

PW8001

HIOKI

PW8001-01 PW8001-11
PW8001-02 PW8001-12
PW8001-03 PW8001-13
PW8001-04 PW8001-14
PW8001-05 PW8001-15
PW8001-06 PW8001-16

MATLAB ツールキット
ユーザズマニュアル

パワーアナライザ



JA

Dec. 2022 Edition 1
PW8001A969-00 22-12H

- ✓ 本取扱説明書は、コマンドに関する部分のみを扱っています。
- ✓ PW8001 の使用前に、必ず PW8001 取扱説明書をお読みください。
- ✓ PW8001 の通信設定に関しては、PW8001 取扱説明書「9 PC との接続」をご覧ください。
- ✓ 本取扱説明書の内容につきましては万全を期していますが、ご不明な点や誤りなどお気づきになった点がございましたら、本社コールセンターまたは最寄りの営業拠点までご連絡ください。

目次

1 概要	1
2 使用準備	1
3 Ethernet 接続により MATLAB 上で PW8001 を制御する	2
4 tcpipPW8001 クラス	4
メンバ関数	4

1 概要

本ソフトウェアは、MATLAB で Ethernet 接続した弊社パワーアナライザ PW8001 を制御するための MATLAB スクリプト(tcpipPW8001.p)により構成されます。

なお、本ツールキットは MATLAB R2022a 以降で使用することを推奨します。また、Ethernet 以外の通信インターフェース(GPIB, RS-232C) を使用し、MATLAB で PW8001 を制御する場合には、本ツールキットは使用できません。

2 使用準備

まず、MatlabToolkitForPW8001.zip に本マニュアルファイル(パワーアナライザ PW8001Matlab ツールキットユーザーズマニュアル.pdf) とスクリプトファイル(tcpipPW8001.p)があることを確認し、適当な場所に配置してください。配置した場所を、MATLAB の検索パスに追加してください。

※MATLAB は The MathWorks, Inc. の登録商標です。

3 Ethernet 接続により MATLAB 上で PW8001 を制御する

本ツールキットが提供する tcpipPW8001 クラスを使用することにより、Ethernet 接続した PW8001 に MATLAB から通信コマンド(詳細は次ページ以降を参照)を送信してその応答を受信します。

実行例を図 1、図 2 に示しました。図 1 ではアナログ波形データを取得しています。以下に、その手順を記します。

1. MATLAB で PW8001 の IP アドレスと通信タイムアウト時間を設定し、オブジェクトを生成します。
2. 生成されたオブジェクトと PW8001 の Ethernet 接続を確立します。
3. MATLAB から PW8001 へ通信コマンドを送信して、PW8001 から通信コマンドに対する応答を受信しています。
4. PW8001 から取得したいアナログ波形を 1 つ選び、そのデータの最大値と最小値を取り込みます。
5. 取得したデータを plot 関数で表示します。
6. Ethernet 接続を切断します。

MATLAB のコマンドウィンドウで 1.から 6.の手順を実行すると、以下の処理になります。

```
>> [obj, flag] = tcpipPW8001("192.168.10.11", 15); %PW8001 の IP アドレス、タイムアウト時間を 15 秒に設定して、  
                                                %Ethernet 接続するための TCP/IP オブジェクトを生成します  
  
>> obj.open; %PW8001 に Ethernet 接続をします  
>> obj.send("*IDN?"); %*IDN?コマンドを送信します  
>> obj.receive; %コマンドの応答を受信します  
>> [samplingSpeed, storageLength, storageMode, waveDataMax,...  
    waveDataMin, flag] = obj.DownloadAnalogWaveData("U1"); %U1 の電圧波形データを取り込みます  
>> plot(waveDataMax); %U1 の電圧波形の最大値のデータ列をプロットします  
>> obj.close; %Ethernet 接続を切断します
```

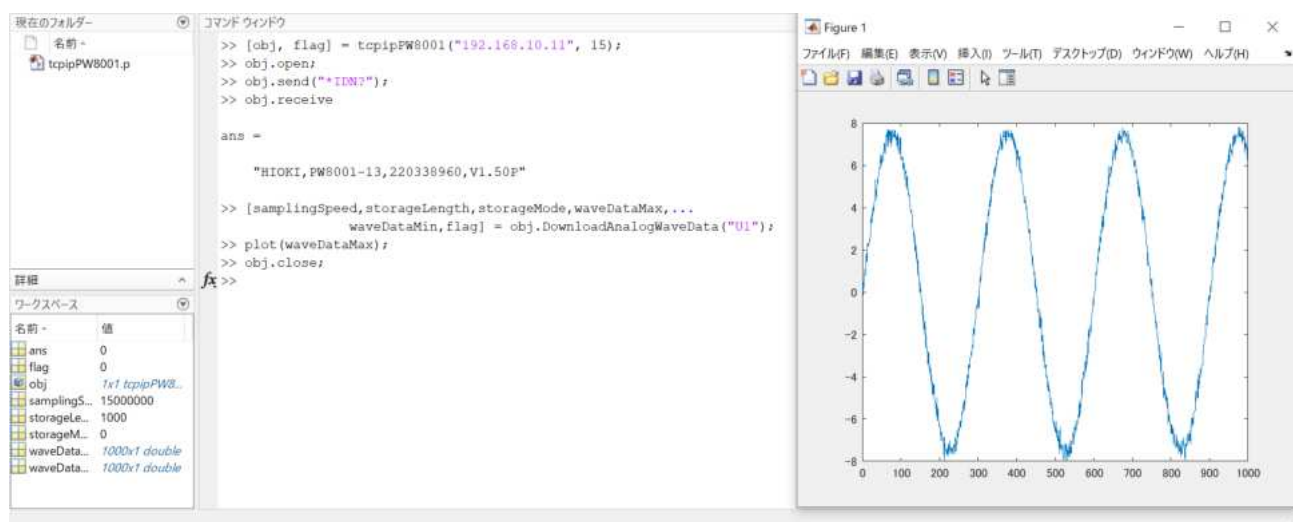


図 1: アナログ波形データを取得する実行例 (Ethernet 接続により MATLAB で PW8001 を制御する)

次に図2ではパルス波形データを取得しています。パルス波形の取得はPW8001に、モーターオプションが装着されているときのみ有効です。以下に、その手順を記します。

1. MATLAB で PW8001 の IP アドレスと通信タイムアウト時間を設定し、オブジェクトを生成します。
2. 生成されたオブジェクトと PW8001 の Ethernet 接続を確立します。
3. MATLAB から PW8001 へ通信コマンドを送信して、PW8001 から通信コマンドに対する応答を受信しています。
4. PW8001 からパルス入力に設定されているモーターチャンネルの情報と、それら全チャンネルのデータの最大値と最小値を取り込みます。
5. パルス入力に設定されているモーターチャンネルのデータから1つチャンネルを指定して、パルス波形データの最大値もしくは最小値を選択し、取り込みます。
6. 取得したデータを plot 関数で表示します。
7. Ethernet の接続を切断します。

MATLAB のコマンドウィンドウで 1.から 7.の手順を実行すると、以下の処理になります。

```
>> [obj, flag] = tcpipPW8001("192.168.10.11", 15); %PW8001 の IP アドレス、タイムアウト時間を 15 秒に設定して、  
                                         %Ethernet 接続するための TCP/IP オブジェクトを生成します  
  
>> obj.open; %PW8001 に Ethernet 接続をします  
>> obj.send("*IDN?"); %*IDN?コマンドを送信します  
>> obj.receive; %コマンドの応答を受信します  
>> [samplingSpeed, storageLength, storageMode, logicCH,...  
    waveDataMax, waveDataMin, flag] = obj.DownloadLogicWaveData;  
%パルス入力に設定されている全波形分を含んだデータを取得します  
>> [chWaveData, flag] = ...  
    obj.ExtractLogicChWaveData("CHA", waveDataMax); %CHA のパルス波形データの最大値を取り込みます  
>> plot(chWaveData); %CHA のパルス波形データの最大値のデータ列をプロットします  
>> obj.close; %Ethernet 接続を切断します
```

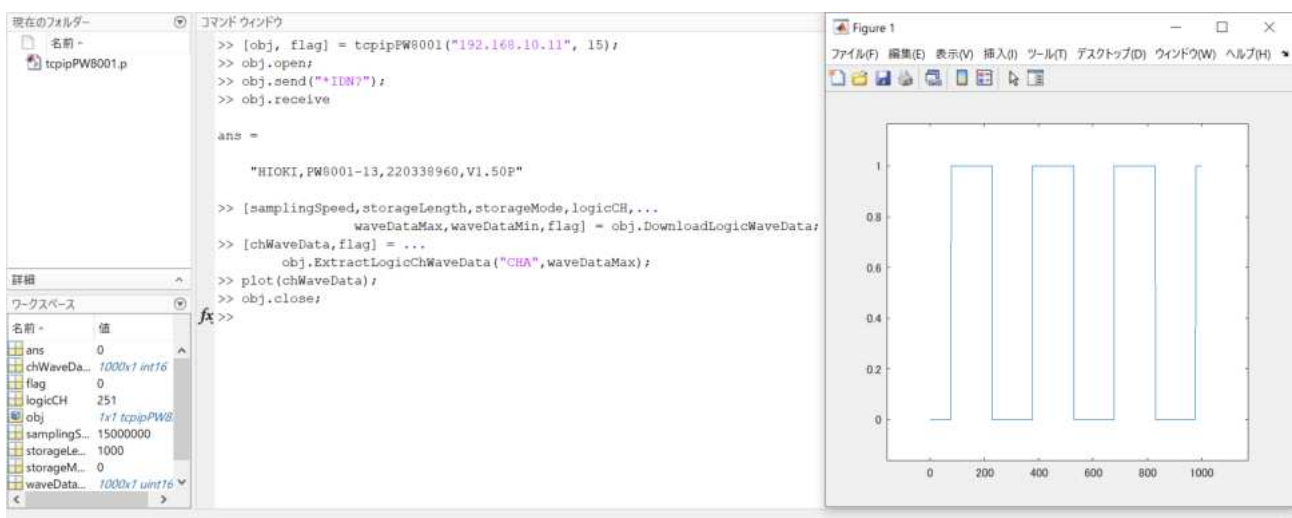


図 2: パルス波形データを取得する実行例 (Ethernet 接続により MATLAB で PW8001 を制御する)

4 tcpipPW8001 クラス

Ethernet 接続により MATLAB 上で PW8001 を制御するためのクラスです。本クラスを介し、PW8001 と通信コマンドの送受信や波形データの取得ができます。

メンバ関数

```
[obj, flag] = tcpipPW8001(ipAddr, timeOut);
```

引数	ipAddr: 接続先のPW8001のIPアドレスの文字列 timeOut: タイムアウト時間(秒)の数値
戻り値	obj: 生成されたtcpipPW8001のオブジェクト flag: オブジェクトを生成できた場合は「0」、異常終了した場合には「1」もしくは事前定義されたMATLABのエラーメッセージを表示
機能	tcpipPW8001 クラスオブジェクトのコンストラクタです。PW8001 に接続するための tcpipPW8001 オブジェクトを生成し、フラグを返します。
使用例	[obj, flag] = tcpipPW8001("192.168.10.11", 15); %PW8001のIPアドレスを"192.168.10.11"、タイムアウト時間を15秒に設定して、 %Ethernet接続するためのTCP/IPオブジェクトを生成します

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

引数	なし
戻り値	ipAddress: 接続先のPW8001のIPアドレスの文字列 flag: IPアドレスを取得できた場合は「0」、異常終了した場合には「1」もしくは事前定義されたMATLABのエラーメッセージを表示
機能	接続先の PW8001 の IP アドレスの文字列とフラグを返します。
使用例	obj.ipAddr;

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

引数	なし
戻り値	aTime: PW8001との通信のタイムアウト時間(秒) flag: タイムアウト時間を取得できた場合は「0」、異常終了した場合には「1」もしくは事前定義されたMATLABのエラーメッセージを表示
機能	PW8001 との通信のタイムアウト時間(秒)とフラグを返します。
使用例	obj.timeout;

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

引数	aTimeout: PW8001との通信のタイムアウト時間を設定する変数
戻り値	flag: タイムアウト時間を設定できた場合は「0」、異常終了した場合には「1」もしくは事前定義されたMATLABのエラーメッセージを表示
機能	PW8001 との通信のタイムアウト時間(秒)を設定し、フラグを返します。
使用例	obj.setTimeout(15); %PW8001との通信のタイムアウト時間を15秒に設定します

flag = obj.open

引数	なし
戻り値	flag: 生成されたオブジェクトとPW8001の接続を確立できた場合は「0」、異常終了した場合には「1」もしくは事前定義されたMATLABのエラーメッセージを表示
機能	生成されたオブジェクトと PW8001 の接続を確立し、フラグを返します。
使用例	obj.open;

flag = obj.close

引数	なし
戻り値	flag: 生成されたオブジェクトとPW8001との接続を切断できた場合は「0」、異常終了した場合には「1」もしくは事前定義されたMATLABのエラーメッセージを表示
機能	生成されたオブジェクトと PW8001 の接続を切断し、フラグを返します。
使用例	obj.close;

flag = obj.send(command)

引数	command: PW8001に送信する通信コマンド文字列
戻り値	flag: 通信コマンドを設定し、PW8001に送信できた場合は「0」、異常終了した場合には「1」もしくは事前定義されたMATLABのエラーメッセージを表示
機能	PW8001 に通信コマンド文字列を送信し、フラグを返します。
使用例	obj.send(" *IDN?"); %通信コマンド" *IDN?"をPW8001に送信します

[str, flag] = obj.receive

引数	なし
戻り値	str: MATLAB上から送信した通信コマンドに対するPW8001の応答文字列 flag: PW8001から応答を受信できた場合は「0」、異常終了した場合には「1」もしくは事前定義されたMATLABのエラーメッセージを表示
機能	送信した通信コマンドに対する PW8001 の応答文字列を受信し、フラグを返します。
使用例	obj.receive;

[samplingSpeed, storageLength, storageMode, waveDataMax,... waveDataMin, flag] = obj.DownloadAnalogWaveData(chName)	
引数	chName: PW8001から取得したい以下の対象波形名の文字列 (U1, U2, U3, U4, U5, U6, U7, U8, I1, I2, I3, I4, I5, I6, I7, I8, CHA, CHC, CHE, CHGのいずれか) (ただし、CHA, CHC, CHE, CHGはアナログ波形に限る)
戻り値	samplingSpeed: 波形データのサンプリング速度 storageLength: 波形データのポイント数 storageMode: 波形データのストレージモード (ピーク圧縮のとき0、単純間引きのとき1を数値で返す) waveDataMax: 指定したアナログ波形データの最大値を格納する配列 waveDataMin: 指定したアナログ波形データの最小値を格納する配列 flag: PW8001からアナログ波形データを取得できた場合は「0」、異常終了した場合には「1」もしくは事前定義されたMATLABのエラーメッセージを表示
機能	PW8001 から指定したアナログ波形データの最大値と最小値を取得し、フラグを返します。
使用例	[samplingSpeed, storageLength, storageMode, waveDataMax, waveDataMin, flag] = obj.DownloadAnalogWaveData("U1"); %U1の電圧波形データの最大値と最小値を取り込みます

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

引数

なし

samplingSpeed: 波形データのサンプリング速度
storageLength: 波形データのポイント数
storageMode: 波形データのストレージモード
(ピーク圧縮のとき0、単純間引きのとき1を数値で返す)
logicCH: パルス入力に設定されているモーターチャンネル
※logicCHとモーターチャンネルのビット対応

ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
CHH	CHG	CHF	CHE	CHD	CHC	CHB	CHA

waveDataMax: パルス入力に設定されているモーターチャンネル全波形分を含んだデータの最大値を格納する配列
waveDataMin: パルス入力に設定されているモーターチャンネル全波形分を含んだデータの最小値を格納する配列
※waveDataMax, waveDataMinとモーターチャンネルのビット対応

ビット 15	ビット 14	ビット 13	ビット 12	ビット 11	ビット 10	ビット 9	ビット 8
CHA	CHB	CHC	CHD	CHE	CHF	CHG	CHH
ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
-	-	-	-	-	-	-	-

flag: PW8001からパルス入力に設定されているモーターチャンネル全波形分を含んだデータを取得できた場合は「0」、異常終了した場合には「1」もしくは事前定義されたMATLABのエラーメッセージを表示

戻り値

機能

PW8001 からパルス入力に設定されている全波形分を含んだデータを取得し、フラグを返します。

使用例

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

```
[chWaveData, flag] = obj.ExtractLogicChWaveData(chName, waveData)
```

引数	chName: パルス入力に設定されているモーターチャンネル名の文字列 (CHA, CHB, CHC, CHD, CHE, CHF, CHG, CHHのいずれか)
	waveData: DownloadLogicWaveData関数で取得したwaveDataMax、waveDataMinのいずれか
戻り値	chwaveData: 指定したモーターチャンネルのパルス波形データの最大値または最小値を格納する配列 flag: PW8001から指定したモーターチャンネルのパルス波形データを取得できた場合は「0」、異常終了した場合には「1」もしくは事前定義されたMATLABのエラーメッセージを表示
機能	PW8001 から指定したモーターチャンネルのパルス波形データの最大値または最小値を取得し、フラグを返します。
使用例	[chWaveData, flag] = ExtractLogicChWaveData(obj, "CHA", waveDataMax); %CHAのパルス波形データの最大値を取り込みます

HIOKI

www.hioki.co.jp/

本社 〒386-1192 長野県上田市小泉 81

製品のお問い合わせ

 **0120-72-0560**

TEL 0268-28-0560 FAX 0268-28-0569

9:00 ~ 12:00, 13:00 ~ 17:00
土・日・祝日を除く

info@hioki.co.jp

修理・校正のお問い合わせ

ご依頼はお買上店（代理店）または最寄りの営業拠点まで
お問い合わせはサービス窓口まで

TEL 0268-28-1688 cs-info@hioki.co.jp

国内拠点



2103 JA

編集・発行 日置電機株式会社

Printed in Japan

- ・CE 適合宣言は弊社ウェブサイトからダウンロードできます。
- ・本書の記載内容を予告なく変更することがあります。
- ・本書には著作権により保護される内容が含まれます。
- ・本書の内容を無断で転記・複製・改変することを禁止します。
- ・本書に記載されている会社名・商品名などは、各社の商標または登録商標です。