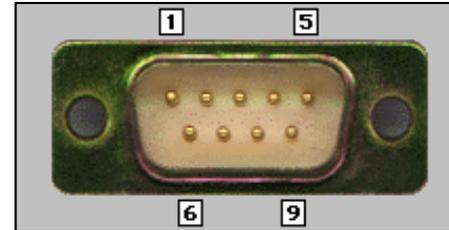
RS-232 Cable



Serial Hardware Connection

- **RS-232**

- DCE or DTE configurations
- 9-pin or 25-pin



- **RS-422**

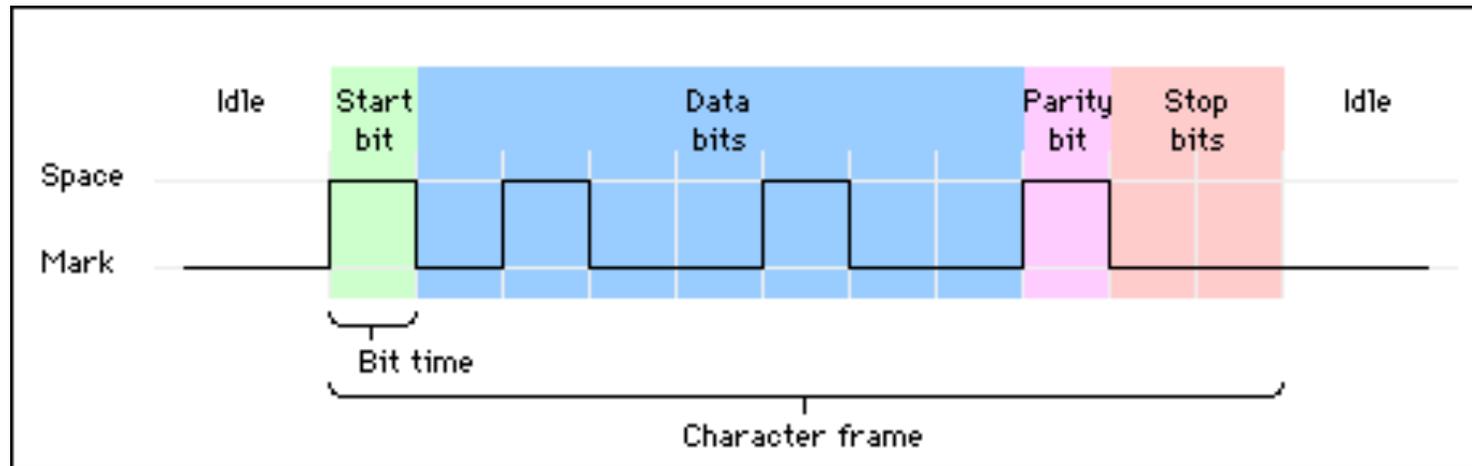
- DCE or DTE
- 8-pin

- **RS-485**

- Multidrop

Pin	DTE	DCE
1 DCD	Input	Output
2 RxD	I	O
3 TxD	O	I
4 DTR	O	I
5 Com	-	-
6 DSR	I	O
7 RTS	O	I
8 CTS	I	O
9 RI	I	O

Serial Communication



Terminology

- Baud rate – bits per second
- Data bits – inverted logic and LSB first
- Parity – optional error-checking bit
- Stop bits – 1, 1.5, or 2 inverted bits at data end
- Flow control – hardware and software handshaking options

Front Panel to Access USB

VISA resource name
COM1

Serial Settings

baud rate
9600

data bits
8

parity
None

stop bits
1.0

flow control
None 0

XON/XOFF Characters

XON
DC1

XOFF
DC3

End Read on Termination Character?

End Write on Termination Character?

Termination Character
xA
0xA is equivalent to a '\n' character, which is a line feed.

Write

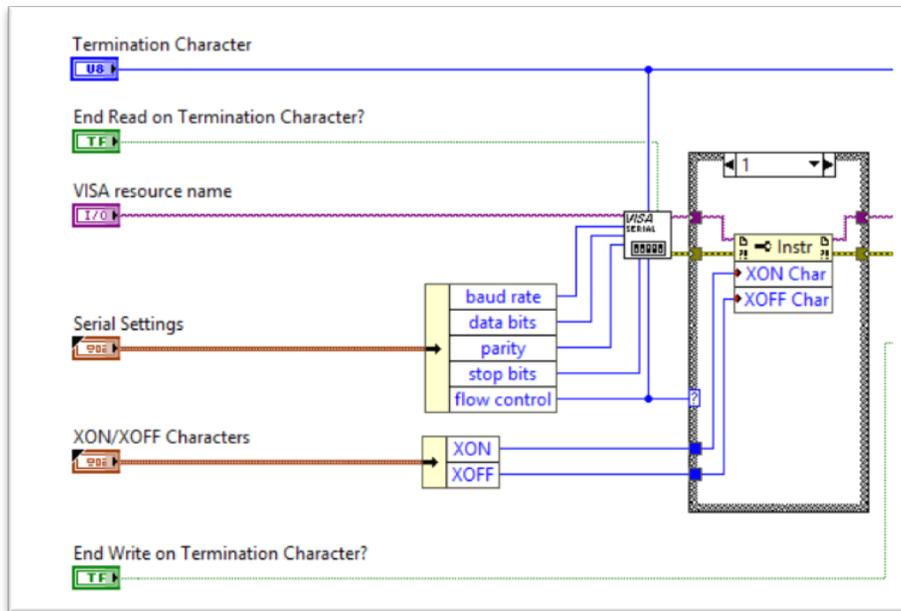
Command
*IDN?
n

Read Count
1000

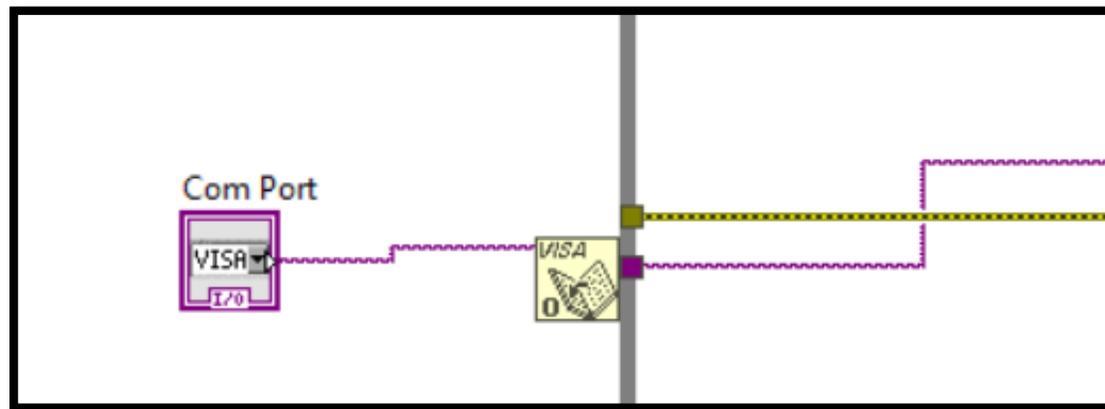
Read

Response
n

Stop

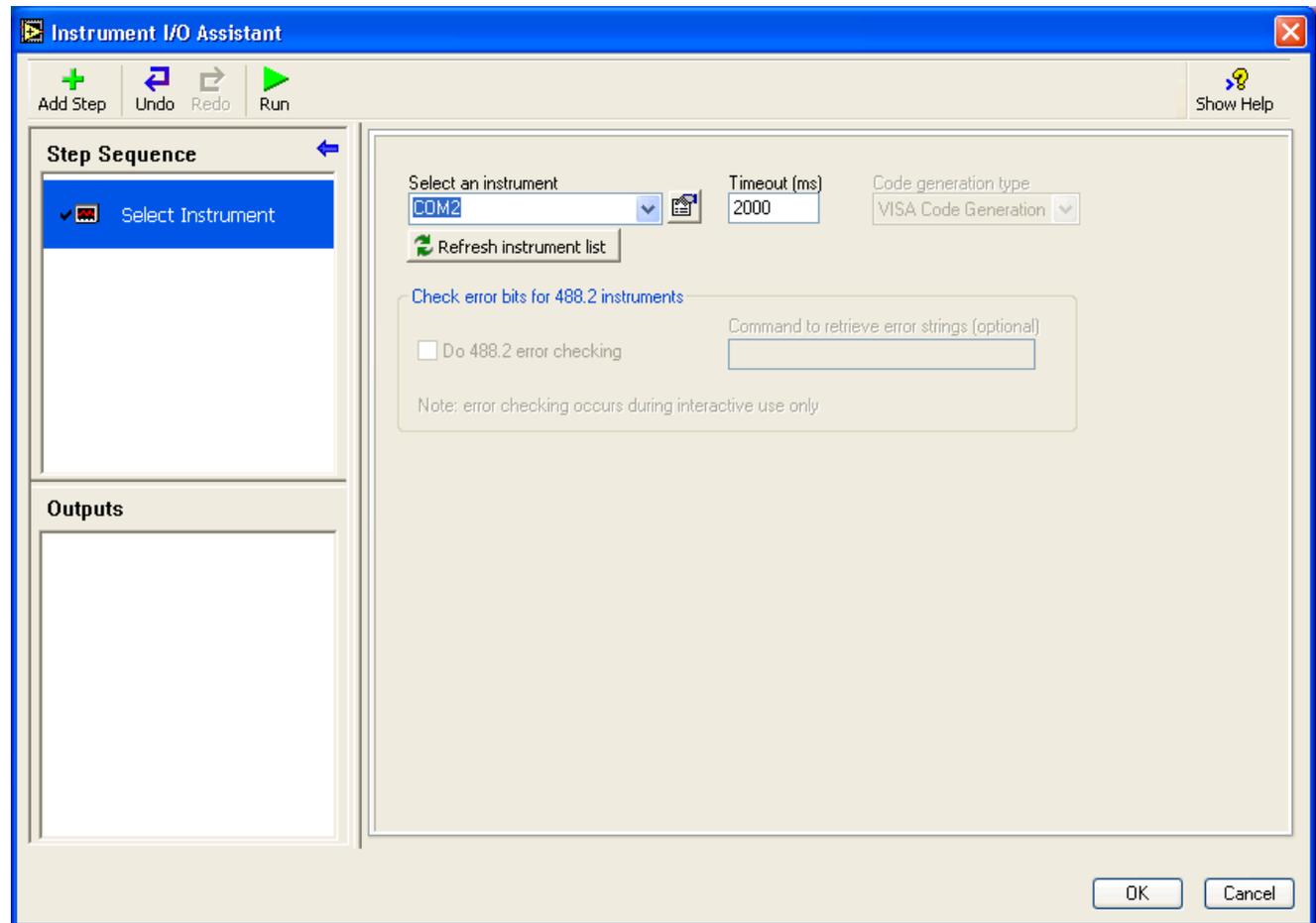


The Virtual Instrument Software Architecture (**VISA**) is a standard for configuring, programming,



Using the Instrument I/O Assistant with Serial

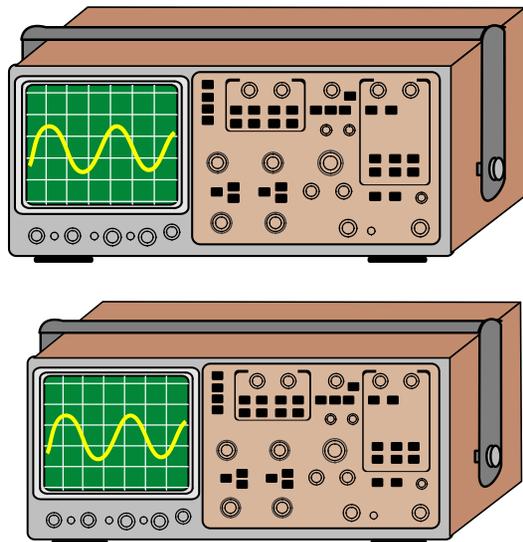
- Select COMX as the instrument address
- Use the I/O Assistant as done with GPIB



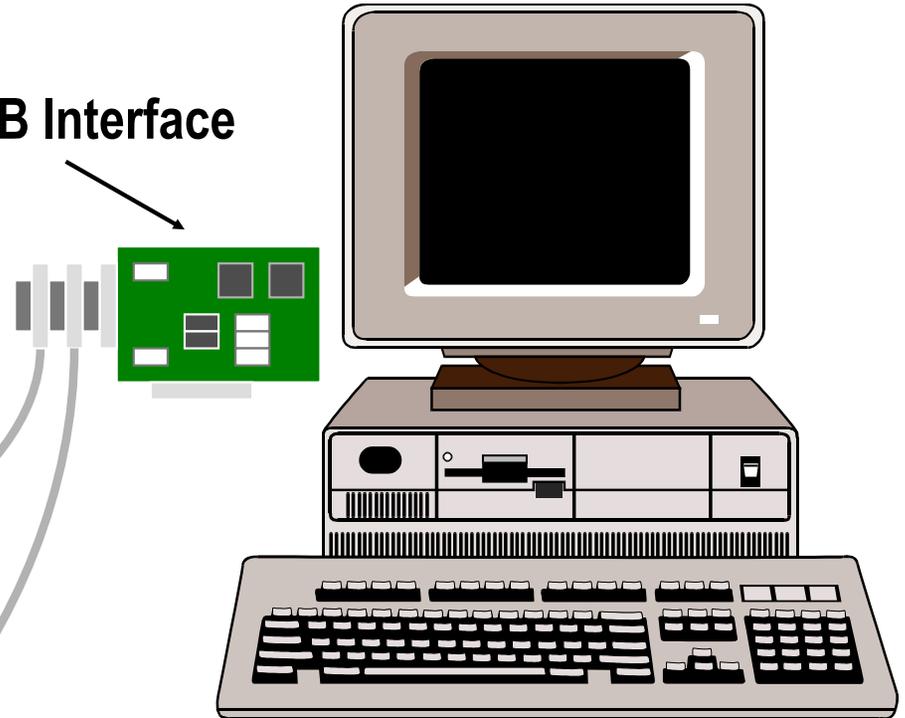
GPIB Communication

IEEE 488 was created as HP-IB (Hewlett-Packard Interface Bus) and is commonly called GPIB (**General Purpose Interface Bus**).

GPIB Instruments

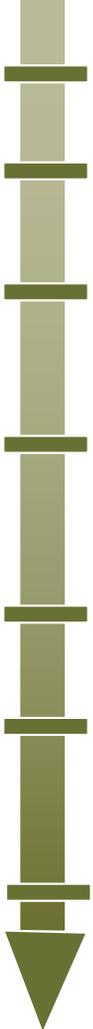


GPIB Interface



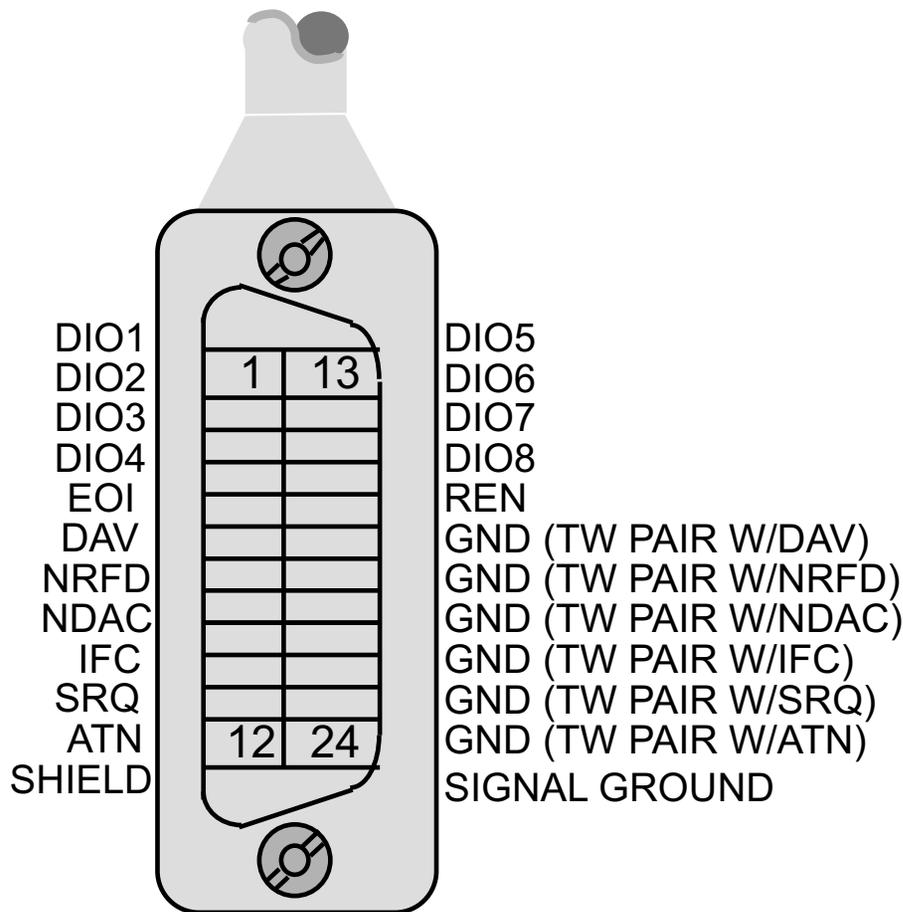
GPIB Cable

Standards Introduction



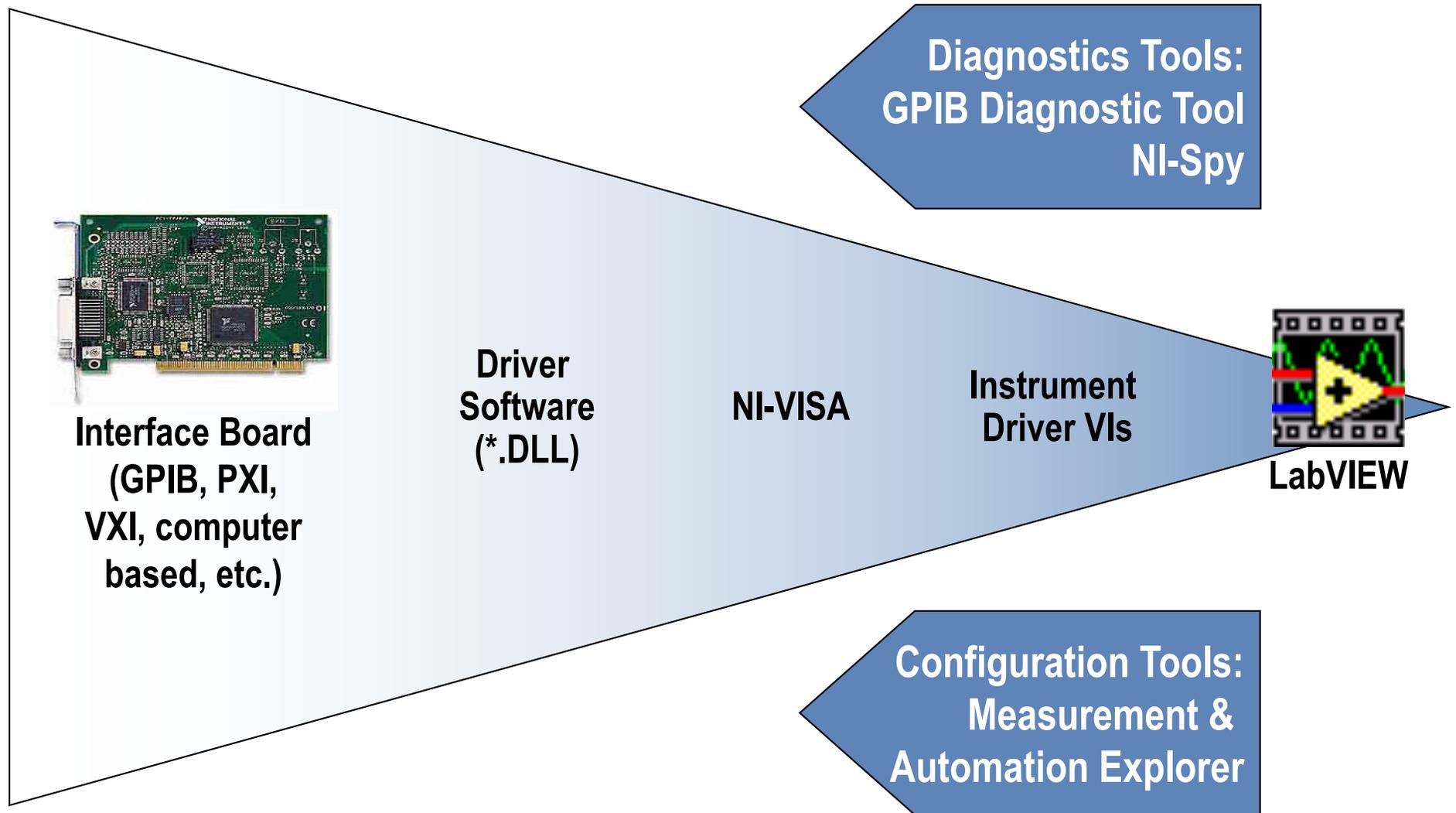
1965	HP designs HP-IB (Hewlett Packard Interface Bus)
1975	HP-IB becomes IEEE 488
1987	IEEE 488.2 adopted IEEE 488 becomes IEEE 488.1
1990	SCPI (Standard Commands for Programmable Instruments) added to IEEE 488.2
1992	IEEE 488.2 revised
1993	HS488 proposed
1999	HS488 approved

GPIB Hardware Specifications

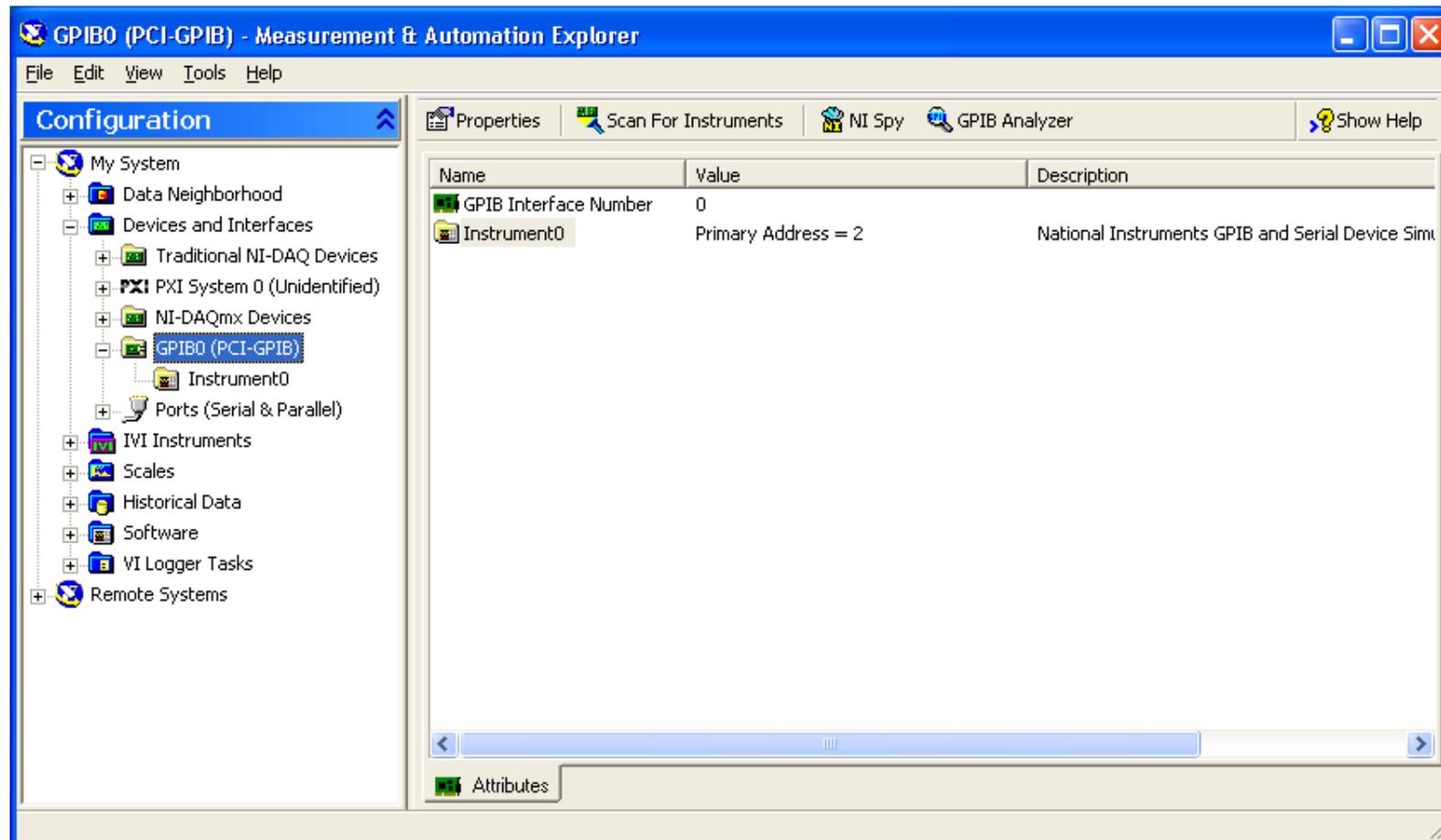


- Max cable length between devices = 4 m (2 m average)
- Max cable length = 20 m
- Max number of devices = 15 (2/3 powered on)

GPIB Software Architecture — Windows



Configuring GPIB Board and Instruments



Measurement & Automation Explorer (MAX)

What is the Instrument I/O Assistant?

- Accessed through a LabVIEW Express VI
- Sets up device communication and data parsing step by step through a configuration interface

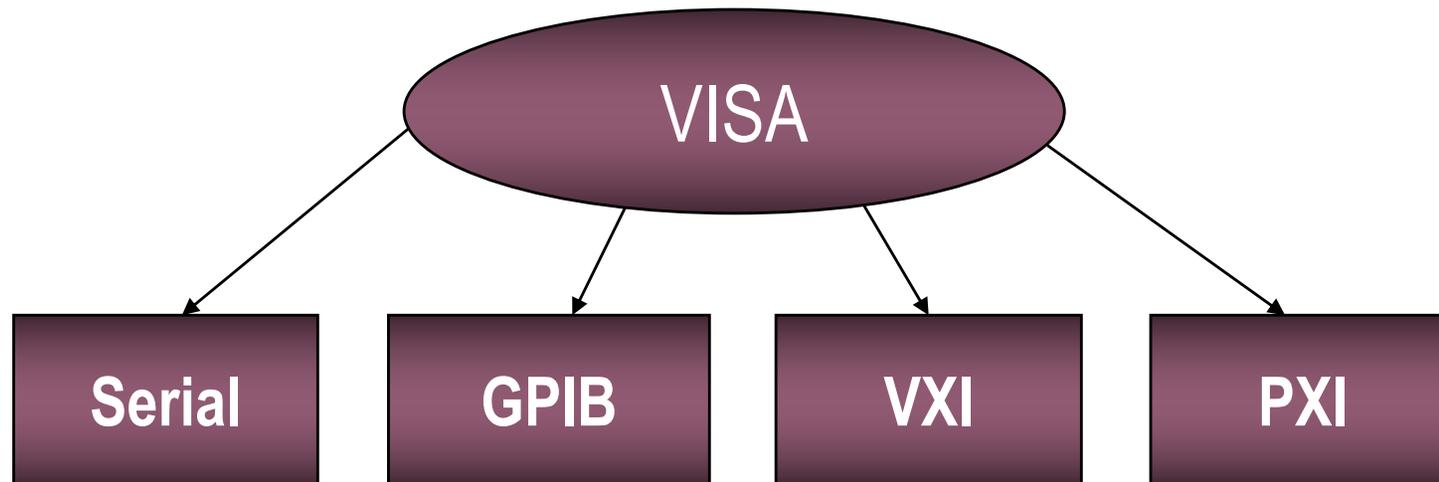


Communicating with an Instrument

The screenshot shows the 'Instrument I/O Assistant' window. The 'Step Sequence' pane on the left has 'Query and Parse' selected. The main area shows a command 'IDN?' entered in the 'Enter a command' field. The 'Termination character' is set to '\n'. Below this, there are tabs for 'Auto parse' and 'Parsing help', with 'Auto parse' selected. A table displays the binary and ASCII representations of the received data. The 'Bin and ASCII' dropdown is set to 'Bin and ASCII', and the 'Byte order' is 'Big Endian (Motorola)'. The 'Selected Token Settings' section shows a token named 'Token' with a 'String' data type and a character count of 61. The 'Value' field displays the text: 'National Instruments GPIB and Serial Device Simulator Rev B.1'.

Byte index	Binary representation	ASCII representation
000000000000:	4E 61 74 69 6F 6E 61 6C 20 49 6E 73	National Ins
000000000012:	74 72 75 6D 65 6E 74 73 20 47 50 49	truments GPI
000000000024:	42 20 61 6E 64 20 53 65 72 69 61 6C	B and Serial
000000000036:	20 44 65 76 69 63 65 20 53 69 6D 75	Device Simu

Virtual Instrument Software Architecture



- Platform independent
- VISA is the backbone of the IVI and Plug & Play Instrument Drivers
- Interface independent
- Must know SCPI command set to program directly with VISA

VISA Terminology

- **Resource**—Instrument, Serial Port, or Parallel Port
- **Session**—Connection to a Resource
- **Instrument Descriptor**—Resource location
 - Format: Interface Type::Address::INSTR

– Examples:

GPIB0:1::INSTR
GPIB0:4::INSTR
GPIB0:10::INSTR
ASRL1::INSTR
ASRL2::INSTR
ASRL3::INSTR
ASRL10::INSTR

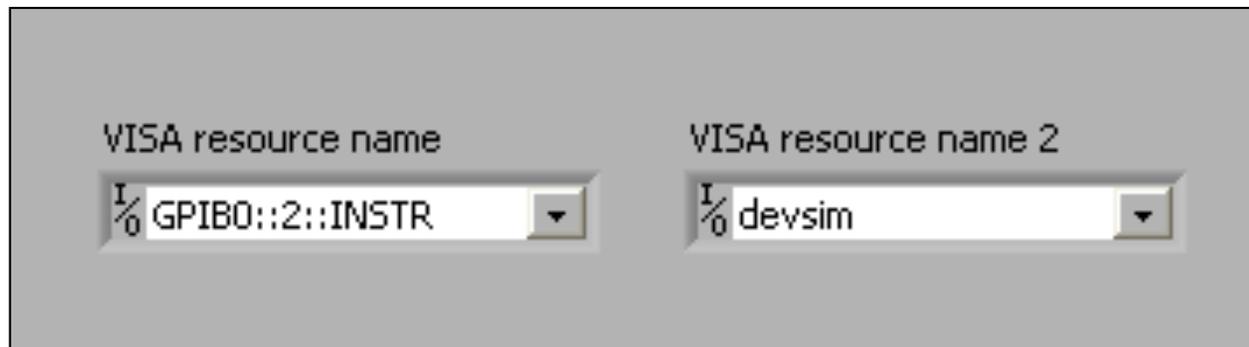
Instrument Descriptor Syntax

- Resource Name contains interface info
- VISA Aliases also work

Interface	Resource Name Grammar
Serial	ASRL [board] [:: INSTR]
GPIB	GPIB [board] :: <i>primary address</i> [:: INSTR]
VXI	VXI [board] :: <i>VXI logical address</i> [:: INSTR]
GPIB-VXI	GPIB-VXI [board] [:: <i>GPIB-VXI primary address</i>] :: <i>VXI logical address</i> [:: INSTR]

VISA Resource Name

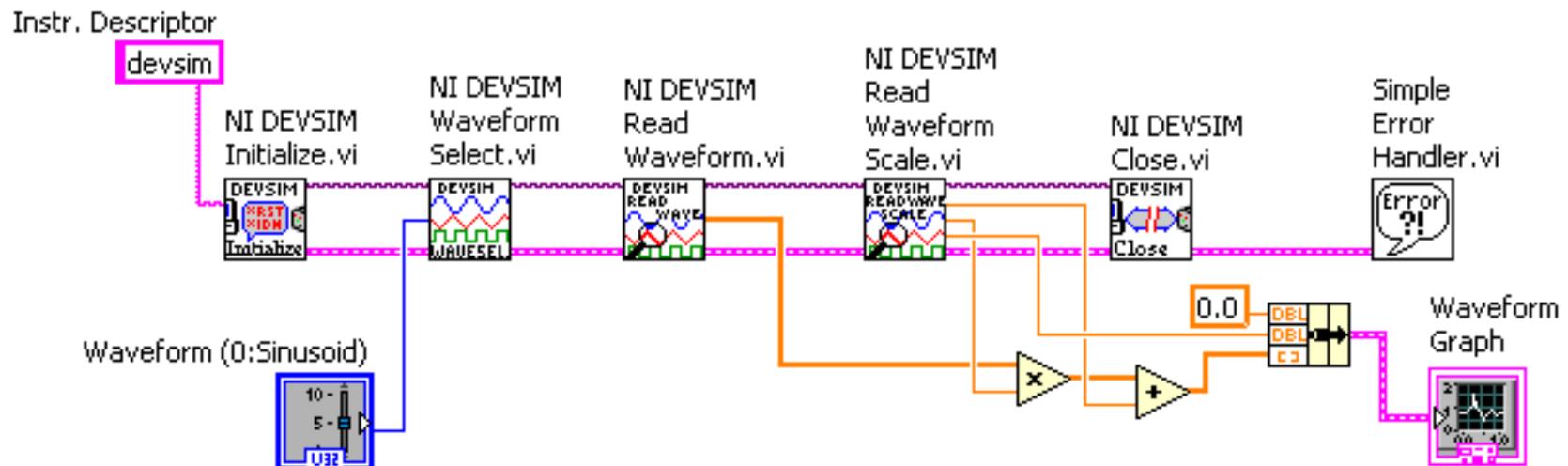
- Exact name and location of the instrument
- Use the VISA Resource Name control
- You can specify the full resource name of the VISA Alias



The image shows a screenshot of a software interface with two dropdown menus. The first dropdown is labeled "VISA resource name" and contains the text "GPIB0::2::INSTR". The second dropdown is labeled "VISA resource name 2" and contains the text "devsim". Both dropdowns have a small downward-pointing arrow on the right side.

Instrument Drivers

- More than 1200 LabVIEW Instrument drivers
- Programming simplified to high-level API



Installing and Finding Instrument Drivers

- Drivers available at ni.com/idnet
- Install the instrument driver VI Library into **LabVIEW 7.0\instr.lib** directory
- Access drivers from **Functions»Input»Instrument Drivers** subpalette



IDNET - Instrument Driver Network

NI Developer Zone 

Search:

Instrument Drivers

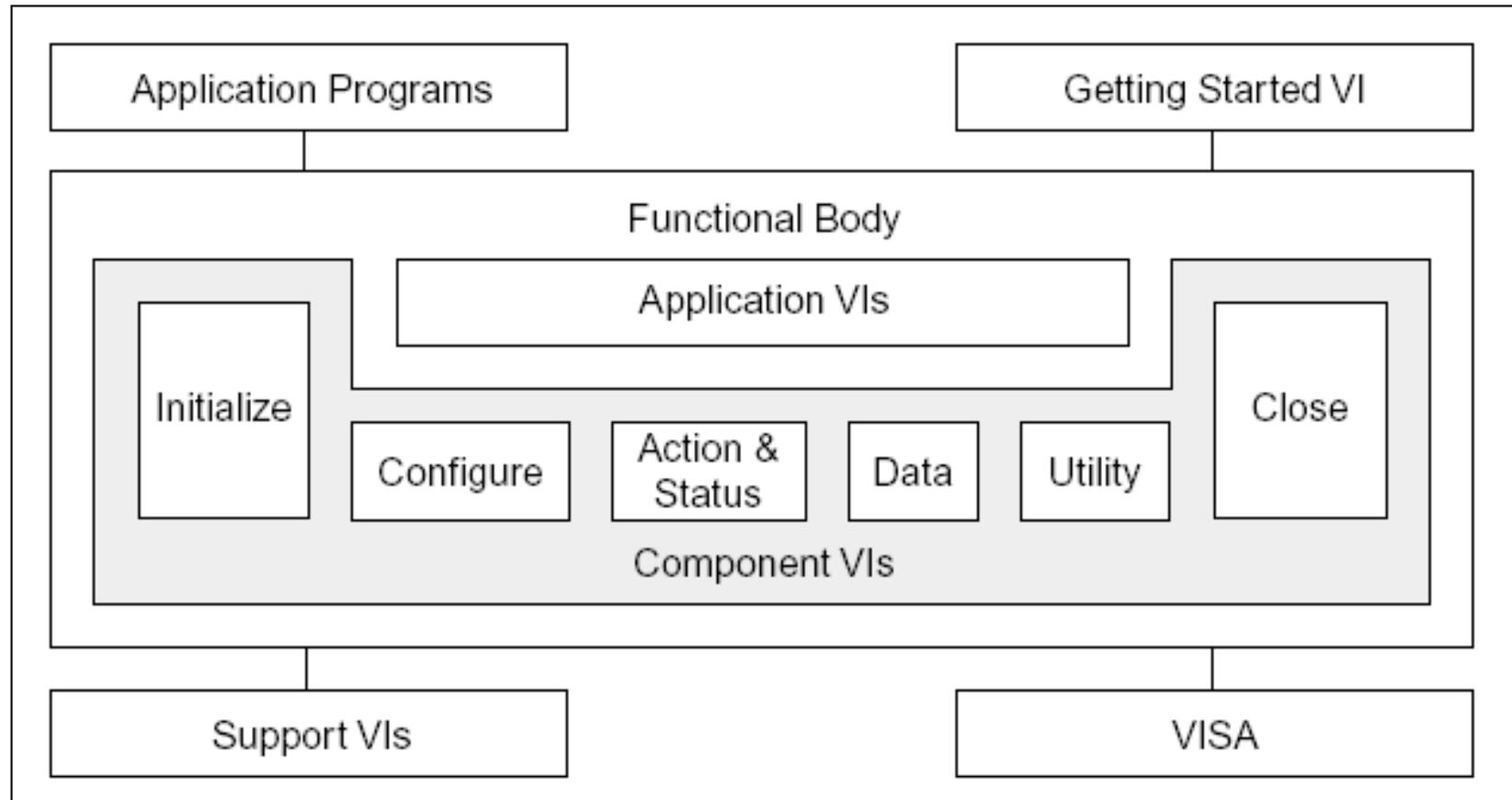
<< NI Developer Zone

- Instrument Driver Network
 - Download
 - Learn
 - Develop
 - Other Driver Resources
 - Driver Request

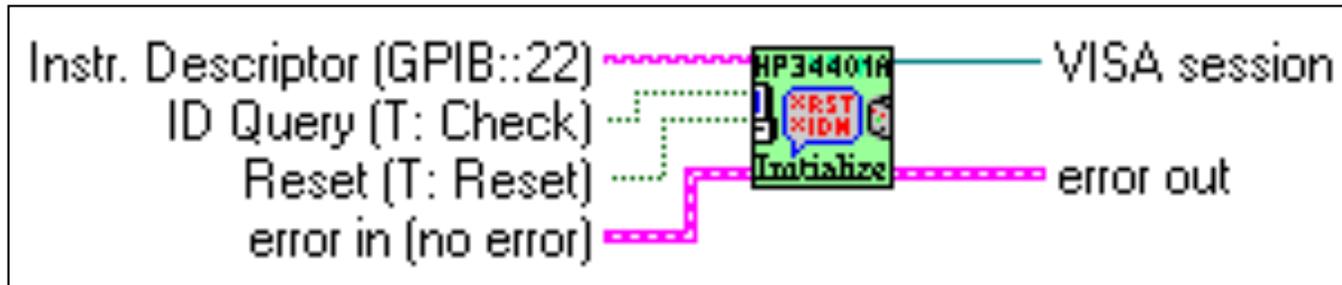
 View Cart

- Learn about drivers
- Get help with developing drivers
- Submit your driver to the network
- Download drivers

Instrument Driver Model



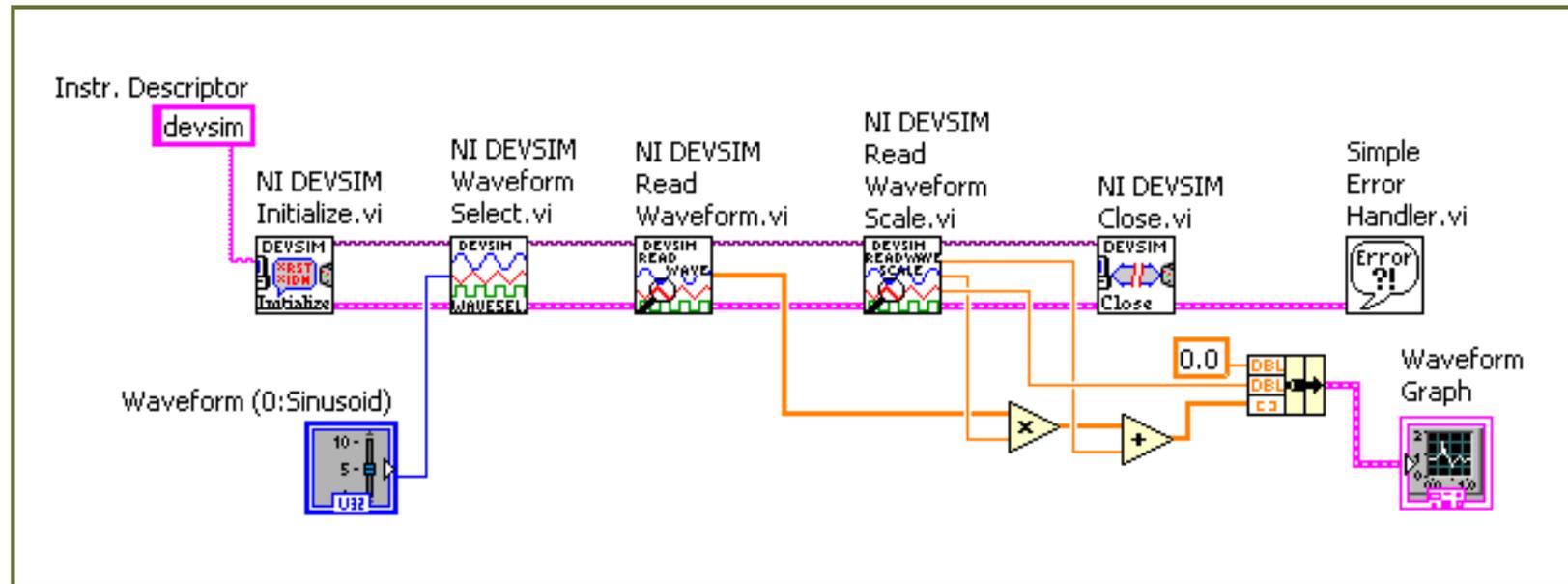
Instrument Driver Inputs and Outputs



HP34401A Initialize.vi

- Instrument Descriptor
- VISA Sessions
 - A connection or link to a specific instrument
 - Created after instrument is initialized
 - Used throughout VI whenever you communicate with that specific instrument
- Error cluster

Putting It All Together



- Initialize instrument
- Do operation(s)
- Close instrument
- Check error status

Summary

- LabVIEW can communicate with any instrument that connects to your computer if you know the interface type
- Use the Measurement & Automation Explorer (MAX) to detect, configure, and test your GPIB interface and instruments
- Use the Instrument I/O Assistant for easy and fast GPIB and serial programming.
- An instrument driver eliminates the need for your to have detailed knowledge of the specific strings used by an instrument
- Instrument Library – more than 2000 instruments supported
- Instrument driver VIs share a common hierarchy and come with an example to help you get started