

JupyterLab チュートリアル: Python と USB-201 を使用するデータ収集に関して

1. JupyterLab の紹介

[JupyterLab](#) は Jupyter Notebook 用のオープンソース開発環境 (IDE) です。Jupyter Notebook は、主に、プログラムコード、コードの実行結果、ドキュメントや解説、グラフ化を 1 つのファイルにまとめる時に使用されます。コードはセル単位で実行可能ですので、コード操作とその結果の確認が容易です。JupyterLab は複数の Notebook を別々のタブで開くことができます。Web ブラウザでタブページごとに複数のサイトを開くケースをイメージしてください。起動時にワークスペースファイルを読み取り、どの Notebook を開くべきかを判断します。本資料で、JupyterLab の概要紹介と、[USB-201](#) を制御して実際のアナログデータを取得・表示する方法を説明します。

2. JupyterLab のメリット

JupyterLab の特徴の一つが、ドキュメント、コード、実行結果を組み合わせたことができる点です。直感的に理解できるインターフェースは初心者でも使いやすく、同時に経験豊富な開発者や研究者のニーズに応える高度な機能も持っています。そのため、データサイエンスなどの分野で使用された実績のあるツールです。

また、双方向性も特徴の一つです。チャンク単位でのコード実行し、その直後に出力を確認できる機能は、試行錯誤しながら問題解決する反復的な手法に最適で

す。この実践的な手法はコードのアルゴリズム改良や複数の実験を行う際において特に有効です。

さらに、JupyterLab はクロスプラットフォームに対応しています。Windows、macOS、Linux などのさまざまなオペレーティングシステム上で動作します。この汎用性とブラウザベースという特徴により、導入へのハードルが下がり、利便性が向上します。

3. アプリケーション

JupyterLab の汎用性は非常に高く、さまざまな分野で使用可能です。

データサイエンスや機械学習の分野では、前処理、モデリング、視覚化を行うための包括的なツールとして使用されています。

実験データのシミュレーション・分析・解析する際にも JupyterLab は優れた能力を発揮します。これは研究の現場でよく見る光景です。

教育の現場では教材作成の際に使用可能です。

アナリストは、実行、レポート準備、そしてダッシュボードを構築するために JupyterLab を利用しています。

ソフトウェア開発者は、アルゴリズムのプロトタイプ作成やワークフローを文書化する際に JupyterLab を使用しています。

4. USB-201 での使用例

次に、USB-201 を使用してデータを取得する方法について説明します。

このケースでは Python を使用します。

まず、Python がインストールされていることを確認します。次に USB-201 でのデータ取得を設定するために Python クラスを作成します。クラスオブジェクトを作成すると、USB-201 への接続が試行され USB-201 を検出できると通信が開始されます。その後、サンプルレートとチャンネルあたりのサンプル数を設定します。データ取得手順として 2D 配列 (行、列) を返し、データをプロットするルーチンを用意しています。Matplotlib と IPython の Display モジュールは、2D 配列をプロットとし、配列を行と列の表形式で表示します。

チャンネルの番号はチャンネル 0 からチャンネル 3 で固定されています。

BACKGROUND モードを使用して、チャンネルあたり 2000 サンプルに制限すると、必要数のサンプルが返されるまでコードをブロックします。

ただし、BACKGROUND モードはコードをブロックしますが、Notebook は次のセルを終了したかのように強調表示します。この動作は、完了までに数秒以上かかるデータ取得で確認できます。

Notebook 全体は GitHub で入手できます。

[USB-201-Jupyter/USB-201-Demo.ipynb at main · Diligent/USB-201-Jupyter.](#)

以下は USB-201 からデータを取得する方法です。

Notebook の最初のセルは、USB-201 からデータを取得するためのコードを含むクラスオブジェクトを定義します。構造は次のとおりです。

```

Class daq:
    __init__
        Initializes internal variables
        Calls open_device
    __del__
        Closes the device and frees allocated memory
    open_device(name)
    set_samples_per_channel(samples)
    set_sample_rate(rate)
    array = get_data()
    plot_data(array)

```

```

: display(markdown("# Welcome to my USB-201 Demonstration"))

mcc = daq("USB-201", rate=1000, samples=200)

new_number_of_samples = mcc.set_samples_per_channel(number_of_samples=2000)
new_rate = mcc.set_sample_rate(sample_rate = 2000)

daq_data = mcc.get_data()
mcc.plot_data(daq_data)

h=display(display_id=2)
df = pd.DataFrame(data=daq_data[0:mcc.points_per_channel], columns=['Ch0', 'Ch1', 'Ch2', 'Ch3'])
h.update(df)

del mcc

```

mcc = daq("USB-201", rate=1000, samples=200) で、クラスオブジェクトを作成し、サンプルレートとチャンネルあたりのサンプル数を設定します。

次に、クラスメソッドを使用して、チャンネルあたりのサンプル数とサンプルレートを設定します。

```
new_number_of_samples = mcc.set_samples_per_channel(number of samples = 2000)
```

```
new_rate = mcc.set_sample_rate(sample_rate = 2000)
```

次に、以下のコマンドを使用して、取得データを含む 2D 配列を取得します。

```
daq_data = mcc.get_data()
```

daq_data はデータをプロットする plot メソッドに渡されます。

IPython Display モジュールは、データを見やすく表示するために使用できます。これが有効な期間中は、データ取得を複数回実行できます。たとえば、ループ機能を使用して追加データを収集できます。セルを離れる前に、オブジェクトを破棄してデバイスを閉じ、割り当てられたメモリを解放します。

Welcome to my USB-201 Demonstration

Found USB-201 board number = 0

Serial number: 1FFC8D1

Product type: 0x113

Samples per channel 200

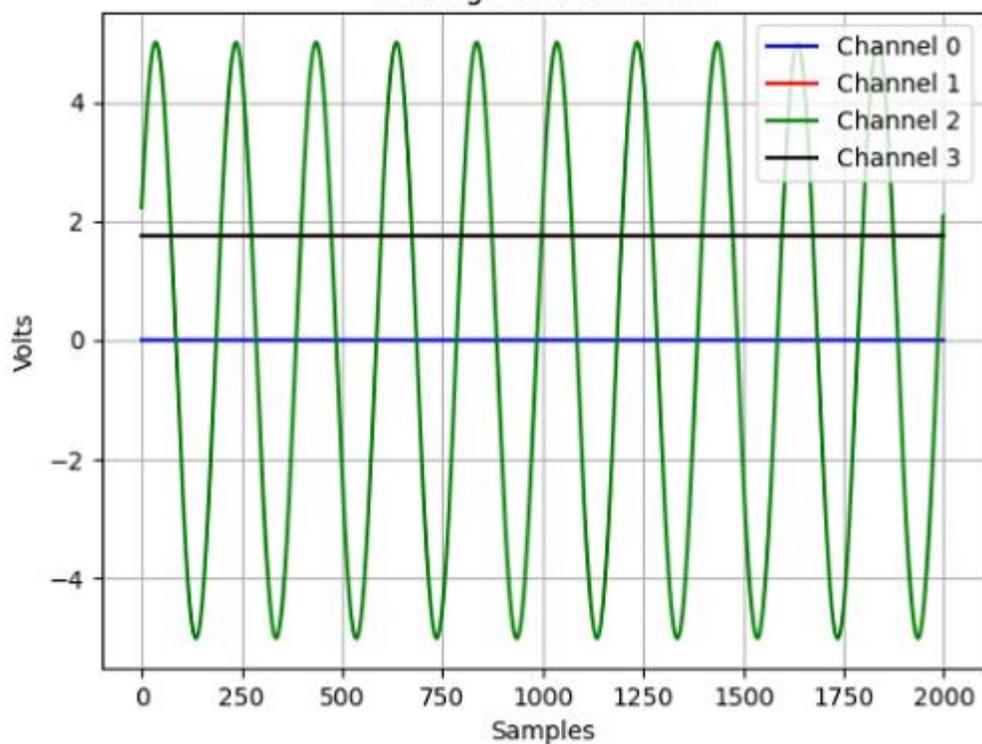
Sample rate is 1000

Samples per channel changed to 2000

Sample rate changed to 2000 Hz

Scan completed successfully.

Plotting Demonstration



	Ch0	Ch1	Ch2	Ch3
0	-0.001978	1.74919	2.232256	1.753897
1	-0.001978	1.74919	2.375219	1.753897
2	-0.001978	1.74919	2.513076	1.753897
3	-0.001978	1.74919	2.645827	1.753897
4	-0.001978	1.74919	2.778579	1.753897
...
1995	-0.001978	1.74919	1.502124	1.753897
1996	-0.001978	1.74919	1.650193	1.753897
1997	-0.001978	1.74919	1.798262	1.753897
1998	-0.001978	1.74919	1.941225	1.753897
1999	-0.001978	1.74919	2.084187	1.753897

2000 rows × 4 columns

USB-201 closed...

上記の手順により、JupyterLab で USB-201 を制御し、シミュレーションデータを取得・表示できます。USB-200 シリーズの詳細については、以下をご参照ください。

https://www.nf-techno.co.jp/products/mcc/usb_ana_14bit.html#