NE-USB USB Communication
Sample Program
(Windows Python)

Contents

1. Overview

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

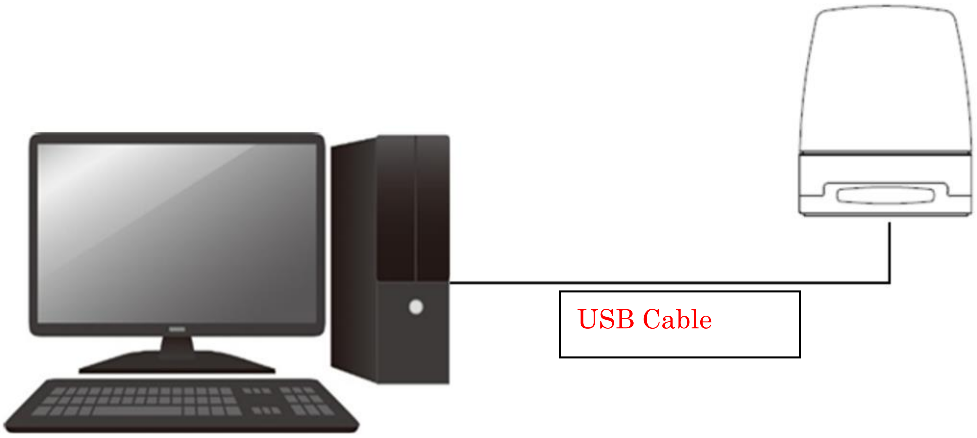
The programs are intended to control the unit using Python without the use of DLLs provided by PATLITE

This program is only a sample and additional design for abnormalities are necessary.

1.1. System Overview

The system configuration diagram of this program is as follows.

This program controls one NE-USB by USB Communication.



2. Development Environment

The development environment of the sample program is shown below.

2.1. Windows Evironment

Development Environment		Remarks
Development OS	Windows10	
Development Language	Python	3.10 or later
Package	PyUSB	1.2.1 or later
Library	libusb	1.2.1 or later

2.1.1. Environment

・Installation of libusb

Download libusb binaries from GitHub.

*As of 2022/03/30, the current version is v 1.0.2 5

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

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

* Administrator privileges are required.

・ Creating a virtual environment

Python virtual environment so that it does not affect the system environment, and install the package for USB operation.

Start a command prompt, move to the working folder, and then 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 the package (PyUSB) for USB operation

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

・Execution of the sample program and termination of the virtual environment

Execution of sample program

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

End of virtual environment

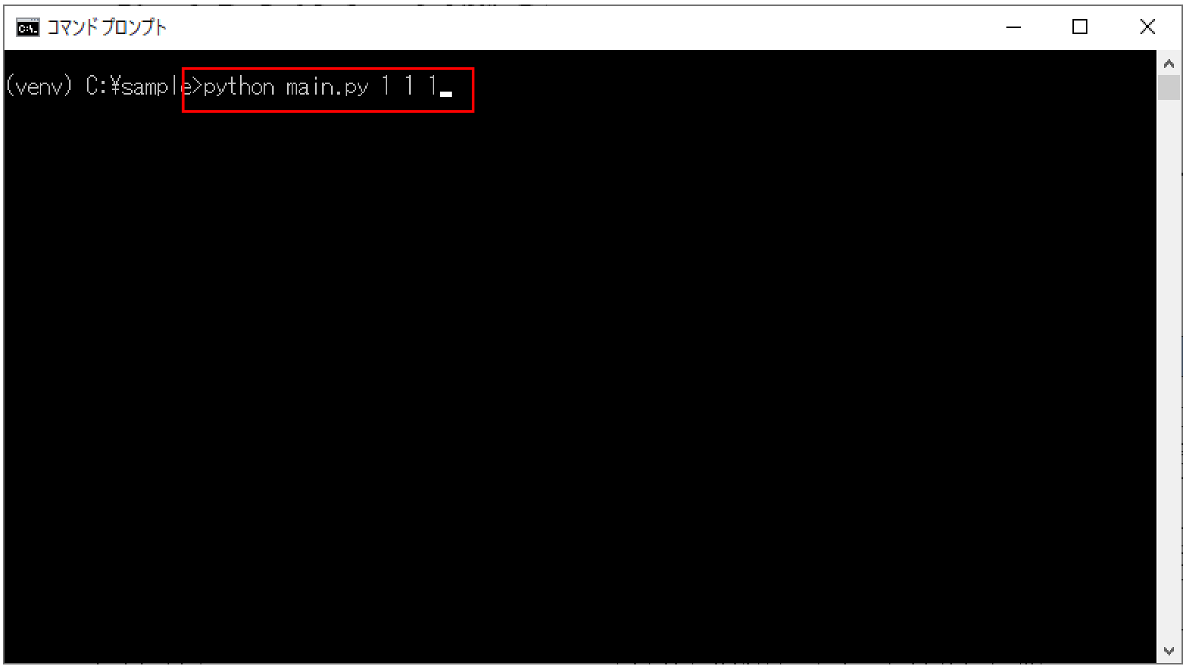
```
(venv) > venv¥Scripts¥deactivate.bat
 >
```

After that, you can execute the sample program by performing “Enable virtual environment”.

3. Sample Source Overview

3.1. Command Operation

At the Command Prompt, specify the command line arguments to execute commands for each operation.



```
(venv) C:¥sample>python main.py 1 1 1
```

3.1.1. Command List

Command Name	Description
LED Control	Set LED color and LED pattern to display and activate it.
Control Alarm Pattern	Set the alarm pattern and the number of cycles.
Control Alarm Volume	Set alarm volume and activate it.
Control Alarm Pattern and Volume	Set alarm pattern, number of times, and volume and activate it.
Connection Display Settings	Change the display settings when connecting.
Acquire input status of Touch sensor (Only for NE-ST-USB/NE-WT-USB)	Message display the input status of the touch sensor.
Reset	Turn off all LED units and stop the alarm.

3.1.2. LED Control

Execute the command with the following command line arguments.

No.	Command Line Arguments	Value
1	Command Identifier	1
2	LED Color	Off: 0 Red: 1 Green: 2 Amber: 3 Blue: 4 Purple: 5 Sky Blue: 6 White: 7 No change: 15
3	LED Pattern	Off: 0 Lit: 1 LED pattern 1: 2 LED pattern 2: 3 LED pattern 3: 4 LED pattern 4: 5 LED pattern 5: 6 LED pattern 6: 7 No change: 15

Example：python main.py 1 1 1

3.1.3. Control Alarm Pattern

Execute the command with the following command line arguments.

No .	Command Line Arguments	Value
1	Command Identifier	2
2	Alarm Pattern	Stop: 0 Sounding (Continuous): 1 Sweep sound: 2 Intermittent sound: 3 Weak caution sound: 4 Strong caution sound: 5 Twinkle, Twinkle Little Star s: 6 London Bridge: 7 No change: 15
3	Alarm Continuous Operation and Number of Cycles	Continuous operation: 0 Number of cycles: 1 to 1 4 No change: 15

Example：python main.py 2 1 1

3.1.4. Control Alarm Volume

Execute the command with the following command line arguments.

No .	Command Line Arguments	Value
1	Command Identifier	3
2	Alarm Volume	Mute: 0 Volume: 1-9 Maximum volume: 10 No change: 15

Example：python main.py 3 1

3.1.5. Control Alarm Pattern and Volume

Execute the command with the following command line arguments.

No.	Command Line Arguments	Value
1	Command Identifier	4
2	Alarm Pattern	Stop: 0 Sounding (Continuous): 1 Sweep sound: 2 Intermittent sound: 3 Weak caution sound: 4 Strong caution sound: 5 Twinkle, Twinkle Little Star: 6 London Bridge: 7 No change: 15
3	Alarm Continuous Operation and Number of Cycles	Continuous operation: 0 Number of cycles: 1 to 1 4 No change: 15
4	Alarm Volume	Mute: 0 Volume: 1-9 Maximum volume: 10 No change: 15

Example：python main.py 4 1 3 5

3.1.6. Connection display settings

Execute the command with the following command line arguments.

No.	Command Line Arguments	Value
1	Command Identifier	5
2	Connection Display Settings	OFF：0 ON：1

Example：python main.py 5 0

3.1.7. Acquire input status of Touch sensor（only for NE-ST-USB/NE-WT-USB）

Execute the command with the following command line arguments.

No.	Command Line Arguments	Value
1	Command Identifier	6

Example：python main.py 6

Output the status of Command Prompt.

・When touch sensor input status is OFF：touch sensor input = OFF

・When touch sensor input status is ON：touch sensor input = ON

3.1.8. Reset

Execute the command with the following command line arguments.

No.	Command Line Arguments	Value
1	Command Identifier	7

Example：python main.py 7