LR-USB USB Communication Sample Program (Windows Python)

Description

1. Overview

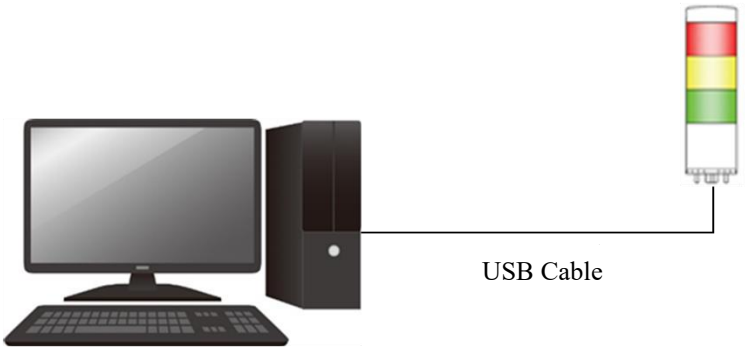
This is an outline of sample programming to control LR-USB via USB communication.

The programs are intended to control the unit using Microsoft Windows Python without the use of DLLs provided by PATLITE.

1.1. System Overview

The system configuration diagram of this program is as follows.

The sample program controls one LR-USB by USB communication.



2. Development environment

The development environment of the sample program is shown below.

2.1. Windows Requirements

Development Environment		Remarks
Development OS	Windows 10	
Development Language	Python	3.6 Or later
Package	PyUSB	1.1.1 Or later
Library	Libusb	1.0.24 Or later

2.1.1. Building an Environment

□Installing libusb

Download binaries from libusb's GitHub.

※The latest version as of 2021/07/05 is v1.0.24

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

Unzip VS2019¥MS64 ¥dll¥ libusb-1.0. dll in the compressed file and place it in C:¥Windows¥System32.

※Must have administrator status

□Creating a Virtual Environment

Create a Python virtual environment so as not to affect the system environment, and install package for USB operation.

Launch Command Prompt, navigate to the working folder, and create a virtual environment

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

Enable Virtual Environment

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

※When the virtual environment is enabled, "(venv)" is displayed at the beginning of the command prompt.

Standard Package Update

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

Install PyUSB to operate the USB device.

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

□Running the Sample Program and Exiting the Virtual Environment

Executing sample programs

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

Exiting Virtual Environment

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

Now, sample programs can be executed by enabling the virtual environment.

2.2. Linux Requirements

Development Evironment		Remarks
Development OS	Ubuntu	18.04
Development Language	Python	3.6 or later

Package	PyUSB	1.1.1 or later
Library	Libusb	1.0.21-2

2.2.1. Building an Environment

□Installing libusb

Ubunut (18.04) installs it as part of the standard package. If, for some reason, it is not installed, install it with apt-get command.

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

□Installing Packages for Python3 Virtual Environment

To create a Python3 virtual environment, install the package with apt-get command.

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

□Creating a Virtual Environment

Create a Python virtual environment so as not to affect the system environment, and install the package for USB operation.

Start the terminal, navigate to the working directory, and create a virtual environment.

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

Enabling Virtual Environment

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

※When virtualization is enabled, "(venv)" is displayed at the beginning of the screen.

Standard Package Update

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

Install USB operation package

```
(venv) $ python3 -m pip install pyusb
```

□Running the Sample Program and Exiting the Virtual Environment

Executing sample programs

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

Closing a Virtual Environment

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

Sample programs can be executed by enabling the virtual environment.

□When Access is Denied when Executing a Sample Program

If "usb. core.USBError: [Errno 13] Access denied (insufficient permissions" is displayed when a sample program is executed, it should be executed with root privilege because there is no access privilege to the USB device.

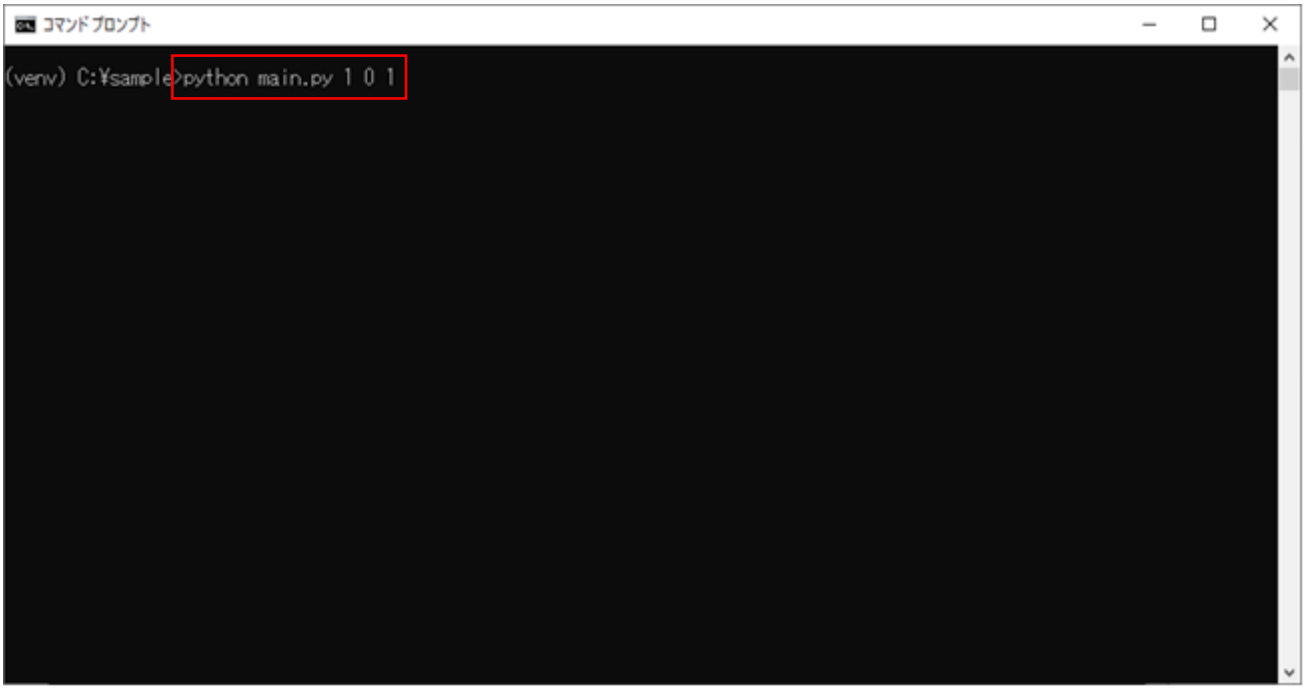
```
(venv) $ sudo venv/bin/python3 main.py
```

※User has been configured to run sudo.

3. Sample Source Overview

3.1. Command Operation

Open Command Prompt to execute commands for individual actions by specifying command line arguments.



3.1.1. Command List

Command Name	Description
Controls LED Unit	Set LED color and LED pattern to display and activate it.
Controls Several LED Units	Set multiple LED colors and LED patterns to display and activate them.
Alarm Controlled by Alarm Pattern	Set alarm pattern and activate it.
Alarm Controlled by Alarm Pattern and Scale	Set alarm scale and pattern and activate it.
Reset	Turn off all LED units and stop the alarm.

3.1.2. LED Unit Control

Execute command with the following command line arguments

No.	Command Line Argument	Value
1	Command ID	1
2	LED Unit Color	Red: 0 Amber: 1 Green: 2 Blue: 3 White: 4
3	LED Pattern	Off: 0

		Continuous On: 1 LED Pattern1: 2 LED Pattern2: 3 LED Pattern3: 4 LED Pattern4: 5 No change: 15

Example: python main.py 1 0 1

3.1.3. Control Several LED Units

Execute command with the following command line arguments

No.	Command Line Argument	Value
1	Command ID	2
2	Red LED Pattern	Off: 0 Continuous On: 1 LED Pattern1: 2 LED Pattern2: 3 LED Pattern3: 4 LED Pattern4: 5 No change: 15
3	Amber LED Pattern	
4	Green LED Pattern	
5	Blue LED Pattern	
6	White LED Pattern	

Example: python main.py 2 1 2 3 4 5

3.1.4. Alarm Controlled by Alarm Pattern

Execute command with the following command line arguments

No.	Command Line Argument	Value
1	Command ID	3
2	Alarm Pattern	Stop: 0 Sounding (Continuous): 1 Alarm Pattern 1: 2 Alarm r Pattern 2: 3 Alarm r Pattern 3: 4 Alarm Pattern 4: 5 No change: 15
3	Alarm Continuous Operation and Number of Cycles	Continuous operation: 0 Number of cycles: 1 to 15

Example: python main.py 3 1 15

3.1.5. Alarm Control by Alarm Pattern and Scale

Execute command with the following command line arguments

No.	Command Line Argument	Value

1	Command ID	4
2	Alarm Pattern	Stop: 0 Sounding (Continuous): 1 Alarm Pattern 1: 2 Alarm r Pattern 2: 3 Alarm r Pattern 3: 4 Alarm Pattern 4: 5 No change: 15
3	Alarm Continuous Operation and Number of Cycles	Continuous operation: 0 Number of cycles: 1 to 15
4	Sound A Alarm Scale	Stop: 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 Default value of sound A: D7: 14 Default value of sound B: (Stop): 15
5	Sound B Alarm Scale	

Example: python main.py 4 1 15 1 13

3.1.6. Reset

Execute command with the following command line arguments

No.	Command Line Argument	Value
1	Command ID	5

Example: python main.py 5