Logitech Gaming LED SDK
V1.01

Overview and Reference

Contents

Overview

The Logitech Gaming LED SDK enables applications such as games to control the backlight LEDs on supported Logitech gaming mice and keyboards.

The user has the option to block games from changing the lighting via a setting in the Logitech Gaming Software (version 8.35 and newer).

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 LED SDK interacts with supported Logitech devices on behalf of the games.

SDK Package

The following files are included:

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

Requirements

The Logitech Gaming LED 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 LED 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 LogitechLed.h and LogitechLed.lib to access LogitechLed.dll

The application can include LogitechLed.h and link to LogitechLed.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 LogitechLed.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).

Available colors

Different devices have different capabilities. They range from full RGB support to single color only. Details for supported devices are found further below in "Features of lighting-capable Logitech Gaming mice and keyboards".

The SDK has a single function to set the backlighting color and takes values for R(ed), G(reen) and B(lue). The way it deals with single color devices is to take whichever of the R, G, and B values is the highest and apply it. The highest value effectively defines the brightness of the single color devices. This is important to remember, because if for example rotating through colors, the game should make sure to alternate the maximum numbers as it rotates so that the effect on a single color device would be noticeable too.

Multiple clients using the SDK at the same time

The SDK allows only one client to control backlighting at any given time. In case two applications try to initialize the SDK, only the first will succeed. The second application's initialization will fail.

Features of lighting-capable Logitech Gaming mice and keyboards

G710

Colors

Single color only. Full resolution. Highest value for R, G or B defines brightness.

G600

Colors

Supports full RGB.

G510

Colors

Supports full RGB.

G110

Colors

Supports full R(ed) and B(lue), but not G(reen). When calling the SDK's LogiLedSetLighting function, values for green will be ignored.

G19

Colors

Supports full RGB.

G105

Colors

Single color only. Full resolution. Highest value for R, G or B defines brightness.

G105 Call Of Duty

Colors

Single color only. Full resolution. Highest value for R, G or B defines brightness.

G300

Colors

Supports red on/off, green on/off, blue on/off, or a combination of the three. When calling the SDK's LogiLedSetLighting function, if the percentage given is below 50, the color will be off, and when above 50, the color will be on.

G11

Colors

Single color only, 3 levels of brightness. When calling the SDK's LogiLedSetLighting function, if the highest RGB percentage given is below 33, the color will be off, if between 33 and 66, the brightness will be low, and when above 66, the brightness will be high.

G13

The SDK treats this device as a keyboard.

Colors

Supports full RGB.

G15 v1

Colors

Single color only, 3 levels of brightness. When calling the SDK's LogiLedSetLighting function, if the highest RGB percentage given is below 33, the color will be off, if between 33 and 66, the brightness will be low, and when above 66, the brightness will be high.

.

G15 v2

Colors

Single color only, 3 levels of brightness. When calling the SDK's LogiLedSetLighting function, if the highest RGB percentage given is below 33, the color will be off, if between 33 and 66, the brightness will be low, and when above 66, the brightness will be high.

Do’s and Don’ts

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

- Don’t call LogiLedSetLighting() immediately after LogiLedInit(). Instead call LogiLedInit() during game initialization, and wait for later for other calls. Reason is that LogiLedInit() will communicate with the Gaming Software via PostMessage to figure out whether the user chose to keep games from changing lighting. And so a little time between the LogiLedInit() and LogiLedSetLighting() is necessary for the communication to complete.
- Remember that some devices have only a single color. They will work fine if flashing a red warning light for example (their color will flash), but if rotating lighting try to make sure that the max value of the three colors goes up and down so that single color devices will have their brightness go up and down.
- Whenever doing a temporary lighting effect do not forget to save the current lighting (via SDK’s LogiLedSaveCurrentLighting function) just before starting the effect, and then restoring the lighting (via SDK’s LogiLedRestoreLighting function) right after the effect is finished. This only applies to user defined effects, the saving-restore lighting is already included in the preset effects.

Sample usage of the SDK

Using header and lib

```
#include "LogitechLed.h"

…

LogiLedInit();
// Be sure to do other things to give some time before calling LogiLedSetLighting()

…

// Save current lighting before starting some temporary effect
LogiLedSaveCurrentLighting(LOGITECH_LED_ALL);

…

int red = …;
int green = …;
int blue = …;

LogiLedSetLighting(LOGITECH_LED_ALL, red, green, blue);

…
// Eventually call effect functions
LogiLedFlashLighting(LOGITECH_LED_ALL, red, green, blue, duration, interval);
…

LogiLedPulseLighting(LOGITECH_LED_ALL, red, green, blue, duration);
```

```
// Restore previously saved lighting when effect is finished
LogiLedRestoreLighting(LOGITECH_LED_ALL);

…

LogiLedShutdown();
```

Using DLL directly

```
// Device types for LogiLedSaveCurrentLighting, LogiLedSetLighting,
LogiLedRestoreLighting
const int LOGITECH_LED_MOUSE = 0x0001;
const int LOGITECH_LED_KEYBOARD = 0x0002;
const int LOGITECH_LED_ALL = LOGITECH_LED_MOUSE | LOGITECH_LED_KEYBOARD;

typedef bool (* LPFNDLLINIT)();
typedef bool (* LPFNDLLSAVECURRENTLIGHTING)(int);
typedef bool (* LPFNDLLSETLIGHTING)(int, int, int, int);
typedef bool (* LPFNDLLRESTORELIGHTING)(int);
typedef bool (* LPFNDLLFLASHLIGHTING)(int, int, int, int, int, int);
typedef bool (* LPFNDLLPULSELIGHTING)(int, int, int, int, int);

typedef void (* LPFNDLLSHUTDOWN)();

LPFNDLLINIT g_lpfnDllInit = NULL;
LPFNDLLSAVECURRENTLIGHTING g_lpfnDllSaveCurrentLighing = NULL;
LPFNDLLSETLIGHTING g_lpfnDllSetLighting = NULL;
LPFNDLLRESTORELIGHTING g_lpfnDllRestoreLighting = NULL;
LPFNDLLFLASHLIGHTING g_lpfnDllFlashLighting = NULL;
LPFNDLLPULSELIGHTING g_lpfnDllPulseLighting = NULL;

LPFNDLLSHUTDOWN g_lpfnDllShutdown = NULL;

…

HINSTANCE logiDllHandle = LoadLibrary(L"LogitechLed.dll");
if (logiDllHandle != NULL)
{
    g_lpfnDllInit = (LPFNDLLINIT)GetProcAddress(logiDllHandle, "LogiLedInit");
    g_lpfnDllSaveCurrentLighing =
(LPFNDLLSAVECURRENTLIGHTING)GetProcAddress(logiDllHandle, "LogiLedSaveCurrentLighting");
    g_lpfnDllSetLighting = (LPFNDLLSETLIGHTING)GetProcAddress(logiDllHandle,
"LogiLedSetLighting");
    g_lpfnDllRestoreLighting = (LPFNDLLRESTORELIGHTING)GetProcAddress(logiDllHandle,
"LogiLedRestoreLighting");
    g_lpfnDllFlashLighting = (LPFNDLLFLASHLIGHTING)GetProcAddress(logiDllHandle,
"LogiLedFlashLighting");
    g_lpfnDllPulseLighting = (LPFNDLLPULSELIGHTING)GetProcAddress(logiDllHandle,
"LogiLedPulseLighting");

    g_lpfnDllShutdown = (LPFNDLLSHUTDOWN)GetProcAddress(logiDllHandle, "LogiLedShutdown");

    g_lpfnDllInit();
}
```

```
// Be sure to do other things to give some time before calling LogiLedSetLighting()

…

// Save current lighting before starting some effect
g_lpfnDllSaveCurrentLighing(LOGITECH_LED_ALL);

…

int red = …;
int green = …;
int blue = …;

g_lpfnDllSetLighting(LOGITECH_LED_ALL, red, green, blue);

…
// Eventually call effects functions

g_lpfnDllFlashLighting(LOGITECH_LED_ALL, red, green, blue, duration, interval);

…

g_lpfnDllPulseLighting(LOGITECH_LED_ALL, red, green, blue, duration);

…

// Restore previously saved lighting when effect is finished
g_lpfnDllRestoreLighting(LOGITECH_LED_ALL);

…

g_lpfnDllShutdown();
```

Reference

LogiLedInit

The **LogiLedInit**() function makes sure there isn't already another instance running and then makes necessary initializations. It saves the current lighting for all connected and supported devices.

```
bool LogiLedInit();
```

Return value

If the function succeeds, it returns true. Otherwise false.
If it returns false, it is either because there is already another application using the SDK, or because of an initialization error.

LogiLedSaveCurrentLighting

The **LogiLedSaveCurrentLighting**() function saves the current lighting so that it can be restored after a temporary effect is finished. For example if flashing a red warning sign for a few seconds, you would call the **LogiLedSaveCurrentLighting**() function just before starting the warning effect.

```
bool LogiLedSaveCurrentLighting(int deviceType);
```

Parameters

- deviceType
  - LOGITECH_LED_MOUSE: change lighting on all connected and supported mice, but not keyboards
  - LOGITECH_LED_KEYBOARD: change lighting on all connected and supported keyboards, but not mice
  - LOGITECH_LED_ALL: change lighting on all connected and supported devices

Return value

If the function succeeds, it returns true. Otherwise false.
The function will return false if **LogiLedInit**() hasn't been called or because the user chose to opt out via a setting in the Logitech Gaming Software.

LogiLedSetLighting

The **LogiLedSetLighting**() function sets the lighting on connected and supported devices.

```
bool LogiLedSetLighting(int deviceType, int redPercentage, int greenPercentage, int
bluePercentage);
```

Parameters

- deviceType
  - LOGITECH_LED_MOUSE: change lighting on all connected and supported mice, but not keyboards
  - LOGITECH_LED_KEYBOARD: change lighting on all connected and supported keyboards, but not mice
  - LOGITECH_LED_ALL: change lighting on all connected and supported devices
- redPercentage: amount of red. Range is 0 to 100.
- greenPercentage: amount of green. Range is 0 to 100.
- bluePercentage: amount of blue. Range is 0 to 100.

Return value

If the function succeeds, it returns true. Otherwise false.
The function will return false if **LogiLedInit**() hasn't been called or because the user chose to opt out via a setting in the Logitech Gaming Software.

Remarks

DO NOT call this function immediately after LogiLedInit(). Instead leave a little bit of time after LogiLedInit().
For devices that only support a single color, the highest percentage value given of the three colors will define the brightness.

LogiLedRestoreLighting

The **LogiLedRestoreLighting**() function restores the last saved lighting. It should be called after a temporary effect is finished. For example if flashing a red warning sign for a few seconds, you would call the **LogiLedRestoreLighting** () function right after the warning effect is finished.

```
bool LogiLedRestoreLighting(int deviceType);
```

Parameters

- deviceType
  - LOGITECH_LED_MOUSE: change lighting on all connected and supported mice, but not keyboards
  - LOGITECH_LED_KEYBOARD: change lighting on all connected and supported keyboards, but not mice
  - LOGITECH_LED_ALL: change lighting on all connected and supported devices

Return value

If the function succeeds, it returns true. Otherwise false.
The function will return false if **LogiLedInit**() hasn't been called or because the user chose to opt out via a setting in the Logitech Gaming Software.

LogiLedFlashLighting

The **LogiLedFlashLighting** () function saves the current lighting, plays the flashing effect on the targeted devices and, finally, restores the saved lighting.

```
bool LogiLedFlashLighting (int deviceType , int redPercentage, int greenPercentage, int
bluePercentage, int milliSecondsDuration, int milliSecondsInterval);
```

Parameters

- deviceType
  - LOGITECH_LED_MOUSE: change lighting on all connected and supported mice, but not keyboards
  - LOGITECH_LED_KEYBOARD: change lighting on all connected and supported keyboards, but not mice
  - LOGITECH_LED_ALL: change lighting on all connected and supported devices
- redPercentage: amount of red. Range is 0 to 100.
- greenPercentage: amount of green. Range is 0 to 100.
- bluePercentage: amount of blue. Range is 0 to 100.
- milliSecondsDuration : duration of the effect in milliseconds
- milliSecondsInterval : duration of the flashing interval in milliseconds

Return value

If the function succeeds, it returns true. Otherwise false.
The function will return false if **LogiLedInit**() hasn't been called or because the user chose to opt out via a setting in the Logitech Gaming Software.

LogiLedPulseLighting

The **LogiLedPulseLighting** () function saves the current lighting, plays the pulsing effect on the targeted devices and, finally, restores the saved lighting.

```
bool LogiLedPulseLighting(int deviceType, int redPercentage, int greenPercentage, int
bluePercentage, int milliSecondsDuration);
```

Parameters

- deviceType
    - LOGITECH_LED_MOUSE: change lighting on all connected and supported mice, but not keyboards
    - LOGITECH_LED_KEYBOARD: change lighting on all connected and supported keyboards, but not mice
    - LOGITECH_LED_ALL: change lighting on all connected and supported devices
- redPercentage: amount of red. Range is 0 to 100.
- greenPercentage: amount of green. Range is 0 to 100.
- bluePercentage: amount of blue. Range is 0 to 100.
- milliSecondsDuration : duration of the effect in milliseconds

Return value

If the function succeeds, it returns true. Otherwise false.
The function will return false if **LogiLedInit**() hasn't been called or because the user chose to opt out via a setting in the Logitech Gaming Software.

LogiLedShutdown

The **LogiLedShutdown** () function restores the last saved lighting and frees memory used by the SDK.

```
void LogiLedShutdown();
```

End-User License Agreement for Logitech Gaming LED SDK

This End-User License Agreement for Logitech Gaming LED 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 LED software development kit, which includes computer software and related media and documentation (hereinafter "Logitech Gaming LED SDK"). By using this Logitech Gaming LED 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 LED 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 LED 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 LED 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 LED 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 LED 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 LED 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 LED 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 LED 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 LED SDK and all intellectual property rights therein. The Logitech Gaming LED 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 LED 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 LED 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 LED 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 LED 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 LED 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 LED SDK outside of the United States, you agree that you will not re-export the Logitech Gaming LED 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 LED 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 LED SDK shall automatically terminate and the Logitech Gaming LED SDK must be returned to Logitech or all copies of the Logitech Gaming LED SDK destroyed. You may also terminate this Agreement at any time by destroying all copies of the Logitech Gaming LED 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 LED 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 LED SDK and supersedes all prior or contemporaneous understandings, communications or agreements, written or oral, regarding such subject matter.