



Using the Logitech Gaming LCD SDK with C#

© 2014 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 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
Making the LCD SDK work in your C# program.....	3
Steps	3

Overview

The Logitech Gaming LCD SDK enables applications such as games to control the LCDs on supported Logitech gaming mice and keyboards.

It's built as a C++ DLL, but it can be easily integrated in a C# assembly, using P/Invoke and function marshaling.

Please refer to the Logitech SDK's Doc\LogitechGamingLCDSDK.pdf for details on the SDK's functionality.

Making the LCD SDK work in your C# program

The following steps show how to make the Logitech SDK work with a C# program. Please adapt the steps to your game for things to work.

Steps

1. Create a SDK C# wrapper class as follows :

LogitechGSDK.cs

```
using System.Collections;
using System.Runtime.InteropServices;
using System.Collections.Specialized;
using System;

public class LogitechGSDK {
    //LCD SDK
    public const int LOGI_LCD_COLOR_BUTTON_LEFT = (0x00000100);
    public const int LOGI_LCD_COLOR_BUTTON_RIGHT = (0x00000200);
    public const int LOGI_LCD_COLOR_BUTTON_OK = (0x00000400);
    public const int LOGI_LCD_COLOR_BUTTON_CANCEL = (0x00000800);
    public const int LOGI_LCD_COLOR_BUTTON_UP = (0x00001000);
    public const int LOGI_LCD_COLOR_BUTTON_DOWN = (0x00002000);
    public const int LOGI_LCD_COLOR_BUTTON_MENU = (0x00004000);

    public const int LOGI_LCD_MONO_BUTTON_0 = (0x00000001);
    public const int LOGI_LCD_MONO_BUTTON_1 = (0x00000002);
    public const int LOGI_LCD_MONO_BUTTON_2 = (0x00000004);
    public const int LOGI_LCD_MONO_BUTTON_3 = (0x00000008);

    public const int LOGI_LCD_MONO_WIDTH = 160;
    public const int LOGI_LCD_MONO_HEIGHT = 43;

    public const int LOGI_LCD_COLOR_WIDTH = 320;
    public const int LOGI_LCD_COLOR_HEIGHT = 240;

    public const int LOGI_LCD_TYPE_MONO = (0x00000001);
    public const int LOGI_LCD_TYPE_COLOR = (0x00000002);

    [DllImport("LogitechLcdEnginesWrapper", CharSet = CharSet.Unicode,
        CallingConvention = CallingConvention.Cdecl)]
```

```

    public static extern bool LogiLcdInit(String friendlyName, int lcdType);

    [DllImport("LogitechLcdEnginesWrapper ", CharSet = CharSet.Unicode,
CallingConvention = CallingConvention.Cdecl)]
    public static extern bool LogiLcdIsConnected(int lcdType);

    [DllImport("LogitechLcdEnginesWrapper ", CharSet = CharSet.Unicode,
CallingConvention = CallingConvention.Cdecl)]
    public static extern bool LogiLcdIsButtonPressed(int button);

    [DllImport("LogitechLcdEnginesWrapper ", CharSet = CharSet.Unicode,
CallingConvention = CallingConvention.Cdecl)]
    public static extern void LogiLcdUpdate();

    [DllImport("LogitechLcdEnginesWrapper ", CharSet = CharSet.Unicode,
CallingConvention = CallingConvention.Cdecl)]
    public static extern void LogiLcdShutdown();

    // Monochrome LCD functions
    [DllImport("LogitechLcdEnginesWrapper ", CharSet = CharSet.Unicode,
CallingConvention = CallingConvention.Cdecl)]
    public static extern bool LogiLcdMonoSetBackground(byte [] monoBitmap);

    [DllImport("LogitechLcdEnginesWrapper ", CharSet = CharSet.Unicode,
CallingConvention = CallingConvention.Cdecl)]
    public static extern bool LogiLcdMonoSetText(int lineNumber, String text);

    // Color LCD functions
    [DllImport("LogitechLcdEnginesWrapper ", CharSet = CharSet.Unicode,
CallingConvention = CallingConvention.Cdecl)]
    public static extern bool LogiLcdColorSetBackground(byte [] colorBitmap);

    [DllImport("LogitechLcdEnginesWrapper ", CharSet = CharSet.Unicode,
CallingConvention = CallingConvention.Cdecl)]
    public static extern bool LogiLcdColorSetTitle(String text, int red , int green ,
int blue );

    [DllImport("LogitechLcdEnginesWrapper ", CharSet = CharSet.Unicode,
CallingConvention = CallingConvention.Cdecl)]
    public static extern bool LogiLcdColorSetText(int lineNumber, String text, int red,
int green, int blue);
}

```

2. Call the functions from the wrapper from your C# code as follows:

```

LogitechGSDK.LogiLcdInit("TEST", LOGI_LCD_TYPE_MONO | LOGI_LCD_TYPE_COLOR);
LogitechGSDK.LogiLcdColorSetTitle("Testing", 255,0,0);
LogitechGSDK.LogiLcdMonoSetText(0,"testing");
LogitechGSDK.LogiLcdShutdown();
pixelMatrix = new byte[LOGI_LCD_COLOR_WIDTH * LOGI_LCD_COLOR_HEIGHT * 4];
//fill this array with your image
LogitechGSDK.LogiLcdColorSetBackground(pixelMatrix);

```

Using the LCD SDK with C#

3. Copy Logitech SDK's Lib\GameEnginesWrapper\x86\LogitechLcdEnginesWrapper.dll to your c# 32bit executable path
4. Copy Logitech SDK's Lib\GameEnginesWrapper\x64\LogitechLcdEnginesWrapper.dll to your c# 64bit executable path
5. Compile and run your program

For questions/comments, email devtechsupport@logitech.com