LOG Storm Studio

C library user's guide

Table of Contents

References

[]

History

Version	Date	Description
1.00	May, 11th, 2012	Initial revision

1 Introduction

LogStorm.dll is a C DLL used in LOG Storm Studio software. This library contains the functions required to configure and control LOG Storm device.

Users wishing to develop their own interface application, automate tasks or integrate LOG Storm control within other environments are enabled to do so with simple C function calls.

This user's guide describes the functions that are made available in LogStorm.dll library.

2 LogStorm.dll Library Description

2.1 Functions Quick Reference Table

Table 1 gives the list of the functions callable from LogStorm.dll.

Function prototype	Description
int log_CreateInstance(void);	Creates an instance of the LOG Storm library.
void log_DeleteInstance(int Handle);	Deletes an instance of the LOG Storm library.
void log_SelectInstance(int Handle);	Selects an instance of the LOG Storm library.
int log_ScanDev(unsigned char *pType, unsigned int *pID, unsigned char *pSerNum, bool *pInUse);	Scans the USB bus for available LOG Storm devices.
int log_Connect(char *pSerNum, unsigned short SupplyVoltage);	Connects to an available LOG Storm device.
int log_Disconnect(void);	Disconnects from an LOG Storm device.
int log_LoadPrjFile(char *pFileName, bool CheckSyntax);	Loads a LOG Storm project file.
int log_Run(bool Blocking);	Runs LOG Storm to capture data.
int log_Stop(void);	Stops LOG Storm capture.
int log_GetState(void);	Retrieves the current state of the LOG Storm device.
int log_SetStateCallback(void *pObj, void *pFct);	Defines a callback for the systems status.

Table 1: Functions quick reference

2.2 Functions Details

int log_CreateInstance(void);

Parameters: none

Returns: A handle to the initialized library instance.

Description: Creates an LOG Storm instance and initializes the LOG Storm library. This function must be called a first time before any other function, to ensure proper operation of the library.

A new LOG Storm library instance is created every time the function is called. The user application can select between the created instances with the log_SelectInstance function. Multiple instances are useful when controlling multiple LOG Storm devices from one user application.

void log_DeleteInstance(int Handle);

Parameters: **Handle** : Handle to an instance of a previously created library instance.

Returns: -

Description: Deletes the library instance corresponding to the supplied handle. The library instance must first be created with log_CreateInstance.

void log_SelectInstance(int Handle);

Parameters: **Handle** : Handle to an instance of a previously created library instance.

Returns: -

Description: Selects the library instance corresponding to the supplied handle. The library instance must first be created with log_CreateInstance.

int log_ScanDev(unsigned char *pType, unsigned int *pID, unsigned char *pSerNum, bool *pInUse);

Parameters: **pType** : Array receiving the type of connection for each LOG Storm device. The type is always USB and is equal to 0.

pID : Array receiving the vendor ID and product ID for each LOG Storm device. The vendor ID is always 0x1CC4 and product ID is always 0x0401.

pSerNum : Array receiving the serial number for each LOG Storm device. Each serial number is 11 bytes wide.

pInUse : Array receiving the connection status of the device. If true, the device is already in use and is not available for connection.

Returns: The number of LogStorm devices found when the value is between 0 and 127, other values correspond with one of the return codes described in paragraph 2.3.

Description: Scans the USB bus for connected LOG Storm devices. The three parameters are pointers to three buffers.

The connection type is defined by one char (= 1 byte), hence the minimum size of the first buffer must be equal the the number of connected LogStorm devices times one byte.

The ID is defined by one int (= 4 bytes), hence the minimum size of the second buffer must be equal the the number of connected LogStorm devices times four bytes.

The serial number is defined by eleven bytes, hence the minimum size of the last buffer must be equal the the number of connected LogStorm devices times eleven bytes.

The maximum number of devices allowed on a USB bus is 127. Hence, allocating respectively 1*127 bytes, 4*127 bytes and 11*127 bytes for pType, pID and pSerNum will always work.

The function returns the number of LOG Storm devices found. This also indicates the “fill level” of the three parameters. If 3 devices were found, the buffer will respectively contain 3 bytes, 12 bytes and 33 bytes of valid data.

int log_Connect(char *pSerNum, unsigned short SupplyVoltage);

Parameters: **pSerNum** : Pointer to an eleven byte buffer containing the serial number of the LogStorm device to connect to.

SupplyVoltage : Defines the user interface supply voltages. Following values are valid:

- 3300 : supply voltage between 3.30V and 2.91V
- 2500 : supply voltage between 2.90V and 2.16V
- 1800 : supply voltage between 2.15V and 1.66V
- 1500 : supply voltage between 1.65V and 1.39V
- 1250 : supply voltage between 1.38V and 1.25V

Returns: See return codes paragraph 2.3.

Description: Connects to the LogStorm device defined by the serial number. During connection, the device is loaded with its configuration defined by the supply voltage.

int log_Disconnect(void);

Parameters: none

Returns: See return codes paragraph 2.3.

Description: Disconnects the LogStorm device from the library.

int log_LoadPrjFile(char *pFileName, bool CheckSyntax);

Parameters: **pFileName** : Pointer to a string containing the file name.

CheckSyntax : Forces to execute a syntax check only, the project file isn't be loaded. To load the project file, this field must be set to false.

Returns: See return codes paragraph 2.3.

Description: Loads a project file. This function must always be called at least once before running a capture.

int log_Run(bool Blocking);

Parameters: **Blocking** : When set to true, the function call is blocking till the end of the capture.

Returns: See return codes paragraph 2.3.

Description: Runs LOG Storm capture with the settings defined in the project file. This is fully equivalent to pressing the “Run” button in the graphical user interface.

int log_Stop(void);

Parameters: none

Returns: See return codes paragraph 2.3.

Description: Stops a running job.

int log_GetState(void);

Parameters: none

Returns: The state value is split into three fields. The first field corresponds with bits 7 to 0 and indicate Log Storm's sequencer status. The second field (bits 15 to 8) contain event flags. The final field (bits 31 to 16) contain the number of overflows in the last job.

The current state of LogStorm (bit 7 to 0) are defined as follows:

- 0x00 : Idle
- 0x01 : Initialising (configuring the library for a capture)
- 0x02 : Waiting start event
- 0x03 : Running
- 0x04 : Post-fetching
- 0x05 : Downloading data

The event flags (bit 15 to 8) are defined as follows:

- 0x01 : Status file updated
- 0x02 : Device disconnected

Description: Retrieves the current state of the LogStorm device.

int log_SetStateCallback(void *pObj, void *pFct);

Parameters: **pObj** : Pointer to an instance of a user object.

	pFct : Callback function.
Returns:	See return codes paragraph 2.3.
Description:	Defines a function to be called on a state change (see log_GetState for the existing states). The callback prototype is: void StateCallback(void *pObj, unsigned int State)

2.3 Functions Return Codes

Return code	Meaning	Action required
0x00000000	No error	
USB Device Driver		
0x80000002	No valid USB device found	Check if your LOG Storm device is connected to the USB port.
0x80000004	Failed to open USB driver	Check if you have properly installed the USB driver. Check if your LOG Storm device is properly connected to the USB port.
0x8000000A	USB device already used	Disconnect the device from an other LOG Storm library or program.
0x80050001	Failed to load bin file	Check if your working directory contains all the *.bin files provided with LOG Storm Studio.
0x80050001	Failed to load bin file	Check if the bin file corresponding to the selected voltage is present in the execution directory.
Licensing		
0x00080001	No valid license	1) Check if you have received your license file. If you don't have it, mailto: support@byteparadigm.com to request your license file. You MUST provide your LOG Storm unit serial number. It is located at the back of your device. 2) Install your license with LOG Storm Studio GUI. Please click here to know how to install license. 3) If you still receive once of these error codes, please contact Byte Paradigm support (support@byteparadigm.com).
0x00080002	Invalid license features	
0x00080003	License not found in license file	
0x80080001	Failed to open license file	
0x80080002	License decoding failed	
0x80080003	Failed to find license section	
Logging		
0x80200002	Log file not opened	Soft cannot create log file in roaming directory. Check users directory properties: **Windows 7 users**: the roaming directory is located here: **C:\Users\<user>\AppData\Roaming\ByteParadigm\LogStormStudio** Windows XP users: the 'roaming directory' is located here: **C:\Documents and Settings\<user>\Application Data\ByteParadigm\LogStormStudio** Contact your IT manager to set your permissions properly.
Configuration File		
0x80400001	Failed to open the project file	The project file is locked or does not exist. Check status.
0x80400002	Failed to create project file log	Check if the log file destination is accessible.
0x80400003	Project file contains errors	Check project file syntax.
0x80400004	Project file not defined	Specified project file is NULL. Please correct.
0x80400005	Project file name exceeds 256 characters	Please shorten project file name (including path).
Application		
0x80500001	No device connected	You attempted to execute an operation with LOG Storm device be you failed to connect it properly. Please first use log_Connect function to connect a device.
0x80500002	Application already connected to a device	You already connected a device. If you want to connect another device, disconnect it and connect the new device. Use log_Connect to connect and log_Disconnect functions.

0x80500004	No project file loaded	You attempted to run a capture whereas no project is defined.
0x80500005	Capture already running	You attempted to run a capture that is already running.
0x80500006	Job not running	You attempted to stop a capture that was not running.
Output Files		
0x80600001	Failed to open the data file	The LOG Storm library is unable to create or modify files and directories. Make sure you have the right permissions.
0x80610001	Failed to open the data file	
0x80620001	Failed to open the trigger file	
0x80630001	Failed to open the overflow file	
0x80640001	Failed to open the status file	
0x806F0001	Failed to create the output directory	
0x806F0002	Output directory already exists	
0x806F0003	Failed to create the result directory	
0x806F0004	Result directory already exists	
C API		
0x80900001	Unable to find instance on C API	You called a function without having created an instance first. Please use log_CreateInstance first.

If you receive an error code that is not listed above, please contact Byte Paradigm support to report it: support@byteparadigm.com