Vaunix LSG 64 Bit SDK Release Notes

The Vaunix LSG 64 bit DLL uses the same general architecture as the LSG 32 bit DLL, with the same functions names, although the DLL file name is different, and several new functions have been added to the DLL. In addition, please note that the documentation for fnLSG_GetPowerLevel in the original LSG DLL manual is incorrect, see below for a better description and the new function fnLSG_GetPowerLevelAbs.

The Windows 64 bit DLL for the LSG signal generators is named vnx_fsynth.dll, and its corresponding library file is vnx_fsynth.lib. The DLL provides ANSI-C style entry points, and is intended to operate on 64 bit AMD or Intel platforms running Microsoft's Windows 7 or later. The DLL does require the presence of the Visual Studio 2013 redistributables, which are already present on most versions of Windows. If they are not present on your system, you can download and install them from Microsoft's web site.

You can use the DLL from applications written in any programming language that supports calls to native code Windows DLLs. The Vaunix LSG 64 bit DLL can be called from managed code applications, like those generated by Microsoft's compilers that output .net code using the language features provided by Microsoft to call native code DLLs. The LSG 64 bit DLL can also be used from programming environments such as Matlab or LabView, or from languages such as Python or TcL.

New Functions

Note - function names are prefixed with a different macro in the 64 bit version of the DLL, so function declarations differ from those listed in the 32 bit version's documentation. See the vnx_LSG_api.h include file for the function declarations. The names of the functions are unchanged from the 32 bit version of the DLL to simplify porting of applications between 32 bit and 64 bit environments

VNX_LSG_API int fnLSG_GetDLLVersion();

This function returns the version of the DLL, encoded into a 32 bit integer where the top 16 bits are flags representing features, which are unused and reserved as of now. The lower 16 bits represent the version in Major.Minor format. Thus version 1.0 is represented by 0x00000100, version 1.1 would be represented by 0x00000101.

VNX_LSG_API int fnLSG_GetModelNameA(DEVID deviceID, char *ModelName);
VNX_LSG_API int fnLSG_GetModelNameW(DEVID deviceID, wchar_t *ModelName);

Both ANSI and Unicode versions of the functions are provided with names that are consistent with Microsoft's system for managing character encoding in applications. In applications being

built in Visual Studio, the development environment will select the correct function based on the project settings. Use the original name, fnLSG_GetModelName in these development environments.

VNX_LSG_API int fnLSG_GetPowerLevel(DEVID deviceID);

This function returns the relative power setting, based on the maximum power output of the LSG device in .25 db units. Thus, for an LSG signal generator with a +10 dBm output, set to +10 dBm, this function would return 0. For the same signal generator set to +5 dBm, this function would return 20.

VNX_LSG_API int fnLSG_GetPowerLevelAbs(DEVID deviceID);

This function returns the absolute output power setting of the LSG device in .25 db units. Thus, for an LSG signal generator with a +10 dBm output, set to +10 dBm, this function would return 40. For the same signal generator set to -20 dBm, this function would return -80.