

DA 出力 Python モジュール

K01DAZ-PyModule マニュアル

1. はじめに

本 DA 出力 Python モジュール、K01DAZ-PyModule は、PC に USB 経由接続された DA ユニットに対して、Python プログラム、電圧出力、波形出力、デジタル入出力機能を提供するものです。以下タートル工業社の DA ユニットに対応しています。

品番	分解能	速度	出力電圧	ch 数	メモリ
TUSB-K01DAZ	14bit	20kHz	±2.5V	1	4kW

■組み込み型ローコストDAコンバータ TUSB-K01DAZ



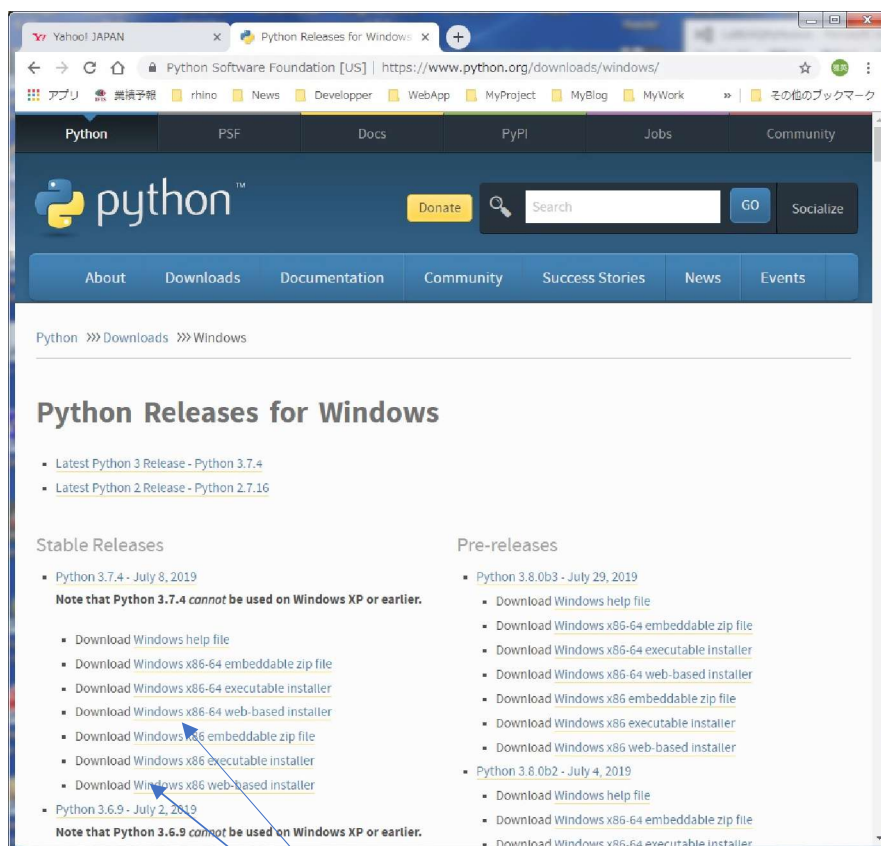
モジュールには初期化、デバイスオープン、デバイスクローズ、DA 出力、波形出力、デジタル入出力等の関数が収められています。

PC の Windows7, 8, 10 32/64bit の OS に対応しています。また本モジュールは .Net framework4.5 が必要です。

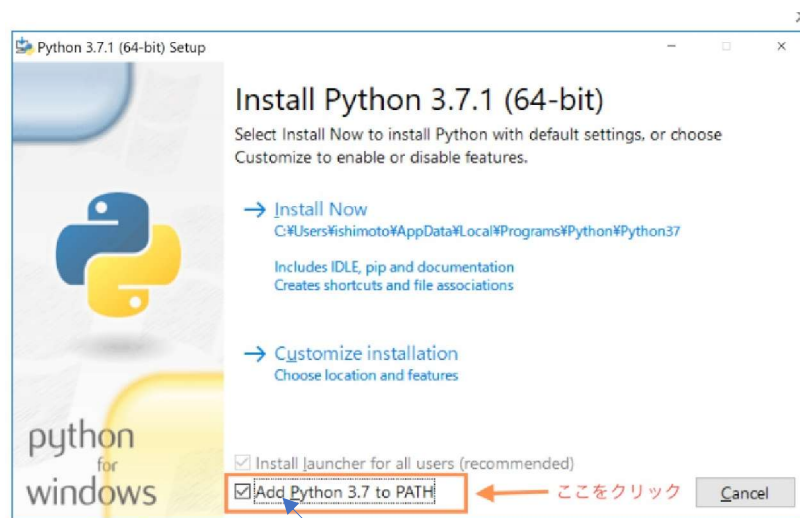
2. インストールおよび環境設定

以下のサイトから Windows PCへ Python をダウンロード、インストールします。

<https://www.python.org/downloads/windows/32bit>

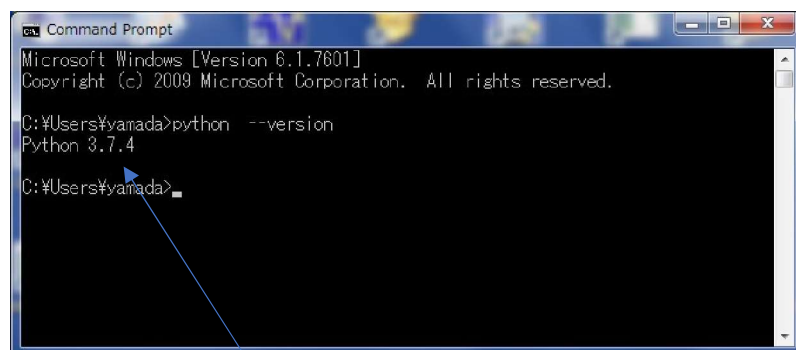


32bit または 64bit、インストール対象 PC の OS に合わせてダウンロード、インストールを実行します。



ここのチェックマークは必ずオンにして、インストールを実行します。

完了後、以下、コマンドプロンプトで、インストール、およびバージョンを確認します。

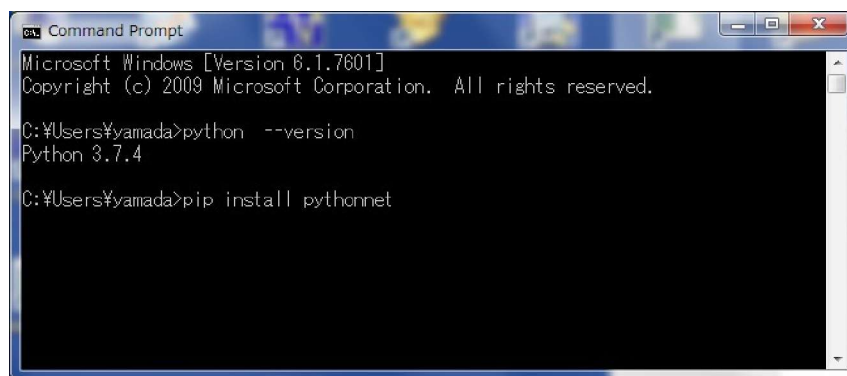


バージョンが表示されます。

Windows P CへPython ドットネットのインストール

本モジュールは C#で開発されており、これを Python から呼び出し可能とするため、Pythonnet(Python for .net - GitHub)のインストールが必要です。

コマンドプロンプトから pip コマンドでインストールします。以下画面です。



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\yamada>python --version
Python 3.7.4

C:\Users\yamada>pip install pythonnet
```

CD 内のファイル構成は以下です。

Documents	>本マニュアル、DA 出力 Python モジュールマニュアル
pyModule	>DA 出力 Python モジュール、LabdaqDaModule.dll
pySample	>SampleLabdaqDa.py DA 出力サンプルプログラム
	>SampleLabdaqWave.py 波形出力サンプルプログラム
	>SampleLabdaqDio.py DIO 入出力サンプルプログラム

CD 内 pyModule フォルダの LabdaqDaModule.dll を python のプログラムがあるフォルダにコピーします。

以上で python から本 DA 出力モジュールが使用できます。

3. DA 出力 Python モジュール

以下のように clr をインポートして、呼び出します

```
import clr
clr.AddReference('LabdaqDaModule')
from LabdaqDaModule import LabdaqDa
```

PyModule の呼び出し手順は

インスタンスの作成

```
labDa = LabdaqDa()
```

モジュールの初期化

```
ret = labDa.Initial(osmode)
```

osmode は、ターゲット工業ドライバの 32bit 版をご使用の場合は 0、64bit 版をご使用の場合は 1 を指定します。

デバイスのオープン

```
ret = labDa.OpenDa(Id)
```

使用するデバイスの Id を 0-15 の範囲で指定します。Id は DA ユニットの背面にあります。

|
DA 電圧出力処理

DA 波形出力処理

デジタル入出力処理等

|
デバイスのクローズ

```
labDa.CloseDa(Id)
```

モジュールの開放

```
labDa.Terminate()
```

上記が基本の手順です、特に、デバイスのクローズ、モジュールの開放の終了処理をしないで、プログラムを終了すると次回、起動しない場合があります。この場合、一度、DA ユニットの PC から切り離してください。終了処理は必須です。

4、DA 出力 Python モジュール関数一覧

定義定数

```
#const define
```

```
OS_32BIT = 0
```

```
OS_64BIT = 1
```

```
DA_DATA_MIN = -8192
```

```
DA_DATA_NAX = 8191
```

```
DA_RANGE_MIN = -2.5
```

```
DA_RANGE_MAX = 2.5
```

```
DA_DATA_LEN_MAX = 4096
```

```
CLK_INTERVAL_MIN = 50000 #nsec 50usec
```

```
CLK_INTERVAL_MAX = 1000000000 #nsec 1000msec
```

```
DA_STATUS_VOLT = 0
```

```
DA_STATUS_WAVE = 1
```

```
DIO_DIR_INPUT = 0
```

```
DIO_DIR_OUTPUT = 1
```

```
DIO_STATUS_INPUT = 0
```

```
DIO_STATUS_LO_OUTPUT = 1
```

```
DIO_STATUS_HI_OUTPUT = 2
```

```
DIO_STATUS_CLOCK = 3
```

モジュールの初期化

```
ret = labDa.Initial(osmode)
```

osmode : タートル工業ドライバの 32bit 版をご使用の場合は 0、64bit 版
をご使用の場合は 1 を指定します。

ret : True, False

デバイスのオープン

ret = labDa.OpenDa(Id)

Id : 使用するデバイスの Id を 0-15 の範囲で指定します。Id は DA ユニットの背面にあります。

ret: True, False

DA 電圧出力処理

ret = labDa.OutDa(Id, daData)

Id : 使出力する DA の Id を指定します。

daData : 出力電圧のデータを指定します。変換式は

$$dvalue = (volt/2.5)*8192.0$$

daValue の範囲は-8192~8191

ret : 0 で正常、その他はエラー

DA 波形出力処理

波形データのセット

ret, WaveData = labDa.SetWaveData(Id, WaveData)

Id : 使出力する DA の Id を指定します。

WaveData : 出力波形のデータを指定します。変換式は

$$dvalue = (volt/2.5)*8192.0$$

daValue の範囲は-8192~8191

波形データ数は最大 4096

ret : 0 で正常、その他はエラー

波形出力開始

ret = labDa.StartWaveOut(Id, clockmode, clockdiv)

Id : 使出力する DA の Id を指定します。

clockmode : 0 で内部クロック、1 で外部クロック

clockdiv:50000 から 1000000000 の範囲で nsec で指定します。

ret : 0 で正常、その他はエラー

以下の波形出力停止を実行しない限り、繰り返し、波形は出力されます。

DA ステータス取得

ret, status = labDa.GetDaStatus(Id, status)

Id: 使出力する DA の Id を指定します。

status: DA のステータスが返されます。0 は電圧出力、1 は波形出力

ret: 0 で正常、その他はエラー

波形出力停止

ret = labDa.StopWaveOut(Id)

Id: 使出力する DA の Id を指定します。

ret: 0 で正常、その他はエラー

デジタル入出力処理

DIO 入力、または出力の設定

ret = labDa.SetDioDir(Id, dionum, diodir)

Id: 使出力する DA の Id を指定します。

dionum: 0 または 1 で指定します。

diodir: 0 で入力、1 で出力

ret: 0 で正常、その他はエラー

DIO 出力

ret = labDa.OutputDio(Id, dionum, dioout)

Id: 使出力する DA の Id を指定します。

dionum: 0 または 1 で指定します。

dioout: 0 でローレベル、1 でハイレベルを出力します。

ret: 0 で正常、その他はエラー

DIO 入力

ret, input0, input1 = labDa.InputDio(Id, input0, input1)

Id: 使出力する DA の Id を指定します。

input0: DIO 入力 0 の状態が返ってきます。

0 でローレベル、1 でハイレベル。

Input1: DIO 入力 1 の状態が返ってきます。

0 でローレベル、1 でハイレベル。

ret: 0 で正常、その他はエラー

デバイスのクローズ

```
ret = labDa.CloseDa(Id)
```

Id : 使用するデバイスの Id を 0-15 の範囲で指定します。Id は DA ユニットの背面にあります。

ret: True, False

モジュールの開放

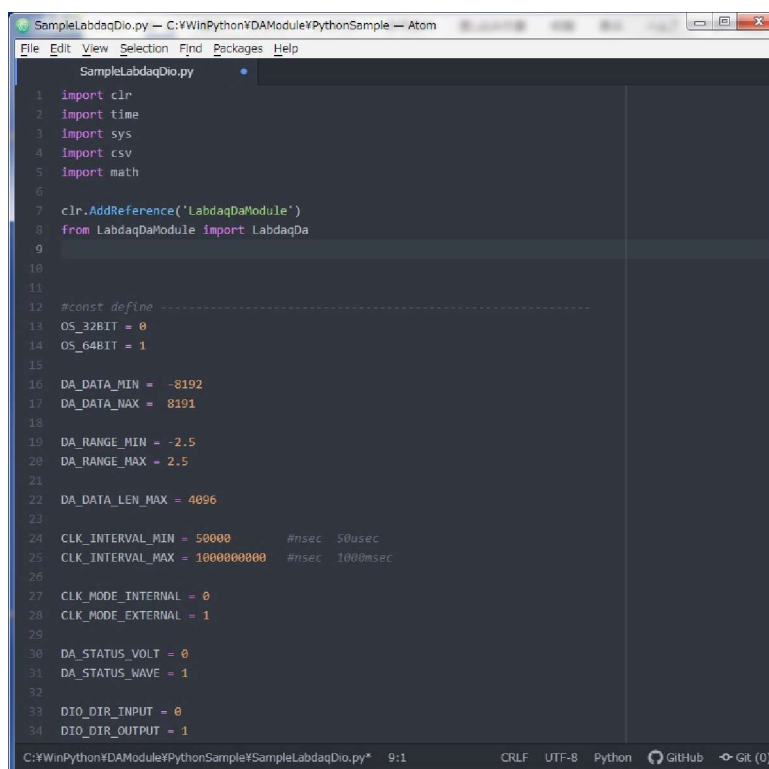
```
ret = labDa.Terminate()
```

ret: True, False

5、サンプルプログラム

pySample フォルダに、本 DA 出力モジュールをすべて使用した 3 個のサンプルが収められています。

SampleLabdaqDa.py	DA 出力サンプルプログラム
SampleLabdaqWave.py	波形出力サンプルプログラム
SampleLabdaqDio.py	DIO 入出力サンプルプログラム



```
SampleLabdaqDio.py
1 import clr
2 import time
3 import sys
4 import csv
5 import math
6
7 clr.AddReference('LabdaqDaModule')
8 from LabdaqDaModule import LabdaqDa
9
10
11
12 #const define -----
13 OS_32BIT = 0
14 OS_64BIT = 1
15
16 DA_DATA_MIN = -8192
17 DA_DATA_MAX = 8191
18
19 DA_RANGE_MIN = -2.5
20 DA_RANGE_MAX = 2.5
21
22 DA_DATA_LEN_MAX = 4096
23
24 CLK_INTERVAL_MIN = 50000 #msec 50usec
25 CLK_INTERVAL_MAX = 100000000 #msec 1000msec
26
27 CLK_MODE_INTERNAL = 0
28 CLK_MODE_EXTERNAL = 1
29
30 DA_STATUS_VOLT = 0
31 DA_STATUS_WAVE = 1
32
33 DIO_DIR_INPUT = 0
34 DIO_DIR_OUTPUT = 1
```

計測テスト、グラフチャートソフトウェア開発

eLaBNET

お問い合わせは089-957-2243

info@labnet.co.jp

株式会社 松山アドバンス 愛媛県松山市古川西 2 丁目 11-24

TEL 089-957-2243

FAX 089-958-2143

www.elabnet.jp