last update 6-28-17

libLL_USB30x.so and sample apps coded in C (gcc compiler) and python for Linux - overview

Library "libLL_USB30x.so" built and tested in Oracle VM VirtualBox, within Win7, running ubuntu 14.04 LTS. Built with installed library "**lusb1.0**" and library, "**pthread 1.1**."

The library makes use of many of the functions mentioned in the documentation for our Windows DLL API which can be found at our website at:

lawsonlabs.com web link

The functions currently available can be found in the "main.h" header file distributed with the library code, and recognized by the "EX_" preceding the rest of the function name. There are also many other functions within the library which wrap, extend, and are called by those functions, but can also be called by themselves. All are shown within the "main.h" header file and can be included, as can any of the variables, by simply declaring them as "extern" within your C application code, or the usual way for python type access as shown in the sample python code. For example when coded in C for a function:

```
extern BOOLEAN EX_ConnectAllDevices ( BYTE* pbNumDevices, USHORT* pDevList, double* pdblRates );
```

or for a variable:

```
extern SCAN_STRUCT scanStruct;
```

There is an example app that makes use of most of the functionality within the library, coded in C, one much simpler also coded in C, and a simple app coded in python. The first requires the Linux "**ncurses**" library to be installed since it makes extensive use of it, while the second C app and python app do not since they simply perform some basic calls into our 30x USB library and prints the results directly to the console screen with no user interaction. The simple apps are hard-coded at the top of function "main" to make use of our Model 302 hardware with device ID 5206. The bigger app accepts any device ID, prompted by a menu-available option. Both apps will almost certainly need to be recompiled within your environment anyhow, unless it matches nearly exactly mine, described higher up.

The library code, as well as each of the sample C apps, contain build commands shown at the top of their respective "header" files. The command line compilation strings for the library are self-explanatory as to where the library is to be placed. The python sample, has no header and is executed in the typical way at the command line.