GRAPHTEC

データロガーからの瞬時値を文字列形式で受信！

テキスト形式やJSON形式で受信可能なインターフェースコマンドのご紹介

概要

データをバイナリ形式で受信する場合、転送データ容量は少なくなりますが、
バイナリ形式から物理値(*V や***℃)への変換が必要となり、プログラムコーディングが多く必要です。**

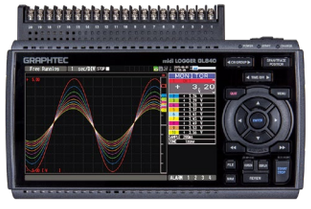
そこで！

テキスト形式「:MEAS:OUTP:ONECSV?<改行>」　JSON形式「:MEAS:OUTP:ONEJSON?<改行>」

これらのインターフェースコマンドを使用すると、変換後の数値をそのまま取得することができます！

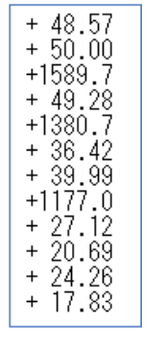
※ 瞬時値のみ取得可能です。　※ 機種によって一部コマンドが異なる場合があります。

バイナリ形式のコマンド



GL840

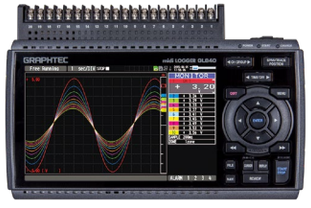
取得したバイナリ形式のデータ



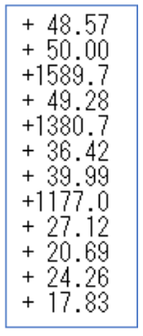
テキスト形式

画面表示

テキスト形式・JSON形式のコマンド



GL840



取得したテキスト形式データ

画面表示

GRAPHTEC

JSON形式とは

{} や[] などの括弧を使用して、CHごとに区切る記述方法です。
各プログラミング言語にて、JSONの整形(パース）を行うことができます。

各言語の整形例

JavaScript

```
const value = JSON.parse(jsonData);
```

python

```
value = json.loads(json_data)
```

C#

```
var value = JObject.Parse(jsonData);
```

実際のインターフェースコマンドの応答内容

:MEAS:OUTP:ONECSV?

瞬時値をカンマ区切りで取得できます。文字列の途中に改行はありません。

```
:MEAS:OUTP:ONECSV? ->
2022/06/16 14:34:41.235,+ 48.57,+ 50.00,+1589.7,+ 49.28,+1380.7,+ 36.42,+ 39.99,+1177.0,
+ 27.12,+ 20.69,+ 24.26,+ 17.83,+ 11.40,+ 14.97,+  8.54,+  2.11,-  4.32,-  0.75,-  7.18,
- 13.61,LLLLLLLLLL,LLLLLLLLLL,LLLL
```

:MEAS:OUTP:ONEJSON?

JSON形式で取得できます。文字列の途中に改行はありません。
瞬時値だけでなく、各CHの単位やアラームなど詳細な情報を取得可能です。

※右図は取得したJSON形式の文字列に改行を入れて整形したものです。
(一部のCHのみ)


```
{
  id : "234"
  datas : [
    0 : {
      name : "date"
      items : {
        value : "2022/06/16 14:36:01.235"
      }
    }
    1 : {
      name : "ch"
      items : {
        ch : "1"
        value : "-26.60"
        unit : "V"
        annot : " CH1"
        alarm : "L"
        color : "#E02830"
      }
    }
    2 : {
      name : "ch"
      items : {
        ch : "2"
        value : "-26.55"
        unit : "V"
        annot : " CH2"
        alarm : "L"
        color : "#0068B8"
      }
    }
```


対象機種

コマンド	GL100	GL220 GL820	GL240	GL840	GLT400	GL900	GL980 GL2000	GL7000	GL7000 Plus
:MEAS:OUTP:ONECSV?	-	-	○ V1.54以降	○ V1.60以降	○ V1.01以降	-	-	-	○ V2.00以降
:MEAS:OUTP:DATA:ONECSV?	-	-	-	-	-	-	○ V1.01以降	-	-
:MEAS:OUTP:ONEJSON?	-	-	○ V1.54以降	○ V1.60以降	○ V1.01以降	-	-	-	○ V2.00以降

GRAPHTEC

コード例(.NET Framework C# の場合)

本体とUSB接続し、1秒ごとに本体データをJSON形式で取得する例です。
※SDKに含まれるサンプルプログラムを使用しています。

```
// USB接続のパラメータ設定
Param param = new Param();
param.UsbId.usbId = 0; // USB ID0番

// 本体に接続
var devIo = new GtcDevIo.DevIo();
var flag = devIo.Open(param);
if (flg == false) return;

for (var i = 0; i < 100; i++)
{
     var json = devIo.SendQuery(":MEAS:OUTP:ONEJSON?");
     var value = JOBject.Parse(json);
     Thread.Sleep(1000);     // 1秒待つ
}
```

詳細資料をご希望の場合

弊社ホームページよりソフトウェア開発キット(SDK)の申請をお願いいたします。(無償提供)

ホームページ URL

▶▶ https://www.graphtec.co.jp/site_download/sdk/sdk-instrument.html