Vaunix Technology Corporation

Lab Brick® Family of Digital Attenuators

Ethernet Based API User Manual

Revision A1

NOTICE

Vaunix has prepared this manual for use by Vaunix Company personnel and customers as a guide for the customized programming of Lab Brick products. The drawings, specifications, and information contained herein are the property of Vaunix Technology Corporation, and any unauthorized use or disclosure of these drawings, specifications, and information is prohibited; they shall not be reproduced, copied, or used in whole or in part as the basis for manufacture or sale of the equipment or software programs without the prior written consent of Vaunix Technology Corporation.

Table of Contents

1.0 OVERVIEW

The LabBrick Programmable Attenuator Win32 SDK supports developers who want to control LabBrick programmable attenuators from Windows programs, or who want to control the attenuators from LabVIEW[1] or other National Instruments programming environments. The SDK includes a dll which provides a Win32 API to find, initialize, and control the attenuators, along with header files and an example Win32 C program which demonstrates the use of the API.

2.0 USING THE SDK

The SDK consists of a dll, named VNX_Eth_Attn64.dll, along with this documentation, a C style header file, a library file for linking to the dll, and a VC 6 example program. Unzip the SDK into a convenient place on your hard disk, and then copy the dll and library file into the directory of the executable program you are creating. Add the header file (ldadrvr.h) to your project and include it with the other header files in your program. Make sure that the linker directives include the path of the library file.

3.0 PROGRAMMING

3.1 Overall Strategy and API Architecture

The API provides functions for identifying how many and what type of LabBrick attenuators are connected to the system, initializing attenuators so that you can send them commands and read their state, functions to control the operation of the attenuators, and finally a function to close the software connection to the attenuator when you no longer need to communicate with it.

The API can be operated in a test mode, where the functions will simulate normal operation but will not actually communicate with the hardware devices. This feature is provided as a convenience to software developers who may not have a LabBrick attenuator with them, but still want to be able to work on an applications program that uses the LabBrick. Of course it is important to make sure that the API is in its normal mode in order to access the actual hardware!

Be sure to call fnLDA_SetTestMode(FALSE), unless of course you want the API to operate in its test mode.

3.2 Status Codes

All of the set functions return a status code indicating whether an error occurred.
The values of the status error are defined in the ldadrvr.h header file.

3.3 Functions – Selecting the Device

VNX_ATTEN_API void fnLDA_SetTestMode(bool testmode)

Set testmode to FALSE for normal operation. If testmode is TRUE the dll does not communicate with the actual hardware, but simulates the basic operation of the dll functions. It does not simulate the operation of attenuation ramps generated by the actual hardware, but it does simulate the behavior of the functions used to set the parameters for the ramps.

VNX_ATTEN_API void fnLDA_Init(void)
This function will be used to initialize LDA device data structures.

VNX_ATTEN_ LDASTATUS fnLDA_InitDevice(char* deviceip)

This function is used to open the device interface socket connection over ethernet to the attenuator and initialize the dll's copy of the device's settings. If the fnLDA_InitDevice function succeeds, then you can use the various fnLDA_Get* functions to read the attenuator's settings. This function will fail, and return an error status if the attenuator has already been opened by another program.

VNX_ATTEN_API LDASTATUS fnLDA_CloseDevice(char* deviceip)

This function closes the device socket interface to the attenuator. It should be called when your program is done using the attenuator.

VNX_ATTEN_API LDASTATUS fnLDA_CheckDeviceReady(char* deviceip)

This function will be used to check whether device is ready to get/set the parameters of the LDA attenuator device.

3.4 Functions – Setting parameters on the Attenuator

VNX_ATTEN_API LDASTATUS fnLDA_SetWorkingFrequency(char* deviceip, int frequency)

This function is used to set the current operation frequency of the device. Set frequency value should be in 100kHz resolution.

VNX_ATTEN_API LDASTATUS fnLDA_SetChannel(char* deviceip, int channel)

This function is used the set the current channel of the device.

VNX_ATTEN_API LDASTATUS fnLDA_SetAttenuation(char* deviceip, int attenuation)

This function is used to set the attenuation level of the programmable attenuator. The attenuation setting is encoded as an integer where each increment represents .05db(Factor 20) of attenuation. The encoding is:

attenuation * 20(.05db) = Attenuation in db

For example, attenuation = 200 for 10db of attenuation, 2000 for 100db of attenuation.

VNX_ATTEN_API LDASTATUS fnLDA_SetRampStart(char* deviceip, int rampstart)

This function sets the attenuation level at the beginning of an attenuation ramp or sweep. The encoding of rampstart, the attenuation level, is the same as the fnLDA_SetAttenuation function.

VNX_ATTEN_API LDASTATUS fnLDA_SetRampEnd(char* deviceip, int rampstop)

This function sets the attenuation level at the end of an attenuation ramp or sweep. The encoding of rampstop, the attenuation level, is the same as the fnLDA_SetAttenuation function.

VNX_ATTEN_API LDASTATUS fnLDA_SetAttenuationStep(char* deviceip, int attenuationstep)

This function sets the size of the attenuation step that will be used to generate the attenuation ramp or sweep. The encoding of attenuationstep, is the same as the fnLDA_SetAttenuation function. The smallest attenuation step size is 2 or .5 db.

VNX_ATTEN_API LDASTATUS fnLDA_SetDwellTime(char* deviceip, int dwelltime)

This function sets the length of time that the attenuator will dwell on each attenuation step while it is generating the attenuation ramp. The dwelltime variable is encoded as the number of milliseconds to dwell at each level. The resolution is in milliseconds.

For example dwelltime = 900 for 90msec

VNX_ATTEN_API LDASTATUS fnLDA_SetIdleTime(char* deviceip, int idletime);

This function sets the length of time that the attenuator will wait at the end of an attenuation ramp before beginning the ramp again when the ramp mode is set to SWP_REPEAT. The idletime variable is encoded as the number of milliseconds to dwell at each level. . The resolution is in milliseconds.

VNX_ATTEN_API LDASTATUS fnLDA_SetHoldTime (char* deviceip, int holdtime);

This function sets the time delay between the first and second phases of a ramp.
The holdtime variable is encoded as the number of milliseconds to hold. The resolution is in milliseconds.

VNX_ATTEN_API LDASTATUS fnLDA_SetRampDirection(char* deviceip, bool up)

This function is used to set the direction of the attenuation ramp. To create a ramp with increasing attenuation, set up = TRUE. Note that the ramp start attenuation value must be less than the ramp end attenuation value for a ramp with increasing attenuation. For a ramp with decreasing attenuation the ramp start value must be greater than the ramp end value.

VNX_ATTEN_API LDASTATUS fnLDA_SetRampMode(char* deviceip, bool mode)

This function is used to select either a single ramp or sweep of attenuation values, or a repeating series of ramps. If mode = TRUE then the ramp will be repeated, if mode = FALSE the ramp will only happen once.

VNX_ATTEN_API LDASTATUS fnLDA_StartRamp(char* deviceip, bool go)
This function is used to start and stop the attenuation ramps. If go = TRUE the attenuator will begin sweeping, FALSE stops the sweep.

VNX_ATTEN_API LDASTATUS fnLDA_SetRampBidirectional(char* deviceip, bool bidir_enable)
This function is used set the ramp directional mode. If bidir_enable = TRUE the Bi-directional mode sweep will be turned on, FALSE stops the bidirectional sweep.

VNX_ATTEN_API LDASTATUS fnLDA_SaveSettings(char* deviceip)

The LabBrick attenuators can save their settings, and then resume operating with the saved settings when they are powered up. Set the desired parameters, then use this function to save the settings.

VNX_ATTEN_API LDASTATUS fnLDA_SetAttenuationStepTwo(char* deviceip, int attenuationstep2)
This function sets the step size for the second phase of an attenuation ramp.

VNX_ATTEN_API LDASTATUS fnLDA_SetDwellTimeTwo(char* deviceip, int dwelltime2)

This function sets the dwell time for each step on the second phase of an attenuation ramp.

VNX_ATTEN_ LDASTATUS fnLDA_SetProfileElement(char* deviceip, int index, int attenuation)

This function sets the value of a profile element. The index runs from zero to the maximum profile length minus 1. PROFILE_MAX is currently 100. The attenuation value is encoded in .05db steps.

VNX_ATTEN_API LDASTATUS fnLDA_SetProfileCount(char* deviceip, int profilecount)

This function sets the number of elements in the profile that will be used. It must be greater than zero and less than PROFILE_MAX, the maximum profile length.

VNX_ATTEN_API LDASTATUS fnLDA_SetProfileIdleTime(char* deviceip, int idletime)

This function sets the idle time after a profile is played before the profile is played again in repeating profile mode.

VNX_ATTEN_API LDASTATUS fnLDA_SetProfileDwellTime(char* deviceip, int dwelltime)

This function sets the time duration of each element in the profile during playback. The dwelltime is specified in milliseconds.

VNX_ATTEN_API LDASTATUS fnLDA_StartProfile(char* deviceip, int mode)[1]

This function starts the playback of a profile. A mode value of 1 plays the profile once, a mode value of 2 plays the profile repeatedly.

3.5 Functions – Reading parameters from the Attenuator

All Get function calls takes two arguments one pointer pointing to the device ip string and other is the response data pointer.

VNX_ATTEN_API LDASTATUS fnLDA_GetModelName(char* deviceip, char *respdata)

This function is used to get the model name of the attenuator.

VNX_ATTEN_API LDASTATUS fnLDA_GetSerialNumber(char* deviceip, int* respdata)

This function is used to get the serial number of the attenuator.

VNX_ATTEN_API LDASTATUS fnLDA_GetSoftwareVersion(char* deviceip, char* respdata)

This function is used to read the software version of the device.

VNX_ATTEN_API LDASTATUS fnLDA_GetIPMode(char* deviceip, int* respdata)

This function is used to read the IP mode configuration of the device. Response data “0” represents the “Static” mode, “1” represents the “DHCP” mode.

VNX_ATTEN_API LDASTATUS fnLDA_GetIPAddress(char* deviceip, char* respdata)

This function is used to read the IP address of the device.

VNX_ATTEN_API LDASTATUS fnLDA_GetNetmask(char* deviceip, char* respdata)

This function is used to read the netmask of the device.

VNX_ATTEN_API LDASTATUS fnLDA_GetGateway (char* deviceip, char* respdata)

This function is used to read the gateway address of the device.

1

VNX_ATTEN_API LDASTATUS fnLDA_GetWorkingFrequency(char* deviceip, int* respdata)

This function is used to read the current frequency value of the device. The return value is in 100kHz resolution.

VNX_ATTEN_API LDASTATUS fnLDA_GetMinWorkingFrequency(char* deviceip, int* respdata)

This function is used to read the minimum frequency of the device. The return value is in 100kHz resolution.

VNX_ATTEN_API LDASTATUS fnLDA_GetMaxWorkingFrequency(char* deviceip, int* respdata)

This function is used to read the maximum frequency of the device. The return value is in 100kHz resolution.

VNX_ATTEN_API LDASTATUS fnLDA_GetChannel(char* deviceip, int* respdata)

This function is used to read the current channel of the device.

VNX_ATTEN_API LDASTATUS fnLDA_GetMaxAttenuation(char* deviceip, int* respdata)

This function is used to read the maximum attenuation value of the device. The return value is in 0.05db units.

VNX_ATTEN_API LDASTATUS fnLDA_GetMinAttenuation(char* deviceip, int* respdata)

This function is used to read the minimum attenuation value of the device. The return value is in 0.05db units.

VNX_ATTEN_API LDASTATUS fnLDA_GetAttenuation(char* deviceip, int* respdata)

This function reads the current attenuation setting of the device. When an attenuation ramp is active this value will change dynamically to reflect the current setting of the device. The return value is in .05 db units.

VNX_ATTEN_API LDASTATUS fnLDA_GetRampStart(char* deviceip, int* respdata)

This function reads the current attenuation ramp start value setting of the device. The return value is in .05db units.

VNX_ATTEN_API LDASTATUS fnLDA_GetRampEnd(char* deviceip, int* respdata)

This function reads the current attenuation ramp end setting of the device. The return value is in .05db units.

VNX_ATTEN_API LDASTATUS fnLDA_GetDwellTime(char* deviceip, int* respdata)

This function reads the current dwell time for each step on the attenuation ramp in milliseconds. The return value is in 100msec resolution.

VNX_ATTEN_API LDASTATUS fnLDA_GetIdleTime(char* deviceip, int* respdata)

This function reads the idle time, which is the delay between attenuation ramps when the device is in the repeating ramp mode, in milliseconds. The return value is in 100msec resolution.

VNX_ATTEN_API LDASTATUS fnLDA_GetHoldTime(char* deviceip, int* respdata)

This function reads the hold time, the time delay between the first and second phases of a ramp in milliseconds. The return value is in 100msec resolution.

4.0 PROGRAMMING SUPPORT

Lab Brick programming support is available from Vaunix Technology Corporation. Please contact our technical support group by email - LabBrickSupport@Vaunix.com.
Vaunix Technology also offers custom programming solutions. Send us your requirements to receive a fixed rate project quotation.
Thank you for using our Lab Brick products.