Application Note

Using MATLAB Applications with MS269xA Signal Analyzer

MS2690A/MS2691A/MS2692A Signal Analyzer

Outline
This document explains how to set up the MATLAB application software for communication with the MS269xA Signal Analyzer family. This is the first step in developing MATLAB applications for the MS269xA. The reader should have a basic knowledge of MATLAB and should be familiar with the MS269xA remote control (SCPI) interface.

The MATLAB programs in this reference material are only examples and are not guaranteed. Anritsu does not support these programs. They are offered to assist development only.

I. Introduction

MATLAB is a very powerful computational, analysis, and simulation tool used widely in universities and industry. It has many convenient functions and features, including SIMULINK, Communications Toolbox, Signal Processing Toolbox and Instrument Control Toolbox, that are very useful for testing and measurement (T&M). Developing MATLAB applications for T&M equipment like the MS269xA Signal Analyzer can help engineers verify their designs and greatly reduce test times when configuring and evaluating a test environment.

II. Communications Interface between MATLAB and MS269xA

The first step in developing MATLAB applications for the MS269xA is getting MATLAB and the MS269xA to communicate. The following code shows an example of communicating with the MS269xA using MATLAB commands.

```
% example1: connecting to the MS269xA
% Create Interface Object
% In this example the interface used is
% VISA-TCP/IP by National Instruments
ipaddress = '192.168.221.3';
msInstr = visa('ni', ['TCPIP::' ipaddress '::INSTR']);
% Now we try to connect to the MS269xA
try
    fopen(msInstr);
    msgbox('Connecting to MS269xA successful');
catch
    errordlg('Cannot connect to MS269xA. Check connections and network settings');
    return;
end

% If successfully connecting to MS269xA,
% we can now send remote control commands (SCPI)
% using the fprintf() command ...
```

```
fprintf(msInstr, '*IDN?');
% ...and receive responses using the fscanf() command
response1 = fscanf(msInstr);
response1

% query() command can also be used and is
% equivalent to the combined fprintf() and fscanf() commands
response2 = query(msInstr, '*IDN?');
response2

% fprintf(), fscanf() and query() are the basic commands
% used to communicate with the MS269xA

% ...more lines of SCPI commands here...

% After we finished communicating with the MS269xA,
% It is necessary to close the connection used.
fclose(msInstr);
% additional clean up procedures
% deleting the interface object variable used
% and removing it from the workspace
delete(msInstr)
clear msInstr
```

First, the interface object used for communications is created. In this example, the interface object is National Instruments' VISA-TCP/IP, which we recommend. Remember to install the VISA drivers beforehand. In this example, the VISA driver provided by National Instruments should already be installed. The syntax for creating interface objects using VISA is:

```
INTERFACE_OBJ = ('VENDOR', 'RSRCNAME')
```

'VENDOR' is the VISA vendor. (It is `ni` in this example, which stands for National Instruments). Other VISA vendors may be `agilent` (Agilent Technologies) or `tek` (Tektronix). The parameter *'RSRCNAME'* describes the interface being used. In this example, the TCP/IP interface is used and its corresponding syntax is:

```
TCPIP[board]::remote_host[::lan_device_name]::INSTR
```

Parameters in square brackets [ ] may be omitted. The `remote_host` parameter describes the Host name or IP address of the instrument. In this example, the MS269xA has the IP address 192.168.221.3. To check the IP address of the MS269xA, press the "**System Config**" button and then press the "**F1 (Interface Settings)**" soft key on the MS269xA front panel. The screen will display the current IP address.

Other interfaces, such as GPIB and USB, can be used with VISA. For more information about using VISA interface objects in MATLAB, input `instrhelp visa` at the MATLAB command window.

After creating the interface object, the connection is opened using the `fopen(INTERFACE_OBJ)` command where INTERFACE_OBJ is the created interface object. When the connection is opened successfully, SCPI commands can be sent and responses received using the `fprintf(INTERFACE_OBJ, 'COMMAND')` and `fscanf(INTERFACE_OBJ)` commands. The `query(INTERFACE_OBJ, 'COMMAND')` command can also be used and is equivalent to the combined `fprintf` and `fscanf` commands. When communications with the MS269xA are completed, the connection is closed using the `fclose(INTERFACE_OBJ)` command.

III. Communicating with MS269xA using Test & Measurement Tool

The Test & Measurement Tool of the Instrument Control Toolbox can also be used to communicate with the MS269xA. To start, input `tmtool` at the MATLAB command window to open the Test & Measurement Tool window. Using this tool, create an interface object as in the previous example using MATLAB commands. To communicate with the MS269xA, first create an interface object by right clicking "Interface Object" at the left side of the Test & Measurement Tool pane and select “Create New Interface Object”. Next, set the interface object parameters to the appropriate values matching the MS269xA to connect. After creating the interface object, the instrument can be connected to start sending and receiving commands. The following figures show how to create an interface object using the Test & Measurement Tool of the Instrument Control Toolbox.

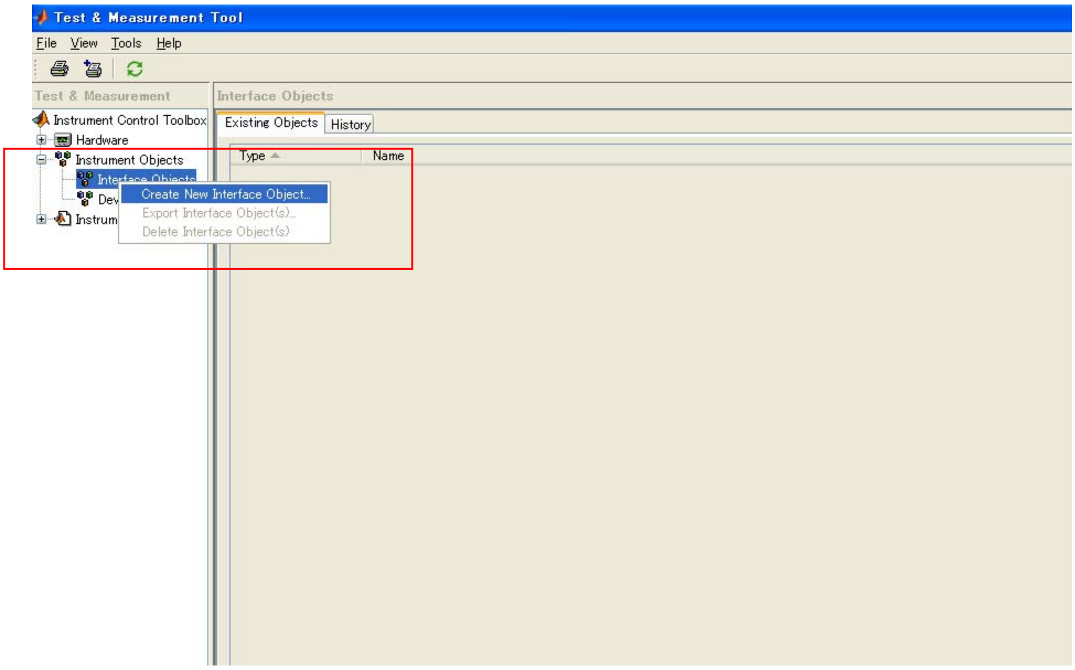

Fig. 1 Creating Interface Object Using Test & Measurements Tool

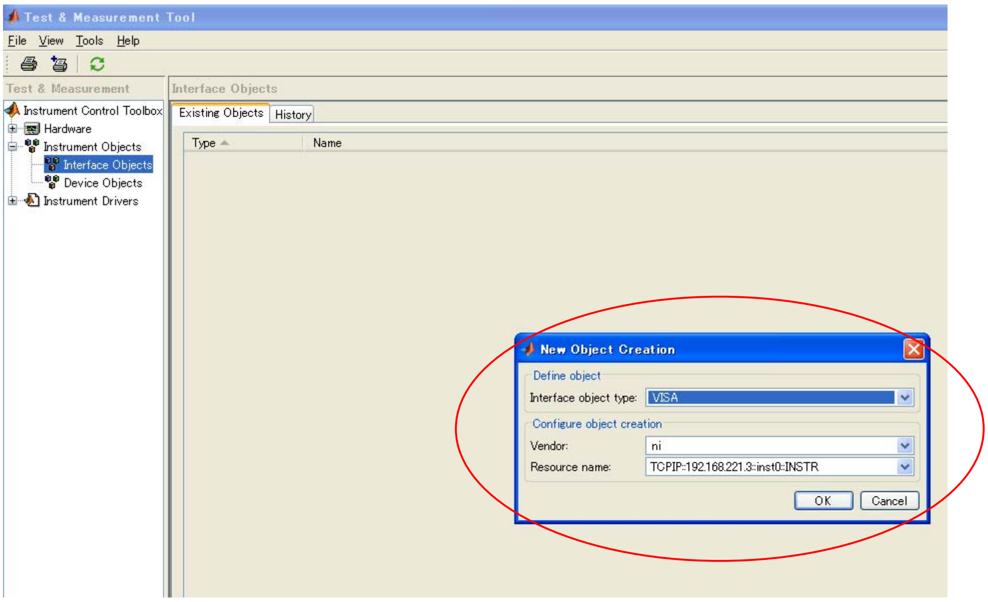

Fig. 2 Setting Interface Object Parameters

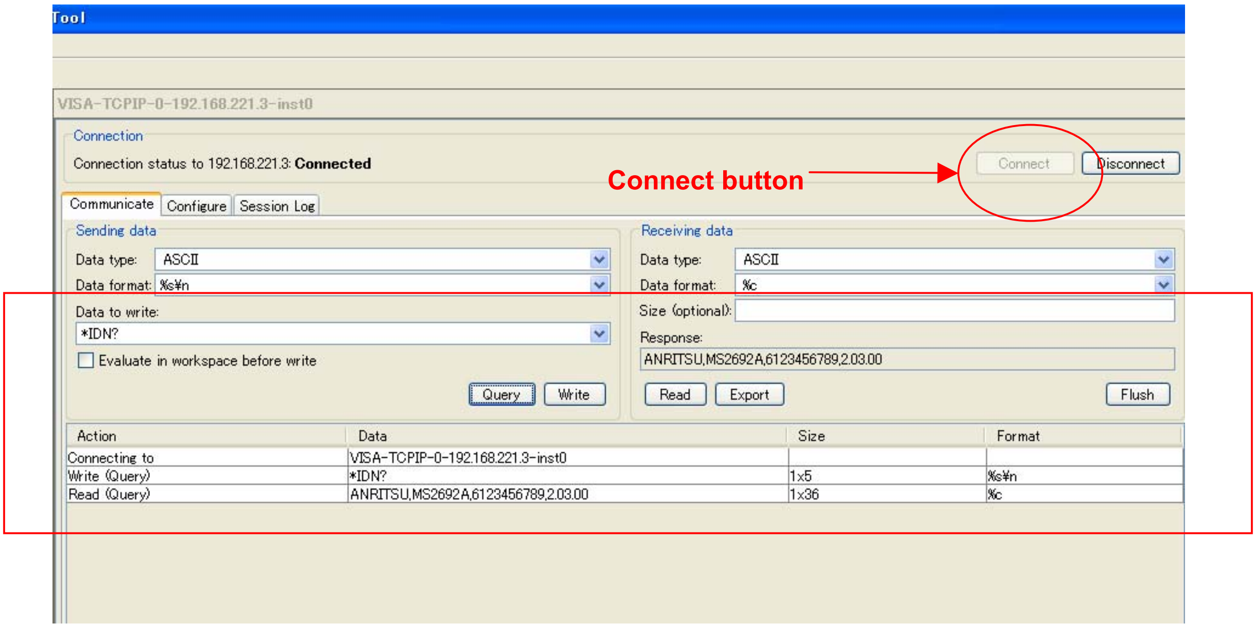

Fig. 3 Connecting and Communicating with Instrument

IV. Transferring Files between MATLAB and MS269xA

When connected to the MS269xA over a LAN, file transfers are straightforward when the copy source and copy destination folders are shared on the network. The file transfer command is `copyfile('source', 'destination')`. For example, the following code copies the file named **IQ_Tx.wvd** in the current directory of MATLAB to a shared folder named **Convert_IQproducer** on the MS269xA with IP address 192.168.221.3.

```
% This copies a file (IQ_Tx.wvd) from the current
% MATLAB directory to an MS269xA network shared
% directory named Convert_IQproducer

ipaddress = '192.168.221.3';
copyfile('IQ_Tx.wvd' , ['\\' ipaddress '\Convert_IQproducer']);
```

Note that the folder **Convert_IQproducer** on the MS269xA must be shared on the network.
The example below copies the file named **IQ_digitized.xml** in the MS269xA folder named **Signal Analyzer** shared on the network to the current MATLAB directory.

```
% This copies a file (IQ_digitized.xml) from the MS269xA
% network shared directory named Signal Analyzer
% to the current MATLAB directory

ipaddress = '192.168.221.3';
copyfile(['\\' ipaddress '\Signal Analyzer\IQ_digitized.xml'] ,  './');
```

As in the first example, the folder named **Signal Analyzer** on the MS269xA must be shared on the network. In addition, the firewall settings for the remote PC (running MATLAB) and MS269xA must be off. (*NOTE:* The MS269xA default firewall setting is Off.)

For more information about using the `copyfile` command, input `help copyfile` at the MATLAB command window.

V. Importing IQ data to MATLAB

MS269xA Digitize Function

The MS269xA Digitize function captures waveforms and stores them as IQ data. It is very useful for further data analysis using commercial tools like MATLAB. The following code shows how to execute the MS269xA Digitize function using MATLAB commands.

```
% example2: shows how to execute the
% MS269xA digitize function using MATLAB

% Create interface object
ipaddress = '192.168.221.3';
msInstr = visa('ni', ['TCPIP::' ipaddress '::INSTR']);

% Connect to MS269xA
try
    fopen(msInstr);
    msgbox('Connecting to MS269xA successful');
catch
    errordlg('Cannot connect to MS269xA. Check connections and network settings');
    return;
end

% Sets the controlled application to the Signal Analyzer
fprintf(msInstr, 'INST SIGANA');

% Sets the Trigger source.
% In this example the Trigger source is set to the
% Signal Generator (SG) option of the MS269xA.
% This trigger source is only valid
% when the SG option is installed.
fprintf(msInstr, 'TRIG:SOUR SG');

% Sets the Center frequency and span
fprintf(msInstr, 'FREQ:CENT 1GHZ');
fprintf(msInstr, 'FREQ:SPAN 10MHZ');

% Sets the analysis start time
fprintf(msInstr, 'CALC:ATIM:STAR 0S');

% Sets the length of analysis time
fprintf(msInstr, 'CALC:ATIM:LENG 10MS');

% Sets the Signal Analyzer trace display
fprintf(msInstr, 'TRAC:MODE SPEC');

% Initiate single measurement
fprintf(msInstr, 'INIT:MODE:SING');
pause(3);

% Specify the Digitized output filename
dgznme = 'IQ_digitize';

% Execute digitize function and
% save digitized data to MS269xA`s D: drive
fprintf(msInstr, ['MMEM:STOR:IQD "' dgznme '", D']);

% close connection
fclose(msInstr);
```

In this example, the signal sent from the MS269xA Signal Generator option is captured using the Digitize function. Before executing the Digitize function, the signal analysis parameters, such as trigger, carrier frequency, frequency span, analysis time, etc., are set. Other parameters like output sampling rate may be varied as well. In this example, the sampling rate of the IQ data output to the file is set automatically to twice the frequency span (2 x 10 MHz = 20 MHz). After successfully completing digitizing, the XML and DGZ files are saved to the specified system drive. (In this example, **IQ_digitize.xml** and **IQ_digitize.dgz** are saved to drive D: of the MS269xA.)

<u>Extracting IQ Data using MATLAB</u>

To use the IQ data of the DGZ file, it must be extracted using MATLAB functions. The contents of the XML and DGZ files are shown in the following figure.

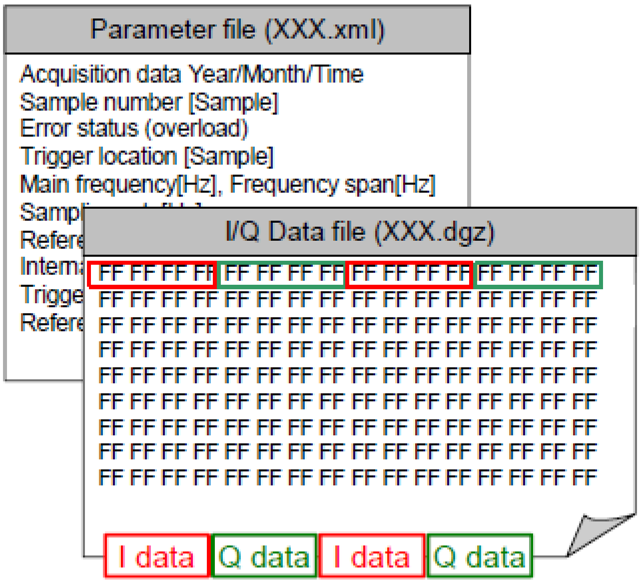

Fig. 4 XML and DGZ files

The first step in extracting the IQ data from the DGZ file is to copy the XML and DGZ files from the MS269xA to the MATLAB current directory using the `copyfile` command of the file transfer function described above.

After transferring the XML and DGZ files, the next step is to extract the trigger information in the XML file. This information is required to extract the required IQ data values from the DGZ file

(NOTE: At digitizing with the MS269xA trace display set to Spectrum, redundant samples are added before and after the section of the waveform saved for calculation of FFT process. When capturing 10 ms of a signal at a sampling rate of 20 MHz, the number of samples saved in the DGZ file is 200,000 +α, and is not exactly 200,000. Consequently, the position of the required signal must be specified using the trigger information.)

The following code shows how to extract the trigger position from the XML file.

```
% example3: parses the XML file named IQ_digitized.xml
% located on the current MATLAB directory to extract
% the Trigger point information
dgznme = 'IQ_digitize';
root = xmlread(['' dgznme '.xml']);
root_chld = root.getChildNodes;
siganaproj = root_chld.item(1);
siganaproj_chld = siganaproj.getChildNodes;
params = siganaproj_chld.item(3);
params_chld = params.getChildNodes;
Trig_pos = params_chld.item(13);
Trig_pos_value = Trig_pos.getAttribute('Value');
trigger = str2double(Trig_pos_value);
trigger = trigger + 1;
```

After executing the above commands, the trigger position is described in the `trigger` variable, which can then be used to extract the required IQ data from the DGZ file easily as shown below.

```
% example3 continued...
% Reads the I and Q data from the DGZ file
% using the triggering information obtained
% and saves them as the variables I_data and Q_data
fid = fopen(['' dgznme '.dgz'] , 'r');
data = fread(fid,'float=>float');
I_captured = data(1:2:end);
Q_captured = data(2:2:end);
% this example extracts 200,000 IQ sample points
I_data = I_captured(trigger:trigger+199999);
Q_data = Q_captured(trigger:trigger+199999);
fclose(fid);
```

Executing the above code saves the IQ data to the MATLAB workspace.

Anritsu

Specifications are subject to change without notice.

Anritsu Corporation
5-1-1 Onna, Atsugi-shi, Kanagawa, 243-8555 Japan
Phone: +81-46-223-1111
Fax: +81-46-296-1264

• **U.S.A.**
Anritsu Company
1155 East Collins Blvd., Suite 100, Richardson,
TX 75081, U.S.A.
Toll Free: 1-800-267-4878
Phone: +1-972-644-1777
Fax: +1-972-671-1877

• **Canada**
Anritsu Electronics Ltd.
700 Silver Seven Road, Suite 120, Kanata,
Ontario K2V 1C3, Canada
Phone: +1-613-591-2003
Fax: +1-613-591-1006

• **Brazil**
Anritsu Eletrônica Ltda.
Praca Amadeu Amaral, 27 - 1 Andar
01327-010-Paraiso-São Paulo-Brazil
Phone: +55-11-3283-2511
Fax: +55-11-3288-6940

• **Mexico**
Anritsu Company, S.A. de C.V.
Av. Ejército Nacional No. 579 Piso 9, Col. Granada
11520 México, D.F., México
Phone: +52-55-1101-2370
Fax: +52-55-5254-3147

• **U.K.**
Anritsu EMEA Ltd.
200 Capability Green, Luton, Bedfordshire, LU1 3LU, U.K.
Phone: +44-1582-433200
Fax: +44-1582-731303

• **France**
Anritsu S.A.
16/18 avenue du Québec-SILIC 720
91961 COURTABOEUF CEDEX, France
Phone: +33-1-60-92-15-50
Fax: +33-1-64-46-10-65

• **Germany**
Anritsu GmbH
Nemetschek Haus, Konrad-Zuse-Platz 1
81829 München, Germany
Phone: +49-89-442308-0
Fax: +49-89-442308-55

• **Italy**
Anritsu S.p.A.
Via Elio Vittorini 129, 00144 Roma, Italy
Phone: +39-6-509-9711
Fax: +39-6-502-2425

• **Sweden**
Anritsu AB
Borgafjordsgatan 13, 164 40 KISTA, Sweden
Phone: +46-8-534-707-00
Fax: +46-8-534-707-30

• **Finland**
Anritsu AB
Teknobulevardi 3-5, FI-01530 VANTAA, Finland
Phone: +358-20-741-8100
Fax: +358-20-741-8111

• **Denmark**
Anritsu A/S
Kirkebjerg Allé 90, DK-2605 Brøndby, Denmark
Phone: +45-72112200
Fax: +45-72112210

• **Spain**
Anritsu EMEA Ltd.
Oficina de Representación en España
Edificio Veganova
Avda de la Vega, n° 1 (edf 8, pl 1, of 8)
28108 ALCOBENDAS - Madrid, Spain
Phone: +34-914905761
Fax: +34-914905762

• **Russia**
Anritsu EMEA Ltd.
Representation Office in Russia
Tverskaya str. 16/2, bld. 1, 7th floor.
Russia, 125009, Moscow
Phone: +7-495-363-1694
Fax: +7-495-935-8962

• **United Arab Emirates**
Anritsu EMEA Ltd.
Dubai Liaison Office
P O Box 500413 - Dubai Internet City
Al Thuraya Building, Tower 1, Suit 701, 7th Floor
Dubai, United Arab Emirates
Phone: +971-4-3670352
Fax: +971-4-3688460

• **Singapore**
Anritsu Pte. Ltd.
60 Alexandra Terrace, #02-08, The Comtech (Lobby A)
Singapore 118502
Phone: +65-6282-2400
Fax: +65-6282-2533

• **India**
Anritsu Pte. Ltd.
India Branch Office
Unit No. S-3, Second Floor, Esteem Red Cross Bhavan,
No. 26, Race Course Road, Bangalore 560 001, India
Phone: +91-80-32944707
Fax: +91-80-22356648

• **P.R. China (Hong Kong)**
Anritsu Company Ltd.
Units 4 & 5, 28th Floor, Greenfield Tower, Concordia Plaza,
No. 1 Science Museum Road, Tsim Sha Tsui East,
Kowloon, Hong Kong
Phone: +852-2301-4980
Fax: +852-2301-3545

• **P.R. China (Beijing)**
Anritsu Company Ltd.
Beijing Representative Office
Room 1515, Beijing Fortune Building,
No. 5, Dong-San-Huan Bei Road,
Chao-Yang District, Beijing 10004, P.R. China
Phone: +86-10-6590-9230
Fax: +86-10-6590-9235

• **Korea**
Anritsu Corporation, Ltd.
8F Hyunjuk Building, 832-41, Yeoksam Dong,
Kangnam-ku, Seoul, 135-080, Korea
Phone: +82-2-553-6603
Fax: +82-2-553-6604

• **Australia**
Anritsu Pty. Ltd.
Unit 21/270 Ferntree Gully Road, Notting Hill,
Victoria 3168, Australia
Phone: +61-3-9558-8177
Fax: +61-3-9558-8255

• **Taiwan**
Anritsu Company Inc.
7F, No. 316, Sec. 1, Neihu Rd., Taipei 114, Taiwan
Phone: +886-2-8751-1816
Fax: +886-2-8751-1817

Please Contact:

080423

Printed on Recycled Paper

No. MS269xA-E-F-6-(1.01) 公知 Printed in Japan 2008-7 AKD