

NE-USB USB 通信
サンプルプログラム
(Windows Python)

内容

NE-USB USB 通信 サンプルプログラム (Windows Python)	1
1. 概要	3
1.1. システム概要	3
2. 開発環境	3
2.1. Windows 環境	3
2.1.1. 環境構築	4
3. サンプルソース概要	5
3.1. コマンド操作説明	5
3.1.1. コマンド一覧	5
3.1.2. LED を制御	6
3.1.3. ブザーパターンでブザーを制御	7
3.1.4. 音量でブザーを制御	7
3.1.5. ブザーパターンと音量でブザーを制御	8
3.1.6. 接続表示の設定	8
3.1.7. タッチセンサ入力状態を取得 (NE-ST-USB/NE-WT-USB のみ)	9
3.1.8. リセット	9

1. 概要

NE-USB を USB 通信で制御するための、サンプルプログラムの概要を記載する。

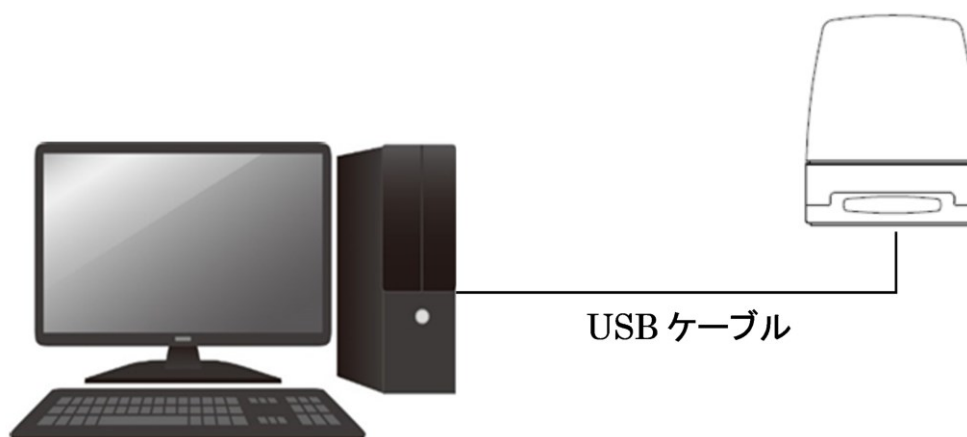
本プログラムは、Python での制御をおこなうことを目的としている。

なお、本プログラムはあくまでもサンプルであるため、異常系に関しては別途設計する必要がある。

1.1. システム概要

本プログラムのシステム構成図は以下の通り。

本プログラムでは、1 台の NE-USB の機器を USB 通信で制御を行う。



2. 開発環境

サンプルプログラムの開発環境を以下に示す。

2.1. Windows 環境

開発環境		備考
開発 OS	Windows10	
開発言語	Python	3.10 以降
パッケージ	PyUSB	1.2.1 以降
ライブラリ	libusb	1.0.25 以降

2.1.1. 環境構築

・libusb のインストール

libusb の GitHub からバイナリをダウンロードする。

※2022/03/30 現在のバージョンは、v1.0.25

<https://github.com/libusb/libusb/releases>

圧縮ファイル内の VS2019¥MS64¥dll¥libusb-1.0.dll を解凍し、C:¥Windows¥System32 に配置する。

※管理者権限が必要です。

・仮想環境の作成

システムの環境に影響を与えないように Python の仮想環境を作成し、USB 操作用のパッケージをインストールする。

コマンドプロンプトを起動し、作業フォルダに移動してから仮想環境を作成

```
> python -m venv venv
```

仮想環境を有効化

```
> venv¥Scripts¥activate.bat  
(venv) >
```

※仮想環境が有効化されるとコマンドプロンプトの先頭に“(venv)”が表示される。

標準パッケージのアップデート

```
(venv) > python -m pip install -U pip setuptools
```

USB 操作用のパッケージ(PyUSB)をインストール

```
(venv) > python -m pip install pyusb
```

・サンプルプログラムの実行と仮想環境の終了

サンプルプログラムの実行

```
(venv) > python main.py
```

仮想環境の終了

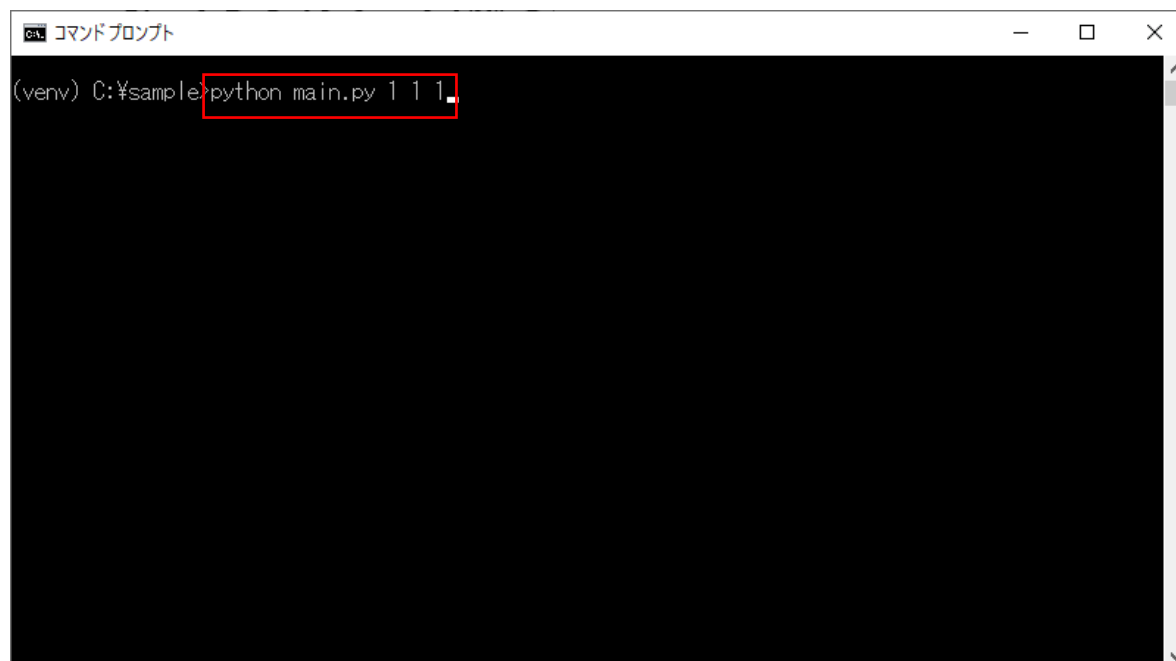
```
(venv) > venv¥Scripts¥deactivate.bat  
>
```

以降、「仮想環境の有効化」を行えばサンプルプログラムの実行が行える。

3. サンプルソース概要

3.1. コマンド操作説明

コマンドプロンプト上では、コマンドライン引数を指定することで各動作のコマンドを実行される。



3.1.1. コマンド一覧

コマンド名	内容
LED を制御	LED 色と LED パターンを指定して点灯、パターン点灯させる
ブザーパターンでブザーを制御	ブザーパターンと回数を指定してブザーを吹鳴させる
音量でブザーを制御	音量を指定してブザー音量を変更する
ブザーパターンと音量でブザーを制御	ブザーパターンと回数と音量を指定してブザーを吹鳴する
接続表示の設定	接続時の表示の設定を変更する
タッチセンサ入力状態取得	タッチセンサの入力状態をメッセージ表示する
リセット	LED を消灯し、ブザーを停止させる

3.1.2. LED を制御

以下のコマンドライン引数を指定して、コマンドを実行する。

No.	設定パラメータ	値
1	コマンド識別子	1
2	LED 色	消灯:0 赤:1 緑:2 黄:3 青:4 紫:5 空:6 白:7 現状の設定を維持:15
3	LED パターン	消灯:0 点灯:1 LED パターン 1:2 LED パターン 2:3 LED パターン 3:4 LED パターン 4:5 LED パターン 5:6 LED パターン 6:7 現状の設定を維持:15

例: python main.py 1 1 1

3.1.3. ブザーパターンでブザーを制御

以下のコマンドライン引数を指定して、コマンドを実行する。

No.	設定パラメータ	値
1	コマンド識別子	2
2	ブザーパターン	停止:0 吹鳴(連続):1 スweep音:2 断続音:3 弱注意音:4 強注意音:5 きらきら星:6 ロンドン橋:7 現状の設定を維持:15
3	ブザーの連続動作・回数動作	連続動作:0 回数動作:1~14 現状の設定を維持:15

例: python main.py 2 1 1

3.1.4. 音量でブザーを制御

以下のコマンドライン引数を指定して、コマンドを実行する。

No.	設定パラメータ	値
1	コマンド識別子	3
2	ブザー音量	消音:0 音量:1~9 最大音量:10 現状の設定を維持:15

例: python main.py 3 1

3.1.5. ブザーパターンと音量でブザーを制御

以下のコマンドライン引数を指定して、コマンドを実行する。

No.	設定パラメータ	値
1	コマンド識別子	4
2	ブザーパターン	停止:0 吹鳴(連続):1 スweep音:2 断続音:3 弱注意音:4 強注意音:5 きらきら星:6 ロンドン橋:7 現状の設定を維持:15
3	ブザーの連続動作・回数動作	連続動作:0 回数動作:1~14 現状の設定を維持:15
4	ブザー音量	消音:0 音量:1~9 最大音量:10 現状の設定を維持:15

例: python main.py 4 1 3 5

3.1.6. 接続表示の設定

以下のコマンドライン引数を指定して、コマンドを実行する。

No.	設定パラメータ	値
1	コマンド識別子	5
2	接続表示の設定	OFF:0 ON:1

例: python main.py 5 0

3.1.7. タッチセンサ入力状態を取得 (NE-ST-USB/NE-WT-USB のみ)

以下のコマンドライン引数を指定して、コマンドを実行する。

No.	設定パラメータ	値
1	コマンド識別子	6

例: python main.py 6

コマンドプロンプトに状態が出力される。

- ・タッチセンサ入力状態が OFF の場合: touch sensor input = OFF
- ・タッチセンサ入力状態が ON の場合: touch sensor input = ON

3.1.8. リセット

以下のコマンドライン引数を指定して、コマンドを実行する。

No.	設定パラメータ	値
1	コマンド識別子	7

例: python main.py 7