

TSA-U24 の測定値を読み取る簡単なサンプル

この説明書では、USB ケーブルで PC に接続した TSA-U24 へ Python から RD コマンドを送信し、現在の測定値を約 1 秒間隔で表示する方法を説明します。

TSA-U24 用の専用 DLL や通信ライブラリは使用しません。PYTHON の一般的なシリアル通信ライブラリを使い、TSA-U24 へコマンドを直接送信します。

プログラムをできるだけ簡単にするため、通信エラーやタイムアウトなどは細かく分類せず、すべて「エラー発生」と表示します。

このサンプルの前提

- TSA-U24 の設定はあらかじめ完了している
- TSA-U24 は USB ケーブルで PC に接続されている
- TSA-U24 の Chk Sum Mode は OFF に設定されている
- 使用する COM ポート番号が分かっている
- Windows PC で使用する

1. インストール

① PYTHON のインストール

Python をインストールしていない場合は、Python 公式サイトから Windows 版をダウンロードしてインストールします。

Python 公式サイト:

<https://www.python.org/downloads/windows/>

インストーラを起動したときに、「Add python.exe to PATH」という項目が表示される場合は、チェックを入れてからインストールしてください。

インストール後、Windows の「コマンドプロンプト」を開き、次のコマンドを入力します。

```
python --version
```

Python のバージョン（例：Python 3.x.x）が表示されればインストール完了です。

※ 環境によっては python の代わりに py コマンドを使用します。（py -version）

② シリアル通信ライブラリのインストール

このサンプルでは、Python から COM ポートを使用するために pySerial を使用します。

コマンドプロンプトで次のコマンドを実行してください。

```
python -m pip install pyserial
```

python コマンドではなく py コマンドを使用する環境では、次のようにします。

```
py -m pip install pyserial
```

2. サンプルプログラムの作成

① PYTHON ファイルを作成する

メモ帳や任意のテキストエディタを開き、次のプログラムを入力します。

```
import serial
import time

COM_PORT = "COM5"
INTERVAL_SEC = 1.0
try:
    ser = serial.Serial(
        port=COM_PORT,
        baudrate=115200,
        bytesize=serial.EIGHTBITS,
        parity=serial.PARITY_NONE,
        stopbits=serial.STOPBITS_ONE,
        timeout=1.0,
        rtscts=True
    )
    print("TSA-U24 データ受信開始")
    print("Ctrl+C で終了します")
    while True:
        try:
            # 受信バッファに古いデータが残っていた場合は捨てる
            ser.reset_input_buffer()
            # TSA-U24 へ RD + CR を送信
            ser.write(b"RD\r")
            # CR まで受信
            data = ser.read_until(b"\r")
            if not data:
                raise Exception()
            # 受信データを文字列へ変換
            line = data.decode("ascii").rstrip("\r")
            # RD 応答らしいかだけ簡単に確認
            if not line.startswith("RD "):
                raise Exception()
            print(line)
        except Exception:
            print("エラー発生")
            time.sleep(INTERVAL_SEC)
    except KeyboardInterrupt:
        print("\n 終了します")
    except Exception:
        print("エラー発生")
finally:
    try:
        ser.close()
    except Exception:
        pass
```

ファイル名を、例えば次のようにして保存します。

```
tsa_u24_simple.py
```

拡張子が「.txt」ではなく、「.py」になっていることを確認してください。

② COM ポート番号を変更する

プログラム冒頭の次の部分を、実際に TSA-U24 が接続されている COM ポート番号へ変更します。

```
COM_PORT = "COM5"
```

例えば TSA-U24 が COM3 の場合は、

```
COM_PORT = "COM3"
```

とします。

ここはお使いになっている PC によって異なるため、必ず確認してください。

なお、TSA-U24 本体がどの COM ポートに接続されているかは、弊社サイトからダウンロードできるサンプルアプリケーション（TSA-Utility）や、Windows の「デバイス マネージャー」によって確認することができます。

③ 測定間隔を変更する

このサンプルでは約 1 秒ごとに RD コマンドを送信します。

```
INTERVAL_SEC = 1.0
```

例えば 2 秒ごとに読みたい場合は、

```
INTERVAL_SEC = 2.0
```

とします。

通常の簡単なデータ確認では、1 秒程度から始めることをおすすめします。

④ プログラムを実行する

コマンドプロンプトで、作成した Python ファイルが保存されているフォルダへ移動します。

その後、次のように実行します。

```
python tsa_u24_simple.py
```

正常に通信できると、例えば次のように表示されます。

```
TSA-U24 データ受信開始
Ctrl+C で終了します
RD 12.345 mA OK VALID
RD 12.346 mA OK VALID
RD 12.344 mA OK VALID
```

表示される値や単位は、TSA-U24 に設定されている Scale Factor、Scale Offset、Unit String などによって変わります。

プログラムを終了するときは、Ctrl+C を押してください。

⑤ 変更する場所

通常、ユーザーが変更する必要があるのは主に次の 2 か所です。

```
COM_PORT = "COM5"  
INTERVAL_SEC = 1.0
```

COM_PORT TSA-U24 が接続されている COM ポート番号を指定します。
INTERVAL_SEC RD コマンドを送信する間隔を秒単位で指定します。

⑥ 注意点

● Chk Sum Mode

このサンプルは、TSA-U24 の Chk Sum Mode が OFF の場合を前提としています。

TSA-U24 本体で次の設定になっていることを確認してください。

Chk Sum Mode = OFF

Chk Sum Mode が CHECK または VERIFY の場合は、コマンドや応答にチェックサムが付加されるため、この簡易サンプルのままでは使用できません。

● USB 通信条件

TSA-U24 の USB Virtual COM Port は次の通信条件を使用します。

Baud rate : 115200 bps

Data bits : 8 bit

Parity : None

Stop bits : 1 bit

Flow Ctrl : Hardware

そのため、サンプルプログラムでは次の指定を行っています。

```
baudrate=115200  
bytesize=serial.EIGHTBITS  
parity=serial.PARITY_NONE  
stopbits=serial.STOPBITS_ONE  
rtscts=True
```

特に USB 接続ではハードウェアフロー制御を使用するため、rtscts=True は変更しないでください。

- エラー表示

このサンプルはプログラムを簡単にすることを優先しています。

そのため、例えば下に示すような場合すべて「エラー発生」とだけ表示します。

- TSA-U24 が接続されていない
- COM ポート番号が間違っている
- 他のソフトウェアが COM ポートを使用している
- 応答がタイムアウトした
- 予想していないデータを受信した
- 通信中に USB ケーブルが抜かれた

正式なアプリケーションを作成する場合は、それぞれのエラーを適切に判定する必要があります。

3. このサンプルで行っていること

このプログラムの基本処理は次の通りです。

```
COM ポートを開く
↓
"RD<CR>" を送信
↓
CR まで 1 行受信
↓
画面に表示
↓
約 1 秒待つ
↓
繰り返す
```

TSA-U24 の RD コマンドに対する通常の応答は、次の形式です。

RD <data> <unit> <stat> <valid> <CR>

例:

RD 12.345 mA OK VALID <CR>

<stat> には次のいずれかが返ります。

OK
UNDER
OVER

<valid> には次のいずれかが返ります。

VALID
INVALID

このサンプルでは、受信した 1 行をそのまま画面に表示しています。

4. 補足

このサンプルは、TSA-U24 の通信プロトコルを完全に実装するものではありません。

チェックサム処理、詳細なエラー判定、再接続、リトライ、デバイス自動検索などは省略しています。

TSA-U24 を自作プログラムから簡単に利用するための、最小限の通信例として使用してください。

通信コマンド、通信方法についての詳細は、TSA-U24 取扱説明書をご参照ください。