Application Note

Anritsu envision : ensure

Using MATLAB Applications with Signal Analyzer

Signal Analyzer MS2850A/MS2840A/MS2830A/MS269xA

Table of Contents

1 Outline

This document explains how to set up the MATLAB application software for communication with Anritsu's Signal Analyzer MS2850A/MS2840A/MS2830A/MS269xA family. This is the first step in developing MATLAB applications for the Signal Analyzer. The reader should have a basic knowledge of MATLAB and should be familiar with the Signal Analyzer 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.

'Signal Analyzer' in this document means Anritsu's Signal Analyzer MS2850A/MS2840A/MS2830A/MS269xA series. Some description requires installation of Signal Generator option (-020/120) with the Signal Analyzer.

2 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 Signal Analyzer can help engineers verify their designs and greatly reduce test times when configuring and evaluating a test environment.

3 Communications Interface between MATLAB and Signal Analyzer

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

```
% example1: connecting to the Signal Analyzer
% 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 Signal Analyzer
try
    fopen(msInstr);
    msgbox('Connecting to Signal Analyzer successful');
catch
    errordlg('Cannot connect to Signal Analyzer. Check connections and network settings');
    return;
end

% If successfully connecting to Signal Analyzer,
% 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 Signal Analyzer

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

% After we finished communicating with the Signal Analyzer,
% 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 Signal Analyzer has the IP address 192.168.221.3. To check the IP address of the Signal Analyzer, press the "**System Config**" button and then press the "**F1 (Interface Settings)**" soft key on the Signal Analyzer 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 Signal Analyzer are completed, the connection is closed using the `fclose(INTERFACE_OBJ)` command.

4 Communicating with Signal Analyzer using Test & Measurement Tool

The Test & Measurement Tool of the Instrument Control Toolbox can also be used to communicate with the Signal Analyzer. 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 Signal Analyzer, 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 Signal Analyzer 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.

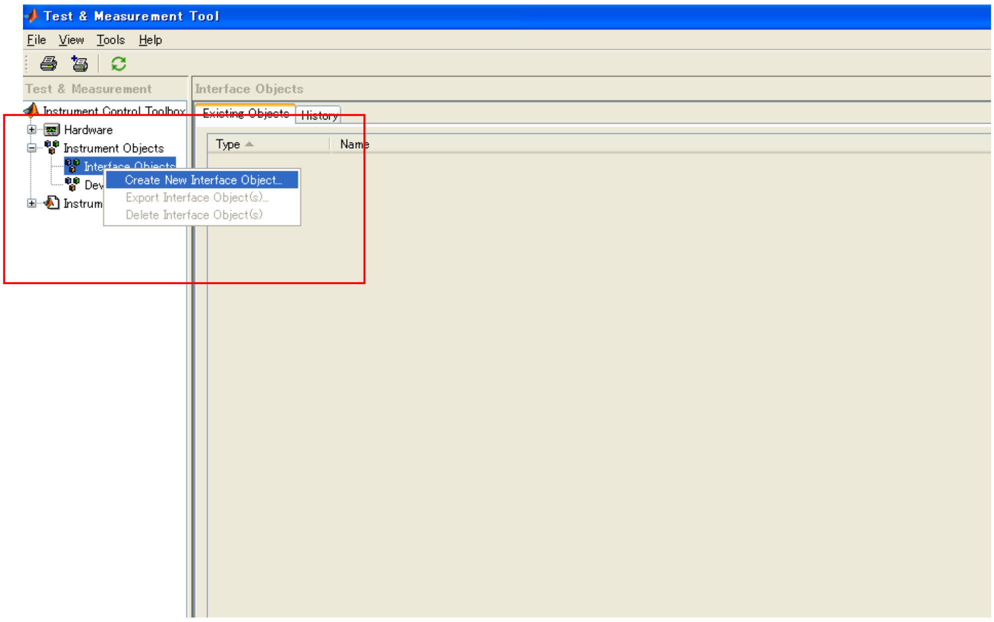
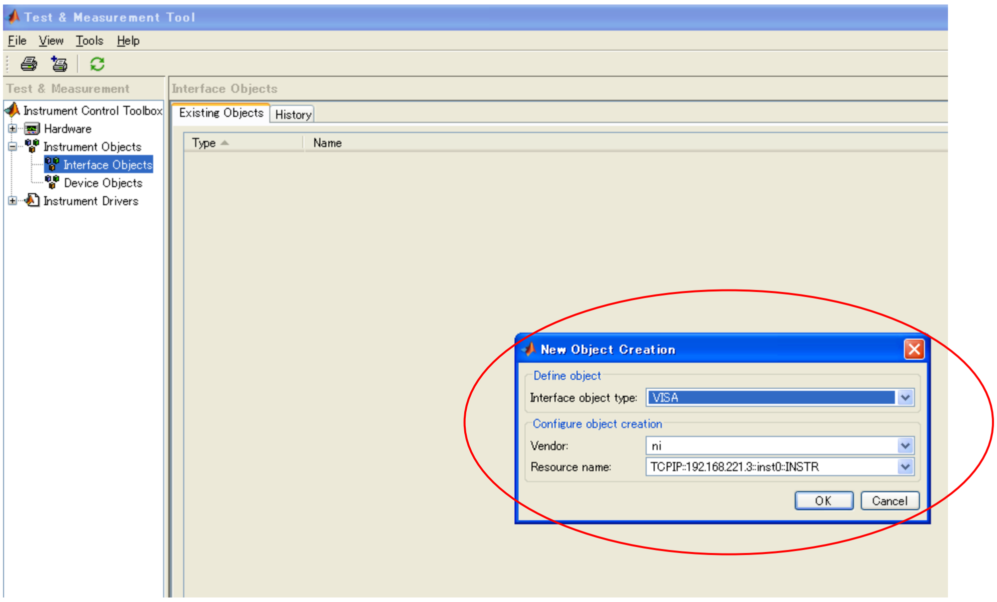


Fig. 1 Creating Interface Object Using Test & Measurements Tool

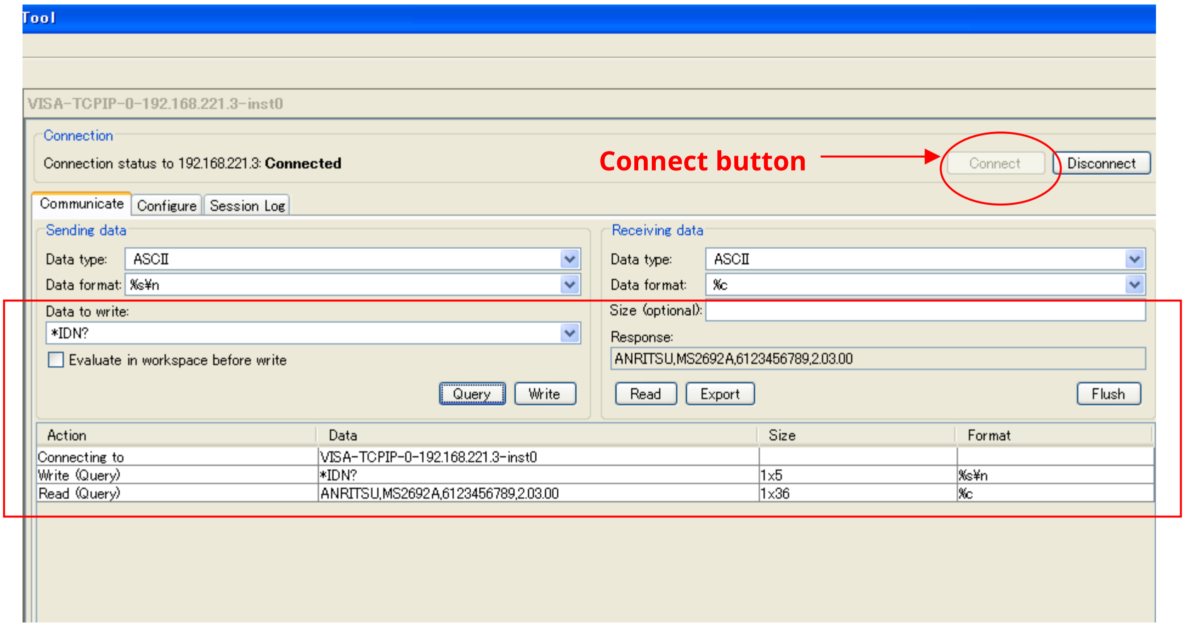


Interface object type: VISA

Vendor: ni

Resource name: TCPIP::192.168.221.3::inst0::INSTR

Fig. 2 Setting Interface Object Parameters



VISA-TCPIP-0-192.168.221.3-inst0

Connection status to 192.168.221.3: **Connected**

Data type: ASCII

Data format: %s¥n

Data to write: *IDN?

☐ Evaluate in workspace before write

Data type: ASCII

Data format: %c

Size (optional):

Response: ANRITSU,MS2692A,6123456789,2.03.00

Action	Data	Size	Format
Connecting to	VISA-TCPIP-0-192.168.221.3-inst0		
Write (Query)	*IDN?	1x5	%s¥n
Read (Query)	ANRITSU,MS2692A,6123456789,2.03.00	1x36	%c

Fig. 3 Connecting and Communicating with Instrument

5 Transferring Files between MATLAB and Signal Analyzer

When connected to the Signal Analyzer 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 Signal Analyzer with IP address 192.168.221.3.

```
% This copies a file (IQ_Tx.wvd) from the current
% MATLAB directory to an Signal Analyzer 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 Signal Analyzer must be shared on the network.
The example below copies the file named **IQ_digitized.xml** in the Signal Analyzer folder named **Signal Analyzer** shared on the network to the current MATLAB directory.

```
% This copies a file (IQ_digitized.xml) from the Signal Analyzer
% 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 Signal Analyzer must be shared on the network. In addition, the firewall settings for the remote PC (running MATLAB) and Signal Analyzer must be off. (*NOTE:* The Signal Analyzer default firewall setting is Off.)
For more information about using the `copyfile` command, input `help copyfile` at the MATLAB command window.

6 Importing IQ data to MATLAB

Signal Analyzer Digitize Function

The Signal Analyzer 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 Signal Analyzer Digitize function using MATLAB commands.

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

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

% Connect to Signal Analyzer
try
    fopen(msInstr);
    msgbox('Connecting to Signal Analyzer successful');
catch
    errordlg('Cannot connect to Signal Analyzer. 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 Signal Analyzer.
% 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 Signal Analyzer`s D: drive
fprintf(msInstr, ['MMEM:STOR:IQD "' dgznme '", D']);

% close connection
fclose(msInstr);
```

In this example, the signal sent from the Signal Analyzer 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 Signal Analyzer.)

Extracting IQ Data using MATLAB

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.

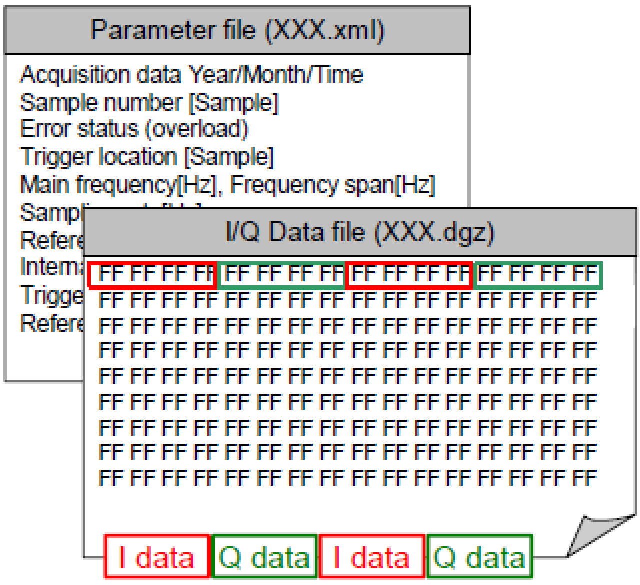


Fig. 4 XML and DGZ files

The first step in extracting the IQ data from the DGZ file is to copy the XML and DGZ files from the Signal Analyzer 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 Signal Analyzer 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 +〈, 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.

Specifications are subject to change without notice.

● United States
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 Eletronica Ltda.
Praça Amadeu Amaral, 27 - 1 Andar
01327-010 - Bela Vista - Sao Paulo - SP
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

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

● France
Anritsu S.A.
12 avenue du Québec, Bâtiment Iris 1- Silic 612,
91140 VILLEBON SUR YVETTE, 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.r.l.
Via Elio Vittorini 129, 00144 Roma, Italy
Phone: +39-6-509-9711
Fax: +39-6-502-2425

● Sweden
Anritsu AB
Kistagången 20B, 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
Torveporten 2, 2500 Valby, Denmark
Phone: +45-7211-2200
Fax: +45-7211-2210

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

● Spain
Anritsu EMEA Ltd.
Representation Office in Spain
Edificio Cuzco IV, Po. de la Castellana, 141, Pta. 5
28046, Madrid, Spain
Phone: +34-915-726-761
Fax: +34-915-726-621

● United Arab Emirates
Anritsu EMEA Ltd.
Dubai Liaison Office
902, Aurora Tower,
P O Box: 500311- Dubai Internet City
Dubai, United Arab Emirates
Phone: +971-4-3758479
Fax: +971-4-4249036

● India
Anritsu India Private Limited
2nd & 3rd Floor, #837/1, Binnamangla 1st Stage,
Indiranagar, 100ft Road, Bangalore - 560038, India
Phone: +91-80-4058-1300
Fax: +91-80-4058-1301

● Singapore
Anritsu Pte. Ltd.
11 Chang Charn Road, #04-01, Shriro House
Singapore 159640
Phone: +65-6282-2400
Fax: +65-6282-2533

● P.R. China (Shanghai)
Anritsu (China) Co., Ltd.
Room 2701-2705, Tower A,
New Caohejing International Business Center
No. 391 Gui Ping Road Shanghai, 200233, P.R. China
Phone: +86-21-6237-0898
Fax: +86-21-6237-0899

● P.R. China (Hong Kong)
Anritsu Company Ltd.
Unit 1006-7, 10/F., Greenfield Tower, Concordia Plaza,
No. 1 Science Museum Road, Tsim Sha Tsui East,
Kowloon, Hong Kong, P.R. China
Phone: +852-2301-4980
Fax: +852-2301-3545

● Japan
Anritsu Corporation
8-5, Tamura-cho, Atsugi-shi, Kanagawa, 243-0016 Japan
Phone: +81-46-296-6509
Fax: +81-46-225-8352

● Korea
Anritsu Corporation, Ltd.
5FL, 235 Pangyoyeok-ro, Bundang-gu, Seongnam-si,
Gyeonggi-do, 13494 Korea
Phone: +82-31-696-7750
Fax: +82-31-696-7751

● Australia
Anritsu Pty. Ltd.
Unit 20, 21-35 Ricketts Road,
Mount Waverley, Victoria 3149, 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

1706

Printed on Recycled Paper

Printed in Japan 公知 2017-7 MG No. MS269xA-E-F-6-(3.00)