



Logitech Gaming LCD SDK V1.01

Overview and Reference

© 2013 Logitech. Confidential

The Logitech Gaming LCD SDK, including all accompanying documentation, is protected by intellectual property laws. All use of the Logitech Gaming LCD SDK is subject to the License Agreement found in the "Logitech Gaming LCD SDK License Agreement" file and at the end of this document. If you do not agree to the terms and conditions of the License Agreement, you must immediately return any documentation, the accompanying software and all other material provided to you by Logitech. All rights not expressly granted by Logitech are reserved.

Contents

Overview	3
SDK Package	3
Requirements	3
Interfacing with the SDK	3
Using LogitechLcd.h and LogitechLcd.lib to access LogitechLcd.dll	3
Using LogitechLcd.dll directly	3
LCD equipped Logitech Gaming devices.....	4
Color LCD, <i>resolution of 320 by 240 pixels, full RGBA</i>	4
G19	4
Monochrome LCD, <i>resolution of 160 by 43 pixels</i>	4
G510	4
G13	5
G15 v1	5
G15 v2	6
Do's and Don'ts	6
Sample usage of the SDK	6
Using header and lib	6
Using DLL directly	7
Reference.....	9
General Functions	9
LogiLcdInit	9
LogiLcdIsConnected	9
LogiLcdIsButtonPressed	9
LogiLcdUpdate.....	10
LogiLcdShutdown	10
Monochrome Lcd Function	10
LogiLcdMonoSetBackground	10
LogiLcdMonoSetText.....	11
Color LCD functions.....	11
LogiLcdColorSetBackground	11
LogiLcdColorSetTitle	13
LogiLcdColorSetText	13
End-User License Agreement for Logitech Gaming LCD SDK	15

Overview

The Logitech Gaming LCD SDK enables applications such as games to display game specific information on Logitech devices that have a monochrome or color LCD.

The main goal is to display useful information that is not already being displayed on the game screen. For example the game can display on the LCD information that usually is only accessing when hitting tab or when going to the game menu.

Another option is to remove information from the game screen (e.g. current score, timer, frame rate, etc), and move it to the LCD instead.

The SDK is a Windows based API for C/C++ programmers. Games based on the Microsoft Win32 API do not access hardware directly. Instead, the Logitech Gaming LCD SDK interacts with supported Logitech devices on behalf of the games.

SDK Package

The following files are included:

- LogitechLcd.h: C/C++ header file containing function prototypes
- LogitechLcd.lib: companion lib file to access DLL exported functions (32 and 64 bit)
- LogitechLcd.dll: library of SDK functions (32 and 64 bit)

Requirements

The Logitech Gaming LCD SDK can be used on the following platforms:

- Windows XP SP2 (32-bit and 64-bit)
- Windows Vista (32-bit and 64-bit)
- Windows 7 (32-bit and 64-bit)
- Windows 8 (32-bit and 64-bit)

The Logitech Gaming LCD SDK is a C based interface and is designed for use by C/C++ programmers. Familiarity with Windows programming is required.

Interfacing with the SDK

Using LogitechLcd.h and LogitechLcd.lib to access LogitechLcd.dll

The application can include LogitechLcd.h and link to LogitechLcd.lib (see "Sample usage of the SDK" further below or sample program in Samples folder). Installation folder for the DLL needs to be the same as the main executable, or needs to be part of the Path in the system environment.

Using LogitechLcd.dll directly

Alternatively the game can use the DLL directly by loading it via [LoadLibrary](#), and accessing its functions using [GetProcAddress](#) (see "Sample usage of the SDK" further below or sample program in Samples folder).

LCD equipped Logitech Gaming devices

Color LCD, resolution of 320 by 240 pixels, full RGBA.

G19



Monochrome LCD, resolution of 160 by 43 pixels

G510



G13



G15 v1



G15 v2



Do's and Don'ts

These are a few guidelines that may help you implement 'better' support in your game:

- For color use a splash screen when the game starts up.
- For color have a nice background image to take full advantage of the RGBA LCD
- Don't just display information on the LCD that is already being displayed on the main game screen. Instead display information that is not readily available for the user to see. Instead display information he can only see when hitting tab or going to the menu.
- Use the LCD to unclutter the main screen. You can move information such as score, timer, framerate and even the minimap to the LCD.
- Write support for both the color and monochrome LCDs, as both have an important user base.
- Text displayed on the LCD is fixed-width, so you can easily create multiple columns that always align correctly.
- If you want to create custom screens, draw your own bitmaps and update the background LCD bitmap up to 60 times/second.
- Use the buttons to create multiple pages or add functionality to the LCD.
- If you are working with Unreal Engine and DLLBind, and want to set the background, use the UDK specific functions, don't use those in any other application.

Sample usage of the SDK

Using header and lib

```
#pragma comment(lib, "LogitechLcd.lib")
#include "LogitechLcd.h"
```

...

```
//The parameters indicate the name of the app and the type of the target display
LogiLcdInit(text, type);

...

//this is your main loop
while(!done){
    //Every frame of your main loop you have to call the update function
    LogiLcdUpdate();

    //You can now call any of the function defined in LogitechLcd.h header file
}

LogiLcdShutdown ();
```

Using DLL directly

```
// LCD types
#define LOGI_LCD_TYPE_MONO      (0x00000001)
#define LOGI_LCD_TYPE_COLOR    (0x00000002)

//LCD Monochrome buttons
#define LOGI_LCD_MONO_BUTTON_0 (0x00000001)
#define LOGI_LCD_MONO_BUTTON_1 (0x00000002)
#define LOGI_LCD_MONO_BUTTON_2 (0x00000004)
#define LOGI_LCD_MONO_BUTTON_3 (0x00000008)

//LCD Color buttons
#define LOGI_LCD_COLOR_BUTTON_LEFT   (0x00000100)
#define LOGI_LCD_COLOR_BUTTON_RIGHT (0x00000200)
#define LOGI_LCD_COLOR_BUTTON_OK     (0x00000400)
#define LOGI_LCD_COLOR_BUTTON_CANCEL (0x00000800)
#define LOGI_LCD_COLOR_BUTTON_UP     (0x00001000)
#define LOGI_LCD_COLOR_BUTTON_DOWN   (0x00002000)
#define LOGI_LCD_COLOR_BUTTON_MENU   (0x00004000)

//LCD Monochrome size
const int LOGI_LCD_MONO_WIDTH = 160;
const int LOGI_LCD_MONO_HEIGHT = 43;

//LCD Color size
const int LOGI_LCD_COLOR_WIDTH = 320;
const int LOGI_LCD_COLOR_HEIGHT = 240;

typedef BOOL (* LPFNDDLINIT)(wchar_t*, int );
typedef BOOL (* LPFNDDLISCONNECTED)(int);
typedef BOOL (* LPFNDDLISBUTTONPRESSED)(int);
typedef void (* LPFNDDLUPDATE)();
typedef void (* LPFNDLLSHUTDOWN)();
typedef BOOL (* LPFNDDLMONOSETBACKGROUND)(BYTE*);
typedef BOOL (* LPFNDDLMONOSETTEXT)(int, wchar_t*);
typedef BOOL (* LPFNDDLCCOLORSETBACKGROUND)(BYTE*);
```

```

typedef BOOL (* LPFNDLLCOLORSETTITLE)(wchar_t*, int, int, int);
typedef BOOL (* LPFNDLLCOLORSETTEXT)(int, wchar_t*, int, int, int);

LPFNDLLINIT LogiLcdInit = NULL;
LPFNDLLISCONNECTED LogiLcdIsConnected = NULL;
LPFNDLLISBUTTONPRESSED LogiLcdIsButtonPressed = NULL;
LPFNDLLUPDATE LogiLcdUpdate = NULL;
LPFNDLLSHUTDOWN LogiLcdShutdown = NULL;
LPFNDLLMONOSETBACKGROUND LogiLcdMonoSetBackground = NULL;
LPFNDLLMONOSETTEXT LogiLcdMonoSetText = NULL;
LPFNDLLCCOLORSETBACKGROUND LogiLcdColorSetBackground = NULL;
LPFNDLLCOLORSETTITLE LogiLcdColorSetTitle = NULL;
LPFNDLLCOLORSETTEXT LogiLcdColorSetText = NULL;

...

HINSTANCE logiDllHandle = LoadLibrary(L"LogitechLcd.dll");

if (logiDllHandle != NULL)
{
    LogiLcdInit = (LPFNDLLINIT)GetProcAddress(logiDllHandle, "LogiLcdInit");
    LogiLcdIsConnected = (LPFNDLLISCONNECTED)GetProcAddress(logiDllHandle,
"LogiLcdIsConnected");
    LogiLcdIsButtonPressed = (LPFNDLLISBUTTONPRESSED)GetProcAddress(logiDllHandle,
"LogiLcdIsButtonPressed");
    LogiLcdUpdate = (LPFNDLLUPDATE)GetProcAddress(logiDllHandle, "LogiLcdUpdate");
    LogiLcdShutdown = (LPFNDLLSHUTDOWN)GetProcAddress(logiDllHandle,
"LogiLcdShutdown");
    LogiLcdMonoSetBackground = (LPFNDLLMONOSETBACKGROUND)GetProcAddress(logiDllHandle,
"LogiLcdMonoSetBackground");
    LogiLcdMonoSetText = (LPFNDLLMONOSETTEXT)GetProcAddress(logiDllHandle,
"LogiLcdMonoSetText");
    LogiLcdColorSetBackground =
(LPFNDLLCCOLORSETBACKGROUND)GetProcAddress(logiDllHandle, "LogiLcdColorSetBackground");
    LogiLcdColorSetTitle = (LPFNDLLCOLORSETTITLE)GetProcAddress(logiDllHandle,
"LogiLcdColorSetTitle");
    LogiLcdColorSetText = (LPFNDLLCOLORSETTEXT)GetProcAddress(logiDllHandle,
"LogiLcdColorSetText");
    //The parameters indicate the name of the app and the type of the target display
    ret = LogiLcdInit(text, type);
}

//this is your main loop
while(!done){
    //Every frame of your main loop you have to call the update function
    LogiLcdUpdate();

    //You can now call any of the function defined in LogitechLcd.h header file
}

LogiLcdShutdown ();

```

Reference

Generic Functions

LogiLcdInit

The **LogiLcdInit** () function makes necessary initializations. You must call this function prior to any other function in the library.

```
bool LogiLcdInit(wchar_t* friendlyName, int lcdType);
```

Parameters

- friendlyName : the name of your applet, you can't change it after initialization.
- lcdType : defines the type of your applet lcd target, it can be one of the following :
 - LOGI_LCD_TYPE_MONO
 - LOGI_LCD_TYPE_COLOR
 - If you want to initialize your applet for both LCD types just use
LOGI_LCD_TYPE_MONO | LOGI_LCD_TYPE_COLOR

Return value

If the function succeeds, it returns true. Otherwise false.

LogiLcdIsConnected

The **LogiLcdIsConnected** () function checks if a device of the type specified by the parameter is connected.

```
bool LogiLcdIsConnected(int lcdType);
```

Parameters

- lcdType : defines the lcd type to look for, it can be one of the following :
 - LOGI_LCD_TYPE_MONO
 - LOGI_LCD_TYPE_COLOR
 - If you want to look for both LCD types just use
LOGI_LCD_TYPE_MONO | LOGI_LCD_TYPE_COLOR

Return value

If a device supporting the lcd type specified is found, it returns true. If the device has not been found or the LogiLcdInit function has not been called before, returns false.

LogiLcdIsButtonPressed

The **LogiLcdIsButtonPressed** () function checks if the button specified by the parameter is being pressed.

```
bool LogiLcdIsButtonPressed(int button);
```

Parameters

- button : defines the button to check on, it can be one of the following :
 - LOGI_LCD_MONO_BUTTON_0
 - LOGI_LCD_MONO_BUTTON_1
 - LOGI_LCD_MONO_BUTTON_2

- LOGI_LCD_MONO_BUTTON_3
- LOGI_LCD_COLOR_BUTTON_LEFT
- LOGI_LCD_COLOR_BUTTON_RIGHT
- LOGI_LCD_COLOR_BUTTON_OK
- LOGI_LCD_COLOR_BUTTON_CANCEL
- LOGI_LCD_COLOR_BUTTON_UP
- LOGI_LCD_COLOR_BUTTON_DOWN
- LOGI_LCD_COLOR_BUTTON_MENU

Return value

If the button specified is being pressed it returns true. Otherwise false.

Notes

The button will be considered pressed only if your applet is the one currently in the foreground.

LogiLcdUpdate

The **LogiLcdUpdate** () function updates the lcd display.

```
void LogiLcdUpdate();
```

Notes

You have to call this function every frame of your main loop, to keep the lcd updated.

LogiLcdShutdown

The **LogiLcdShutdown** () function kills the applet and frees memory used by the SDK.

```
void LogiLcdShutdown();
```

Monochrome Lcd Function

LogiLcdMonoSetBackground

The **LogiLcdMonoSetBackground** () function sets the specified image as background for the monochrome lcd device connected.

```
bool LogiLcdMonoSetBackground(BYTE monoBitmap[]);
```

Parameters

- monoBitmap: the array of pixels that define the actual monochrome bitmap

The array of pixels is organized as a rectangular area, 160 bytes wide and 43 bytes high. Despite the display being monochrome, 8 bits per pixel are used here for simple manipulation of individual pixels. To learn how to use GDI drawing functions efficiently with such an arrangement, see the sample code.

The pixels are arranged in the following order

byte 0 (0,0)	byte 1 (1,0)	byte 2 (2,0)	...	byte 157 (157,0)	byte 158 (158,0)	byte 159 (159,0)
byte 160 (0,1)	byte 161 (1,1)	byte 162 (2,1)	...	byte 317 (157,1)	byte 318 (158,1)	byte 319 (159,1)
...
byte 6560 (0,41)	byte 6561 (1,41)	byte 6562 (2,41)	...	byte 6717 (157,41)	byte 6718 (158,41)	byte 6719 (159,41)
byte 6720 (0,42)	byte 6721 (1,42)	byte 6722 (2,42)	...	byte 6877 (157,42)	byte 6878 (158,42)	byte 6879 (159,42)

Return value

True if it succeeds, false otherwise.

Notes

The image size must be 160x43 in order to use this function. The SDK will turn on the pixel on the screen if the value assigned to that byte is ≥ 128 , it will remain off if the value is < 128 .

LogiLcdMonoSetText

The **LogiLcdMonoSetText** () function sets the specified text in the requested line on the monochrome lcd device connected.

```
bool LogiLcdMonoSetText(int lineNumber, wchar_t* text);
```

Parameters

- lineNumber: the line on the screen you want the text to appear. The monochrome lcd display has 4 lines, so this parameter can be any number from 0 to 3.
- text: defines the text you want to display

Return value

True if it succeeds, false otherwise.

Color LCD functions

LogiLcdColorSetBackground

The **LogiLcdColorSetBackground** () function sets the specified image as background for the color lcd device connected.

```
bool LogiLcdColorSetBackground(BYTE colorBitmap[]);
```

Parameters

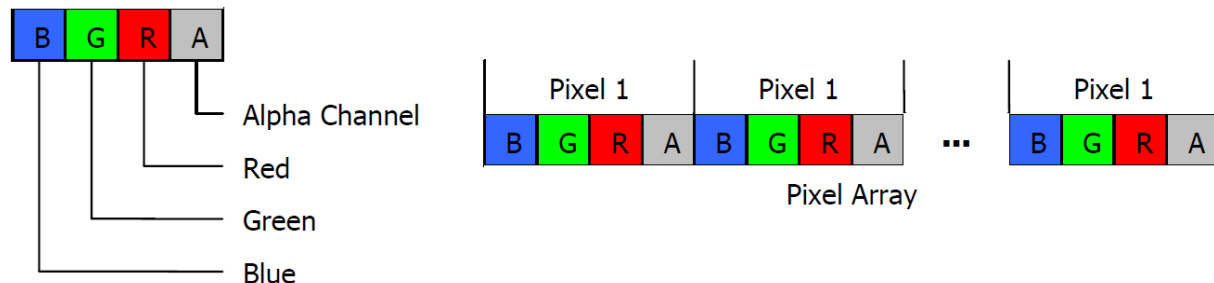
- colorBitmap: the array of pixels that define the actual color bitmap

The array of pixels is organized as a rectangular area, 320 bytes wide and 240 bytes high. Since the color lcd can display the full RGB gamma, 32 bits per pixel (4 bytes) are used. The size of the colorBitmap array has to be $320 \times 240 \times 4 = 307200$ therefore. To learn how to use GDI drawing functions efficiently with such an arrangement, see the sample code.

The pixels are arranged in the following order

byte 0-3 (0,0)	byte 4-7 (1,0)	byte 8-11 (2,0)	...	byte 1268-1271 (317,0)	byte 1272-1275 (318,0)	byte 1276-1279 (319,0)
byte 1280-1283 (0,1)	byte 1284-1287 (1,1)	byte 1288-1291 (2,1)	...	byte 2549-2552 (317,1)	byte 2553-2555 (318,1)	byte 2556-2559 (319,1)
...
byte 304640-304643 (0,238)	byte 304646-304649 (1,238)	byte 304650-304653 (2,238)	...	byte 305908-305911 (317,238)	Byte 305912-305915 (318,238)	byte 305916-305919 (319,238)
byte 305920-305923 (0,239)	byte 305924-305927 (1,239)	byte 305928-305931 (2,239)	...	byte 307188-307191 (317,239)	byte 307192-307195 (318,239)	byte 307196-307199 (319,239)

32 bit values are stored in 4 consecutive bytes that represent the RGB color values for that pixel. These values use the same top left to bottom right raster style transform to the flat character array with the exception that each pixel value is specified using 4 consecutive bytes. The illustration below shows the data arrangement for these RGB quads.



Each of the bytes in the RGB quad specify the intensity of the given color. The value ranges from 0 (the darkest color value) to 255 (brightest color value)

Return value

True if it succeeds, false otherwise.

Notes

The image size must be 320x240 in order to use this function.

LogiLcdColorSetTitle

The **LogiLcdColorSetTitle** () function sets the specified text in the first line on the color lcd device connected. The font size that will be displayed is bigger than the one used in the other lines, so you can use this function to set the title of your applet/page.

```
bool LogiLcdColorSetTitle(wchar_t* text, int red, int green, int blue);
```

Parameters

- text: defines the text you want to display as title
- red, green, blue: this lcd can display a full RGB color gamma, you can define the color of your title using this parameters. Values between 0 and 255 are accepted. The default value for this parameters is 255, so if you don't specify any color, your title will be white.

Return value

True if it succeeds, false otherwise.

LogiLcdColorSetText

The **LogiLcdColorSetText** () function sets the specified text in the requested line on the color lcd device connected.

```
bool LogiLcdColorSetText(int lineNumber, wchar_t* text, int red, int green, int blue);
```

Parameters

- lineNumber: the line on the screen you want the text to appear. The color lcd display has 8 lines for standard text, so this parameter can be any number from 0 to 7.
- text: defines the text you want to display
- red, green, blue: this lcd can display a full RGB color gamma, you can define the color of your text using this parameters. Values between 0 and 255 are accepted. The default value for this parameters is 255, so if you don't specify any color, your text will be white.

Return value

True if it succeeds, false otherwise.

Unreal Engine and DLLBind specific functions

The following functions are specific for Unreal Engine script integration through DLLBind :

- `LogiLcdColorSetBackgroundUDK`
- `LogiLcdColorResetBackgroundUDK`
- `LogiLcdMonoSetBackgroundUDK`
- `LogiLcdMonoResetBackgroundUDK`

The main reason for these different functions is the array size limitation in the UnrealScript language to 2048. As described before the bitmap in the LCDs is represented by an array of bytes. For both the monochrome and the color version the array size is way bigger than that limitation. Using these functions in Unreal Script you will still be able to send bitmaps to the LCDs. For more information on how to use these functions check the document `UdkDLLBindInstructions` in the Doc folder of this package.

REMARK: Don't use this functions if you are not using Unreal Engine and integrating Logitech support using DLLBind technology.

End-User License Agreement for Logitech Gaming LCD SDK

This End-User License Agreement for Logitech Gaming LCD SDK ("Agreement") is a legal agreement between you, either an individual or legal entity ("You" or "you") and Logitech Inc. ("Logitech") for use of the Logitech Gaming LCD software development kit, which includes computer software and related media and documentation (hereinafter "Logitech Gaming LCD SDK"). By using this Logitech Gaming LCD SDK, you are agreeing to be bound by the terms and conditions of this Agreement. If you do not agree to the terms and conditions of this Agreement, promptly return the Logitech Gaming LCD SDK and other items that are part of this product in their original package, or if you have downloaded this software from a Logitech or a Distributor web site, then you must stop using the software and destroy any copies of the software in your possession or control.

1 Grant of License and Restrictions. This Agreement grants You the following rights provided that You comply with all terms and conditions of this Agreement.

- (a) Logitech grants You a limited, non-exclusive, nontransferable license to install and use an unlimited number of copies of the Logitech Gaming LCD SDK on computers. All other rights are reserved to Logitech.
- (b) You shall not reverse engineer, decompile or disassemble any portion of the Logitech Gaming LCD SDK, except and only to the extent that this limitation is expressly prohibited by applicable law.
- (c) At your option, you may provide reasonable feedback to Logitech, including but not limited to usability, bug reports and test results, with respect to the Logitech Gaming LCD SDK. All bug reports, test results and other feedback provided to Logitech by You shall be the property of Logitech and may be used by Logitech for any purpose.
- (d) In the event Logitech, in its sole discretion, elects to provide copies of the Logitech Gaming LCD SDK to more than one individual employed by You (if You are not a single individual), each such individual shall be entitled to exercise the rights granted in this Agreement and shall be bound by the terms and conditions herein.

2 Updates. Logitech is not obligated to provide technical support or updates to You for the Logitech Gaming LCD SDK provided to You pursuant to this Agreement. However, Logitech may, in its sole discretion,

provide further pre-release versions, technical support, updates and/or supplements (“Updates”) to You, in which case such Updates shall be deemed to be included in the “Logitech Gaming LCD SDK” and shall be governed by this Agreement, unless other terms of use are provided in writing by Logitech with such Updates.

- 3 Intellectual Property Rights.** The Logitech Gaming LCD SDK is licensed, not sold, to You for use only under the terms and conditions of this Agreement. Logitech and its suppliers retain title to the Logitech Gaming LCD SDK and all intellectual property rights therein. The Logitech Gaming LCD SDK is protected by intellectual property laws and international treaties, including U.S. copyright law and international copyright treaties. All rights not expressly granted by Logitech are reserved.
- 4 Disclaimer of Warranty.** TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, LOGITECH, ITS SUPPLIERS AND DISTRIBUTORS PROVIDE THE LOGITECH GAMING LCD SDK AND OTHER LOGITECH PRODUCTS AND SERVICES (IF ANY) AS IS AND WITHOUT WARRANTY OF ANY KIND. LOGITECH AND ITS SUPPLIERS AND DISTRIBUTORS EXPRESSLY DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD-PARTY RIGHTS WITH RESPECT TO THE LOGITECH GAMING LCD SDK AND ANY WARRANTIES OF NON-INTERFERENCE OR ACCURACY OF INFORMATIONAL CONTENT. NO LOGITECH DISTRIBUTOR, AGENT, OR EMPLOYEE IS AUTHORIZED TO MAKE ANY MODIFICATION, EXTENSION, OR ADDITION TO THIS WARRANTY. Some jurisdictions do not allow limitations on how long an implied warranty lasts, so the above limitation may not apply to you.
- 5 Limitation of Liability.** IN NO EVENT WILL LOGITECH, ITS SUPPLIERS, OR DISTRIBUTORS BE LIABLE FOR ANY COSTS OF PROCUREMENT OF SUBSTITUTE PRODUCTS OR SERVICES, LOST PROFITS, LOSS OF INFORMATION OR DATA, OR ANY OTHER SPECIAL, INDIRECT, CONSEQUENTIAL, OR INCIDENTAL DAMAGES ARISING IN ANY WAY OUT OF THE SALE OF, USE OF, OR INABILITY TO USE THE LOGITECH GAMING LCD SDK OR ANY LOGITECH PRODUCT OR SERVICE, EVEN IF LOGITECH HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO CASE SHALL LOGITECH'S, ITS SUPPLIERS' AND DISTRIBUTORS' TOTAL LIABILITY EXCEED THE ACTUAL MONEY PAID FOR THE LOGITECH PRODUCT OR SERVICE GIVING RISE TO THE LIABILITY.

Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you. The above limitations will not apply in case of personal injury where and to the extent that applicable law requires such liability.

- 6 U.S. Government Rights.** Use, duplication, or disclosure of the software contained in the Logitech Gaming LCD SDK by the U.S. Government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (OCT 1988) FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable. Logitech Inc. 7600 Gateway Blvd, Newark, CA 94560.
- 7 Export Law Assurances.** You agree and certify that neither the Logitech Gaming LCD SDK nor any other technical data received from Logitech will be exported outside the United States except as authorized and as permitted by the laws and regulations of the United States. If you have rightfully obtained the Logitech Gaming LCD SDK outside of the United States, you agree that you will not re-export the Logitech Gaming LCD SDK nor any other technical data received from Logitech, except as permitted by the laws and regulations of the United States and the laws and regulations of the jurisdiction in which you obtained the Logitech Gaming LCD SDK.
- 8 Termination:** This Agreement is effective until terminated. Upon any violation of any of the provisions of this Agreement, or any provisions of any agreement between you and a Distributor, rights to use the Logitech Gaming LCD SDK shall automatically terminate and the Logitech Gaming LCD SDK must be returned to Logitech or all copies of the Logitech Gaming LCD SDK destroyed. You may also terminate this Agreement at any time by destroying all copies of the Logitech Gaming LCD SDK in your possession or control. If Logitech makes a request via public announcement or press release to stop using the copies of the Logitech Gaming LCD SDK, you will comply immediately with this request. The provisions of paragraphs 3, 7, 8 and 12 will survive any termination of this Agreement.
- 9 General Terms and Conditions.** If You are an individual signing this Agreement on behalf of a company, then You represent that You have authority to execute this Agreement on behalf of such company. This Agreement will be governed by and construed in accordance with the laws of the United States and the State of California, without regard to or application of its choice of law rules or principles. If for any reason a

court of competent jurisdiction finds any provision of this Agreement, or portion thereof, to be unenforceable, that provision of the Agreement shall be enforced to the maximum extent permissible so as to affect the intent of the parties, and the remainder of this Agreement shall continue in full force and effect. This Agreement constitutes the entire agreement between You and Logitech respect to the use of the Logitech Gaming LCD SDK and supersedes all prior or contemporaneous understandings, communications or agreements, written or oral, regarding such subject matter.