Automate Your Keysight Test Instruments Using the MATLAB Instrument Control Toolbox

This document will provide step-by-step instructions for using the Instrument Control Toolbox in MATLAB to communicate to a test and measurement instrument. **NOTE:** Although a free trial is available, MATLAB is not free software. A license must be purchased for the MATLAB platform AND the Instrument Control Toolbox for long term use. **NOTE:** The Instrument Control Toolbox is needed in order to communicate to instruments through a VISA.

If you already have the software installed, skip step one.

1. Download MATLAB

MATLAB: http://www.mathworks.com/index.html?s_tid=gn_logo

Instrument Control Toolbox: http://www.mathworks.com/products/instrument/index.html

2. Download and Install a VISA

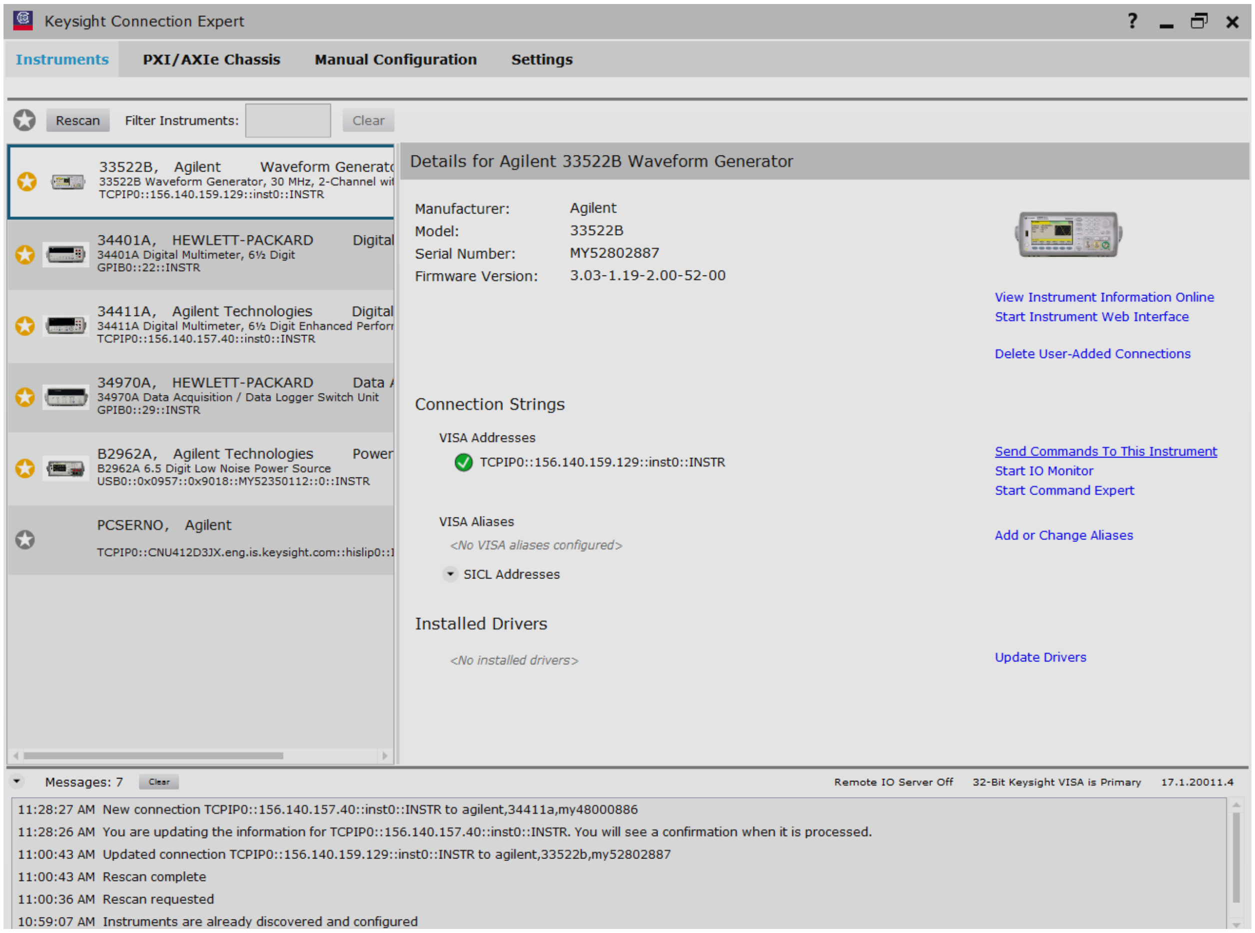
The **V**irtual **I**nstrument **S**oftware **A**rchitecture is a very useful tool when communicating to instruments with a PC. In a nutshell, it takes the commands you are using, and converts them into the proper syntax needed for a specific interface (GPIB, USB, LAN, etc..). The programmer simply needs to specify which interface is being used and the VISA will take care of the rest.

Keysight IO Libraries Suite: http://www.keysight.com/find/iolibs

After downloading and installing the IO Libraries Suite, you will have access to the Keysight Connection Expert:

NOTE: If your PC already has the IO Libraries installed, you do not have to upgrade to the latest version

CAUTION: If installing MATLAB on an instrument that already has the IO Libraries installed, do not update the version. This could potentially cause problems with that instrument.



33522B, Agilent Waveform Generato
33522B Waveform Generator, 30 MHz, 2-Channel wi
TCPIP0::156.140.159.129::inst0::INSTR

34401A, HEWLETT-PACKARD Digital
34401A Digital Multimeter, 6½ Digit
GPIB0::22::INSTR

34411A, Agilent Technologies Digital
34411A Digital Multimeter, 6½ Digit Enhanced Perforr
TCPIP0::156.140.157.40::inst0::INSTR

34970A, HEWLETT-PACKARD Data /
34970A Data Acquisition / Data Logger Switch Unit
GPIB0::29::INSTR

B2962A, Agilent Technologies Power
B2962A 6.5 Digit Low Noise Power Source
USB0::0x0957::0x9018::MY52350112::0::INSTR

PCSERNO, Agilent
TCPIP0::CNU412D3JX.eng.is.keysight.com::hislip0::I

Details for Agilent 33522B Waveform Generator

Manufacturer:	Agilent
Model:	33522B
Serial Number:	MY52802887
Firmware Version:	3.03-1.19-2.00-52-00

Connection Strings

VISA Addresses

✓ TCPIP0::156.140.159.129::inst0::INSTR

VISA Aliases

<No VISA aliases configured>

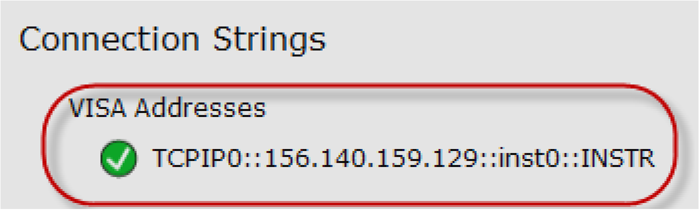
▾ SICL Addresses

Installed Drivers

<No installed drivers>

```
11:28:27 AM  New connection TCPIP0::156.140.157.40::inst0::INSTR to agilent,34411a,my48000886
11:28:26 AM  You are updating the information for TCPIP0::156.140.157.40::inst0::INSTR. You will see a confirmation when it is processed.
11:00:43 AM  Updated connection TCPIP0::156.140.159.129::inst0::INSTR to agilent,33522b,my52802887
11:00:43 AM  Rescan complete
11:00:36 AM  Rescan requested
10:59:07 AM  Instruments are already discovered and configured
```

From this software, you can troubleshoot connections, add/delete your connected instruments, and monitor the IO traffic (among other things). You can also grab the VISA address of the instrument here for your program. This will be needed in order to send commands to the instrument and read data from it within MATLAB:



Connection Strings

VISA Addresses

✓ TCPIP0::156.140.159.129::inst0::INSTR

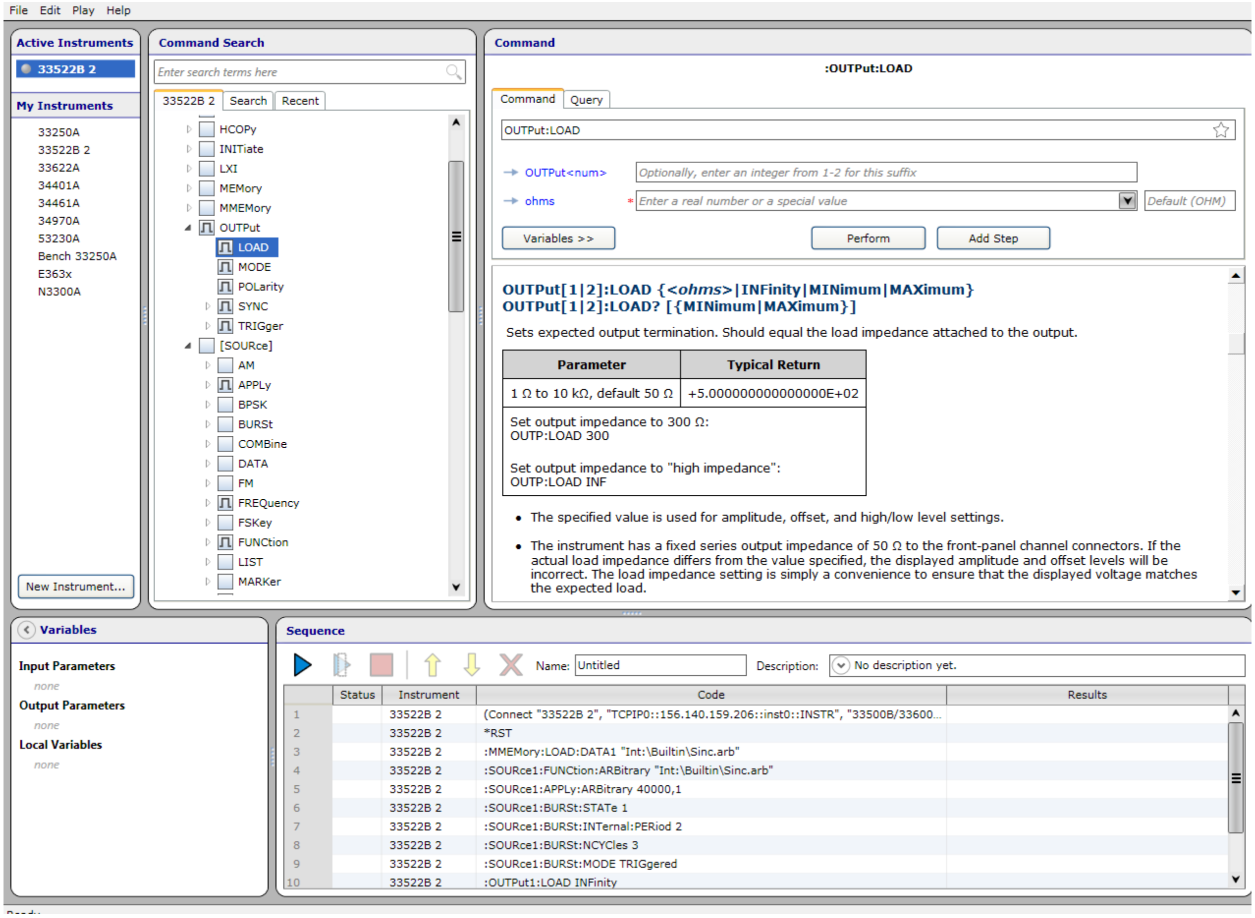
3. OPTIONAL* Download and Install Keysight Command Expert

Keysight Command Expert is a great tool for learning the individual SCPI commands for an instrument and testing the command sequences. There are good descriptions for each command, as well as examples within the GUI. You can then run your SCPI command sequences within Command Expert to make sure the sequence doesn’t cause any instrument errors.

If MATLAB is already installed on your PC at the time of the Command Expert installation, the Command Expert software will detect this, and will automatically install a MATLAB add-on. This add-on is a set of MATLAB functions that run Command Expert sequences in MATLAB.

You can also convert your SCPI command sequence directly to MATLAB code.

Keysight Command Expert: http://www.keysight.com/find/commandexpert



Command

:OUTPut:LOAD

OUTPut:LOAD

OUTPut<num> Optionally, enter an integer from 1-2 for this suffix

ohms * Enter a real number or a special value Default (OHM)

OUTPut[1|2]:LOAD {<*ohms*>|INFinity|MINimum|MAXimum}
OUTPut[1|2]:LOAD? [{MINimum|MAXimum}]

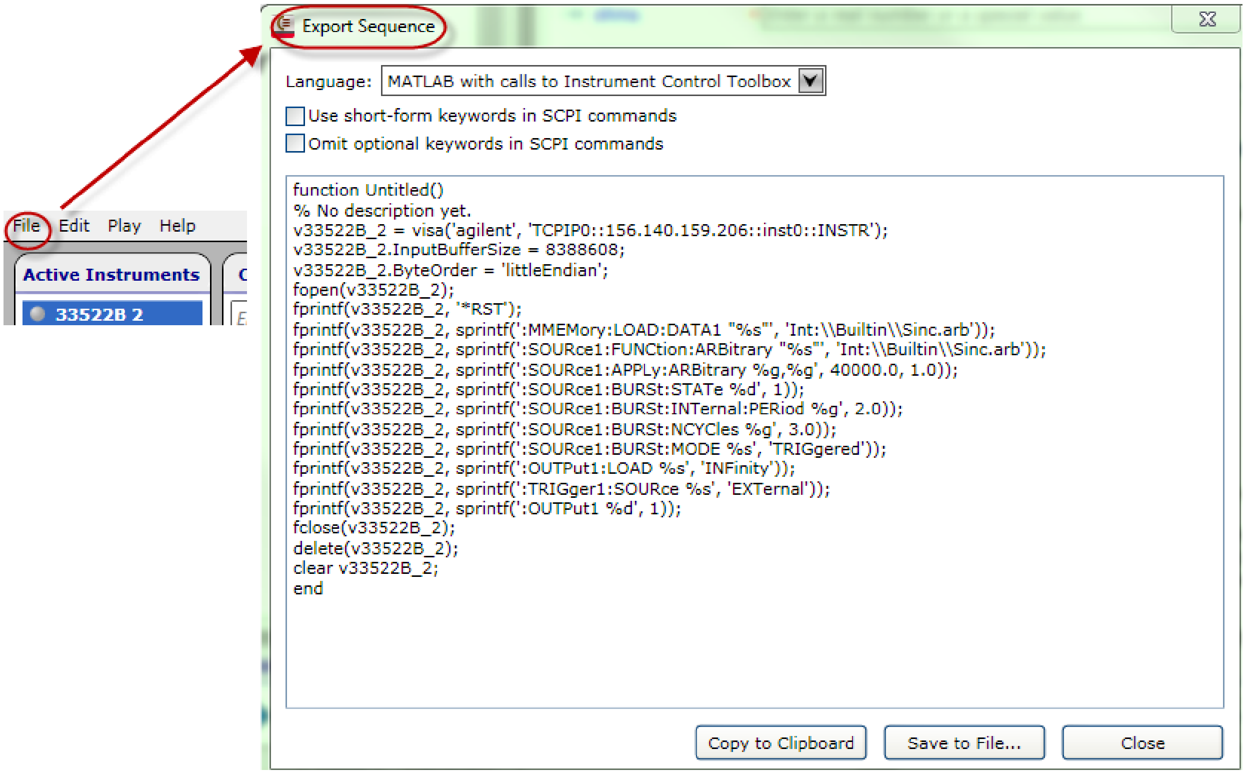
Sets expected output termination. Should equal the load impedance attached to the output.

Parameter	Typical Return
1 Ω to 10 kΩ, default 50 Ω	+5.000000000000000E+02
Set output impedance to 300 Ω: OUTP:LOAD 300 Set output impedance to "high impedance": OUTP:LOAD INF	

- The specified value is used for amplitude, offset, and high/low level settings.
- The instrument has a fixed series output impedance of 50 Ω to the front-panel channel connectors. If the actual load impedance differs from the value specified, the displayed amplitude and offset levels will be incorrect. The load impedance setting is simply a convenience to ensure that the displayed voltage matches the expected load.

Name: Untitled   Description: No description yet.

	Status	Instrument	Code	Results
1		33522B 2	(Connect "33522B 2", "TCPIP0::156.140.159.206::inst0::INSTR", "33500B/33600...	
2		33522B 2	*RST	
3		33522B 2	:MMEMory:LOAD:DATA1 "Int:\Builtin\Sinc.arb"	
4		33522B 2	:SOURce1:FUNCtion:ARBitrary "Int:\Builtin\Sinc.arb"	
5		33522B 2	:SOURce1:APPLy:ARBitrary 40000,1	
6		33522B 2	:SOURce1:BURSt:STATe 1	
7		33522B 2	:SOURce1:BURSt:INTernal:PERiod 2	
8		33522B 2	:SOURce1:BURSt:NCYCles 3	
9		33522B 2	:SOURce1:BURSt:MODE TRIGgered	
10		33522B 2	:OUTPut1:LOAD INFinity	



Language: MATLAB with calls to Instrument Control Toolbox

□ Use short-form keywords in SCPI commands

□ Omit optional keywords in SCPI commands

```
function Untitled()
% No description yet.
v33522B_2 = visa('agilent', 'TCPIP0::156.140.159.206::inst0::INSTR');
v33522B_2.InputBufferSize = 8388608;
v33522B_2.ByteOrder = 'littleEndian';
fopen(v33522B_2);
fprintf(v33522B_2, '*RST');
fprintf(v33522B_2, sprintf(':MMEMory:LOAD:DATA1 "%s"', 'Int:\\Builtin\\Sinc.arb'));
fprintf(v33522B_2, sprintf(':SOURce1:FUNCtion:ARBitrary "%s"', 'Int:\\Builtin\\Sinc.arb'));
fprintf(v33522B_2, sprintf(':SOURce1:APPLy:ARBitrary %g,%g', 40000.0, 1.0));
fprintf(v33522B_2, sprintf(':SOURce1:BURSt:STATe %d', 1));
fprintf(v33522B_2, sprintf(':SOURce1:BURSt:INTernal:PERiod %g', 2.0));
fprintf(v33522B_2, sprintf(':SOURce1:BURSt:NCYCles %g', 3.0));
fprintf(v33522B_2, sprintf(':SOURce1:BURSt:MODE %s', 'TRIGgered'));
fprintf(v33522B_2, sprintf(':OUTPut1:LOAD %s', 'INFinity'));
fprintf(v33522B_2, sprintf(':TRIGger1:SOURce %s', 'EXTernal'));
fprintf(v33522B_2, sprintf(':OUTPut1 %d', 1));
fclose(v33522B_2);
delete(v33522B_2);
clear v33522B_2;
end
```

4. Start MATLAB and Start Programming

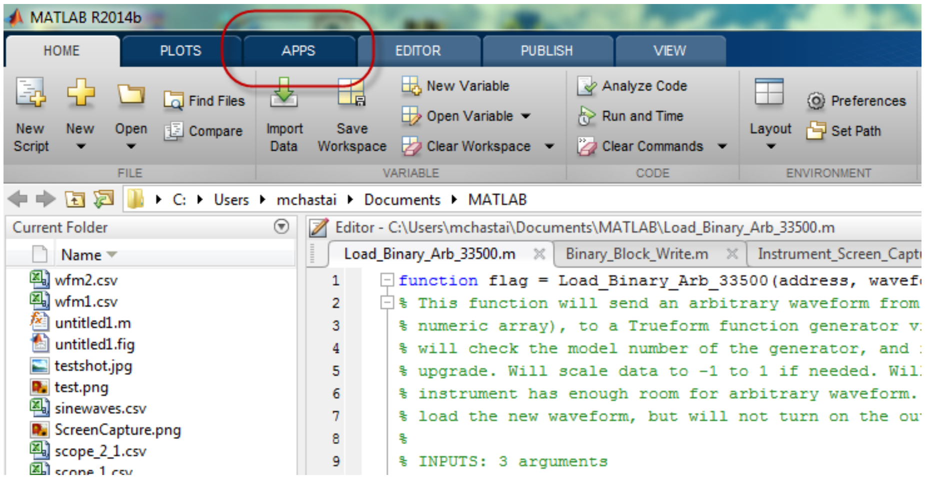
There are a couple ways to in which you can control an instrument with MATLAB:

A. Command Expert

See the previous step. Copy/past code into MATLAB

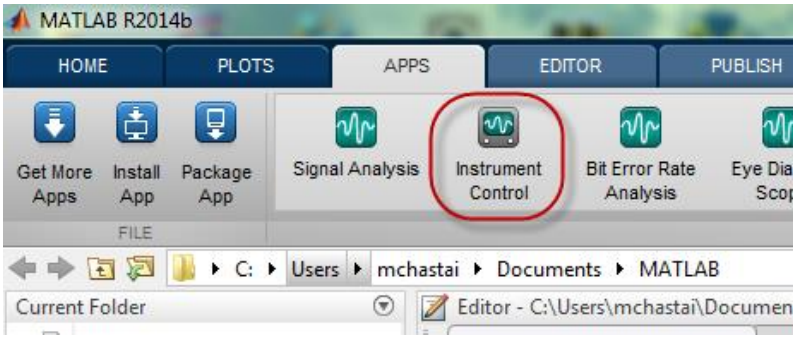
B. Instrument Control Toolbox Application

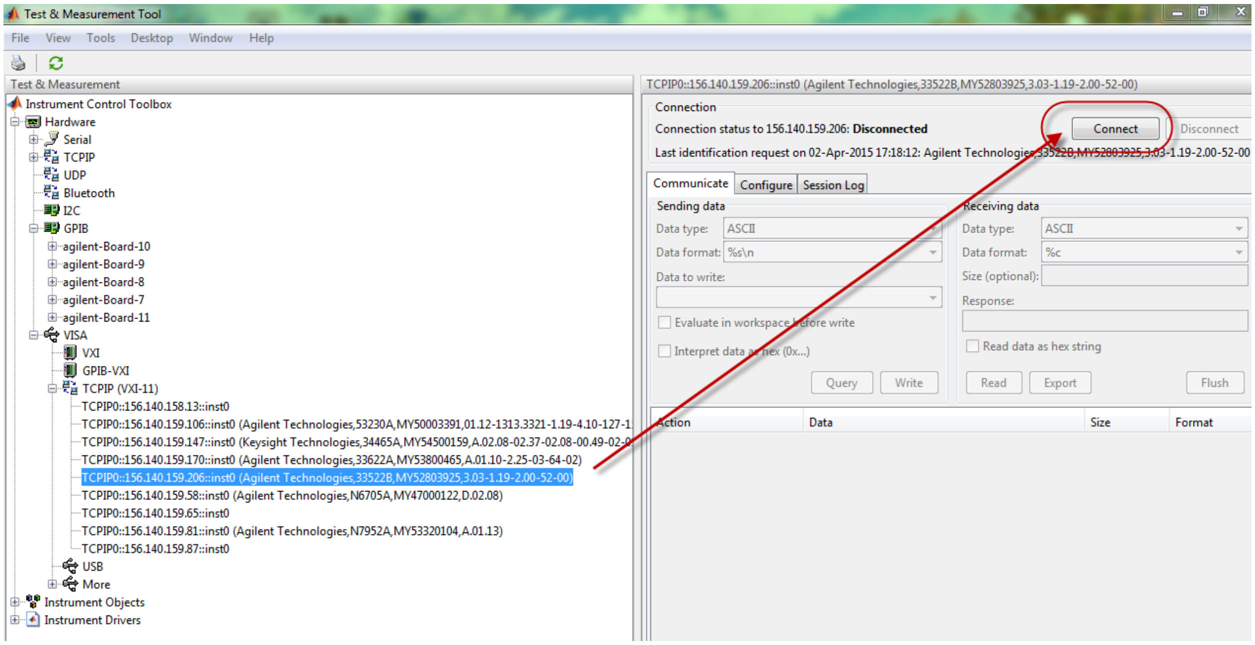
While the Instrument Control Toolbox is necessary in order to use the VISA functions, using the application itself is not. However, using the application can help get you started with using the VISA function and reading/writing with the instrument.



wfm2.csv
wfm1.csv
untitled1.m
untitled1.fig
testshot.jpg
test.png
sinewaves.csv
ScreenCapture.png
scope_2_1.csv
scope_1.csv

```
function flag = Load_Binary_Arb_33500(address, wavef
% This function will send an arbitrary waveform from
% numeric array), to a Trueform function generator v
% will check the model number of the generator, and
% upgrade. Will scale data to -1 to 1 if needed. Wil
% instrument has enough room for arbitrary waveform.
% load the new waveform, but will not turn on the ou
%
% INPUTS: 3 arguments
```





- Instrument Control Toolbox
  - Hardware
    - Serial
    - TCPIP
    - UDP
    - Bluetooth
    - I2C
    - GPIB
      - agilent-Board-10
      - agilent-Board-9
      - agilent-Board-8
      - agilent-Board-7
      - agilent-Board-11
    - VISA
      - VXI
      - GPIB-VXI
      - TCPIP (VXI-11)
        - TCPIP0::156.140.158.13::inst0
        - TCPIP0::156.140.159.106::inst0 (Agilent Technologies,53230A,MY50003391,01.12-1313.3321-1.19-4.10-127-1
        - TCPIP0::156.140.159.147::inst0 (Keysight Technologies,34465A,MY54500159,A.02.08-02.37-02.08-00.49-02-0
        - TCPIP0::156.140.159.170::inst0 (Agilent Technologies,33622A,MY53800465,A.01.10-2.25-03-64-02)
        - TCPIP0::156.140.159.206::inst0 (Agilent Technologies,33522B,MY52803925,3.03-1.19-2.00-52-00)
        - TCPIP0::156.140.159.58::inst0 (Agilent Technologies,N6705A,MY47000122,D.02.08)
        - TCPIP0::156.140.159.65::inst0
        - TCPIP0::156.140.159.81::inst0 (Agilent Technologies,N7952A,MY53320104,A.01.13)
        - TCPIP0::156.140.159.87::inst0
      - USB
      - More
  - Instrument Objects
  - Instrument Drivers

TCPIP0::156.140.159.206::inst0 (Agilent Technologies,33522B,MY52803925,3.03-1.19-2.00-52-00)

Connection

Connection status to 156.140.159.206: **Disconnected**

Last identification request on 02-Apr-2015 17:18:12: Agilent Technologies,33522B,MY52803925,3.03-1.19-2.00-52-00

Sending data

Data type: ASCII

Data format: %s\n

Data to write:

□ Evaluate in workspace before write

□ Interpret data as hex (0x...)

Receiving data

Data type: ASCII

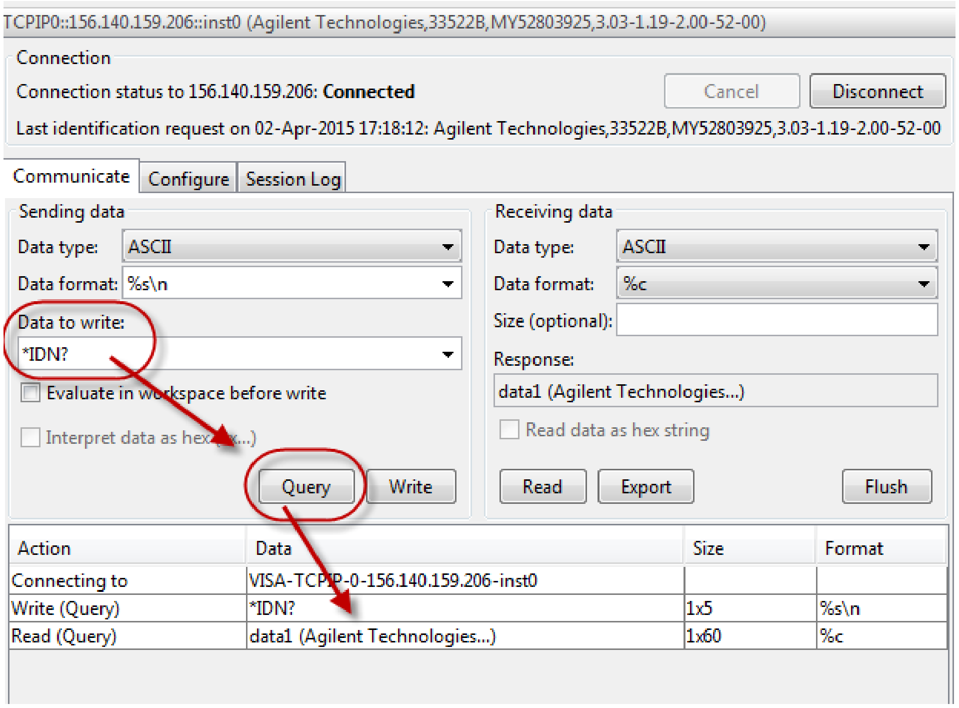
Data format: %c

Size (optional):

Response:

□ Read data as hex string

Action	Data	Size	Format


TCPIP0::156.140.159.206::inst0 (Agilent Technologies,33522B,MY52803925,3.03-1.19-2.00-52-00)

Connection

Connection status to 156.140.159.206: **Connected**

Last identification request on 02-Apr-2015 17:18:12: Agilent Technologies,33522B,MY52803925,3.03-1.19-2.00-52-00

Sending data

Data type: ASCII

Data format: %s\n

Data to write: *IDN?

□ Evaluate in workspace before write

□ Interpret data as hex (x...)

Receiving data

Data type: ASCII

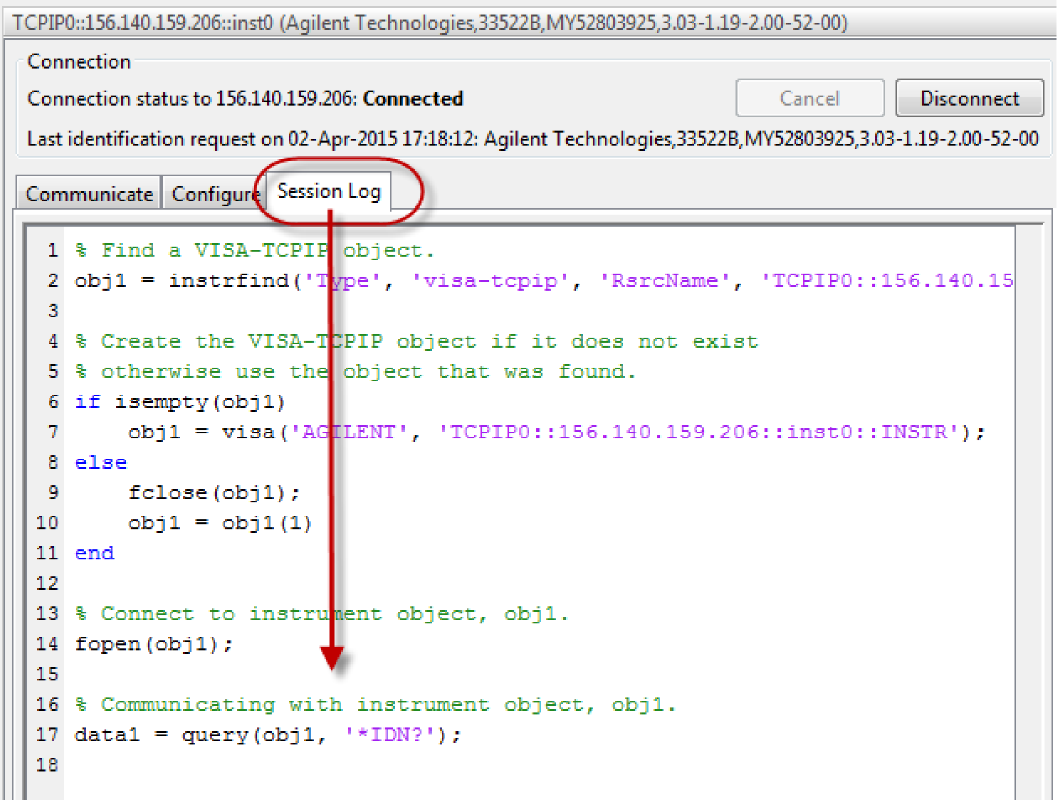
Data format: %c

Size (optional):

Response: data1 (Agilent Technologies...)

□ Read data as hex string

Action	Data	Size	Format
Connecting to	VISA-TCPIP-0-156.140.159.206-inst0		
Write (Query)	*IDN?	1x5	%s\n
Read (Query)	data1 (Agilent Technologies...)	1x60	%c


TCPIP0::156.140.159.206::inst0 (Agilent Technologies,33522B,MY52803925,3.03-1.19-2.00-52-00)

Connection

Connection status to 156.140.159.206: **Connected**

Last identification request on 02-Apr-2015 17:18:12: Agilent Technologies,33522B,MY52803925,3.03-1.19-2.00-52-00

```
% Find a VISA-TCPIP object.
obj1 = instrfind('Type', 'visa-tcpip', 'RsrcName', 'TCPIP0::156.140.15

% Create the VISA-TCPIP object if it does not exist
% otherwise use the object that was found.
if isempty(obj1)
    obj1 = visa('AGILENT', 'TCPIP0::156.140.159.206::inst0::INSTR');
else
    fclose(obj1);
    obj1 = obj1(1)
end

% Connect to instrument object, obj1.
fopen(obj1);

% Communicating with instrument object, obj1.
data1 = query(obj1, '*IDN?');

```

You can copy/paste this code into a MATLAB program from this screen, or save the following code to a ".m" MATLAB file.

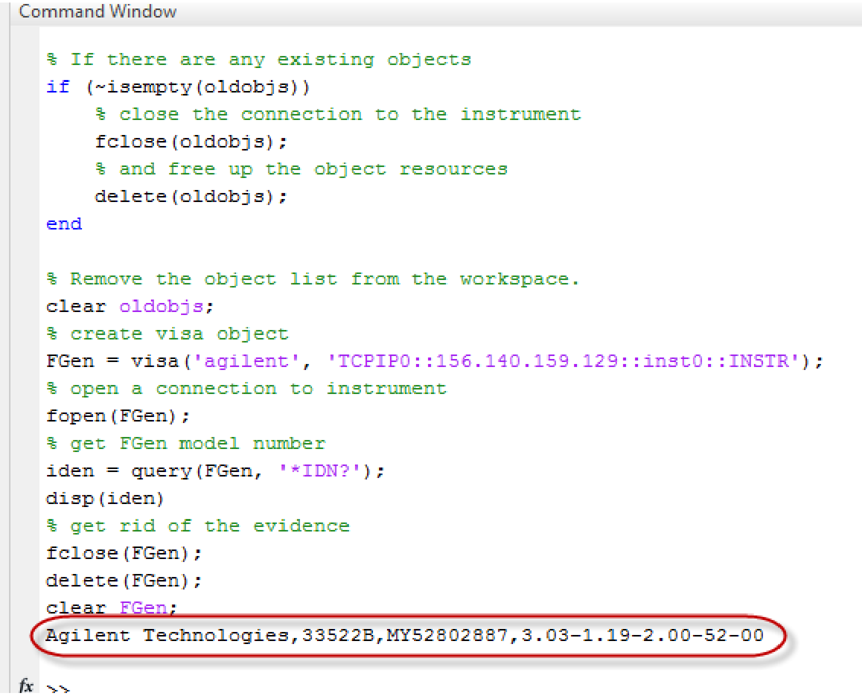
5. Example Code

You can drop the following code into MATLAB to obtain the instrument's identification string. Make sure you change the VISA address. The following code gets the identification string from a Keysight 33500B series function generator.

```
% find all previously created objects
oldobjs = instrfind;

% If there are any existing objects
if (~isempty(oldobjs))
    % close the connection to the instrument
    fclose(oldobjs);
    % and free up the object resources
    delete(oldobjs);
end

% Remove the object list from the workspace.
clear oldobjs;
% create visa object
FGen = visa('agilent', 'TCPIP0::156.140.159.129::inst0::INSTR');
% open a connection to instrument
fopen(FGen);
% get FGen model number
iden = query(FGen, '*IDN?');
disp(iden)
% get rid of the evidence
fclose(FGen);
delete(FGen);
clear FGen;
```



```
% If there are any existing objects
if (~isempty(oldobjs))
    % close the connection to the instrument
    fclose(oldobjs);
    % and free up the object resources
    delete(oldobjs);
end

% Remove the object list from the workspace.
clear oldobjs;
% create visa object
FGen = visa('agilent', 'TCPIP0::156.140.159.129::inst0::INSTR');
% open a connection to instrument
fopen(FGen);
% get FGen model number
iden = query(FGen, '*IDN?');
disp(iden)
% get rid of the evidence
fclose(FGen);
delete(FGen);
clear FGen;
Agilent Technologies,33522B,MY52802887,3.03-1.19-2.00-52-00

fx >>
```