User's Manual

MATLAB Toolkit for AQ63xx series Optical Spectrum Analyzer

YOKOGAWA ◆

1st Edition

Thank you for using the MATLAB toolkit for AQ63xx series Optical Spectrum Analyzer (referred to as MATLAB toolkit for OSA).

This User's Manual contains informative precautions, functions, and operating procedure of the MATLAB toolkit for OSA. To ensure correct use, please read this manual thoroughly before beginning operation.

After reading the manual, keep it in a convenient location for quick reference whenever a question arises during operation.

For information about the handling precautions, functions, and operating procedures of Windows, MATLAB, and Yokogawa AQ63xx series OSA, see the manuals respectively.

Note

- The contents of this manual are subject to change without prior notice as a result of continuing improvements to the software's performance and functions. The figures given in this manual may differ from those that actually appear on your screen.
- Copying or reproducing all or any part of the contents of this manual without the permission of Yokogawa Electric Corporation is strictly prohibited.

Disclaimers

By downloading and installing this software, the customer agrees to all of the following disclaimers.

- Yokogawa bears no liability for any problems occurring as a result of downloading or installing this software.
- Yokogawa bears no responsibility for any damage caused directly or indirectly as a result of using this software.
- This software is provided free of charge, however no unlimited warranty against software defects exists, nor is any claim made that the product is free of all defects whatsoever. Also, Yokogawa is not always able to repair defects ("bugs") in, or respond to questions or inquiries about this software.
- Yokogawa reserves all rights to this software, including but not limited to all property rights, ownership rights, and intellectual property rights.

Trademarks

- MATLAB is a registered trademark of The MathWorks, Inc. in the United States.
- Microsoft, MS-DOS, and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- Adobe and Acrobat are registered trademarks or trademarks of Adobe Systems Incorporated.
- For purposes of this manual, the ® and TM symbols do not accompany their respective registered trademark names or trademark names. Other company and product names are registered trademarks or trademarks of their respective companies.

Revisions

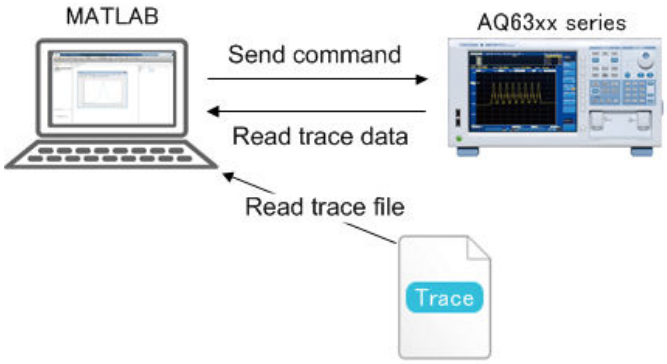
- 1st Edition: January 2017

1st Edition : January 2017 (YMI)

Product Overview

MATLAB toolkit for OSA provides direct data communication easily between MATLAB and the OSA by Yokogawa TMCTL library which is included in this toolkit. This toolkit supports the following features.

- Can be sent remote command from MATLAB to the OSA utilizing the remote interface.
- Can be read trace data into the MATLAB from the OSA utilizing the remote interface.
- Can be read saved trace data file into the MATLAB.
  * GP-IB and Ethernet can be used for the remote interface.



How to control the OSA by MATLAB?

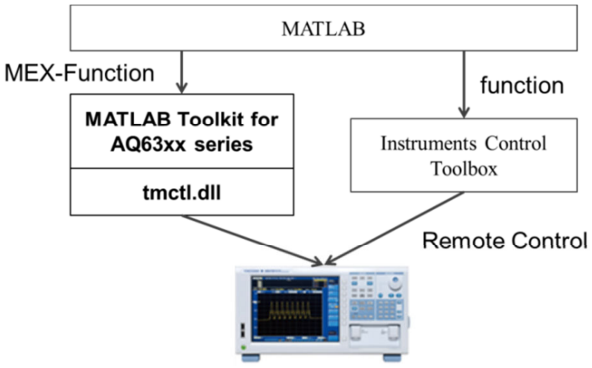
There are two ways in order to control the OSA directly from MATLAB.

- Uses MATLAB Instruments Control Toolbox
- Uses Yokogawa MATLAB toolkit for OSA (this toolkit)

MATLAB Instruments Control Toolbox is one of function group for basic communicating with measuring instruments in MATLAB. It can communicate with many measuring instruments due to support basic communication commands (connect, close, send, receive, etc.).

On the other hand, Yokogawa MATLAB toolkit for OSA can be realized direct data communication between MATLAB and OSA without MATLAB Instruments Control Toolbox. This toolkit is designed to use MATLAB EXTRA FUNCTION (MEX-Function) which supports some basic communication commands and specialized commands for AQ63xx series OSA (ex. execution of single sweep, acquisition of trace data, reading of trace data file, etc.). Detail of each commands are described in the section of MEX-Functions in this manual.

Moreover, this toolkit and MATLAB Instruments Control Toolbox can be used simultaneously.



Note

- The "tmctl.dll" is an API library for remote control of products of Yokogawa.
- This software is designed for converting the commands into MEX-Function by the TMCTL communication function.

PC System Requirements

- **OS**
  Windows 7(32-bit version and 64-bit version)
  Windows 10(32-bit version and 64-bit version)

- **CPU**
  Same as that of MATLAB

- **Memory**
  Same as that of MATLAB

- **Communication Interface**
  GP-IB, Ethernet

- **MATLAB**
  R2011b or later

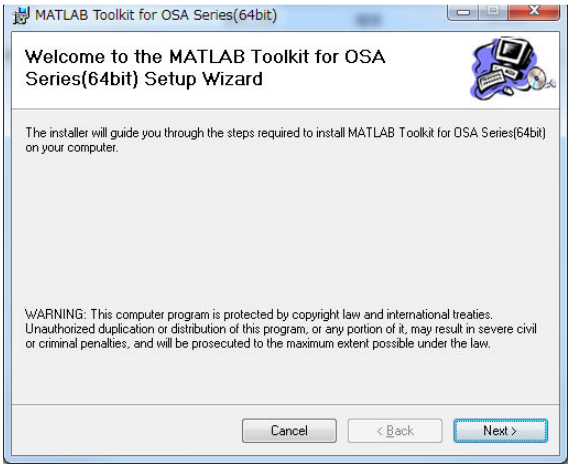
- **OSA**
  AQ637X series

Installing the MATLAB Toolkit for OSA

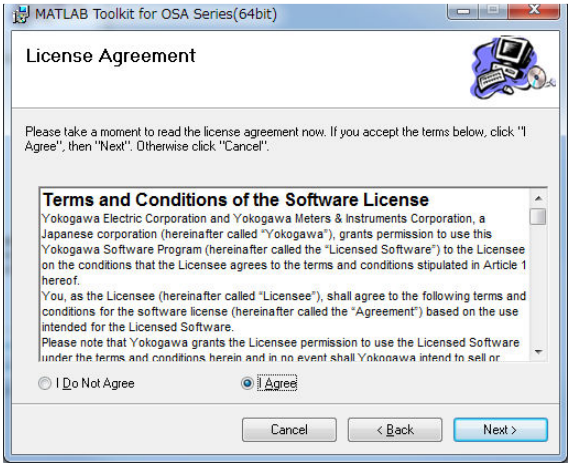
Install procedure

1. Log on to Windows with administrator privileges.
2. Choose the 32bit or 64bit installer according to the environment. The following dialog box opens.

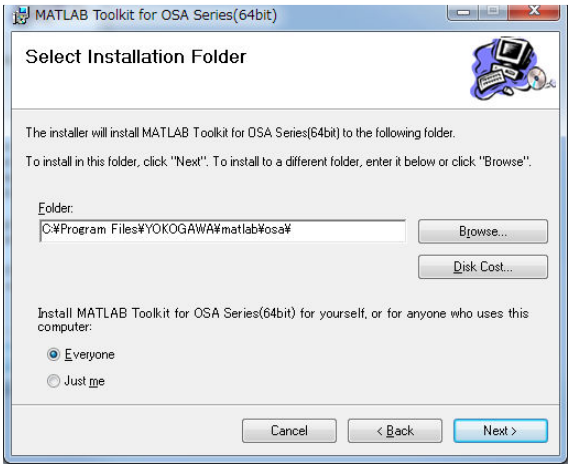
   (Gives an example of the operation using the Windows 7(x64))



3. Click **Next>**.
4. Check the **I Agree** option button, and click **Next>**.



5. Specify the installation destination folder and click **Next>**.

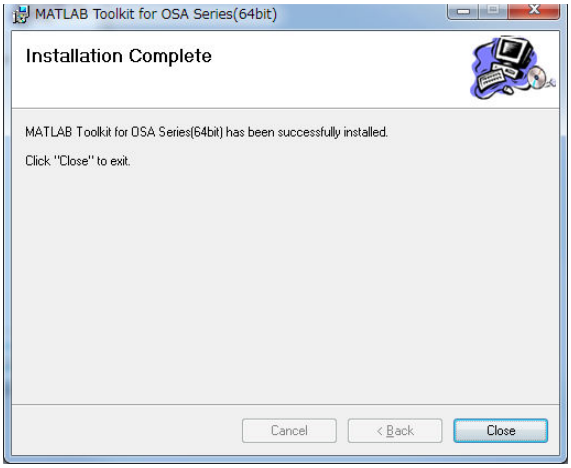


Note

- The default installation destination folders are as follows.

  C:\Program Files\YOKOGAWA\matlab\osa

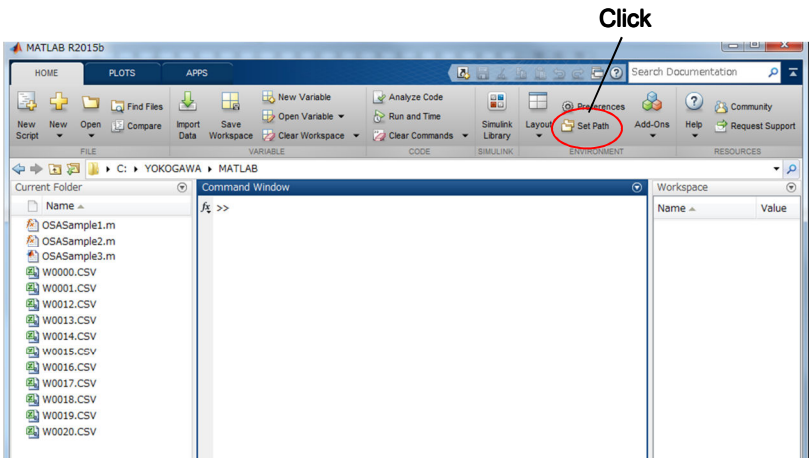
6. Click **Next>** to start the installation.
7. If the installation completes successfully, Click **Close**. Installation is complete.



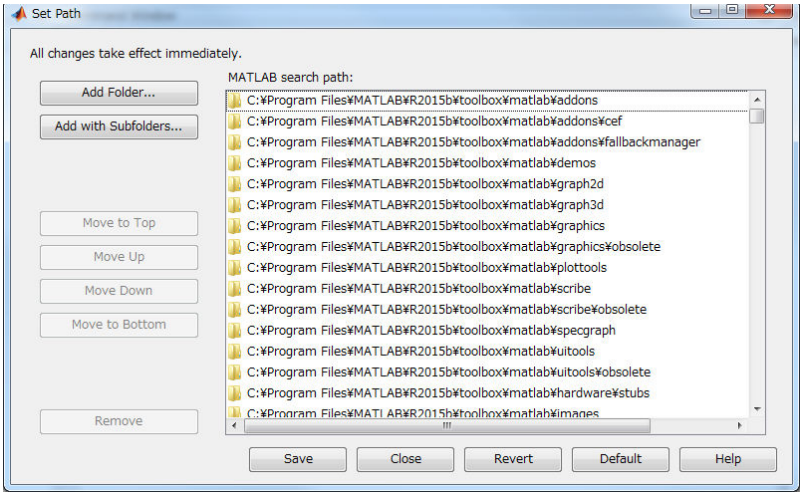
Initial setting of MATLAB

Carry out the procedure below once after installing this software.

1. Start MATLAB.
2. Click **Set Path** of the toolbar. The Set Path dialog box opens.



3. Click the **Add Folder Button** and select the folder in which the MATLAB Toolkit for OSA was installed.

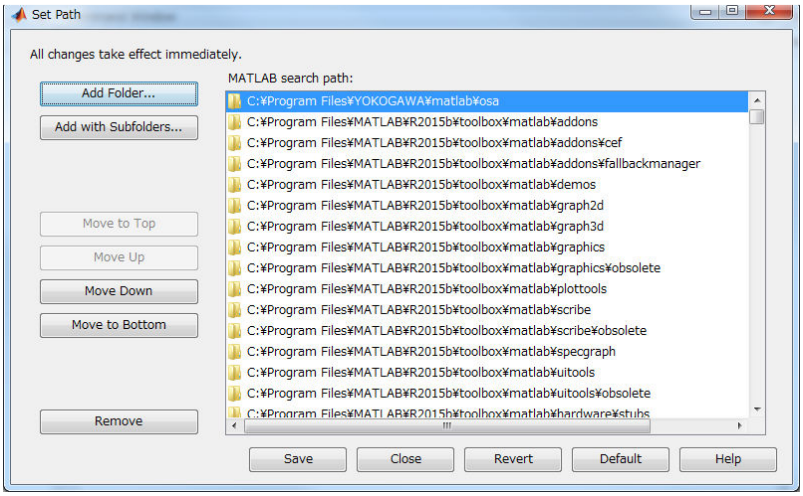


Note

- The default installation destination folders are as follows.

  C:\Program Files\YOKOGAWA\matlab\osa

4. Click the **Save** button and then the **Close** button.



MEX-Functions

When remote control the OSA or retrieve trace data, input MEX-Functions for OSA or OSA communication commands on the Command Window of MATLAB.

List of MEX-Functions

MEX-Function Name	Function
[ret] =mexOSAComStart(wire, adr)	Initializes the line and connects the line to the specified device.
[ret] = mexOSAComEnd()	Closes the line connected to the device.
[ret] = mexOSADeviceClear()	Executes the clearing (SDC) of the selected device.
[ret] = mexOSASend(msg)	Sends a message to the device.
[ret, buf, size] = mexOSAReceive()	Receives a message from the device.
[ret, buf, size] = mexOSASendReceive(msg)	Sends a message and receives a message
[ret] = mexOSASetRen(flag)	Sets the device in remote or local mode. Use of an interface other than GP-IB is limited to Yokogawa products.
[errorID] = mexOSAGetLastError()	Returns the number of the last error that occurred due to a MATLAB command.
help mexOSAGetVersion()	Displays the version information of the software.
[ret] = mexOSASetTimeout(timeout)	Sets the communication timeout time.
[ret] = mexOSASweepSingle()	Execute Single sweep and wait the operation complete.
[ret, data] = mexOSAGetTrace(trace) [ret, data] = mexOSAGetTrace(trace, start, end)	Gets the trace data from the device. If set the start and end, device returns the data from that range.
[ret, cond] = mexOSAGetMeasCond()	Gets the measurement conditions from the device.
[ret] = mexOSASetMeasCond(cond)	Sets the measurement conditions to the device.
[ret, dataX, dataY, cond] = mexOSAReadFile(filename)	Reads the trace data and measurement conditions from CSV file.

1.1.1 [ret] = mexOSAComStart(wire, adr)

Function: Initializes the line and connects the line to the specified device.

Parameters: wire : Line type

The value for each device is as follows:

GP-IB : wire = 1

Ethernet : wire = 4

adr : Character sequence of the line-specific address

Below are the settings for each interface.

GP-IB : adr = “1” to “30” (GP-IB address of the device)

Ethernet : adr = “IP address”,”Port”,”Username”,”Password”

Return value: ret(0 = OK, 1 = ERROR)

Example:

- Case of GP-IB

  [ret] = mexOSAComStart(1, ‘1’);
- Case of Ethernet(Username is “anonymous”)

  [ret] = mexOSAComStart(4, ‘192.168.1.100,10001,anonymous,’);
- Case of Ethernet(Username is “testUser”, and password is “12345678”)
- [ret] = mexOSAComStart(4, ‘192.168.1.100,10001,testUser,12345678,’);

1.1.2 [ret] = mexOSAComEnd()

Function: Closes the line connected to the device.

Parameters: None

Return value: ret(0 = OK, 1 = ERROR)

Description: Closes the line that was opened using mexOSAComStart(initialization function).

Be sure to execute this function when terminating the communication.

Example: [ret] = mexOSAComEnd

1.1.3 [ret] = mexOSADeviceClear()

Function: Executes the clearing (SDC) of the selected device.

A dedicated command for the GP-IB

Parameters: None

Return value: ret(0 = OK, 1 = ERROR)

Description: This function applies only to the device connected to the GP-IB; it does nothing to

a device connected by a different interface.

Example: [ret] = mexOSADeviceClear

1.1.4 [ret] = mexOSASend()

Function: Sends a message to the device.

Parameters: msg : Character sequence of the OSA Communication command

Return value: ret(0 = OK, 1 = ERROR)

Example:

- Case of send “*IDN?”.

  [ret] = mexOSASend(‘*IDN?’);

1.1.5 [ret, buf, size] = mexOSAReceive()

Function:	Receives a message from the device.
Parameters:	None
Return value:	ret(0 = OK, 1 = ERROR)
	buf : Received data buffer
	size : The actual number of received bytes.
Example:	[ret, buf, size] = mexOSAReceive

1.1.6 [ret, buf, size] = mexOSASendReceive(msg)

Function:	Sends a message and receives a message.
Parameters:	msg : Character sequence of the OSA Communication command
Return value:	ret(0 = OK, 1 = ERROR)
	buf : Received data buffer
	size : The actual number of received bytes.
Example:	

- Case of send “*IDN?”.
  [ret, buf, size] = mexOSASendReceive(‘*IDN?’);

1.1.7 [ret] = mexOSASetRen(flag)

Function:	Sets the device in remote or local mode. Use of an interface other than GP-IB is limited to Yokogawa products.
Parameters:	flag : (0 = Local, 1 = Remote)
Return value:	ret(0 = OK, 1 = ERROR)
Description:	For GP-IB, the REN line is set to TRUE/FALSE. Therefore, remote mode is actually enabled when a message is sent to the device.
Example:	[ret] = mexOSASetRen(1)

1.1.8 [errorID] = mexOSAGetLastError()

Function: Returns the number of the last error that occurred due to a MATLAB command (not the error code on the OSA).

Parameters: None

Return value: errorID : Last error number

Description: The list of error IDs is as follows

0x00000000(0)	No error
0x00000001(1)	Timeout
0x00000002(2)	Device Not Found
0x00000004(4)	Open Port Error
0x00000008(8)	Device Not Open
0x00000010(16)	Device Already Open
0x00000020(32)	Controller Not Found
0x00000040(64)	Parameter is illegal
0x00000100(256)	Send Error
0x00000200(512)	Receive Error
0x00000400(1024)	Data is not Block Data
0x00001000(4096)	System Error
0x00002000(8192)	Device ID is Illegal

When the return value of a function including the initialization function is not 0(=OK), this function is used to retrieve the actual error number.

取得されるエラーの内容は TMCTL ライブラリと同様のものです。

Example: [errorID] = mexOSAGetLastError

1.1.9 help mexOSAToolkit

Function: Displays the version information of the software.

Parameters: None.

Return value: None.

Example: help mexOSAToolkit

MATLAB Toolkit for OSA Series
Version *.**

1.1.10 [ret] = mexOSASetTimeout(timeout)

Function: Sets the communication timeout time.

Parameters: timeout : Timeout time(100 ms unit). (0 or 100 to 6553600ms)

If timeout = 0,

GP-IB : Timeout set to infinity.

Ethernet : Timeout not set.

Note

"Infinity" means that the wait time is set to an infinite amount of time. "Not set" means that there is no wait time(responds immediately)

Return value: ret(0 = OK, 1 = ERROR)

Description: For Yokogawa product, set the time greater than equal to 30s.

(Even if the timeout time is set long, the overall performance is not affected.)

Example:

- Case of setting timeout to 30 seconds.

  [ret] = mexOSASetTimeout(300)

1.1.11 [ret] = mexOSASweepSingle()

Function: Execute Single sweep and wait the operation complete.

Parameters: None.

Return value: ret(0 = OK, 1 = ERROR)

Example: [ret] = mexOSASweepSingle

1.1.12 [ret, dataX, dataY] = mexOSAGetTrace(trace) or (trace, start, end)

Function: Gets the trace data from the device.

Parameters: trace : trace name to retrieve data

The setting range is as follows

trace = 'TRA' to 'TRG'

start : First data point to be acquired

end : Last data point to be acquired

Return value: ret(0 = OK, 1 = ERROR)

dataX : Acquired data(Wavelength)

dataY : Acquired data(Power)

Description: If set the start and end, device returns the data from that range. Otherwise this function will acquire the data of the whole range.

Example:

- Case of acquire whole data of trace C.

  [ret, dataX, dataY] = mexOSAGetTrace('TRC')

- Case of acquire data from 300 points to 500 points of trace F.

  [ret, dataX, dataY] = mexOSAGetTrace('TRF', 300, 500)

1.1.13[ret, cond] = mexOSAGetMeasCond()

Function: Gets the measurement conditions from the device.

Parameters: None.

Return value: ret(0 = OK, 1 = ERROR)

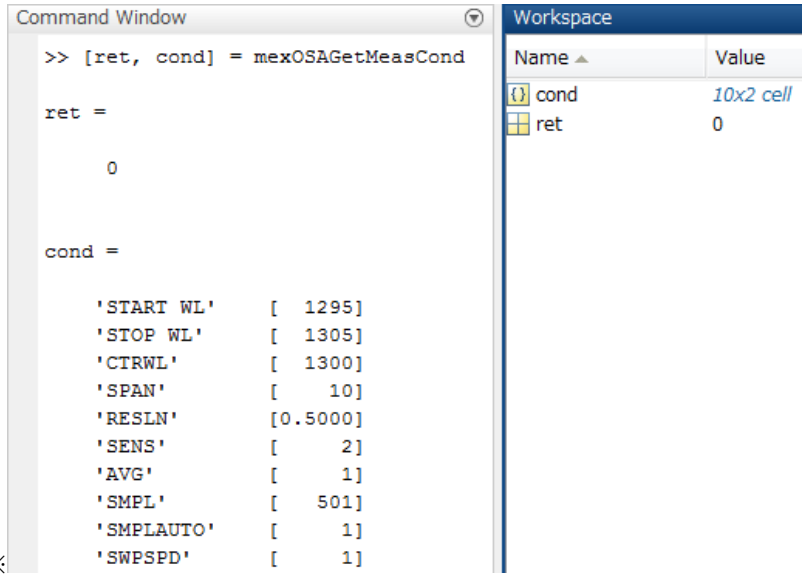
cond : Acquired measurement conditions array.

Description: The structure of “cond” is as follows.

Tag	Value
START WL	Start wavelength
STOP WL	Stop wavelength
CTRWL	Center wavelength
SPAN	Span wavelength
RESLN	Resolution
SENS	Sensitivity 0=NORMAL HOLD 1=NORMAL AUTO 2=MID 3=HIGH1 4=HIGH2 5=HIGH3 6=NORMAL
AVG	Average times
SMPL	Sampling Points
SMPLAUTO	Sampling Points Auto 0=OFF 1=ON
SWPSPD	Sweep Speed 0=1x 1=2x

Example: [ret, cond] = mexOSAGetMeasCond

ex) Output image



※

1.1.14 [ret] = mexOSASetMeasCond(cond)

Function: Sets the measurement conditions to the device.

Parameters: cond : Acquired measurement conditions array.

Return value: ret(0 = OK, 1 = ERROR)

Description: The structure of “cond” is as follows.

Tag	Value
START WL	Start wavelength
STOP WL	Stop wavelength
CTRWL	Center wavelength
SPAN	Span wavelength
RESLN	Resolution
SENS	Sensitivity 0=NORMAL HOLD 1=NORMAL AUTO 2=MID 3=HIGH1 4=HIGH2 5=HIGH3 6=NORMAL
AVG	Average times
SMPL	Sampling Points
SMPLAUTO	Sampling Points Auto 0=OFF 1=ON
SWPSPD	Sweep Speed 0=1x 1=2x

If the setting value is blank or invalid, it is ignored.

Example: [ret] = mexOSASetMeasCond(cond)

1.1.15 [ret, dataX, dataY, cond] = mexOSAFileRead(filename)

Function: Reads the trace data and measurement conditions from CSV file.

Parameters: filename : File name of the data to be read.

Return value: ret(0 = OK, 1 = ERROR)

dataX : Acquired data(Wavelength)

dataY : Acquired data(Power)

cond : Acquired measurement conditions array.

Example: [ret, dataX, dataY, cond] = mexOSAFileRead(‘W0000.CSV’)

How to use the MATLAB Toolkit for OSA

Setting the communication interface of the OSA

Set the remote interface to be used from the front panel of the OSA.

The setting procedure is as follows.

1. Push the [SYSTEM] key.
2. Push the <More 1/4> button.
3. Push the <Remote Interface> button and choose remote interface to use.

Note

- Following the remote interfaces are supported by the MATLAB toolkit for OSA.

  GP-IB

  Ethernet

Execution example on the Command Window

These examples set measurement conditions and executes sweep.

```
>> mexOSAComStart(1, ‘1’);
>> mexOSASend(‘:SENS:WAV:CENT 1550nm’);
>> mexOSASend(':SENS:WAV:SPAN 5nm');
>> mexOSASend(':SENS:BAND 0.1nm');
>> mexOSASend(':SENS:SENS MID');
>> mexOSASend(':SENS:SWE:SPE 2x');
>> mexOSASweepSingle;
>> mexOSAComEnd;
```

Sample programs

Some sample programs are located in the folder in which the MATLAB Toolkit for OSA was installed.

Sample program 1

Below is a program for connection the communication line and sending commands.
Remove the % character from the program line corresponding to the interface type you are using. The sample program is OSASample1.m.
Example [ret, data, size] = OSASample1

```
% Example-1：GPIB　( address = 1 )
ret = mexOSAComStart(1,'1');
% Example-2：Ethernet
%(IP = 192.168.1.100, Port = 10001, Username = yokogawa, Password = 1234)
% ret = mexOSAComStart(4, '192.168.1.100,10001,yokogawa,1234');

% Error Check
if ret ~= 0
    ret = mexOSAGetLastError;
    return;
end;

% set timeout(30sec)
ret = mexOSASetTimeout(300);
if ret ~= 0
      ret = mexOSAGetLastError;
      return;
end;

% sending IDN? & receiving query
ret = mexOSASend('*IDN?');
if ret ~= 0
      ret = mexOSAGetLastError;
      return;
end;

[ret,Buf,Size] = mexOSAReceive();
if ret ~= 0
      ret = mexOSAGetLastError;
      return;
end;

% Places a device in local mode
ret = mexOSASetRen(0);
if ret ~= 0
      ret = mexOSAGetLastError;
      return;
end;
ret = mexOSAComEnd;
if ret ~= 0
      ret = mexOSAGetLastError;
      return;
end;
```

Sample program 2

Below is a program for execute setting the measurement conditions to the OSA.

The sample program is OSASample2.m.

Example ret = OSASample2

```
% Example-1 : GPIB    ( address = 1 )
ret = mexOSAComStart(1,'1');
% Example-2 : Ethernet
% (IP = 192.168.1.100, Port = 10001, Username = yokogawa, Password = 1234)
% ret = mexOSAComStart(4, '192.168.1.100,10001,yokogawa,1234');

% Error Check
if ret ~= 0
    ret = mexOSAGetLastError;
    return;
end;

% set timeout(30sec)
ret = mexOSASetTimeout(300);
if ret ~= 0
      ret = mexOSAGetLastError;
      return;
end;

% Get measurement conditions
[ret, cond] = mexOSAGetMeasCond();
if ret ~= 0
      ret = mexOSAGetLastError;
      return;
end;

% Please change the setting value as necessary.(Blank([]) items are not set.)
startWavelength = 1545.5;
stopWavelength = 1555.5;
centerWavelength = [];
spanWavelength = [];
resolution = 0.5;
sensitivity = 2;                    % MID
averageTimes = 1;
samplingPoints = [];
samplingPointsAuto = 1;         % Enabling the Sampling Points Auto
sweepSpeed = 1;                    % Sweep Speed 2x

% Change the measurement conditions.
setValue = {startWavelength;
                stopWavelength;
                centerWavelength;
                spanWavelength;
                resolution;
                sensitivity;
                averageTimes;
                samplingPoints;
                samplingPointsAuto;
```

```
                    sweepSpeed};
[cond{:, 2}] = setValue{:, 1};

% Set measurement condition
ret = mexOSASetMeasCond(cond);
if ret ~= 0
      ret = mexOSAGetLastError;
      return;
end;

%Close communication port
ret = mexOSAComEnd;
if ret ~= 0
      ret = mexOSAGetLastError;
      return;
end;
```

Sample program 3

Below is a program for execute single sweep and receive data from OSA.

The sample program is OSASample3.m.

Example [ret, dataX, dataY, cond] = OSASample3

```
% Example-1：GPIB　( address = 1 )
ret = mexOSAComStart(1,'1');
% Example-2：Ethernet
%(IP = 192.168.1.100, Port = 10001, Username = yokogawa, Password = 1234)
% ret = mexOSAComStart(4, '192.168.1.100,10001,yokogawa,1234');

% Error Check
if ret ~= 0
    ret = mexOSAGetLastError;
    return;
end;

% set timeout(30sec)
ret = mexOSASetTimeout(300);
if ret ~= 0
      ret = mexOSAGetLastError;
      return;
end;

% set measurement conditions(No error check)
mexOSASend(':TRAC:ACT TRA')
mexOSASend(':SENS:WAV:STAR 1545.000nm');
mexOSASend(':SENS:WAV:STOP 1555.000nm');
mexOSASend(':SENS:BAND 0.05nm');
mexOSASend(':SENS:SENS NORMAL');
mexOSASend(':SENS:SWE:SPE 2x');

% start sweep
ret = mexOSASweepSingle;
if ret ~= 0
      ret = mexOSAGetLastError;
      return;
end;

% receive tracedata
[ret, dataX, dataY] = mexOSAGetTrace('TRA');
if ret ~= 0
      ret = mexOSAGetLastError;
      return;
end;

% receive measurement condition
[ret, cond] = mexOSAGetMeasCond();
if ret ~= 0
      ret = mexOSAGetLastError;
      return;
end;
```

```
% get XY scales unit
[ret, unitX, ~] = mexOSASendReceive(':UNIT:X?');
if ret ~= 0
	ret = mexOSAGetLastError;
	return;
end;

[ret, unitY, ~] = mexOSASendReceive(':DISP:TRAC:Y1:UNIT?');
if ret ~= 0
	ret = mexOSAGetLastError;
	return;
end;

%Close communication port
ret = mexOSAComEnd;
if ret ~= 0
	ret = mexOSAGetLastError;
	return;
end;

% Display trace data to a graph
plot(dataX, dataY);
ylim([-80, 10]);
title('TRA');
xlabel(strcat('Wavelength [', unitX, ']'));
ylabel(strcat('Power [', unitY, ']'));
```

Sample program 4

Below is a program for retrieving the tracedata and measurement conditions from CSV file.

The sample program is OSASample4.m.

Example [ret, dataX, dataY, cond] = OSASample4

```
% Input the Filename
filename = input('filename = ', 's');
[ret, dataX, dataY, cond] = mexOSAFileRead(filename);

% Display trace data to a graph
plot(dataX, dataY);
ylim([-80, 10]);
xlabel(strcat('Wavelength [nm]'));
ylabel(strcat('Power[dBm]'));
```