Power Analyzer PW6001

MATLAB Toolkit User's Manual

Overview

This software package consists of two components: a MATLAB script (binFileReadPW6001.p) that loads binary waveform data (comprised of BIN files) recorded by a Hioki Power Analyzer PW6001 as a MATLAB array, and a MATLAB script (tcpipPW6001.p) and JAR file (TcpipPW6001.jar) that together control the PW6001 from inside MATLAB over an Ethernet connection.

This toolkit supports MATLAB R2016a or later versions. It cannot be used to control the PW6001 from inside MATLAB using any communications interface other than Ethernet (GPIB, RS-232C, etc.).

Getting ready to use the toolkit

Copy the contents of the toolkit folder (binFileReadPW6001.p, tcpipPW6001.p, and TcipipPW6001.jar), which can be found in the MatlabToolkitForPW6001.zip file along with this manual file (manual.pdf), to a suitable location on your computer. Add the location to which you copied the files to the MATLAB search path, and, if you have not already installed the Instrument Control Toolbox from The MathWorks, Inc., for your copy of MATLAB, add the location to TcpipPW6001.jar as a Java class path, too. For more information about how to add search paths and Java class paths, see the MATLAB online manual.

---

*MATLAB is a registered trademark of The MathWorks, Inc.

**Java is a registered trademark of Oracle Corporation.

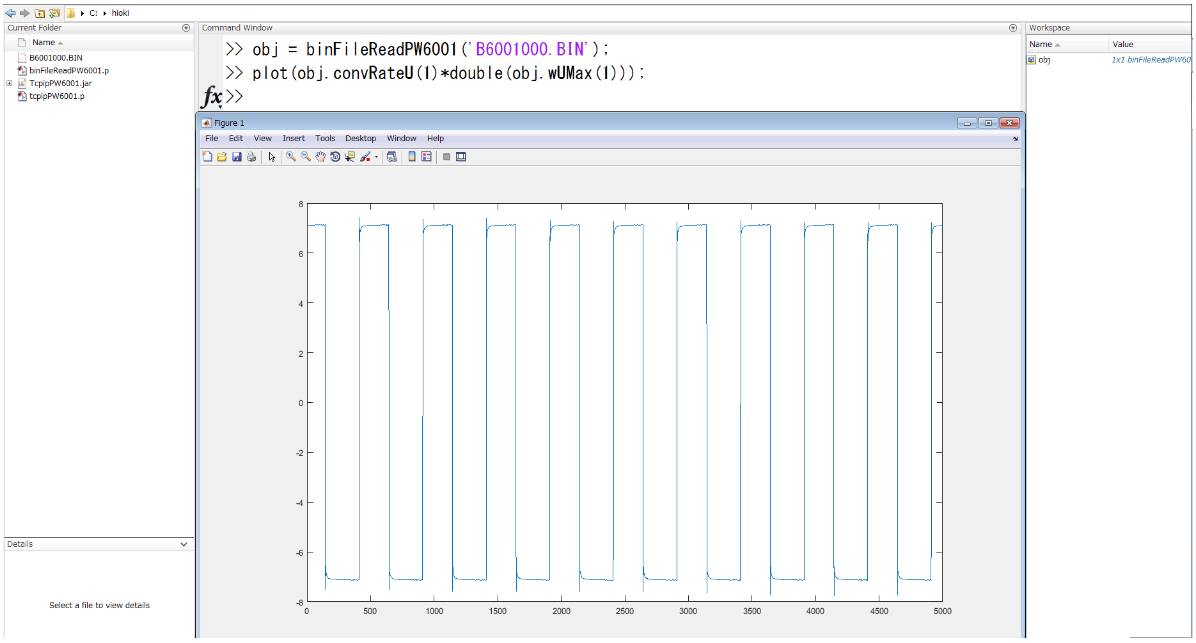
***Ethernet is a registered trademark of Fuji Xerox Co., Ltd.

Loading binary waveform data as a MATLAB array

By using the binFileReadPW6001 class provided by this toolkit, you can easily load binary waveform data (comprised of BIN files) as an array in MATLAB.

In Fig. 1, which provides an example of this process, a binary waveform data file (B6001000.BIN) has been loaded in the workspace, CH 1 voltage scale values and waveform data have been read from the binary file, and the waveform data has displayed using the plot function.

```
>> obj = binFileReadPW6001('B6001000.BIN');% Load binary file.
>> plot(obj.convRateU(1)*double(obj.wUMax(1)));% Plot the CH 1 voltage waveform data.
```



```
>> obj = binFileReadPW6001('B6001000.BIN');
>> plot(obj.convRateU(1)*double(obj.wUMax(1)));
fx >>
```

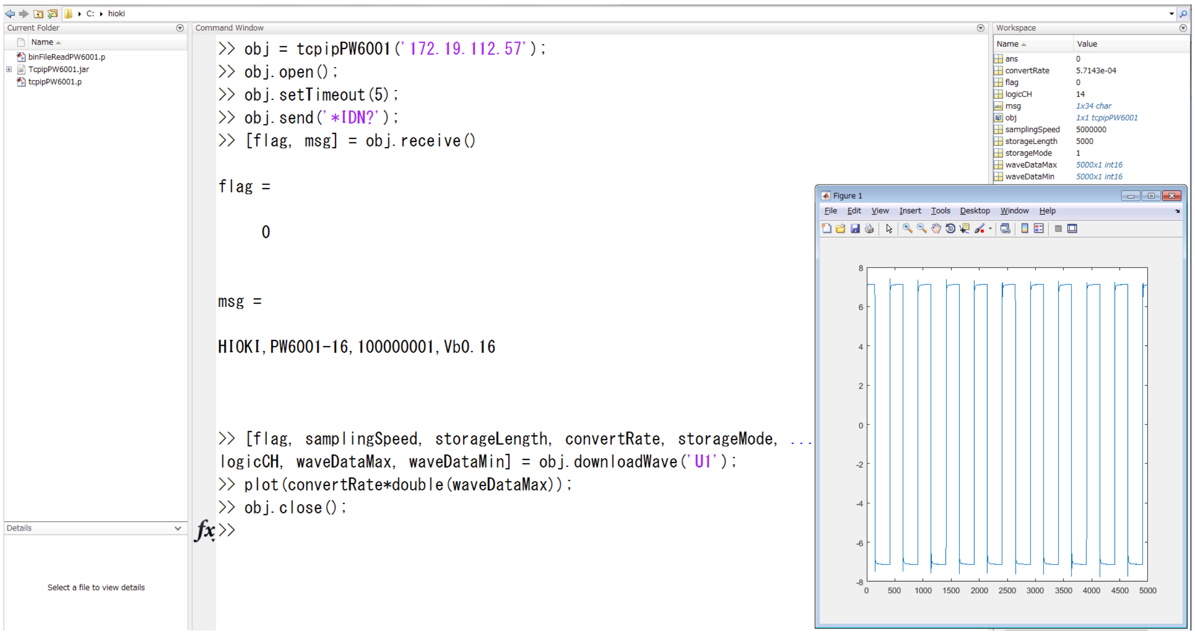
Fig. 1: Example of use for the toolkit (Loading binary waveform data as a MATLAB array).

Controlling a PW6001 from inside MATLAB over an Ethernet connection

By using the tcpipPW6001 class provided by this toolkit, you can send communications commands (for more information, see the Communications Command Manual) from inside MATLAB to a PW6001 that has been connected to the computer via Ethernet, receive responses from the instrument, and download waveform data recorded by the PW6001 as well as FFT analysis results as a MATLAB array.

Fig. 2 provides an example of this process. In this example, the user connects the PW6001 to the computer over Ethernet, sets the timeout time to 5 sec., sends the *IDN? command, and receives the instrument's response. Then he downloads the data arrays for the CH 1 voltage waveform recorded by the PW6001, displays one of them using the plot function, and disconnects the instrument's Ethernet connection.

```
>> obj = tcpipPW6001('172.19.112.57');% Create a TCP/IP object
>> obj.open();% Establish an Ethernet connection with the PW6001
>> obj.setTimeout(5);% Set 5 sec as the timeout time
>> obj.send('*IDN?');% Send the *IDN command
>> [flag, msg] = obj.receive()% Receive the command response
>> [flag, samplingSpeed, storageLength, convertRate, storageMode, ...
logicCH, waveDataMax, waveDataMin] = obj.downloadWave('U1');% Download the waveform data
>> plot(convertRate*double(waveDataMax));% Plot the data array of waveform data
>> obj.close();% Disconnect the Ethernet connection
```



```
>> obj = tcpipPW6001('172.19.112.57');
>> obj.open();
>> obj.setTimeout(5);
>> obj.send('*IDN?');
>> [flag, msg] = obj.receive()

flag =

     0


msg =

HIOKI,PW6001-16,100000001,Vb0.16



>> [flag, samplingSpeed, storageLength, convertRate, storageMode, ...
logicCH, waveDataMax, waveDataMin] = obj.downloadWave('U1');
>> plot(convertRate*double(waveDataMax));
>> obj.close();
fx >>
```

Name	Value
ans	0
convertRate	5.7143e-04
flag	0
logicCH	14
msg	*1x34 char*
obj	*1x1 tcpipPW6001*
samplingSpeed	5000000
storageLength	5000
storageMode	1
waveDataMax	*5000x1 int16*
waveDataMin	*5000x1 int16*

Fig. 2: Example of use for the toolkit (Controlling a PW6001 from inside MATLAB over an Ethernet connection).

Class Reference

binFileReadPW6001 class

This class loads PW6001 binary waveform data in MATLAB. By means of this class, you can easily access settings information and waveform data stored in the form of binary data.

Member functions

obj = binFileReadPW6001(filePath);

Function	Serves as the constructor for binFileReadPW6001 class objects.
Argument(s)	**filePath**: Path for the binary waveform data file you wish to load.
Return value(s)	**obj** : The generated binFileReadPW6001 object.
Description	The function generates a binFileReadPW6001 object and loads the specified binary data file (filePath).

Example of use

```
obj = binFileReadPW6001('B6001000.BIN');% Generate the binFileReadPW6001
                                        % object and load B6001000.BIN.
```

ret = load(filePath);

Function	Loads PW6001 binary waveform data.
Argument(s)	**filePath**: Path for the binary waveform data file you wish to load.
Return value(s)	**ret** : A value of 0 indicating that the binary data was loaded successfully or 1 indicating that the function failed to load the data.
Description	The function loads the specified binary data file (**filePath**).

Example of use

```
obj.load('B6001000.BIN');% Load B6001000.BIN using the binFileReadPW6001
                         % object obj.
```

ret = state();

Function	Returns the state of a binFileReadPW6001 object.
Argument(s)	None.
Return value(s)	**ret** : A value of 0 indicating that the binary data was loaded successfully or 1 indicating that the binFileReadPW6001 object failed to load the data.
Description	The function allows you to check whether a binFileReadPW6001 object has loaded a binary waveform data file successfully.

Example of use

```
obj.state();% Check the state of the binFileReadPW6001 object obj.
```

ret = filePath();

Function	Returns the path of a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	None.
Return value(s)	**ret** : Path of the binary waveform data file that has been loaded.
Description	The function allows you to check the path of a binary waveform data file that has been loaded.

Example of use

```
obj.filePath();% Check the path of the binary waveform data file that
               % has been loaded by the binFileReadPW6001 object obj.
```

ret = sizeStr();

Function	Returns a string indicating the file size of a binary waveform data file that has been loaded by a binFileReadPW6001 object, excluding this string data.
Argument(s)	None.
Return value(s)	**ret** : Size of the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.
Description	The function allows you to check the size of a binary waveform data file that has been loaded.

Example of use

```
obj.sizeStr();% Check the size of the binary waveform data file that has
              % been loaded by the binFileReadPW6001 object obj.
```

ret = model();

Function	Returns the model name string for the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	None.
Return value(s)	**ret** : Model name string for the PW6001 that generated the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.
Description	The function allows you to check the model of the PW6001 that generated a binary waveform data file that has been loaded.

Example of use

```
obj.model();% Check the model name of the PW6001 that generated the binary
            % waveform data file that has been loaded by the
            % binFileReadPW6001 object obj.
```

ret = version();

Function: Returns the firmware version string for the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.

Argument(s): None.

Return value(s): **ret** : Firmware version string for the PW6001 that generated the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.

Description: The function allows you to check the firmware version number of the PW6001 that generated a binary waveform data file that has been loaded.

Example of use

```
obj.version();% Check the firmware version number of the PW6001 that
              % generated the binary waveform data file that has been
              % loaded by the binFileReadPW6001 object obj.
```

ret = comment();

Function: Returns the comment string that was generated when a binary waveform data file that has been loaded by a binFileReadPW6001 object was saved.

Argument(s): None.

Return value(s): **ret** : Comment string that was generated when the binary waveform data file that has been loaded was saved. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.

Description: The function allows you to check the comment string that was generated when a binary waveform data file that has been loaded was saved.

Example of use

```
obj.comment();% Check the comment string that was generated when the binary
              % waveform data file that has been loaded by the
              % binFileReadPW6001 object obj was saved.
```

ret = saveCH();

Function: Returns channel information for waveform data saved in a binary waveform data file that has been loaded by a binFileReadPW6001 object.

Argument(s): None.

Return value(s): **ret** : Channel information for waveform data saved in the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.

Description: The function allows you to check the channels for waveform data saved in a binary waveform data file that has been loaded.

Example of use

```
obj.saveCH();% Check the channels for the waveform data saved in the binary
             % waveform data file that has been loaded by the
             % binFileReadPW6001 object obj.
```

ret = logicCH();

Function	Returns information about the channel set to motor option pulse input on the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	None.
Return value(s)	**ret** : Information about the channel set to motor option pulse input on the PW6001 that generated the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.
Description	The function lets you check the number of the channel set to motor option pulse input on the PW6001 that generated a binary waveform data file that has been loaded.

Example of use

```
obj.logicCH();% Check the number of the channel set to motor option pulse input
              % on the PW6001 that generated the binary waveform data file that
              % has been loaded by the binFileReadPW6001 object obj.
```

ret = abType();

Function	Returns information about the measurement parameters set for motor options CH A and B on the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	None.
Return value(s)	**ret** : Information about the type of measurement parameters set for motor options CH A and B on the PW6001 that generated the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.
Description	The function lets you check the type of measurement parameters set for motor options CH A and B on the PW6001 that generated a binary waveform data file that has been loaded.

Example of use

```
obj.abType();% Check the type of measurement parameters set for the motor
             % options CH A and B on the PW6001 that generated the binary
             % waveform data file loaded by the binFileReadPW6001 object obj.
```

ret = wiring(ch);

Function	Returns the wiring setting string for the channel specified by the argument on the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	**ch** : Channel number (1, 2, · · · , 6).
Return value(s)	**ret** : Wiring setting string for the channel specified by the argument (**ch**) on the PW6001 that generated the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.
Description	The function lets you check the wiring settings on the PW6001 that generated a binary waveform data file that has been loaded.

Example of use

```
obj.wiring();% Check the wiring settings on the PW6001 that generated the
             % binary waveform data file that has been loaded by the
             % binFileReadPW6001 object obj.
```

ret = uRange(ch);

Function	Returns the voltage range for the channel specified by the argument on the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	**ch** : Channel number (1, 2, $\cdots$, 6).
Return value(s)	**ret** : Voltage range for the channel specified by the argument (**ch**) on the PW6001 that generated the binary waveform data file that has been loaded.
Description	The function lets you check the voltage range settings on the PW6001 that generated a binary waveform data file that has been loaded.

Example of use

```
obj.uRange(1);% Check the voltage range setting for CH 1 on the PW6001 that
              % generated the binary waveform data file that has been loaded by
              % the binFileReadPW6001 object obj.
```

ret = iRange(ch);

Function	Returns the current range for the channel specified by the argument on the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	**ch** : Channel number (1, 2, $\cdots$, 6).
Return value(s)	**ret** : Current range for the channel specified by the argument (**ch**) on the PW6001 that generated the binary waveform data file that has been loaded.
Description	The function lets you check the current range settings on the PW6001 that generated a binary waveform data file that has been loaded.

Example of use

```
obj.iRange(1);% Check the current range setting for CH 1 on the PW6001 that
              % generated the binary waveform data file that has been loaded by
              % the binFileReadPW6001 object obj.
```

ret = analogRange(ch);

Function	Returns the voltage range for the channel specified by the argument set to motor option analog DC input on the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	**ch** : Channel number (1: CH A; 2: CH B).
Return value(s)	**ret** : Voltage range for the channel specified by the argument (**ch**) set to motor option analog DC input on the PW6001 that generated the binary waveform data file that has been loaded.
Description	The function lets you check the voltage range setting for motor option analog DC input on the PW6001 that generated a binary waveform data file that has been loaded.

Example of use

```
obj.analogRange(1);% Check the voltage range setting for CH A, which is set to
                   % motor option analog DC input on the PW6001 that generated
                   % the binary waveform data file loaded by the
                   % binFileReadPW6001 object obj.
```

ret = vt(ch);

Function	Returns the VT ratio for the channel specified by the argument on the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	**ch** : Channel number (1, 2, ⋯, 6).
Return value(s)	**ret** : VT ratio for the channel specified by the argument (**ch**) on the PW6001 that generated the binary waveform data file that has been loaded.
Description	The function lets you check the VT ratio setting on the PW6001 that generated a binary waveform data file that has been loaded.

Example of use

```
obj.vt(1);% Check the VT ratio setting for CH 1 on the PW6001 that generated
          % the binary waveform data file that has been loaded by the
          % binFileReadPW6001 object obj.
```

ret = ct(ch);

Function	Returns the CT ratio for the channel specified by the argument on the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	**ch** : Channel number (1, 2, ⋯, 6).
Return value(s)	**ret** : CT ratio for the channel specified by the argument (**ch**) on the PW6001 that generated the binary waveform data file that has been loaded.
Description	The function lets you check the CT ratio setting on the PW6001 that generated a binary waveform data file that has been loaded.

Example of use

```
obj.ct(1);% Check the CT ratio setting for CH 1 on the PW6001 that generated
          % the binary waveform data file that has been loaded by the
          % binFileReadPW6001 object obj.
```

ret = tqScale(ch);

Function	Returns the torque scaling value for the channel specified by the argument on the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	**ch** : Channel number (1: CH A; 2: CH B).
Return value(s)	**ret** : Torque scaling value for the channel specified by the argument (**ch**) on the PW6001 that generated the binary waveform data file that has been loaded.
Description	The function lets you check the motor option torque scaling value setting on the PW6001 that generated a binary waveform data file that has been loaded.

Example of use

```
obj.tqScale(1);% Check the torque scaling value for motor option CH A on the
               % PW6001 that generated the binary waveform data file that has
               % been loaded by the binFileReadPW6001 object obj.
```

ret = speedScale();

Function	Returns the motor option speed scaling value (for CH B) on the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	None.
Return value(s)	**ret** : Motor option speed scaling value on the PW6001 that generated the binary waveform data file that has been loaded by the binFileReadPW6001 object.
Description	The function lets you check the motor option speed scaling value on the PW6001 that generated a binary waveform data file that has been loaded.

Example of use

```
obj.speedScale();% Check the motor option speed scaling value setting on the
                 % PW6001 that generated the binary waveform data file that has
                 % been loaded by the binFileReadPW6001 object obj.
```

ret = deltaConv();

Function	Returns the delta conversion setting on the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	None.
Return value(s)	**ret** : Delta conversion setting information for the PW6001 that generated the binary waveform data file that has been loaded by the binFileReadPW6001 object. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.
Description	The function lets you check the delta conversion setting on the PW6001 that generated a binary waveform data file that has been loaded.

Example of use

```
obj.deltaConv();% Check the delta conversion setting on the PW6001 that
                % generated the binary waveform data file that has been loaded
                % by the binFileReadPW6001 object obj.
```

ret = lpf(ch);

Function	Returns the setting value indicating the cutoff frequency of the low-pass filter for the channel specified by the argument on the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	**ch** : Channel number (1, 2, $\cdots$, 6).
Return value(s)	**ret** : Setting value indicating the cutoff frequency of the low-pass filter for the channel specified by the argument (**ch**) on the PW6001 that generated the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.
Description	The function lets you check the setting value indicating the cutoff frequency of the low-pass filter on the PW6001 that generated a binary waveform data file that has been loaded.

Example of use

```
obj.lpf(1);% Check the setting value indicating the cutoff frequency of the
           % low-pass filter for CH 1 on the PW6001 that generated the binary
           % waveform data file that has been loaded by the binFileReadPW6001
           % object obj.
```

ret = anaLpf();

Function	Returns low-pass filter setting information for motor option analog DC input on the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	None.
Return value(s)	**ret** : Low-pass filter setting information for motor option analog DC input on the PW6001 that generated the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.
Description	The function lets you check the low-pass filter setting for motor option analog DC input on the PW6001 that generated a binary waveform data file that has been loaded.

Example of use

```
obj.anaLpf();% check the low-pass filter setting for motor option analog DC
             % input on the PW6001 that generated the binary waveform data
             % file that has been loaded by the binFileReadPW6001 object obj.
```

ret = logicLpf(ch);

Function	Returns the low-pass filter setting string for the channel specified by the argument for motor option pulse input on the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	**ch** : Channel number (1 : CH A, 2 : CH B, 3 : CH C, 4 : CH D).
Return value(s)	**ret** : low-pass filter setting string (“OFF“, “WEAK”, or “STRONG”) for motor option pulse input for the channel specified with the argument (**ch**) on the PW6001 that generated the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.
Description	The function lets you check the low-pass filter setting for motor option pulse input on the PW6001 that generated a binary waveform data file that has been loaded.

Example of use

```
obj.logicLpf(1);% Check the low-pass filter setting for motor option CH A pulse
                % input on the PW6001 that generated the binary waveform data
                % file that has been loaded by the binFileReadPW6001 object
                % obj.
```

ret = spc();

Function	Returns sensor phase shift setting information for the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	None.
Return value(s)	**ret** : Sensor phase shift setting information for the PW6001 that generated the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.
Description	The function lets you check the sensor phase shift setting on the PW6001 that generated a binary waveform data file that has been loaded.

Example of use

```
obj.spc();% Check the sensor phase shift setting on the PW6001 that generated
          % the binary waveform data file that has been loaded by the
          % binFileReadPW6001 object obj.
```

ret = spcHz(ch);

Function: Returns the frequency setting value used in sensor phase shift for the channel specified by the argument on the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.

Argument(s): **ch** : Channel number (1, 2, · · · , 6).

Return value(s): **ret** : Frequency setting value used in sensor phase shift for the channel specified with the argument (**ch**) on the PW6001 that generated the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.

Description: The function lets you check the frequency setting value used in sensor phase shift on the PW6001 that generated a binary waveform data file that has been loaded.

Example of use

```
obj.spcHz();% Check the frequency setting value used in sensor phase shift on
            % the PW6001 that generated the binary waveform data file that has
            % been loaded by the binFileReadPW6001 object obj.
```

ret = spcDeg(ch);

Function: Returns the angle setting value used in sensor phase shift for the channel specified by the argument on the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.

Argument(s): **ch** : Channel number (1, 2, · · · , 6).

Return value(s): **ret** : Angle setting value used in sensor phase shift for the channel specified with the argument (**ch**) on the PW6001 that generated the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.

Description: The function lets you check the angle setting value used in sensor phase shift on the PW6001 that generated a binary waveform data file that has been loaded.

Example of use

```
obj.spcDeg();% Check the angle setting value used in sensor phase shift on the
             % PW6001 that generated the binary waveform data file that has
             % been loaded by the binFileReadPW6001 object obj.
```

ret = storageMode();

Function: Returns the storage mode (P–P/Deci) of a binary waveform data file that has been loaded by a binFileReadPW6001 object.

Argument(s): None.

Return value(s): **ret** : Storage mode setting value (0: P–P; 1: Deci) of the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.

Description: The function lets you check the storage mode setting of a binary waveform data file that has been loaded.

Example of use

```
obj.storageMode();% Check the storage mode setting of the binary waveform data
                  % file that has been loaded by the binFileReadPW6001 object
                  % obj.
```

ret = smplSpd();

Function	Returns the sampling speed of a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	None.
Return value(s)	**ret** : Sampling speed of the binary waveform data file that has been loaded.
Description	The function lets you check the sampling speed of a binary waveform data file that has been loaded.

Example of use

```
obj.smplSpd();% Check the sampling speed of the binary waveform data file that
              % has been loaded by the binFileReadPW6001 object obj.
```

ret = smplSpdAnalog();

Function	Returns the sampling speed for motor option analog DC input in a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	None.
Return value(s)	**ret** : Sampling speed for motor option analog DC input in the binary waveform data file that has been loaded.
Description	The function lets you check the sampling speed for motor option analog DC input in a binary waveform data file that has been loaded.

Example of use

```
obj.smplSpdAnalog();% Check the sampling speed for motor option analog DC input
                    % in the binary waveform data file that has been loaded by
                    % the binFileReadPW6001 object obj.
```

ret = strgLen();

Function	Returns the number of data points in a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	None.
Return value(s)	**ret** : Number of data points in the databinary waveform data file that has been loaded by a binFileReadPW6001 object.
Description	The function lets you check the number of data points in a binary waveform data file that has been loaded.

Example of use

```
obj.strgLen();% Check the number of data points in the binary waveform data
              % file that has been loaded by the binFileReadPW6001 object obj.
```

ret = strgLenAnalog();

Function	Returns the number of data points for motor option analog DC input in a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	None.
Return value(s)	**ret** : Number of data points for motor option DC input in the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.
Description	The function lets you check the number of data points for motor option DC input in a binary waveform data file that has been loaded.

Example of use

```
obj.strgLenAnalog();% Check the number of data points for the motor option DC
                    % input in the binary waveform data file that has been
                    % loaded by the binFileReadPW6001 object obj.
```

ret = convRateU(ch);

Function	Returns the conversion coefficient for the voltage waveform of the channel specified by the argument in a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	**ch** : Channel number (1, 2, · · · , 6).
Return value(s)	**ret** : Conversion coefficient for the voltage waveform of the channel specified by the argument (**ch**) in the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.
Description	The function lets you acquire the conversion coefficient for the voltage waveform in a binary waveform data file that has been loaded.

Example of use

```
obj.convRateU(1);% Acquire the conversion coefficient for the voltage waveform
                 % of PW6001 CH 1 in the binary waveform data file that has
                 % been loaded by the binFileReadPW6001 object obj.
```

ret = convRateI(ch);

Function	Returns the conversion coefficient for the current waveform of the channel specified by the argument in a binary waveform data file that has been loaded by a binFileReadPW6001 object.
Argument(s)	**ch** : Channel number (1, 2, · · · , 6).
Return value(s)	**ret** : Conversion coefficient for the current waveform of the channel specified by the argument (**ch**) in the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.
Description	The function lets you acquire the conversion coefficient for the current waveform in a binary waveform data file that has been loaded.

Example of use

```
obj.convRateI(1);% Acquire the conversion coefficient for the current waveform
                 % of PW6001 CH 1 in the binary waveform data file that has
                 % been loaded by the binFileReadPW6001 object obj.
```

ret = convRateAnalog(ch);

Function Returns the conversion coefficient for the motor option analog DC waveform of the channel specified by the argument in a binary waveform data file that has been loaded by a binFileReadPW6001 object.

Argument(s) **ch** : Channel number (1: CH A; 2: CH B).

Return value(s) **ret** : Conversion coefficient for the motor option analog DC waveform of the channel specified by the argument (**ch**) on the PW6001 that generated the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.

Description The function lets you acquire the conversion coefficient for the motor option analog DC waveform in a binary waveform data file that has been loaded.

Example of use

```
obj.convRateAnalog(1);% Acquire the conversion coefficient for the motor option
                      % analog DC input CH A waveform in the binary waveform
                      % data file that has been loaded by the binFileReadPW6001
                      % object obj.
```

ret = wUMax(ch);

Function Returns a data array of voltage waveform count values for the channel specified by the argument on the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.

Argument(s) **ch** : Channel number (1, 2, · · · , 6).

Return value(s) **ret** : Data array of voltage waveform count values for the channel specified by the argument (**ch**) in the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.

Description The function lets you acquire a data array of voltage waveform count values in a binary waveform data file that has been loaded.

Example of use

```
obj.wUMax(1);% Acquire a data array of voltage waveform count values for CH 1
             % in the binary waveform data file that has been loaded by the
             % binFileReadPW6001 object obj.
```

ret = wUMin(ch);

Function Returns a data array of voltage waveform count values for the channel specified by the argument on the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.

Argument(s) **ch** : Channel number (1, 2, · · · , 6).

Return value(s) **ret** : Data array of voltage waveform count values for the channel specified by the argument (**ch**) in the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.

Description The function lets you acquire a data array of voltage waveform count values in a binary waveform data file that has been loaded.

Example of use

```
obj.wUMax(1);% Acquire a data array of voltage waveform count values for CH 1
             % in the binary waveform data file that has been loaded by the
             % binFileReadPW6001 object obj.
```

ret = wIMax(ch);

Function Returns a data array of current waveform count values for the channel specified by the argument on the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.

Argument(s) **ch** : Channel number (1, 2, ···, 6).

Return value(s) **ret** : Data array of current waveform count values for the channel specified by the argument (**ch**) in the binary waveform data file that has been loaded. For more information, see the "Waveform Binary Data Format" section in the Power Analyzer PW6001 Instruction Manual.

Description The function lets you acquire a data array of current waveform count values in a binary waveform data file that has been loaded.

Example of use

```
obj.wIMax(1);% Acquire a data array of current waveform count values for CH 1
             % in the binary waveform data file that has been loaded by the
             % binFileReadPW6001 object obj.
```

ret = wIMin(ch);

Function Returns a data array of current waveform count values for the channel specified by the argument on the PW6001 that generated a binary waveform data file that has been loaded by a binFileReadPW6001 object.

Argument(s) **ch** : Channel number (1, 2, ···, 6).

Return value(s) **ret** : Data array of current waveform count values for the channel specified by the argument (**ch**) in the binary waveform data file that has been loaded. For more information, see the "Waveform Binary Data Format" section in the Power Analyzer PW6001 Instruction Manual.

Description The function lets you acquire a data array of current waveform count values in a binary waveform data file that has been loaded.

Example of use

```
obj.wIMin(1);% Acquire a data array of current waveform count values for CH 1
             % in the binary waveform data file that has been loaded by the
             % binFileReadPW6001 object obj.
```

ret = wLMax();

Function Returns a data array for the motor option pulse input waveform in a binary waveform data file that has been loaded by a binFileReadPW6001 object.

Argument(s) None.

Return value(s) **ret** : Data array for the motor option pulse input waveform in the binary waveform data file that has been loaded. For more information, see the "Waveform Binary Data Format" section in the Power Analyzer PW6001 Instruction Manual.

Description The function lets you acquire a data array for a motor option pulse input waveform in a binary waveform data file that has been loaded.

Example of use

```
obj.wLMax();% Acquire the data array for the motor option pulse input waveform
            % in the binary waveform data file that has been loaded by the
            % binFileReadPW6001 object obj.
```

ret = wLMin();

Function: Returns a data array for the motor option pulse input waveform in a binary waveform data file that has been loaded by a binFileReadPW6001 object.

Argument(s): None.

Return value(s): **ret** : Data array for the motor option pulse input waveform in the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.

Description: The function lets you acquire a data array for a motor option pulse input waveform in a binary waveform data file that has been loaded.

Example of use

```
obj.wLMin();% Acquire the data array for the motor option pulse input waveform
            % in the binary waveform data file that has been loaded by the
            % binFileReadPW6001 object obj.
```

ret = wAnalogMax(ch);

Function: Returns a data array of count values for a motor option analog DC input waveform of the channel specified by the argument in a binary waveform data file that has been loaded by a binFileReadPW6001 object.

Argument(s): **ch** : Channel number (1: CH A; 2: CH B).

Return value(s): **ret** : Data array of count values for the motor option analog DC input waveform of the channel specified by the argument in the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.

Description: The function lets you acquire a data array of count values for the motor option analog DC input waveform in a binary waveform data file that has been loaded.

Example of use

```
obj.wAnalogMax(1);% Acquire the data array of count values for the motor option
                  % analog DC input CH A waveform in the binary waveform data
                  % file that has been loaded by the binFileReadPW6001 object
                  % obj.
```

ret = wAnalogMin(ch);

Function: Returns a data array of count values for a motor option analog DC input waveform of the channel specified by the argument in a binary waveform data file that has been loaded by a binFileReadPW6001 object.

Argument(s): **ch** : Channel number (1: CH A; 2: CH B).

Return value(s): **ret** : Data array of count values for the motor option analog DC input waveform of the channel specified by the argument in the binary waveform data file that has been loaded. For more information, see the “Waveform Binary Data Format” section in the Power Analyzer PW6001 Instruction Manual.

Description: The function lets you acquire a data array of count values for the motor option analog DC input waveform in a binary waveform data file that has been loaded.

Example of use

```
obj.wAnalogMin(1);% Acquire the data array of count values for the motor option
                  % analog DC input CH A waveform in the binary waveform data
                  % file that has been loaded by the binFileReadPW6001 object
                  % obj.
```

tcpipPW6001 class

This class is used to control a PW6001 from inside MATLAB over an Ethernet connection. By means of this class, you can send communications commands to a PW6001 and acquire waveform data and FFT spectrum data from the instrument.

Member functions

obj = tcpipPW6001(ipAddr);

Function	Serves as the constructor for tcpipPW6001 class objects.
Argument(s)	**ipAddr** : String consisting of the IP address of the PW6001 to which you wish to connect.
Return value(s)	**obj** : The generated tcpipPW6001 object.
Description	The function generates a tcpipPW6001 object for the purpose of connecting to the PW6001 with the IP address specified as the argument (**ipAddr**).

Example of use

```
obj = tcpip('172.19.112.57');% Generate the tcpipPW6001 object obj in order to
                             % connect to the PW6001 with the IP address 172.
                             % 19.112.57.
```

ret = ipAddr();

Function	Returns a string consisting of the IP address of the PW6001 designated as the destination instrument of a tcpipPW6001 object.
Argument(s)	None.
Return value(s)	**ret** : String consisting of the IP address of the PW6001 designated as the destination instrument of the tcpipPW6001 object.
Description	The function lets you check the IP address of the PW6001 designated as the destination instrument of a tcpipPW6001 object.

Example of use

```
obj.ipAddr();% Check the IP address of the PW6001 designated as the destination
             % instrument of the tcpipPW6001 object obj.
```

ret = timeout();

Function	Returns the timeout time for communications between a tcpipPW6001 object and a PW6001.
Argument(s)	None.
Return value(s)	**ret** : Timeout time (in seconds) for communications between the tcpipPW6001 object and the PW6001.
Description	The function lets you check the timeout time for communications between a tcpipPW6001 object and a PW6001.

Example of use

```
obj.timeout();% Check the timeout time for communications between the
              % tcpipPW6001 object obj and the designated PW6001.
```

ret = open();

Function	Establishes a connection between a tcpipPW6001 object and a PW6001.
Argument(s)	None.
Return value(s)	**ret**: A value of 0 indicating success or 1 indicating an error.
Description	The function establishes a connection between a tcpipPW6001 object and a PW6001.
Example of use	

```
obj.open();% Establish a connection between the tcpipPW6001 object obj and the
           % designated PW6001.
```

ret = close();

Function	Closes a connection between a tcpipPW6001 object and a PW6001.
Argument(s)	None.
Return value(s)	**ret** : A value of 0 indicating success or 1 indicating an error.
Description	The function closes a connection between a tcpipPW6001 object and a PW6001.
Example of use	

```
obj.close();% Close the connection between the tcpipPW6001 object obj and the
            % designated PW6001.
```

ret = setTimeout(time);

Function	Sets the timeout time for communications between a tcpipPW6001 object and a PW6001.
Argument(s)	**time** : The timeout time (in seconds) you wish to set.
Return value(s)	**ret** : A value of 0 indicating success or 1 indicating an error.
Description	The function sets the timeout time for communications between a tcpipPW6001 object and a PW6001 to the time specified by the argument (**time**).
Example of use	

```
obj.setTimeout(5);% Set the timeout time for communications between the
                  % tcpipPW6001 object obj and the designated PW6001 to 5
                  % seconds.
```

ret = send(command);

Function	Sends a communications command (as text) to the PW6001 designated as the destination instrument of a tcpipPW6001 object.
Argument(s)	**command**: The communications command you wish to send.
Return value(s)	**ret** : A value of 0 indicating success or 1 indicating an error.
Description	The function sends the communications command specified by the argument (**command**) to the PW6001 designated as the destination instrument of a tcpipPW6001 object. For more information about communications commands, see the Communications Command Manual.
Example of use	

```
obj.send('*IDN?');% Send the communications command *IDN to the PW6001
                  % designated as the destination instrument for the
                  % tcpipPW6001 object obj.
```

[ret, msg] = receive();

Function	Receives the response (as text) to a communications command that was sent to the PW6001 designated as the destination instrument of a tcpipPW6001 object.
Argument(s)	None.
Return value(s)	**ret** : A value of 0 indicating success or 1 indicating an error. **msg** : String consisting of the response to the communications command that was sent.
Description	The function receives the response to a communications command that was sent to the PW6001 designated as the destination instrument of a tcpipPW6001 object.

Example of use

```
obj.receive();% Receive the response to the communications command sent to the
              % PW6001 designated as the destination instrument of the
              % tcpipPW6001 object obj.
```

**[flag, samplingSpeed, storageLength, convertRate, storageMode, ...
logicCH, waveDataMax, waveDataMin] = downloadWave(chName);**

Function	Acquires waveform data from the PW6001 designated as the destination instrument of a tcpipPW6001 object.	
Argument(s)	**chName**	: See table below.
Return value(s)	**flag**	: A value of 0 indicating success or 1 indicating an error.
	samplingSpeed	: The sampling speed of the waveform data.
	storageLength	: The number of points of waveform data.
	convertRate	: The conversion coefficient of the waveform data.
	storageMode	: The storage mode of the waveform data.
	logicCH	: The channel set to motor option pulse input.
	waveDataMax	: The waveform's data array.
	waveDataMin	: The waveform's data array.
Description	The function acquires waveform data for the channel specified by the argument (**chName**) for the PW6001 designated as the destination instrument of a tcpipPW6001 object. The format of the waveform data (waveDataMax/waveDataMin), storageMode, and logicCH is is the same as that used for binary waveform data. For more information, see the "Waveform Binary Data Format" section in the Power Analyzer PW6001 Instruction Manual.	

Example of use

```
 [flag, samplingSpeed, storageLength, convertRate, storageMode, logicCH, ...
   waveDataMax, waveDataMin] = obj.downloadWave('U1');
% Acquire voltage waveform data for CH1 for the PW6001 designated as the
% destination instrument of the tcpipPW6001 object obj.
```

chName	
'Uj'	Voltage waveform data for CH j ($j = 1, 2, \cdots, 6$).
'Ij'	Current waveform data for CH j ($j = 1, 2, \cdots, 6$).
'A'	Analog DC input waveform data for motor option CH A.
'B'	Analog DC input waveform data for motor option CH B.
'LOGIC'	Motor option pulse (frequency) input waveform data.

[flag, samplingSpeed, fftLength, convertRate, fftCH, fftValid, fftData] = downloadFFT(ch);

Function	Acquires FFT spectrum data for the PW6001 designated as the destination of a tcpipPW6001 object.	
Argument(s)	**ch**	: The channel number targeted for FFT. (1 : Voltage or CH A; 2 : Current or CH B)
Return value(s)	**flag**	: A value of 0 indicating success or 1 indicating an error.
	samplingSpeed	: The sampling speed of the waveform data targeted for FFT.
	fftLength	: The length of the FFT.
	convertRate	: The conversion coefficient of the FFT spectrum data.
	fftValid	: A value of 0 or 1. The FFT spectrum data is only valid when this spectrum data is only valid when this return value is 1.
	fftData	: Data array of FFT spectrum (FFT spectrum of waveform count value data).
Description	The function acquires the FFT spectrum data of the channel specified by the argument (**ch**) for the PW6001 designated as the destination of a tcpipPW6001 object.	

Example of use

```
[flag, samplingSpeed, fftLength, convertRate, fftCH, fftValid, fftData] ...
 = obj.downloadFFT(1);
% Acquire the FFT analysis results data for the PW6001 designated as the
% destination of the tcpipPW6001 object obj.
plot(convertRate*single(fftData));% Plot the FFT spectrum data.
```

Document Revision History

Date	Version	Changes
Oct. 2016	1.0	First edition