

Application Note



5800 Series- VISA Instrument Control

Reference: AN501-02



Introduction

The Aeroflex Integrated Development Environment (AIDE) provides a comprehensive programming platform capable of controlling a vast array of devices in addition to the 5800 Automatic Test Equipment (ATE) hardware.

The Virtual Instrument Software Architecture (VISA) is a Test and Measurement industry standard Application Programming Interface (API) for communicating with devices over specific Input/Output (I/O) interfaces. At present these comprise GPIB (IEEE488), VXI, PXI, cPCI, PCI, Ethernet, USB and Serial (RS232/422) interfaces.

These notes describe how VISA compatible devices may be controlled from the AIDE in a .NET environment.

Functionality

The AIDE communicates with the 5800 ATE via the National Instruments (NI) Multisystem eXtension Interface bus (MXIbus). The API used to control this hardware interface is the NI implementation of VISA. To communicate with GPIB devices an NI IEEE488.2 card is added to the PC controller.

The VISA API provides mechanisms for communicating with both register based and message based devices. A large number of intelligent stand-alone devices are message based, the majority these days being Standard Commands for Programmable Instrumentation (SCPI) compliant.

In a .NET environment the VISA functionality is provided through two .NET Assemblies with the names NationalInstruments.VisaNS and NationalInstruments.Common. By adding a direct reference to each of these in an AIDE program the user can access all the available object types and methods necessary to communicate with a VISA interfaced devices.

Program Example 1 – Message Based Devices

Create a new Program called, for example, AIDEVisa.

Detailed information on the NI VISA API can be obtained from the installed documentation by selecting:-

Start\All Programs\National Instruments\VISA\NI-VISA Help

Or by using a .NET Reflector tool (see Lutz Roeder's at <http://www.aisto.com/roeder/dotnet/>) to examine the public contents of the NI Assemblies. The AIDE also uses .NET Reflection to show the available functionality.

Configuration Data

Expand the Configuration Data section of the program and add the following .NET Assembly References from the .NET Global Assembly Cache (GAC) using the browse facilities provided:-

```
Assembly:NationalInstruments.VisaNS,
```

```
Assembly:NationalInstruments.Common,
```

Add the following .NET Namespace References:-

```
NationalInstruments.VisaNS
```

Global Symbols

Create a new Global Variable of type `MessageBasedSession` called `visaMsgSession` and assign it an `InitialValue = null`. The available NI-VISA data types can be found and selected by clicking the button in the Type property window and pressing the More Types... button and scrolling the subsequent display contents until the required data type can be seen and selected.

Create another new Global Variable of type `string` called `result` and also assign an `InitialValue=null`.

Main Method

Create a new Method called `Main` and reference it in the Program OnStart Property.

We can now create Evaluate statements that can be used to communicate with the majority of message based VISA compatible devices.

GPIB (IEEE 488) Devices

To communicate with a message based GPIB device connected via the first (only) GPIB Interface Card in the PC controller on GPIB Primary Address 2, insert a new Evaluate Method Code statement as follows:-

```
Evaluate visaMsgSession = new  
    MessageBasedSession("GPIB0::2::INSTR")
```

This statement should open a VISA message based session via the GPIB0 interface card to the device on GPIB Primary Address 2.

To send a command to the device now that we have an open session we insert a statement of the form :-

```
Evaluate visaMsgSession.Write("ABC 123")
```

To receive a response from the device, now that we have an open session, we insert a statement of the form:-

```
Evaluate result = visaMsgSession.ReadString()
```

For example, if the target instrument is a relatively old device such as the DATRON 1061 Digital Multi-Meter, the following sequence :-

```
Evaluate visaMsgSession.Write("O0Q1T2F1R0@")
```

```
Evaluate result = visaMsgSession.ReadString()
```

results in the DATRON being set up as shown in the table below.

The returned value will be a series of characters representing the measured resistance value e.g. "1.23456e3" represents a value of 1.23456 kOhm.

Command	Description
O0	Output format (floating point decimal)
Q1	Generate SRQ when data is ready
T2	Respond to Group Execute Trigger or Immediate (@) command.
F1	Function 1 = RESISTANCE MEASUREMENT
R0	Range 0 = AUTORANGE
@	Execute Immediately

For a newer instrument that is SCPI compliant we can positively identify the device by sending the "*IDN?" command, e.g.:-

```
Evaluate visaMsgSession.Write("*IDN?")  
    Evaluate id = visaMsgSession.ReadString()
```

Where `id` is a previously declared `string` variable.

Using an Alias

It is possible, using National Instruments Measurement and Automation eXplorer (MAX) to set up aliases for particular device addresses e.g. GPIB0::2::INSTR could be assigned a value of DATRON.

The initial connection statement could then become:-

```
Evaluate visaMsgSession = new  
    MessageBasedSession("DATRON")
```

Flexible Connection Addressing

Many devices can be controlled over more than one interface. For example the IFR 29xx Series of Radio Test Set can be controlled via the GPIB or a Serial (RS232) connection. Assuming the serial connection is via COM port 5, the VISA device address string becomes "ASRL5::INSTR". The alternative addressing can be stored in the AIDE test program or can be set up externally using NI-MAX so that the physical connection used is immaterial to the operation of the test program.

Note that GPIB devices that do NOT conform to the SCPI specification are not readily identifiable.

Program Example 2 – Register Based Devices

PXI Register Based devices such as the Aeroflex 5800 instrument cards (MFC, PUC, TPR) are only identifiable by their Manufacturer ID Number and Manufacturer Model Number. Additional information is supplied to the National Instruments MXI/VISA environment by providing suitable ".inf" files. These supply the necessary details not available from simply reading the PCI/cPCI/PXI registers on the cards when they are plugged into a suitable backplane.

Having done this, the NI-MAX tool is able to display the devices in a PXI Rack configuration and allow aliases to be set up as for the GPIB and Serial devices.

N.B. VXI devices are also often register based.

Global Symbols

Create a new Global Variable of type RegisterBasedSession called visaRegSession and assign it an InitialValue = null. The available NI-VISA data types can be found and selected by clicking the button in the Type property window and pressing the More Types... button and scrolling the subsequent display contents until the required data type can be seen and selected.

Main Method

We can now create Evaluate statements that can be used to communicate with register based VISA compatible devices.

PXI Devices

To communicate with a register based device connected via an MXI Interface Card connected to a single PXI backplane with the card in PXI slot 7, insert a new Evaluate Method Code statement as follows:-

```
Evaluate visaRegSession = new  
RegisterBasedSession("PXI0::7::INSTR")
```

This statement should open a VISA register based session via the MXI interface card to the device in slot 7 of PXI backplane (rack) 0.

To send a command to the device now that we have an open session we insert a statement of the form:-

```
Evaluate visaRegSession.Out32(PxiBar3,Address,  
Data)
```

Where Address and Data are 32 bit integer values.

To receive a response from the device, now that we have an open session, we insert a statement of the form :-

```
Evaluate reg32bit = visaRegSession.In32(PxiBar3,  
Address)
```

Where Address is a 32 bit integer value and reg32bit is a 32 bit integer variable.

With this level of access the user requires very detailed knowledge of the device architecture and the address spaces used.

Using an Alias

It is again possible, using National Instruments Measurement and Automation eXplorer (MAX), to set up aliases for particular device addresses e.g. PXI0::7::INSTR could be assigned a value of MFC.

The initial connection statement could then become:-

```
Evaluate visaRegSession = new RegisterBased  
Session("MFC")
```

None of the above examples depend on having a specific Device Driver installed for the particular instrument being controlled. The user must have access to the necessary hardware architecture and programming information in order to be able to set up and use the device as required.

For information on controlling instruments via a vendor supplied Device Driver see 5800 Series Application Note AN502-01.

CHINA Beijing

Tel: [+86] (10) 6539 1166
Fax: [+86] (10) 6539 1778

CHINA Shanghai

Tel: [+86] (21) 5109 5128
Fax: [+86] (21) 5150 6112

CHINA Shenzhen

Tel: [+86] (755) 3301 9358
Tel: [+86] (755) 3301 9356

FINLAND

Tel: [+358] (9) 2709 5541
Fax: [+358] (9) 804 2441

FRANCE

Tel: [+33] 1 60 79 96 00
Fax: [+33] 1 60 77 69 22

GERMANY

Tel: [+49] 89 99641 0
Fax: [+49] 89 99641 160

HONG KONG

Tel: [+852] 2832 7988
Fax: [+852] 2834 5364

INDIA

Tel: [+91] 80 [4] 115 4501
Fax: [+91] 80 [4] 115 4502

JAPAN

Tel: [+81] (3) 3500 5591
Fax: [+81] (3) 3500 5592

KOREA

Tel: [+82] (2) 3424 2719
Fax: [+82] (2) 3424 8620

SCANDINAVIA

Tel: [+45] 9614 0045
Fax: [+45] 9614 0047

SINGAPORE

Tel: [+65] 6873 0991
Fax: [+65] 6873 0992

UK Stevenage

Tel: [+44] (0) 1438 742200
Fax: [+44] (0) 1438 727601
Freephone: 0800 282388

USA

Tel: [+1] (316) 522 4981
Fax: [+1] (316) 522 1360
Toll Free: 800 835 2352

As we are always seeking to improve our products, the information in this document gives only a general indication of the product capacity, performance and suitability, none of which shall form part of any contract. We reserve the right to make design changes without notice. All trademarks are acknowledged. Parent company Aeroflex, Inc. ©Aeroflex 2011.

www.aeroflex.com
info-test@aeroflex.com



Our passion for performance is defined by three attributes represented by these three icons: solution-minded, performance-driven and customer-focused.