last update 9-19-17

LL_USB_multi_python Win sample app for Windows – overview

Sample app "LL_USB_multi_python.exe" makes use of our DLL "LL_USB2k.dll" and driver "LL_USB2k.sys" built for WinXP/2k/Vista/7/8/10. The DLL and driver for your system can be found at:

install file download

Install driver as instructed in the documentation accompanying the download, and place the DLL in the same folder as the LL_USB_multi_python.exe application.

The sample makes use of many of the functions mentioned in the documentation for our Windows DLL API which can be found at our website. The link below provides the API as well as much more information that further explains using the functions, and some information concerning interaction with the hardware. Below is the link:

lawsonlabs.com web link

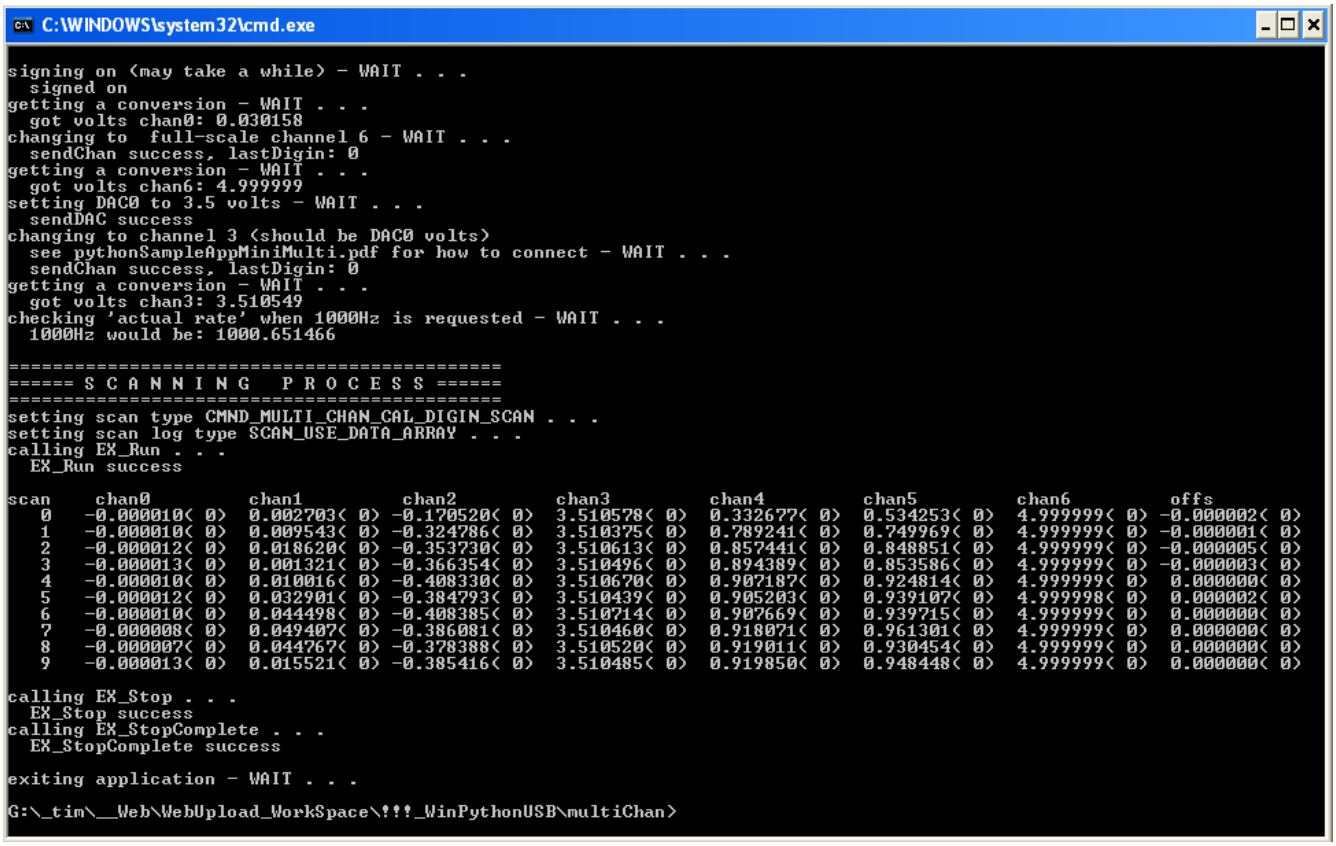
Sample app LL_USB_python.exe makes use of only a handful of the functions available, in order to demonstrate their usage in connecting to the device and performing some basic tasks. The app connects to the device, reads/displays the voltage from the default channel 0, sets channel 6, and then reads/displays the voltage again. Next the app sets DAC 0 to 3.5 volts and changes to channel 3. Please connect DAC 0 to channel 3 (analog out 1 to 4+ and GD to 4-). The app reads the voltage from that channel (should read approx. 3.5 volts). It then demonstrates how to get the "actual rate" the board will run at based on a requested rate, since the board is not capable of always running at the precise rate one might request. It sets a temporary rate variable to 1000Hz and then calls into the library to get what the "actual board rate" would be (approx. 1000.651466Hz) at that requested rate. Note that trying to set the rate, for example with a call to **EX_SendRate(...)** with the rate that was returned by that call, could then set it to a different rate. That returned rate should only be used within an app for timing within the app. Finally, the app does a multi-channel-digin-cal scan with the board running at 100Hz. Since the board automatically initializes at 100Hz, that did not need to be changed. The scan data rate for this scan is about 2.5Hz. For an explanation of how this is calculated, read the first FAQ at ***this link***. The application does 10 scans and then exits in order to keep the entire sample application output on one screen for this documentation.

It makes use of the following function calls within the DLL:

EX_ConnectOneDevice(...)
EX_GetOneConversion(...)
EX_SendChan(...)
EX_SendDAC(...)
EX_GetCalculatedRate(...)
EX_SetScanType(...)
EX_SetDataLogOptions(...)
EX_Run(...)
EX_CheckScanStatus(...)
EX_GetScanDataDbl(...)
EX_Stop(...)
EX_StopComplete(...)

The device ID is hard-coded to 5206 at the top of the "main()" function call, so that will need to be changed to "your" device ID before running your the sample app.

The next page shows how it looks when run. Note that the scan data for channel 3 is the approximate 3.5 volts that was sent to DAC0 to which that channel is connected. Channel 6 is 'full-scale' , 0 is connected to ground, and the others are not connected to anything. Only ten scans are read in order to keep the example (and displayed ouput) simple.


C:\WINDOWS\system32\cmd.exe
signing on (may take a while) - WAIT . . .
signed on
getting a conversion - WAIT . . .
got volts chan0: 0.030158
changing to full-scale channel 6 - WAIT . . .
sendChan success, lastDigin: 0
getting a conversion - WAIT . . .
got volts chan6: 4.999999
setting DAC0 to 3.5 volts - WAIT . . .
sendDAC success
changing to channel 3 (should be DAC0 volts)
see pythonSampleAppMiniMulti.pdf for how to connect - WAIT . . .
sendChan success, lastDigin: 0
getting a conversion - WAIT . . .
got volts chan3: 3.510549
checking 'actual rate' when 1000Hz is requested - WAIT . . .
1000Hz would be: 1000.651466
====== S C A N N I N G P R O C E S S ======
setting scan type CMND_MULTI_CHAN_CAL_DIGIN_SCAN . . .
setting scan log type SCAN_USE_DATA_ARRAY . . .
calling EX_Run . . .
EX_Run success
scan chan0 chan1 chan2 chan3 chan4 chan5 chan6 offs
0 -0.000010(0) 0.002703(0) -0.170520(0) 3.510578(0) 0.332677(0) 0.534253(0) 4.999999(0) -0.000002(0)
1 -0.000010(0) 0.009543(0) -0.324786(0) 3.510375(0) 0.789241(0) 0.749969(0) 4.999999(0) -0.000001(0)
2 -0.000012(0) 0.018620(0) -0.353730(0) 3.510613(0) 0.857441(0) 0.848851(0) 4.999999(0) -0.000005(0)
3 -0.000013(0) 0.001321(0) -0.366354(0) 3.510496(0) 0.894389(0) 0.853586(0) 4.999999(0) -0.000003(0)
4 -0.000010(0) 0.010016(0) -0.408330(0) 3.510670(0) 0.907187(0) 0.924814(0) 4.999999(0) 0.000000(0)
5 -0.000012(0) 0.032901(0) -0.384793(0) 3.510439(0) 0.905203(0) 0.939107(0) 4.999998(0) 0.000002(0)
6 -0.000010(0) 0.044498(0) -0.408385(0) 3.510714(0) 0.907669(0) 0.939715(0) 4.999999(0) 0.000000(0)
7 -0.000008(0) 0.049407(0) -0.386081(0) 3.510460(0) 0.918071(0) 0.961301(0) 4.999999(0) 0.000000(0)
8 -0.000007(0) 0.044767(0) -0.378388(0) 3.510520(0) 0.919011(0) 0.930454(0) 4.999999(0) 0.000000(0)
9 -0.000013(0) 0.015521(0) -0.385416(0) 3.510485(0) 0.919850(0) 0.948448(0) 4.999999(0) 0.000000(0)
calling EX_Stop . . .
EX_Stop success
calling EX_StopComplete . . .
EX_StopComplete success
exiting application - WAIT . . .
G:_tim__Web\WebUpload_WorkSpace\!!!_WinPythonUSB\multiChan>