Control of Mini-Circuits' Portable Test Equipment (PTE) Using VISA

1 - VISA Overview

VISA (Virtual Instrument Software Architecture) is a commonly used I/O (Input/Output) API (Advanced Programming Interface) for communication with test and measurement equipment. It is intended to provide a common interface, allowing test instrumentation from a variety of manufacturers to be controlled from a single software environment, using common processes.

The standard is managed by the IVI foundation (http://ivifoundation.org/), however there are various differing implementations available from test equipment manufacturers. This application note will focus on 2 of the most common VISA software implementations within the RF/Microwave test and measurement market:

- Keysight IO Libraries Suite, using Keysight Connection Expert
- National Instruments (NI) VISA, using NI's Measurement & Automation Explorer

2 - VISA Interface Options

The initial step to controlling a test instrument or device through VISA is creating an interface or "driver"; this is a configuration within the VISA software suite that defines the communication method and address of the physical test instrument.

The simplest method for Mini-Circuits' PTE products is to define the device as a LAN (Local Area Network) instrument (for Ethernet enabled models only). The device must be connected into the Ethernet network and the VISA software interface just needs to be configured with the device's IP address and port number. For Mini-Circuits PTE devices it is best to use port 23, reserved for Telnet communication.

The alternative connection method for Mini-Circuits' PTE devices is to configure as a USB instrument. The advantage is that all Mini-Circuits PTE models support this connection method but the configuration process is a little more involved and requires knowledge of the raw USB connection parameters.

3 - Mini-Circuits PTE Products as LAN Instruments

3.a - Configuring as a LAN Instrument

This section details the process for configuring any Mini-Circuits PTE product to be controlled through VISA as a LAN instrument, using the device's RJ45 port to connect to an Ethernet network.

3.a.i - Configuration using Keysight Connection Expert

1. Connect the PTE device to the local network and note the IP address
2. Launch Keysight Connection Expert
3. Navigate to Manual Configuration > Add New Instruments/Interfaces > LAN Instrument

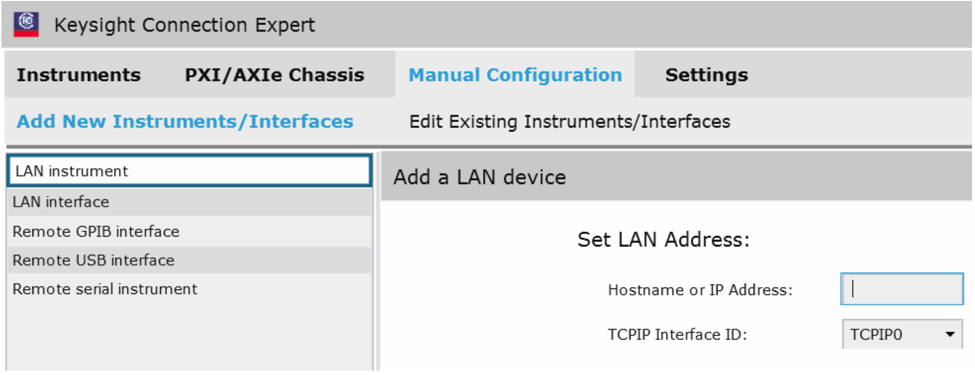
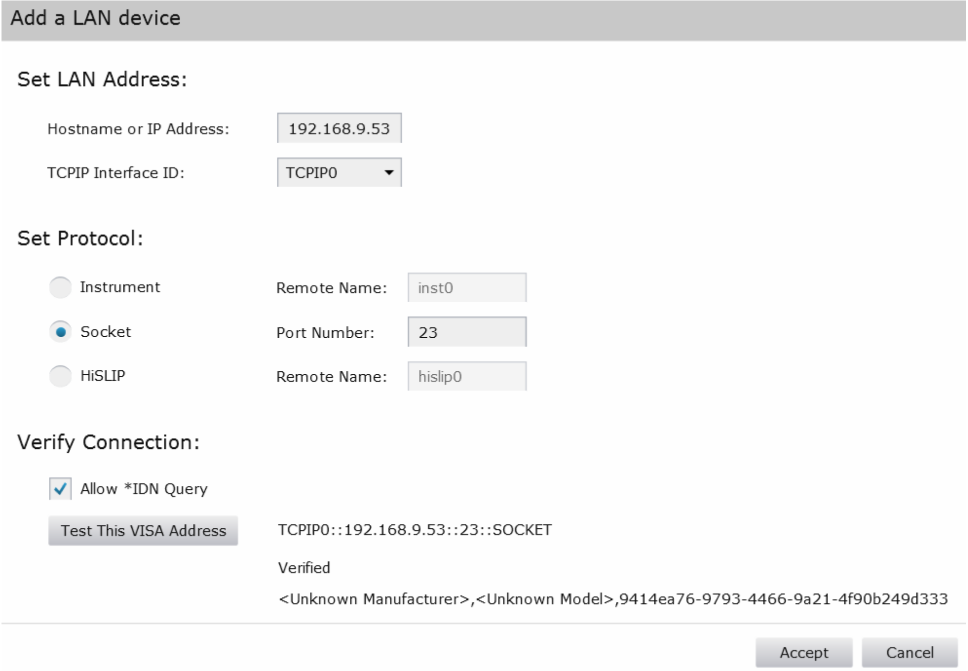


Fig i - Adding a new LAN instrument

4. Enter the IP address of the device in the "Hostname or IP Address" field
5. "TCPIP Interface ID" should usually be left as the default, this is the ID for the LAN itself
6. In the "Set Protocol" section, select "Socket" and enter port number "23"
7. Click "Test This VISA Address" to confirm the connection settings are correct



Add a LAN device

Set LAN Address:

Hostname or IP Address: 192.168.9.53

TCPIP Interface ID: TCPIP0

Set Protocol:

○ Instrument Remote Name: inst0

● Socket Port Number: 23

○ HiSLIP Remote Name: hislip0

Verify Connection:

☑ Allow *IDN Query

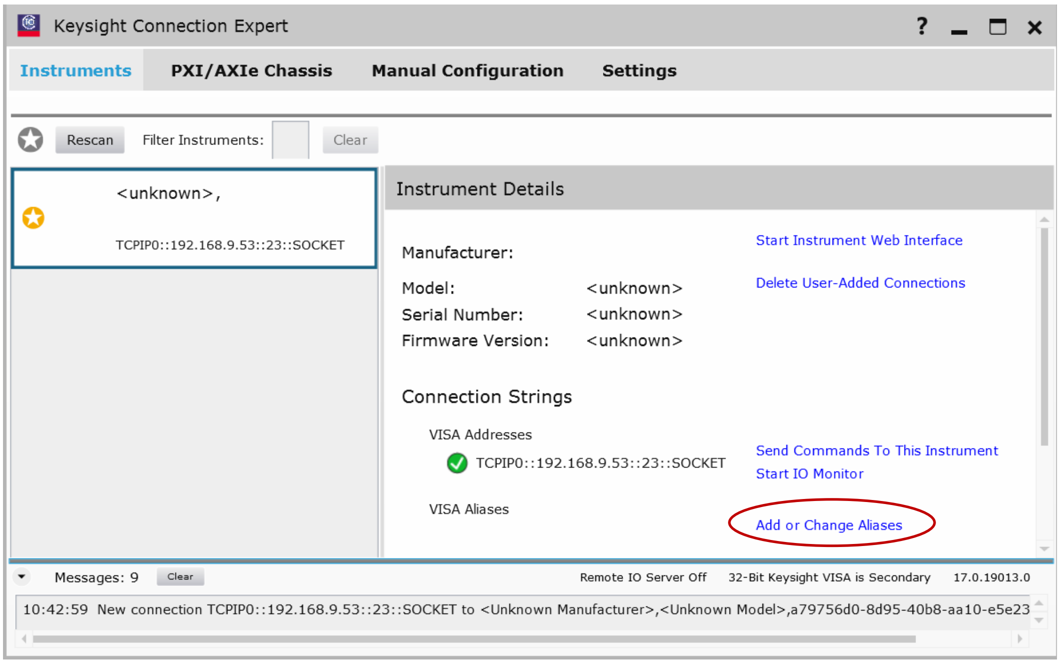
Test This VISA Address TCPIP0::192.168.9.53::23::SOCKET

Verified

<Unknown Manufacturer>,<Unknown Model>,9414ea76-9793-4466-9a21-4f90b249d333

Fig ii - The successfully tested LAN device settings for a power sensor with IP address 192.168.9.53

8. Click Accept to save the configuration and Connection Expert will return to the home screen where the new instrument will be listed along with any other known VISA devices.
9. The Mini-Circuits PTE device will be listed as "Unknown" but will be identifiable by the IP address
10. To add an "alias" in order to make the device more easily identifiable, click on the device and then select "Add or Change Aliases" from the summary screen



<unknown>,
TCPIP0::192.168.9.53::23::SOCKET

Instrument Details

Manufacturer:
Model: <unknown>
Serial Number: <unknown>
Firmware Version: <unknown>

Connection Strings

VISA Addresses
✓ TCPIP0::192.168.9.53::23::SOCKET

VISA Aliases

Messages: 9 — Remote IO Server Off — 32-Bit Keysight VISA is Secondary — 17.0.19013.0

```
10:42:59  New connection TCPIP0::192.168.9.53::23::SOCKET to <Unknown Manufacturer>,<Unknown Model>,a79756d0-8d95-40b8-aa10-e5e23
```

Fig iii - Connection Expert home screen summarising the new LAN instrument (Add or Change Aliases is highlighted)

11. Enter the chosen Alias Name and make sure the correct VISA Address string is selected if there are multiple VISA instruments defined
12. The device is now configured and ready to use with a VISA address that takes the form "[LAN_Interface]::[IP_Address]::[Port]::SOCKET"

3.a.ii - Configuration using NI MAX

1. Connect the PTE device to the local network and note the IP address
2. Launch NI MAX (Measurement & Automation Explorer)
3. In the navigation column on the left of the screen, expand "My System" and then "Devices and Interfaces"
4. Rick-click on "Network Devices" and select "Create New VISA TCP/IP Resource"

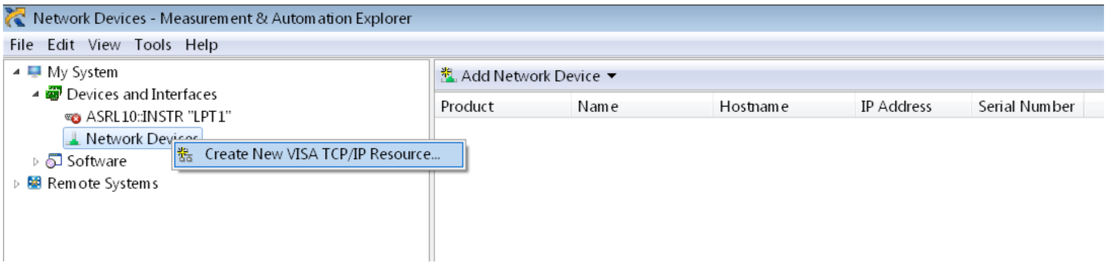
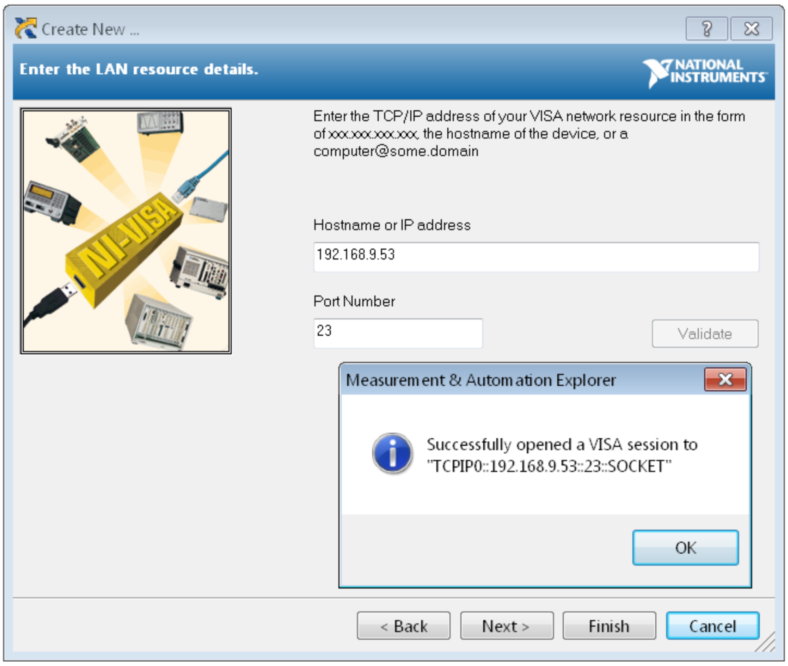


Fig iv - Creating a new VISA TCP/IP Resource in NI MAX

5. Select "Manual Entry of Raw Socket" from the pop-up screen and click Next
6. Enter the IP address and port number "23", then click Validate to test the connection



Enter the TCP/IP address of your VISA network resource in the form of xxx.xxx.xxx.xxx, the hostname of the device, or a computer@some.domain

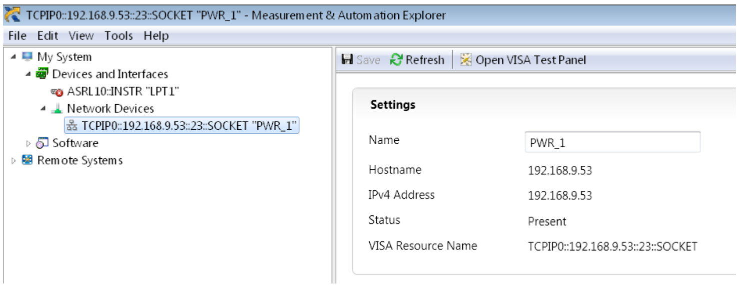
Hostname or IP address: 192.168.9.53

Port Number: 23

Successfully opened a VISA session to "TCPIP0::192.168.9.53::23::SOCKET"

Fig v - Successfully testing the LAN configuration

7. Click Finish to return to the home screen with the new device now listed under the "Network Devices" heading
8. The device configuration has an optional "alias" Name field which can be used to identify the device, just enter a name and click Save


TCPIP0::192.168.9.53::23::SOCKET "PWR_1" - Measurement & Automation Explorer
File Edit View Tools Help
My System
Devices and Interfaces
ASRL10::INSTR "LPT1"
Network Devices
TCPIP0::192.168.9.53::23::SOCKET "PWR_1"
Software
Remote Systems
Save
Refresh
Open VISA Test Panel
Settings

Name	PWR_1
Hostname	192.168.9.53
IPv4 Address	192.168.9.53
Status	Present
VISA Resource Name	TCPIP0::192.168.9.53::23::SOCKET

Fig vi - The LAN device summary with PWR_1 entered as an alias name

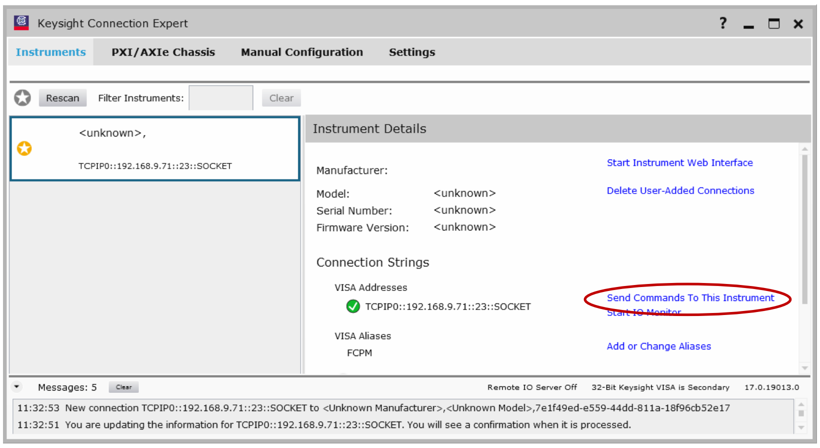
9. The device is now configured and ready to use with a VISA address that takes the form "[LAN_Interface]::[IP_Address]::[Port]::SOCKET"

3.b - Control of a LAN Instrument Using the Keysight / NI Software Suites

The software suites from Keysight and National Instruments both provide a simple interface with which commands can be sent to Mini-Circuits PTE products once they have been configured as LAN instruments.

3.b.i - Sending Commands using Keysight Connection Expert

1. Open Keysight Connection Expert, select the appropriate LAN instrument and click "Send Commands To This Instrument" to open the Interactive IO interface



<unknown>,
TCPIP0::192.168.9.71::23::SOCKET

Manufacturer:
Model: <unknown>
Serial Number: <unknown>
Firmware Version: <unknown>

Connection Strings

VISA Addresses
TCPIP0::192.168.9.71::23::SOCKET

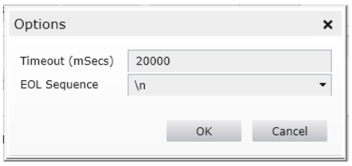
VISA Aliases
FCPM

Remote IO Server Off 32-Bit Keysight VISA is Secondary 17.0.19013.0

```
11:32:53  New connection TCPIP0::192.168.9.71::23::SOCKET to <Unknown Manufacturer>,<Unknown Model>,7e1f49ed-e559-44dd-811a-18f96cb52e17
11:32:51  You are updating the information for TCPIP0::192.168.9.71::23::SOCKET. You will see a confirmation when it is processed.
```

Fig vii - Identifying the LAN instument in Keysight Connection Expert

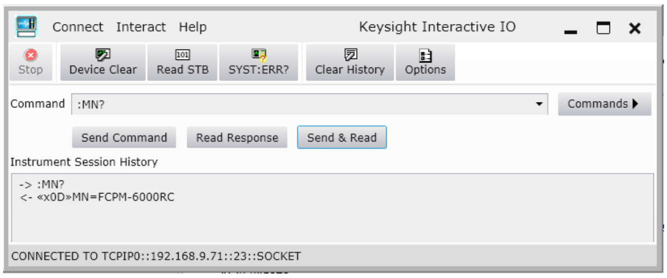
2. Click on the options tab and set the "EOL Sequence to "\n", this is the terminating character to be sent at the end of each command



Timeout (mSecs) 20000
EOL Sequence \n

Fig viii - Setting the EOL Sequence

3. The full list of commands/queries available is listed in the programming manual for the Mini-Circuits PTE product, available for download from the Mini-Circuits website; enter these in the command section and click "Send & Read" to see the response
4. To query the model name of the device, enter ":MN?\n" and then click "Send & Read"



Command :MN?

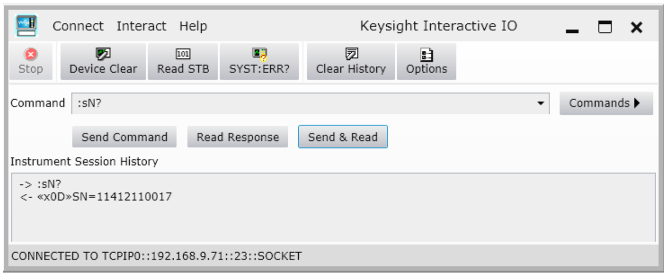
Instrument Session History

```
-> :MN?
<- «x0D»MN=FCPM-6000RC
```

CONNECTED TO TCPIP0::192.168.9.71::23::SOCKET

Fig ix - Requesting the model name; the response is FCPM-6000RC

5. To query the serial number of the device, enter ":SN?\n" and then click "Send & Read"



Command :sN?

Instrument Session History

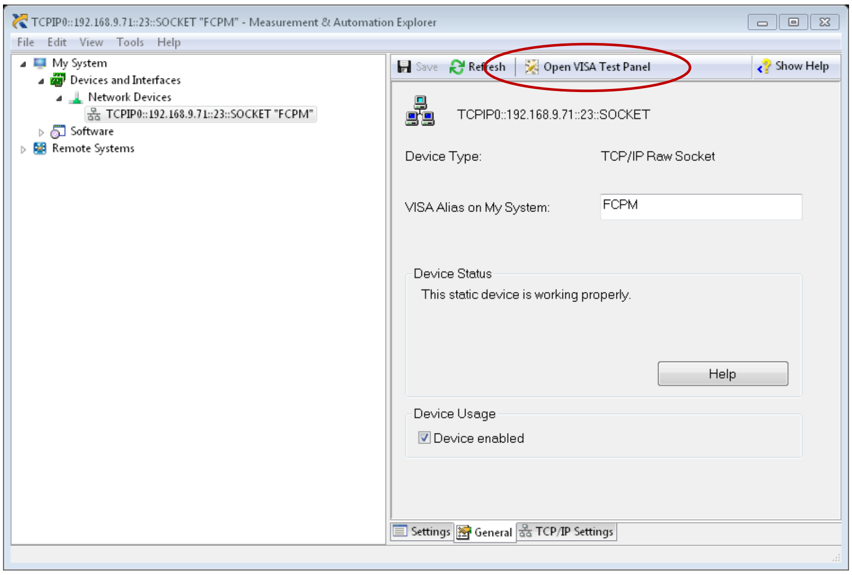
```
-> :sN?
<- «x0D»SN=11412110017
```

CONNECTED TO TCPIP0::192.168.9.71::23::SOCKET

Fig x - Requesting the serial number; the response is 11412110017

3.b.ii - Sending Commands Using NI MAX

1. Open NI MAX, select the appropriate LAN instrument from the list and click "Open VISA Test Panel" to open the interactive communication window



TCPIP0::192.168.9.71::23::SOCKET

Device Type: TCP/IP Raw Socket

VISA Alias on My System: FCPM

Device Status

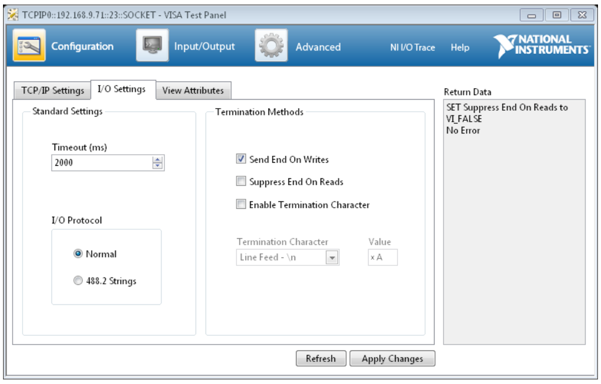
This static device is working properly.

Device Usage

☑ Device enabled

Fig xi - Opening the VISA Test Panel in NI MAX

2. Select the I/O Settings tab of the Configuration screen, uncheck the "Supress End On Reads" tick box and click "Apply Changes"



Standard Settings

Timeout (ms): 2000

I/O Protocol
● Normal
○ 488.2 Strings

Termination Methods

☑ Send End On Writes
□ Suppress End On Reads
□ Enable Termination Character

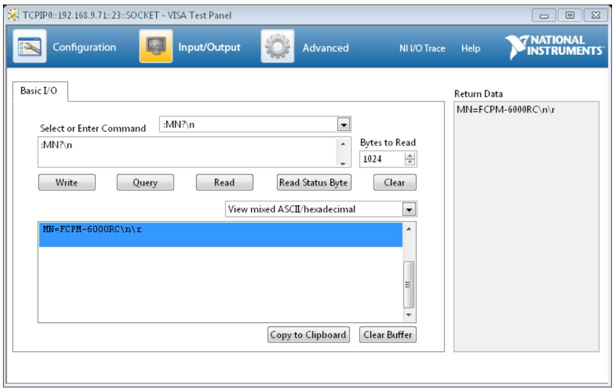
Termination Character: Line Feed - \n
Value: x A

Return Data

```
SET Suppress End On Reads to
VI_FALSE
No Error
```

Fig xii - Uncheck the "Suppress End on Reads" option

3. Select the Input/Output tab in order to send commands/queries to the device
4. Commands can be written in the space below the "Select or Enter Command" label:
    a. The full list of commands available is listed in the programming manual for the Mini-Circuits PTE product, available for download from the Mini-Circuits website
    b. All commands must end with a new line character ("\n")
5. To query the model name of the device, enter ":MN?\n" and then click "Query"



Select or Enter Command: :MN?\n

```
:MN?\n
```

Bytes to Read: 1024

View mixed ASCII/hexadecimal

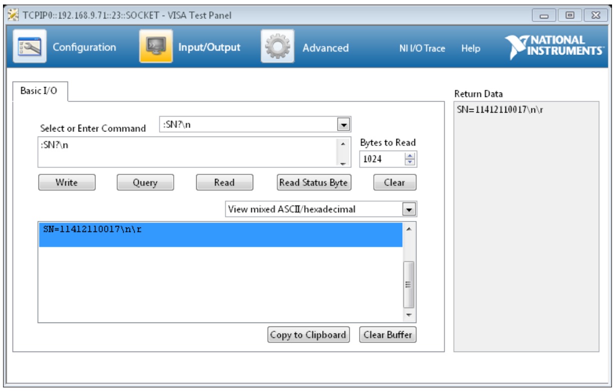
```
MN=FCPM-6000RC\n\r
```

Return Data

```
MN=FCPM-6000RC\n\r
```

Fig xiii - Requesting the model name; the response (FCPM-6000RC) is highlighted

6. To query the serial number of the device, enter ":SN?\n" and then click "Query"



Select or Enter Command: :SN?\n

```
:SN?\n
```

Bytes to Read: 1024

View mixed ASCII/hexadecimal

```
SN=11412110017\n\r
```

Return Data

```
SN=11412110017\n\r
```

Fig xiv - Requesting the serial number; the response (1141211001) is highlighted

3.c - Programmatic Control of a LAN Instrument

It is a simple process to communicate with Mini-Circuits' PTE devices configured as VISA LAN Instruments in most programming environments. Communication is achieved using a message based session defined in the National Instruments API.

This communication process is applicable to all Mini-Circuits' PTE products but detailed examples for each are also available on request.

3.c.i - Example Using VB.NET for Control of a Power Sensor

```
Imports NationalInstruments.VisaNS                       ' Import the VISA namespace

~~~~~~~~~~~~~~~~~~~~~~~

Dim mbSession As MessageBasedSession                     ' Create a new message based session

' Open the message based session using the VISA connection string of the device
mbSession = CType(ResourceManager.GetLocalManager().Open("TCPIP0::192.168.9.71::23::SOCKET"),
                                                       MessageBasedSession)
mbSession.TerminationCharacterEnabled = True             ' Important

Dim textToWrite As String = ""
Dim stSerialNo As String
Dim stModelName As String

textToWrite = ":SN?\n\r"                                 ' The text string to send (get serial no)
textToWrite.Replace("\n", vbLf).Replace("\r", vbCr)      ' Remove escape sequences
stSerialNo = mbSession.Query(textToWrite)                ' The query must be sent twice in VB
stSerialNo = mbSession.Query(textToWrite)

textToWrite = ":MN?\n\r"                                 ' The text string to send (get model name)
textToWrite.Replace("\n", vbLf).Replace("\r", vbCr)      ' Remove the /n/r characters
stModelName = mbSession.Query(textToWrite)               ' The query must be sent twice in VB
stModelName = mbSession.Query(textToWrite)

mbSession.Dispose()                                      ' Close the session

MsgBox(stModelName & " " & stSerialNo)
```

3.c.ii - Example Using C# for Control of a Power Sensor

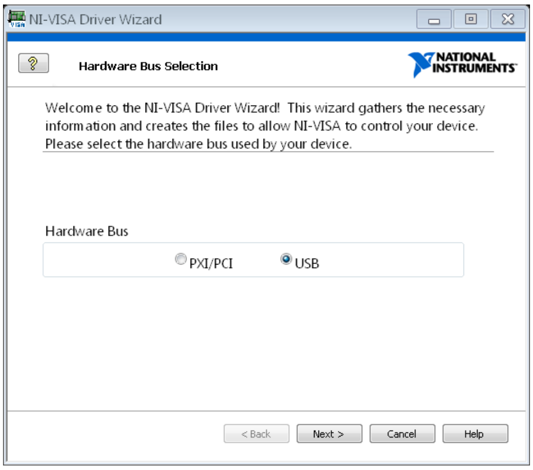
```
using NationalInstruments.VisaNS;                        // Use the VISA namespace

~~~~~~~~~~~~~~~~~~~~~~~

MessageBasedSession mbSession;                           // Create a new message based session

mbSession = (MessageBasedSession)ResourceManager.GetLocalManager().Open(
                                                                     _ "TCPIP0::10.0.6.4::23::SOCKET");

mbSession.TerminationCharacterEnabled = true;            // Important

string textToWrite = ""
string stSerialNo = ""
string stModelName = ""

textToWrite = ":SN?\n\r"                                 // The text string to send (get serial no)
TextToWrite.Replace("\\n", "\n").Replace("\\r", "\r")    // Remove escape sequences
stSerialNo = mbSession.Query(textToWrite);               // Send the query

textToWrite = ":MN?\n\r"                                 // The text string to send (get model name)
TextToWrite.Replace("\\n", "\n").Replace("\\r", "\r")    // Remove escape sequences
stModelName = mbSession.Query(textToWrite);              // Send the query


mbSession.Dispose();                                     // Close the session

MessageBox.Show(stModelName + " " + stSerialNo);
```

4 - Mini-Circuits PTE products as USB Instruments

This section details the process for configuring any Mini-Circuits PTE product to be controlled through VISA as a USB instrument, using the device's USB port to connect to a local computer. Since Mini-Circuits' PTE products comply with the USB HID (Human Interface Device) standard the operating system will automatically assign a driver. By configuring the device as a VISA driver, the standard driver will be replaced and so the Mini-Circuits GUI and DLL files will no longer operate with the PTE. This change can subsequently be undone by updating the driver assigned to the device using the Windows Device Manager.

4.a - Creating the VISA Driver Using NI-VISA Driver Wizard

1. Connect the PTE device to the local computer using the USB connection
2. Launch NI-VISA Driver Wizard from the National Instruments VISA program group
3. Select the USB option and click Next



Welcome to the NI-VISA Driver Wizard! This wizard gathers the necessary information and creates the files to allow NI-VISA to control your device. Please select the hardware bus used by your device.

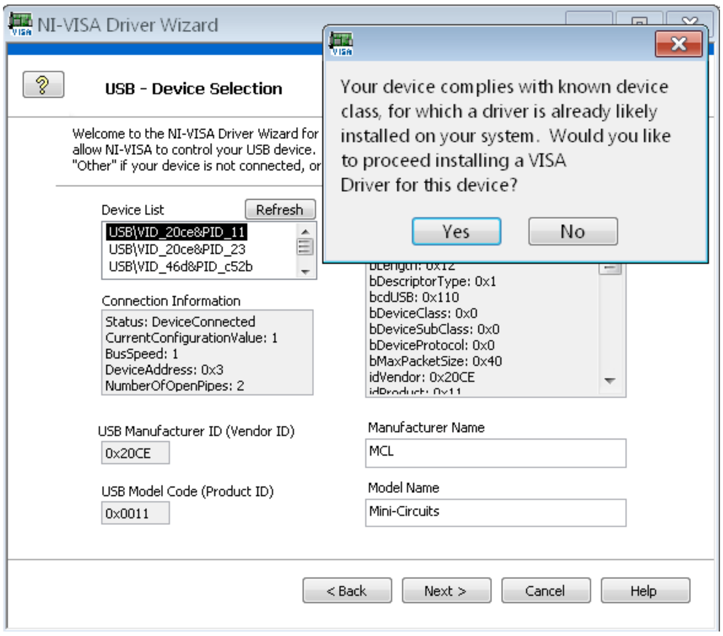
Hardware Bus

○ PXI/PCI ● USB

Fig xv - NI VISA Driver Wizard initial screen

4. Select the correct PTE device from the Device List; the description will begin with *USB\VID_20ce&PID_xx*, where:
   - USB indicates a USB connection
   - VID indicates the vendor ID (*20ce* for Mini-Circuits)
   - PID indicates the product ID:

     10 = frequency counter
     11 = power meter / integrated power & frequency sensor
     12 = signal generator
     22 = switch matrix / switch box / custom switch assembly
     23 = programmable attenuator
5. The Descriptor Information reveals more detail on the USB configuration and the Manufacturer Name will also be populated
6. Click Next and a warning will appear advising that the device already has a driver installed (for USB HID communication), click Yes to acknowledge this and proceed



NI-VISA Driver Wizard

USB - Device Selection

Welcome to the NI-VISA Driver Wizard for
allow NI-VISA to control your USB device.
"Other" if your device is not connected, or

Your device complies with known device class, for which a driver is already likely installed on your system. Would you like to proceed installing a VISA Driver for this device?

Yes No

Device List Refresh

USB\VID_20ce&PID_11
USB\VID_20ce&PID_23
USB\VID_46d8&PID_c52b

Connection Information

Status: DeviceConnected
CurrentConfigurationValue: 1
BusSpeed: 1
DeviceAddress: 0x3
NumberOfOpenPipes: 2

bLength: 0x12
bDescriptorType: 0x1
bcdUSB: 0x110
bDeviceClass: 0x0
bDeviceSubClass: 0x0
bDeviceProtocol: 0x0
bMaxPacketSize: 0x40
idVendor: 0x20CE
idProduct: 0x11

USB Manufacturer ID (Vendor ID): 0x20CE

Manufacturer Name: MCL

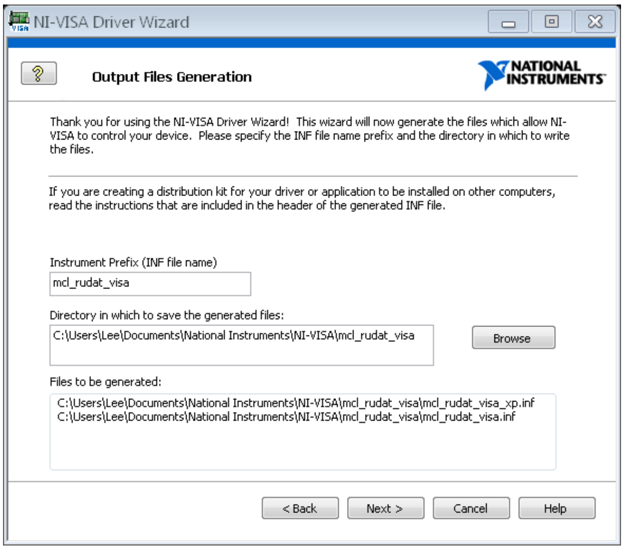
USB Model Code (Product ID): 0x0011

Model Name: Mini-Circuits

< Back | Next > | Cancel | Help

Fig xvi - USB Device Selection screen with warning about pre-installed driver

7. Update the default Instrument Prefix with a more memorable filename for the VISA driver, for example "mcl_rudat_visa", and enter a new directory name if you do not wish to save the files to the default location
8. The Driver Wizard will create 2 drivers, the first with "_xp.inf" appended to the filename for 32-bit operating systems, and the second with ".inf" for 64-bit operating systems; click Next to proceed



NI-VISA Driver Wizard

Output Files Generation

NATIONAL INSTRUMENTS

Thank you for using the NI-VISA Driver Wizard! This wizard will now generate the files which allow NI-VISA to control your device. Please specify the INF file name prefix and the directory in which to write the files.

If you are creating a distribution kit for your driver or application to be installed on other computers, read the instructions that are included in the header of the generated INF file.

Instrument Prefix (INF file name): mcl_rudat_visa

Directory in which to save the generated files: C:\Users\Lee\Documents\National Instruments\NI-VISA\mcl_rudat_visa | Browse

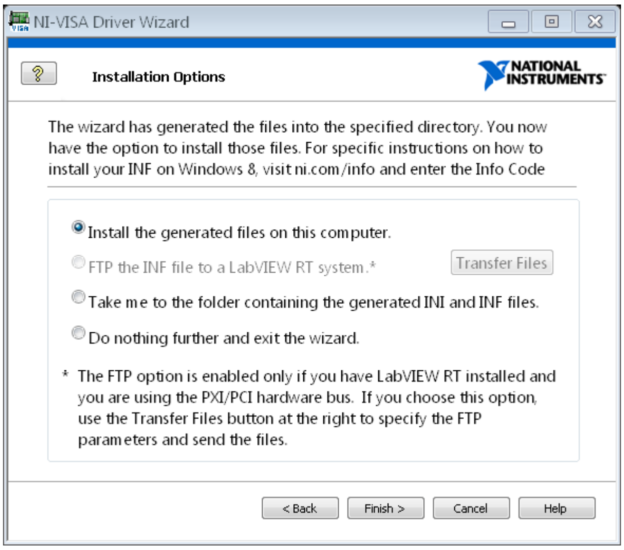
Files to be generated:

C:\Users\Lee\Documents\National Instruments\NI-VISA\mcl_rudat_visa\mcl_rudat_visa_xp.inf
C:\Users\Lee\Documents\National Instruments\NI-VISA\mcl_rudat_visa\mcl_rudat_visa.inf

< Back | Next > | Cancel | Help

Fig xvii - Output Files (drivers) generation screen

9. If the VISA drivers are needed on the same PC on which the wizard was run then select "Install the generated files on this computer" and click Finish



NI-VISA Driver Wizard

Installation Options

The wizard has generated the files into the specified directory. You now have the option to install those files. For specific instructions on how to install your INF on Windows 8, visit ni.com/info and enter the Info Code

● Install the generated files on this computer.

○ FTP the INF file to a LabVIEW RT system.*

○ Take me to the folder containing the generated INI and INF files.

○ Do nothing further and exit the wizard.

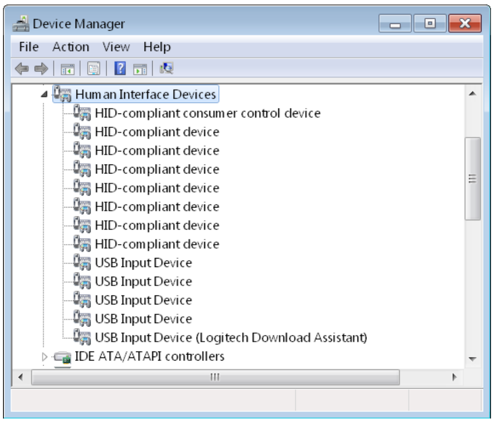
* The FTP option is enabled only if you have LabVIEW RT installed and you are using the PXI/PCI hardware bus. If you choose this option, use the Transfer Files button at the right to specify the FTP parameters and send the files.

Fig xviii - Choose to install the driver or just exit the wizard

10. If the above option was selected, the Windows driver installation wizard will run to install the driver
11. In the event of an error installing the driver, or if the driver is to be installed on a different PC, then follow the next steps

4.b - Installing the USB VISA Driver

1. Connect the PTE device to the local computer using the USB connection
2. Open the Device Manager from the Windows Control Panel and expand the list of Human Interface Devices



Device Manager

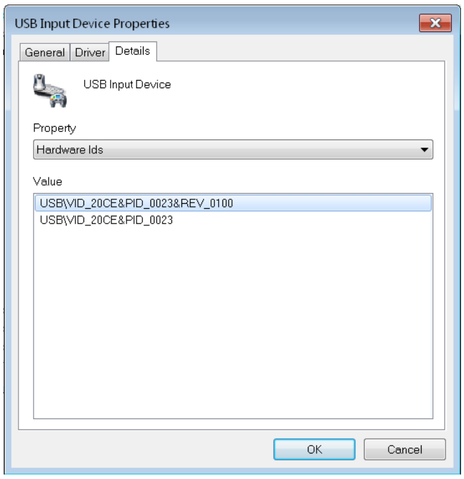
Human Interface Devices
- HID-compliant consumer control device
- HID-compliant device
- HID-compliant device
- HID-compliant device
- HID-compliant device
- HID-compliant device
- HID-compliant device
- HID-compliant device
- USB Input Device
- USB Input Device
- USB Input Device
- USB Input Device
- USB Input Device (Logitech Download Assistant)

IDE ATA/ATAPI controllers

Fig xix - The list of Human Interface Devices in the Windows Device Manager

3. Mini-Circuits PTE devices will be listed as "USB Input Device", alongside any other USB HID devices currently connected to the system
4. To find the correct device, double-click on the "USB Input Device" entries, go to the Details tab and select "Hardware Ids" from the Property list

5. The Value for a Mini-Circuits PTE device will be of the form *USB\VID_20CE&PID_00xx*, where:
    - USB indicates a USB connection
    - VID indicates the vendor ID (*20CE* for Mini-Circuits)
    - PID indicates the product ID:
      10 = frequency counter
      11 = power meter / integrated power & frequency sensor
      12 = signal generator
      22 = switch matrix / switch box / custom switch assembly
      23 = programmable attenuator



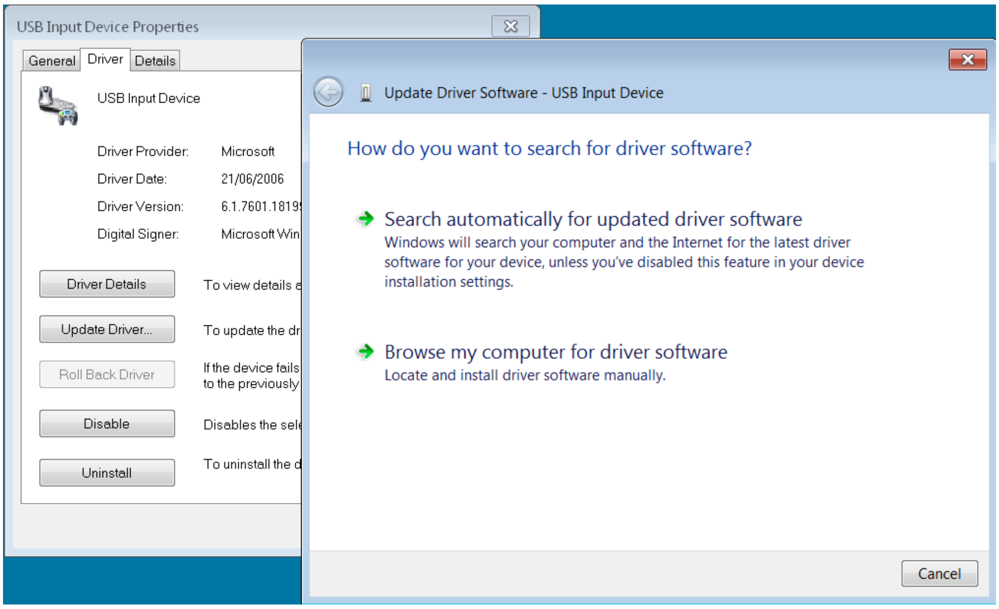
USB Input Device

Property: Hardware Ids

Value:
USB\VID_20CE&PID_0023&REV_0100
USB\VID_20CE&PID_0023

Fig xx - USB Input Device properties window for a Mini-Circuits' programmable attenuator (VID = 20CE, PID = 23)

6. When the correct device is located, select the Driver tab of the Properties window and click Update Driver to install the VISA driver



USB Input Device

Driver Provider: Microsoft
Driver Date: 21/06/2006
Driver Version: 6.1.7601.1819
Digital Signer: Microsoft Win

To view details a
To update the dr
If the device fails to the previously
Disables the sel
To uninstall the d

Update Driver Software - USB Input Device

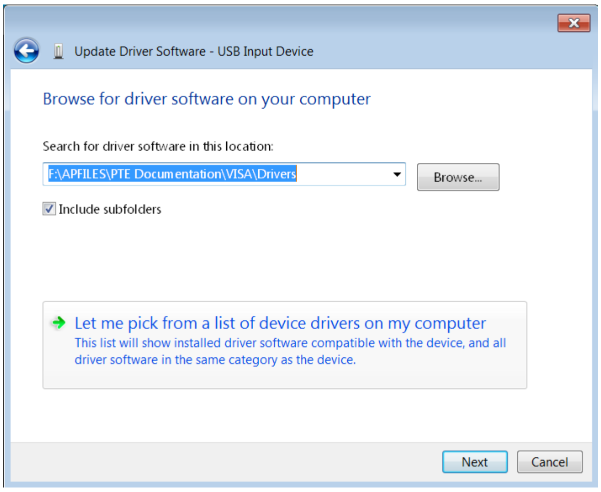
How do you want to search for driver software?

→ Search automatically for updated driver software
Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.

→ Browse my computer for driver software
Locate and install driver software manually.

Fig xxi - Launching the Update Driver wizard

7. Select the option to "Browse my computer for driver software" and then choose "Let me pick from a list of device drivers on my computer"



Update Driver Software - USB Input Device

Browse for driver software on your computer

Search for driver software in this location:

F:\APFILES\PTE Documentation\VISA\Drivers

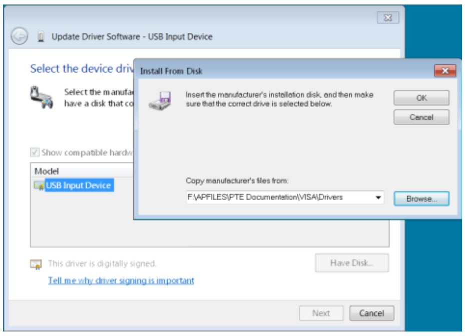
☑ Include subfolders

Let me pick from a list of device drivers on my computer

This list will show installed driver software compatible with the device, and all driver software in the same category as the device.

Fig xxii - Select the location of the VISA driver

8. Click the "Have Disk" button, navigate to the location of the VISA driver file and click OK to accept



Update Driver Software - USB Input Device

Select the device driv

Select the manufa
have a disk that co

☑ Show compatible hardw

Model

USB Input Device

This driver is digitally signed.

Tell me why driver signing is important

Install From Disk

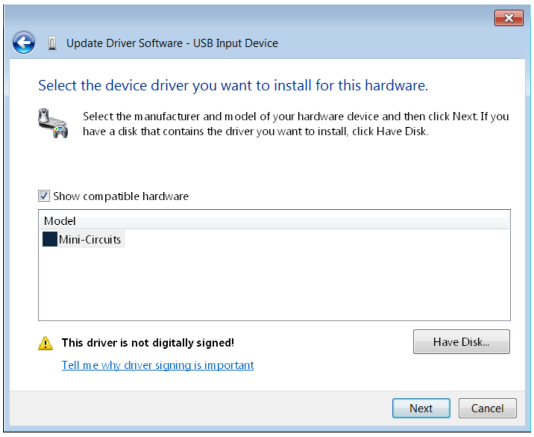
Insert the manufacturer's installation disk, and then make sure that the correct drive is selected below.

Copy manufacturer's files from:

F:\APFILES\PTE Documentation\VISA\Drivers

Fig xxiii - Navigate to the location of the VISA driver

9. The Update Driver selection window will now show "Mini-Circuits" as the model name; click Next to proceed



Update Driver Software - USB Input Device

Select the device driver you want to install for this hardware.

Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk.

☑ Show compatible hardware

Model

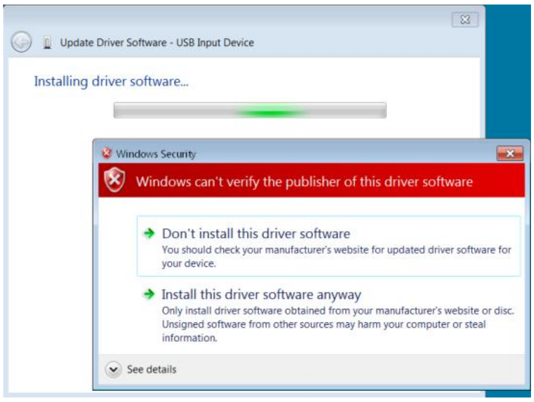
Mini-Circuits

This driver is not digitally signed!

Tell me why driver signing is important

Fig xxiv - The update driver selection

10. A warning may appear to advise that the driver's publisher cannot be verified, select "Install the driver software anyway" to continue



Update Driver Software - USB Input Device

Installing driver software...

Windows Security

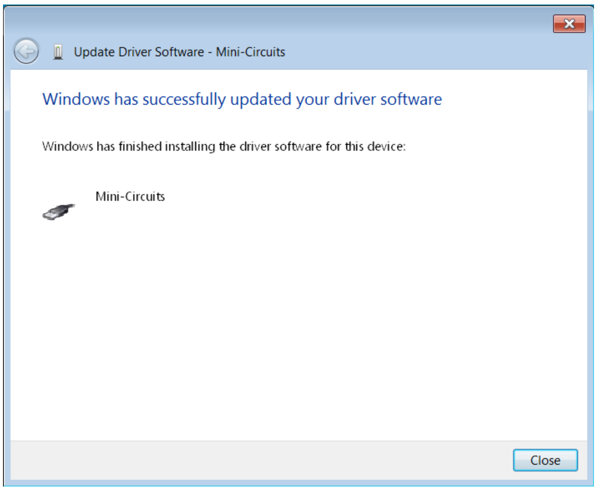
Windows can't verify the publisher of this driver software

Don't install this driver software
You should check your manufacturer's website for updated driver software for your device.

Install this driver software anyway
Only install driver software obtained from your manufacturer's website or disc. Unsigned software from other sources may harm your computer or steal information.

Fig xxv - Driver verification warning

11. Finally a success notification will appear once the driver has been installed for the PTE and the device is ready for VISA communication



Update Driver Software - Mini-Circuits

Windows has successfully updated your driver software

Windows has finished installing the driver software for this device:

Mini-Circuits

Fig xxvi- Final confirmation of driver installation

12. The device should be visible through NI MAX as "USB0::0x20CE::0x00[PID]::NI-VISA-10002::RAW"

4.c - Programmatic Control using the VISA Driver over a USB Connection

Once a USB driver has been configured, the PTE device will have a connection string of the form "USB0::0x20CE::0x00[PID]::NI-VISA-10002::RAW", where [PID] is the device ID. This connection string can be used in a wide range of programming environments, in conjunction with the VISA API provided by National Instruments, to send commands to the device over USB.

Mini-Circuits' PTE devices are based on the USB interrupt pipe whereas VISA uses the USB bulk-in pipe so it is necessary to alter these parameters as one of the first steps in communicating with the device; the steps are:

1. Reference the National Instruments Visa namespace (assuming you are programming in a .net environment)
2. Create the connection to the device using the USB connection string
3. Disable the USB interrupt pointer (set it to value -1)
4. Re-define the USB bulk-in pipe (set it to value 129)

Since these changes can only be made programmatically, it is not possible to control Mini-Circuits PTE through the Agilent and VISA interfaces when using the USB connections.

The commands to set and query the PTE are defined in the programming manuals, available for download from the Mini-Circuits website at http://www.minicircuits.com/support/software_download.html.

The following examples create the VISA connection to a Mini-Circuits power meter using the VISA connection string. The model name and serial number are queried and then a power measurement is returned, compensated for an expected input frequency of 3000 MHz. The list of available commands and queries is explained in the programming manual.

This communication process is applicable to all Mini-Circuits' PTE products but detailed examples for each are also available on request.

4.c.i - Power Sensor Control Using VB.NET

```
Imports NationalInstruments.VisaNS              ' Import the VISA namespace

~~~~~~~~~~~~~~~~~~~~~~~

Dim dev As NationalInstruments.VisaNS.UsbRaw    ' Reference the NI API
Dim StrToSend, StrReturned, ModelName, SerialNo, PowerRead As String

dev = CType(ResourceManager.GetLocalManager.Open("USB0::0x20CE::0x0011::NI-VISA-60001::RAW"), UsbRaw)
dev.InterruptInPipe = -1                        ' Disable the interrupt pointer
dev.BulkInPipe = 129                            ' Re-define the bulk-in pipe
dev.Timeout = 5000                              ' Set a timeout for communication

StrToSend = Chr(104)                            ' Code 104 = Get Model Name
StrReturned = dev.Query(StrToSend, 64)          ' Send the string and receive the response
ModelName = Mid(StrReturned, 2)                 ' Model name starts from the second character

StrToSend = Chr(105)                            ' Code 105 = Get Serial Number
StrReturned = dev.Query(StrToSend, 64)          ' Send the string and receive the response
SerialNo = Mid(StrReturned, 2)                  ' Model name starts from the second character

StrToSend = Chr(102) & Chr(11) & Chr(184)       ' Read Power; 11*256+184=3000 MHz (cal frequency)
StrReturned = dev.Query(StrToSend, 64)          ' Send the string and receive the response
PowerRead = Mid(StrReturned, 2, 7) & " (dBm)"   ' Extract the power reading from the returned string
```

4.c.ii - Power Sensor Control Using C#

```
using NationalInstruments.VisaNS;       // Use the VISA namespace

~~~~~~~~~~~~~~~~~~~~~~~

UsbRaw MyDev;                           // Connect using the VISA connection string
MyDev = (UsbRaw)ResourceManager.GetLocalManager().Open("USB0::0x20CE::0x0011::NI-VISA-60001::RAW");
MyDev.InterruptInPipe = -1;             // Disable the interrupt pointer
MyDev.BulkInPipe = 129;                 // Re-define the bulk-in pipe
MyDev.Timeout = 3000;                   // Set a timeout for communication

byte[] array = { 102, 11, 184 };                        // Read power; 11*256+184=3000 MHz cal
string ToSend = ASCIIEncoding.ASCII.GetString(array);   // Create string to send
string ReadVal = MyDev.Query(ToSend, 64);               // Send command and read response
string PowerRead = ReadVal.Substring(1, 7) + " (dBm)";  // Get the power

array[0] = 104;                                         // 104=Code to read model name
ToSend = ASCIIEncoding.ASCII.GetString(array);
ReadVal = MyDev.Query(ToSend, 64);
string ModelName = ReadVal.Substring(1, 15) ;

array[0] = 105;                                         // 105=Code to read serial number
ToSend = ASCIIEncoding.ASCII.GetString(array);
ReadVal = MyDev.Query(ToSend, 64);
string SerialNo = ReadVal.Substring(1, 15);
```

IMPORTANT NOTICE

This document is provided as an accommodation to Mini-Circuits customers in connection with Mini-Circuits parts only. In that regard, this document is for informational and guideline purposes only. Mini-Circuits assumes no responsibility for errors or omissions in this document or for any information contained herein.

Mini-Circuits may change this document or the Mini-Circuits parts referenced herein (collectively, the "Materials") from time to time, without notice. Mini-Circuits makes no commitment to update or correct any of the Materials, and Mini-Circuits shall have no responsibility whatsoever on account of any updates or corrections to the Materials or Mini-Circuits' failure to do so.

Mini-Circuits customers are solely responsible for the products, systems, and applications in which Mini-Circuits parts are incorporated or used. In that regard, customers are responsible for consulting with their own engineers and other appropriate professionals who are familiar with the specific products and systems into which Mini-Circuits' parts are to be incorporated or used so that the proper selection, installation/integration, use and safeguards are made. Accordingly, Mini-Circuits assumes no liability therefor.

In addition, your use of this document and the information contained herein is subject to Mini-Circuits' standard terms of use, which are available at Mini-Circuits' website at www.minicircuits.com/homepage/terms_of_use.html.

Mini-Circuits and the Mini-Circuits logo are registered trademarks of Scientific Components Corporation d/b/a Mini-Circuits. All other third-party trademarks are the property of their respective owners. A reference to any third-party trademark does not constitute or imply any endorsement, affiliation, sponsorship, or recommendation: (i) by Mini-Circuits of such thirdparty's products, services, processes, or other information; or (ii) by any such third-party of Mini-Circuits or its products, services, processes, or other information.