LR-USB USB Communication Sample Program (Linux C++)

Description

1. Overview

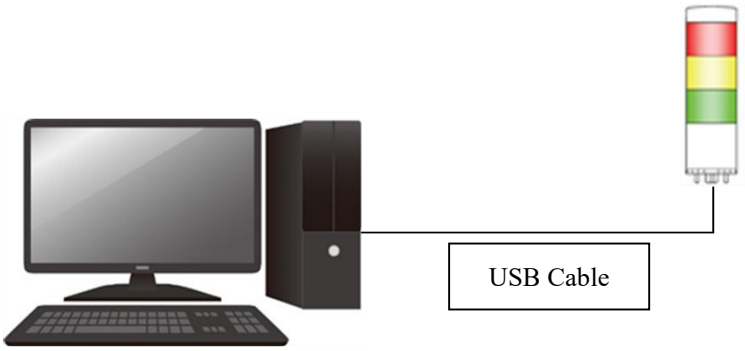
This is an outline of sample programming to control LR-USB via USB communication.

The programs are intended to control the unit using Linux C++ 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. Linux Requirements

Development Environment		Remarks
Development OS	Ubuntu	18.04
Development Language	C++	
App Type	CUI application	
Development Tools	g++	7.5.0

2.1.1. Building an Environment

・Installing libusb

It is installed as a standard-package in Ubunut (18.04). 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
```

・Compiling Sample Programs

Use Makefile in the project folder of the sample program to perform compilation with Make command.

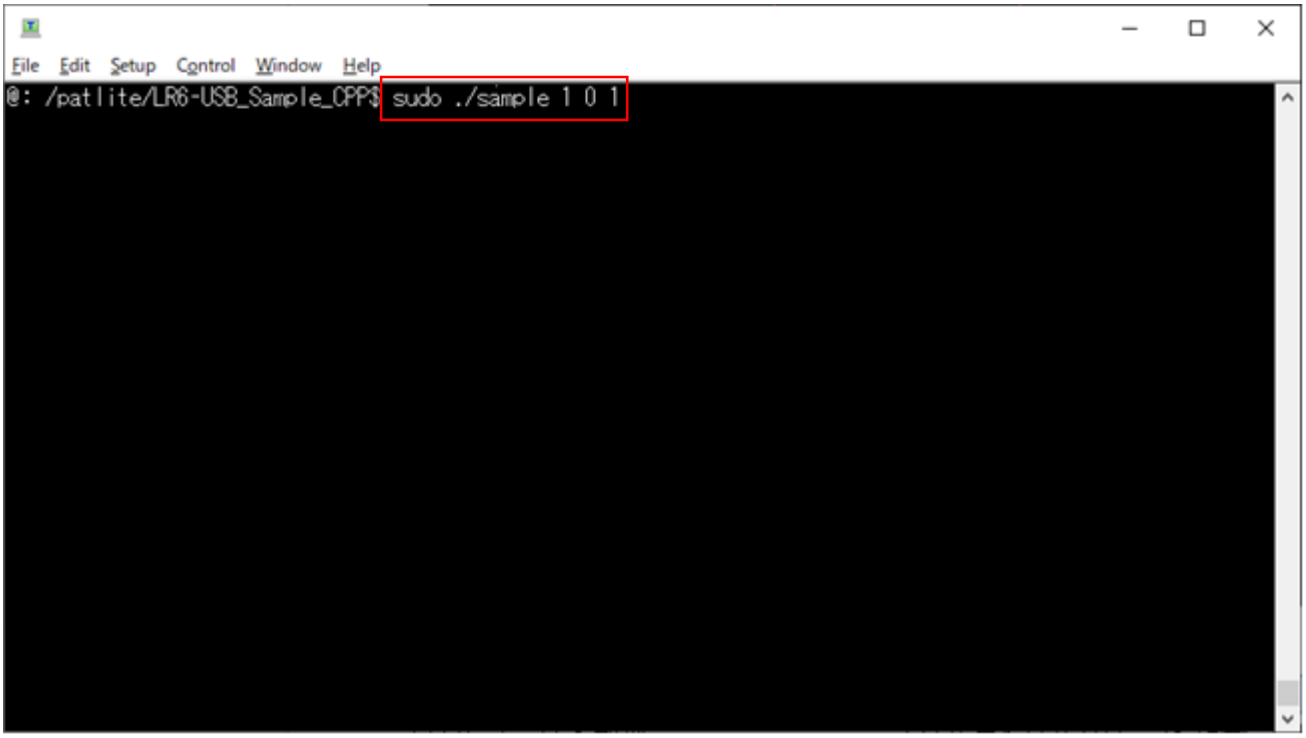
If compilation is successful, a sample is created.

```
$ make
g++ main.cpp -o sample `pkg-config libusb-1.0 --libs`
$ ls
$ Makefile main.cpp sample
```

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: sudo. /sample 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: sudo. /sample 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: sudo. /sample 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 :sudo. /sample 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: sudo. /sample 5