

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

## 内容

LR-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 .....	3
2.2. Linux 環境 .....	4
2.2.1. 環境構築 .....	5
3. サンプルソース概要 .....	7
3.1. コマンド操作説明 .....	7
3.1.1. コマンド一覧 .....	7
3.1.2. LED ユニットを制御 .....	7
3.1.3. 複数の LED ユニットを制御 .....	8
3.1.4. ブザーパターンでブザーを制御 .....	8
3.1.5. ブザーパターンと音階でブザーを制御 .....	8
3.1.6. リセット .....	9

## 1. 概要

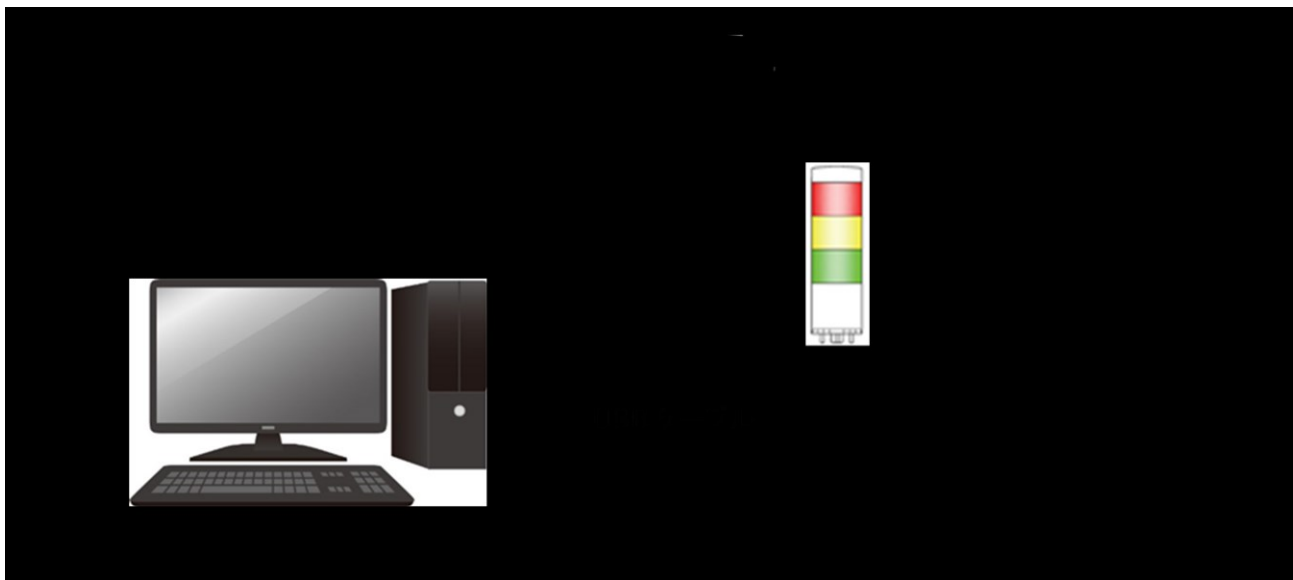
LR-USB を USB 通信で制御するための、サンプルプログラムの概要を記載する。

本プログラムは、パトライトが提供する DLL を使用せずにでの制御をおこなうことを目的としている。

### 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 LR-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	3.6 以降
パッケージ	<a href="#">PyUSB</a>	1.1.1 以降
ライブラリ	<a href="#">libusb</a>	1.0.24 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。

※2021/07/05 現在のバージョンは、v1.0.24

<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。

## 2.2. Linux 環境

開発環境		備考
開発 OS	Ubuntu	18.04
開発言語	Python	3.6 以降
パッケージ	<a href="#">PyUSB</a>	1.1.1 以降

ライブラリ	libusb	1.0.21-2
-------	--------	----------

## 2.2.1. 環境構築

### ・libusb のインストール

Ubuntu(18.04)では標準パッケージとしてインストールされている。何らかの影響でインストールされていない場合は、`apt-get` コマンドにてインストールをする。

```
$ sudo apt-get update
$ sudo apt-get install libusb-1.0.0
```

### ・Python3 仮想環境のパッケージのインストール

Python3 の仮想環境を作成するため、パッケージを `apt-get` コマンドでインストールする。

```
$ sudo apt-get install python3-venv
```

### ・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。

端末を起動し、作業ディレクトリに移動してから仮想環境を作成する。

```
$ python3 -m venv venv
```

仮想環境の有効化

```
$ source venv/bin/activate
(venv) $
```

※仮想環境が有効化されると端末の先頭に“(venv)”が表示される。

標準パッケージのアップデート

```
(venv) $ python3 -m pip install -U pip setuptools
```

USB 操作用のパッケージをインストール

```
(venv) $ python3 -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) $ python3 main.py
```

仮想環境の終了

```
(venv) $ deactivate  
$
```

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。

・サンプル実行時にアクセス拒否される場合

サンプルを実行した際に、「usb.core.USBError: [Errno 13] Access denied (insufficient permissions)」が表示される場合は、USB デバイスへのアクセス権限がないため root 権限にて実行するようにしてください。

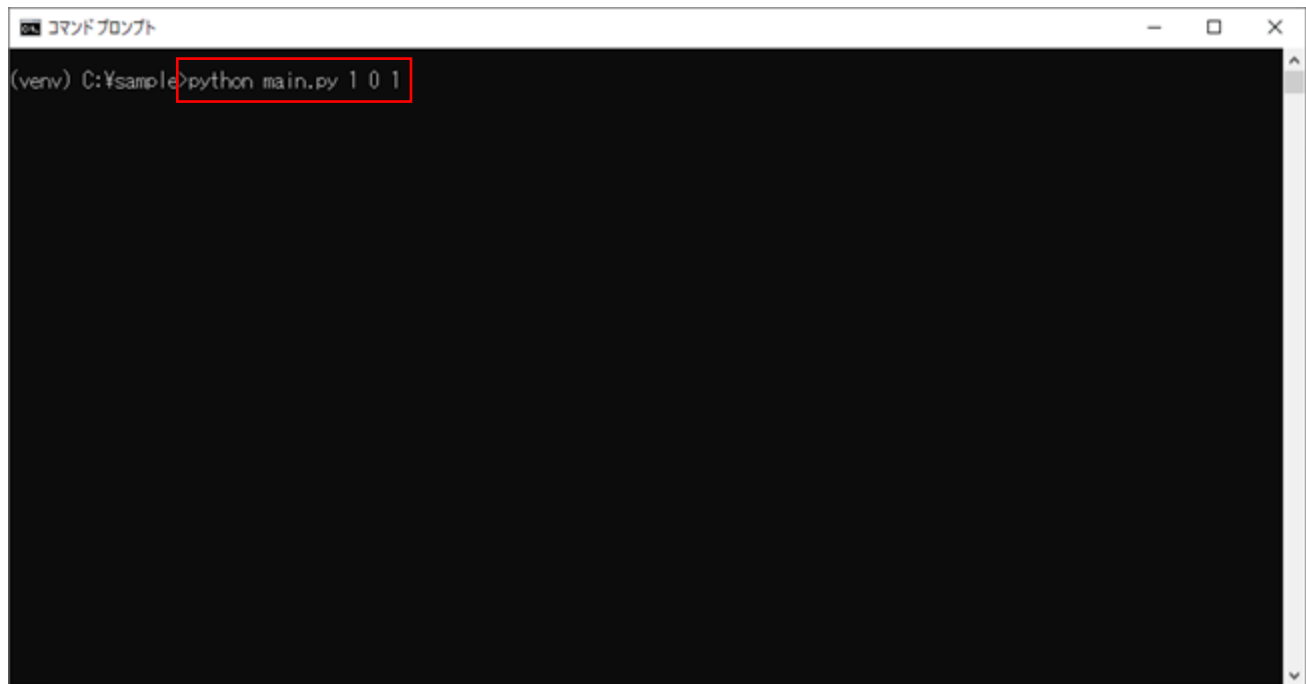
```
(venv) $ sudo venv/bin/python3 main.py
```

※ユーザーが `sudo` を実行できるように設定されていること。

## 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 ユニットを制御	LED 色と LED パターンを指定して点灯、パターン点灯させる
複数の LED ユニットを制御	複数の LED 色と LED パターンを指定してパターン点灯させる
ブザーパターンでブザーを制御	ブザーパターンを指定してブザー吹鳴させる
ブザーパターンと音階でブザーを制御	ブザーの音階とパターンを指定してブザー吹鳴させる
リセット	LED ユニットをすべて消灯し、ブザーを停止させる

#### 3.1.2. LED ユニットを制御

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.	コマンドライン引数	値
1	コマンド ID	1
2	LED ユニット色	赤:0 黄:1 緑:2 青:3 白:4
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 現状の設定を維持:15
--	--	---

例: python main.py 1 0 1

### 3.1.3. 複数の LED ユニットを制御

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.	コマンドライン引数	値
1	コマンド ID	2
2	赤の LED のパターン	消灯:0
3	黄の LED のパターン	点灯:1
4	緑の LED のパターン	LED パターン 1:2
5	青の LED のパターン	LED パターン 2:3
6	白の LED のパターン	LED パターン 3:4 LED パターン 4:5 現状の設定を維持:15

例: python main.py 2 1 2 3 4 5

### 3.1.4. 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.	コマンドライン引数	値
1	コマンド ID	3
2	ブザーパターン	停止:0 吹鳴(連続) :1 ブザーパターン 1:2 ブザーパターン 2:3 ブザーパターン 3:4 ブザーパターン 4:5 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:1~15

例: python main.py 3 1 15

### 3.1.5. ブザーパターンと音階でブザーを制御

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.	コマンドライン引数	値
-----	-----------	---



1	コマンド ID	4
2	ブザーパターン	停止:0 吹鳴(連続) :1 ブザーパターン 1:2 ブザーパターン 2:3 ブザーパターン 3:4 ブザーパターン 4:5 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:1~15
4	音 A のブザー音階	停止:0 A6:1 B ♭ 6:2 B6:3 C7:4 D ♭ 7:5 D7:6 E ♭ 7:7 E7:8 F7:9 G ♭ 7:10 G7:11 A ♭ 7:12 A7:13 音 A のデフォルト値:D7:14 音 B のデフォルト値:(停止) :15
5	音 B のブザー音階	

例: python main.py 4 1 15 1 13

## 3.1.6. 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.	コマンドライン引数	値
1	コマンド ID	5

例: python main.py 5