

PW4001

HIOKI

PW4001-01
PW4001-02
PW4001-03
PW4001-04
PW4001-05

MATLAB Toolkit
User's Manual

POWER ANALYZER



Check for the latest edition and
other language versions.



- ✓ This instruction manual explains the communication commands for the PW4001 Power Analyzer.
- ✓ Before using the PW4001, be sure to read the instruction manual of the PW4001.
- ✓ For details regarding the command settings, please refer to “9 Connecting with Computers” in the instruction manual for the PW4001.
- ✓ Although all reasonable care has been taken in the production of this instruction manual, should you find any points which are unclear or in error, please contact your local distributor or Hioki's website.

EN

Contents

- 1 Overview 1
- 2 Ready for use..... 1
- 3 Control PW4001 on MATLAB via Ethernet connection..... 2
- 4 tcpipPW4001Class..... 4
 - Member Functions..... 4

1 Overview

This software consists of a MATLAB^{*1} script (tcpipPW4001.p) to control our power analyzer PW4001 connected via Ethernet with MATLAB.

It is recommended to use this toolkit with MATLAB R2022a or later. In addition, this toolkit cannot be used to control PW4001 with MATLAB using a communication interface other than Ethernet (USB).

^{*1} Third-party company's trademark

2 Ready for use

First, make sure you have this manual file (Power Analyzer PW4001 Matlab Toolkit User's Manual.pdf) and the script file (tcpipPW4001.p) in MatlabToolkitForPW4001.zip and place them in an appropriate location.

And, add the location to the MATLAB search path.

3 Control PW4001 on MATLAB via Ethernet connection

By using the `tcpipPW4001` class provided by this toolkit, communication commands (see next page for details) are sent from MATLAB to the Ethernet-connected PW4001 and its response is received.

Examples of execution are shown in Figures 1 and 2. In Figure 1, analog waveform data is acquired. The procedure is described below.

1. Specifies the IP address of PW4001 in MATLAB and generates an object.
2. Establishes an Ethernet connection between the generated object and the PW4001.
3. Sends a communication command from MATLAB to PW4001 and receives a response from PW4001 to the communication command.
4. Selects one analog waveform to be acquired from PW4001 and captures the maximum and minimum values of the data.
5. Displays the acquired data with the plot function.
6. Disconnects the Ethernet connection.

Executes steps 1. through 6. in the command window of MATLAB, resulting in the following process.

```
>> [obj, flag] = tcpipPW4001("192.168.10.11");  
% Generates a TCP/IP object for Ethernet connection with specified IP address of PW4001.  
>> obj.open; % Ethernet connection to PW4001  
>> obj.send("*IDN?"); %Sends the *IDN? command  
>> obj.receive; %Receives command response  
>> [samplingSpeed, storageLength, storageMode, waveDataMax,...  
    waveDataMin, flag] = obj.DownloadAnalogWaveData("U1"); % Captures voltage waveform data of U1  
>> plot(waveDataMax); % Plots the data sequence of the maximum value of the voltage waveform of U1  
>> obj.close; %Disconnects the Ethernet connection
```

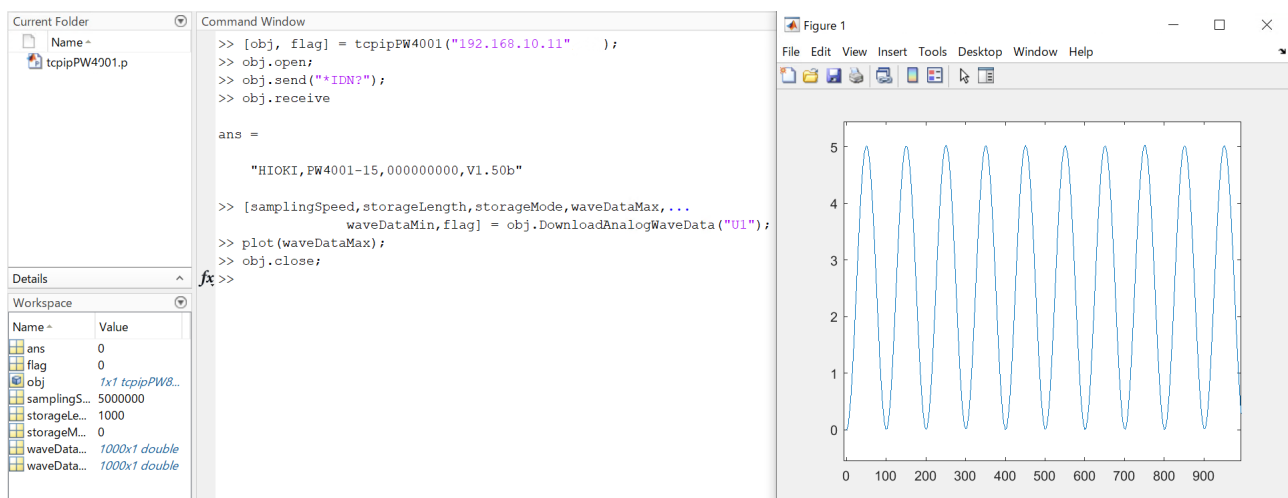


Figure1 :Execution example of acquiring analog waveform data (controlling PW4001 with MATLAB via Ethernet connection)

Next, in Figure 2, pulse waveform data is acquired. The procedure is described below.

1. Specifies the IP address of PW4001 in MATLAB and generates the object.
2. Establishes an Ethernet connection between the generated object and PW4001.
3. Sends a communication command from MATLAB to PW4001 and receives a response from PW4001 to the communication command.
4. Captures information on the motor channels set for pulse input from PW4001, as well as the maximum and minimum data values for all those channels.
5. Selects one channel from the data of the motor channels set for pulse input, selects the maximum or minimum value of the pulse waveform data, and acquires it.
6. Displays the acquired data with the plot function.
7. Disconnects the Ethernet connection.

Executes steps 1. through 7. in the command window of MATLAB, resulting in the following process.

```
>> [obj, flag] = tcpipPW4001("192.168.10.11");
% Generates a TCP/IP object for Ethernet connection with specified IP address of PW4001.
>> obj.open; % Ethernet connection to PW4001
>> obj.send("*IDN?"); %Sends the *IDN? command
>> obj.receive; %Receives command response
>> [samplingSpeed, storageLength, storageMode, logicCH,...
    waveDataMax, waveDataMin, flag] = obj.DownloadLogicWaveData;
% Obtains data including all waveforms set for pulse input.
>> [chWaveData, flag] = obj.ExtractLogicChWaveData("CHA",waveDataMax);
% Captures the maximum value of CHA pulse waveform data
>> plot(chWaveData); % Plots a data sequence of the maximum values of the voltage waveforms of CHA
>> obj.close; % Disconnects the Ethernet connection
```

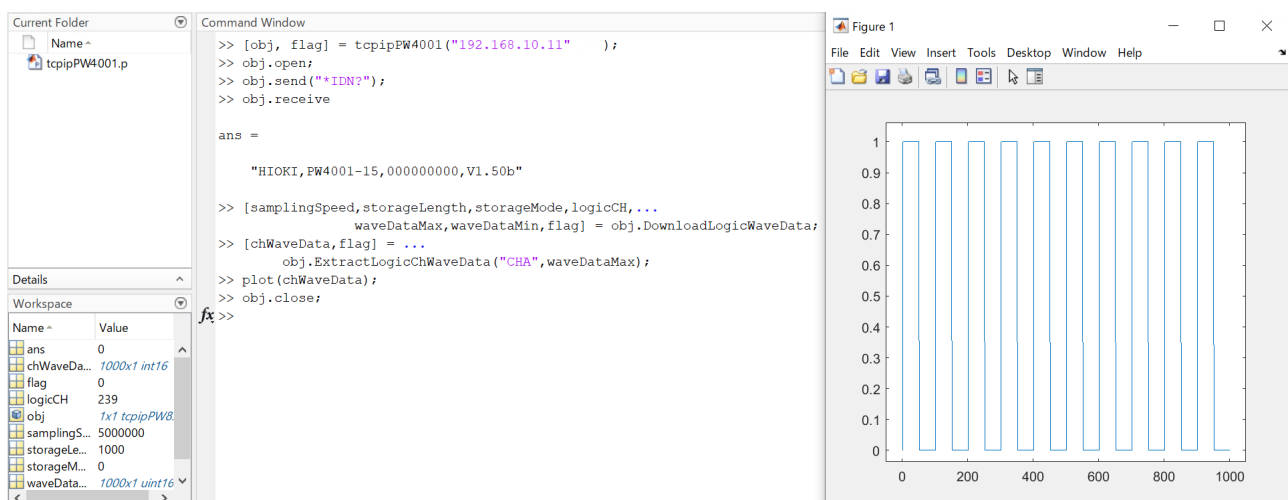


Figure2: Execution example of acquiring pulse waveform data (controlling PW4001 with MATLAB via Ethernet connection)

4 tcpipPW4001 Class

This class is used to control PW4001 with MATLAB via Ethernet connection. Through this class, communication commands can be sent to and received from the PW4001 and waveform data can be acquired.

Member Functions

```
[obj, flag] = tcpipPW4001(ipAddr);
```

Argument	ipAddr: String of IP address of PW4001
Return Value	obj: Generated tcpipPW4001 object flag: "0" if the object could be generated, "1" or a predefined MATLAB error message in case of abnormal termination
Function	Constructor for tcpipPW4001 class object; creates a tcpipPW4001 object to connect to PW4001 and returns the flag.
Example	[obj, flag] = tcpipPW4001("192.168.10.11"); Generates a TCP/IP object for Ethernet connection with specified IP address of PW4001.

```
[ipAddress, flag] = obj.ipAddr;
```

Argument	None
Return Value	ipAddress: String of IP address of PW4001 flag: "0" if IP address could be obtained, "1" or a predefined MATLAB error message in case of abnormal termination
Function	Returns the IP address string for PW4001 and the flag.
Example	obj.ipAddr;

```
[aTime, flag] = obj.timeout
```

Argument	None
Return Value	aTime: Timeout time for communication with PW4001 (sec) flag: "0" if timeout time could be obtained, "1" or a predefined MATLAB error message in case of abnormal termination
Function	Returns the timeout time (in seconds) for communication with PW4001 and the flag.
Example	obj.timeout;

```
flag = obj.setTimeout(aTimeout)
```

Argument	aTimeout: Variable that sets the timeout period for communication with PW4001
Return Value	flag: "0" if timeout time could be set, "1" or a predefined MATLAB error message in case of abnormal termination
Function	Sets the timeout period (in seconds) for communication with PW4001 and returns the flag.
Example	obj.setTimeout(15); % Sets the timeout period for communication with PW4001 to 15 seconds

flag = obj.open

Argument	None
Return Value	flag: "0" if the connection between the generated object and PW4001 can be established, "1" or a predefined MATLAB error message in case of abnormal termination
Function	Establishes a connection between the generated object and PW4001 and returns the flag.
Example	obj.open;

flag = obj.close

Argument	None
Return Value	flag: "0" if the connection between the generated object and the PW4001 can be disconnected, "1" or a predefined MATLAB error message in case of abnormal termination
Function	Disconnects the generated object from PW4001 and returns the flag.
Example	obj.close;

flag = obj.send(command)

Argument	command: Communication command string to be sent to PW4001
Return Value	flag: "0" if the command could be sent to PW4001, "1" or a predefined MATLAB error message in case of abnormal termination
Function	Sends a communication command string to PW4001 and returns the flag.
Example	obj.send("*IDN?"); % Sends the communication command "*IDN?" to PW4001

[str, flag] = obj.receive

Argument	None
Return Value	str: Responses string of PW4001 to a communication command sent from MATLAB flag: "0" if a response is received from PW4001, "1" or a predefined MATLAB error message in case of abnormal termination
Function	Receives the PW4001 response string to the communication command sent and returns the flag.
Example	obj.receive;

[samplingSpeed, storageLength, storageMode, waveDataMax,... waveDataMin, flag] = obj.DownloadAnalogWaveData(chName)	
chName: String of the following target waveform names to be retrieved from PW4001 (U1, U2, U3, U4, I1, I2, I3, I4, CHA or CHC) (However, CHA and CHC are limited to analog waveforms)	
Argument	
Return Value	samplingSpeed: Sampling rate of waveform data storageLength: Number of points of waveform data storageMode: Storage mode of waveform data (Returns 0 for peak compression.) waveDataMax: Array to store the maximum value of the specified analog waveform data waveDataMin: Array to store the minimum value of the specified analog waveform data flag: "0" if analog waveform data could be obtained from PW4001, "1" or a predefined MATLAB error message in case of abnormal termination
Function	Obtains the maximum and minimum values of the specified analog waveform data from the PW4001 and returns the flag.
Example	[samplingSpeed, storageLength, storageMode, waveDataMax, waveDataMin, flag] = obj.DownloadAnalogWaveData("U1"); % Captures the maximum and minimum voltage waveform data of U1

[samplingSpeed, storageLength, storageMode, logicCH,...
waveDataMax, waveDataMin, flag] = obj.DownloadLogicWaveData

Argument

None

samplingSpeed: Sampling rate of waveform data

storageLength: Number of points of waveform data

storageMode: Storage mode of waveform data
(Returns 0 for peak compression.)

logicCH: Motor channel set to pulse input

※Bit correspondence between logicCH and motor channel

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
				CHD	CHC	CHB	CHA

Return Value

waveDataMax: Array to store the maximum value of the data including all waveforms of the motor channel set for pulse input

waveDataMin: Array to store the minimum value of the data including all waveforms of the motor channel set for pulse input

※Bit correspondence between waveDataMax, waveDataMin and motor channel

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
CHA	CHB	CHC	CHD				
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
-	-	-	-	-	-	-	-

flag: "0" if data including all waveforms of the motor channel set for pulse input can be acquired from PW4001, "1" or a predefined MATLAB error message is displayed in case of abnormal termination.

Function

Obtains data including all waveforms set for pulse input from PW4001 and returns the flag.

Example

[samplingSpeed, storageLength, storageMode, logicCH, waveDataMax, waveDataMin, flag] =
obj.DownloadLogicWaveData;

[chWaveData, flag] = obj.ExtractLogicChWaveData(chName, waveData)	
chName: String of motor channel name set for pulse input	
Argument	(CHA, CHB, CHC or CHD)
	waveData: Either waveDataMax or waveDataMin obtained with the DownloadLogicWaveData function
	chwaveData: Array to store the maximum or minimum pulse waveform data for the specified motor
Return	channel
Value	flag: "0" if pulse waveform data of the specified motor channel could be acquired from PW4001, "1" or a predefined MATLAB error message in case of abnormal termination
Function	Obtains the maximum or minimum pulse waveform data of the specified motor channel from PW4001 and returns the flag.
Example	[chWaveData, flag] = ExtractLogicChWaveData(obj, "CHA", waveDataMax); % Captures the maximum value of CHA pulse waveform data

HIOKI

www.hioki.com/



**All regional
contact
information**

HIOKI E.E. CORPORATION

81 Koizumi, Ueda, Nagano 386-1192 Japan

2402 EN

Edited and published by HIOKI E.E. CORPORATION

Printed in Japan

- Contents subject to change without notice.
- This document contains copyrighted content.
- It is prohibited to copy, reproduce, or modify the content of this document without permission.
- Company names, product names, etc. mentioned in this document are trademarks or registered trademarks of their respective companies.

Europe only

- EU declaration of conformity can be downloaded from our website.
- Contact in Europe: HIOKI EUROPE GmbH
Helfmann-Park 2, 65760 Eschborn, Germany hioki@hioki.eu