Plexon Inc

Matlab Offline Files SDK

8/15/2013

Revision History (most recent last):			
DATE	REVISION	DESCRIPTION	ORIGINATOR
5/30/2013	1.0.0	Initial Version	Harvey Wiggins
6/20/2013	1.1.0	Added Read Block functionality	Al Bruns
8/15/2013	1.2.0	Added Note about Visual Studio 2005 C++ Library requirement	Al Bruns

Plexon Inc.
6500 Greenville Ave. # 700 LB33
Dallas TX 75206

Tel: 214 369 4957
Fax: 214 369 1775
www.plexon.com

Plexon Data File Import for Matlab

This package allows importing the data from Plexon .plx, pl2 or .ddt files into the Matlab workspace. It also allows writing continuous data from Matlab into a .ddt file.

The package contains:

1. Matlab Executable files (mexPlex.mexw32 and mexPlex.mexw64) that can be used in Windows to read .plx and .ddt files.
2. Compiled Matlab function files (.p files) to read .pl2 files. These files can be used in all operating systems supported by Matlab.
3. Matlab function files (.m files) that call mexPlex executable files (to read .plx and .ddt files) or call .p files (to read .pl2 files).
4. C++ source code that is used to compile mexPlex files in Windows and Linux.

Note: Matlab Executable files, the .m and .p files need to be installed in a path known to Matlab.

Basic Functions

The following routines provide basic functionality to read plx and ddt files.
See the .m files for usage and additional notes.

plx_info – Reads count information about a plx file.
plx_information - Reads extended header information from a .plx file
plx_chan_filters - Reads channel filter settings for each spike channel
plx_chan_gains - Reads channel gains for each channel from a .plx file
plx_chan_thresholds - Reads channel thresholds from a .plx file
plx_chan_names - Reads spike channel names from a .plx file
plx_chanmap – Returns DSP channel number for each spike channel

plx_ts– Reads spike timestamps for the specified DSP channel and unit
plx_waves- Reads waveform data for the specified DSP channel and unit.
plx_waves_v - Reads waveform data for the specified DSP channel and unit. Waveform data are represented in millivolts.

plx_vt_interpret - Takes output of **plx_event_ts** and produces an array of coordinates.

plx_adchan_freqs - Reads the per-channel frequencies for analog channels
plx_adchan_gains - Reads analog channel gains from a .plx file
plx_adchan_names - Reads analog channel names from a .plx file
plx_adchan_samplecounts - Reads analog channel sample counts for .PLX file.
plx_ad_chanmap – Returns raw channel number for each analog channel

plx_ad – Reads A/D data for the specified A/D channel

plx_ad_span - Reads a span of the A/D data for the specified A/D channel
plx_ad_v - Reads A/D data for the specified A/D channel. A/D data are represented in millivolts.
plx_ad_span_v - Reads a span of the A/D data for the specified A/D channel. A/D data are represented in millivolts.
plx_ad_gap_info – Similar to **plx_ad**/**plx_ad_v** but returns only general information without A/D data.

plx_event_names - Reads event type names from a .plx file
plx_event_ts- Reads event data for the specified external event
plx_event_chanmap – Returns raw channel number for each event channel

plx_close - Closes any open .plx files

ddt - Reads data from a .ddt file..
ddt_v - Reads data from a .ddt file. A/D data are represented in millivolts.
ddt_write_v - Writes data to .ddt file. A/D data are represented in millivolts.

plx_mexplex_version - Returns the version of mexPlex library

How to Read Data from PL2 Files

All existing **plx_** routines work with pl2 files. For example,

```
[adfreq, n, ts, fn, ad] = plx_ad_v('C:\PlexonData\test1.pl2', 0);
```

will return the data for the first analog channel of pl2 file C:\PlexonData\test1.pl2. This means that you can use your existing Matlab scripts to process both plx and pl2 files.

Please note that you can now also use channel names in all the routines that return channel data:

```
[adfreq, n, ts, fn, ad] = plx_ad_v(filename,  'FP01' );
```

You can also use PL2-specific routines to read data from pl2 files. See the .m files for usage and additional notes.

PL2GetFileIndex – Reads file headers and other general info from a .pl2 file.
PL2Ad – Reads A/D data for the specified A/D channel.
PL2AdBySource – Reads A/D data for the specified A/D channel. The channel is specified using its source id.
PL2AdSpan – Reads a span of A/D data for the specified A/D channel.
PL2AdBySource – Reads a span of A/D data for the specified A/D channel.
PL2AdTimeSpan – Reads a time span of A/D data for the specified A/D channel.

PL2AdTimeSpanBySource – Reads a time span of A/D data for the specified A/D channel.
PL2EventTs – Reads event data for the specified external event.
PL2EventTsBySource – Reads event data for the specified external event.
PL2StartStopTs – Reads event data for the start/stop external event.
PL2Ts – Reads spike timestamps for the specified spike channel.
PL2TsBySource – Reads spike timestamps for the specified spike channel.
PL2Waves – Reads spike timestamps and waveforms for the specified spike channel.
PL2WavesBySource – Reads spike timestamps and waveforms for the specified spike channel.
PL2ReadFirstDataBlock – Reads the first data block in the file.
PL2ReadNextDataBlock – Reads the next data block in the file.

Notes:

1. The original versions of the MATLAB file import routines were written as pure .m scripts, with no underlying DLL. The DLL-based approach is thus several hundred times faster than the previous .m script-based approach. These old .m scripts are included into this distribution as the old_scripts.zip file. These old .m files may be useful for instructive purposes or for those running MATLAB on non-Windows platforms. Otherwise, there is no reason to unzip the file. Note that these old .m files are no longer maintained, and new features and bug fixes will be not be added to them. Also note: when reading .plx or .ddt files into MATLAB on non-Windows platforms, be aware that there may be a byte-ordering issue. Plexon files are written with little-endian byte ordering, and MATLAB by default expects binary files with the system native byte-ordering, which may be big-endian on Mac platforms.
2. Conceptually, channels in .plx data files will be ‘collapsed’ into a contiguous block of n channels, regardless of their original numbering, and n is reported back as a return value in some calls. All access to the channel’s data through the API is via its new, ‘collapsed’ index 1..n. If there are gaps in the channel numbers in the original .plx file, the channels names returned from plx_chan_names (which have also been collapsed) can be used to identify the channel.
3. When running these functions on a Windows PCs, if you get an error message including text similar to: `The application has failed to start because its side-by-side configuration is incorrect`, a likely cause is that the Microsoft Visual C++ 2005 SP1 run time redistributable library is not installed on the computer. If you are using a 32-bit version of Windows, download and install:
   http://www.microsoft.com/en-us/download/details.aspx?id=5638
   On an x64 PC, the appropriate package to install is here:
   http://www.microsoft.com/en-us/download/details.aspx?id=18471
   Other packages may be appropriate for your particular Operating System and computer.