Contents

Overview

The Cooler Master Maker Toolbox is a software development kit that gives you complete access to the code behind LED lighting on all of our peripheral products. With the controls in your hands and the ability to retrieve system data from your PC, your lighting can be programmed to change, for example, according to the music you play or the speed of your processor. Create a whole host of effects using basic C++ knowledge.

Share your profiles with other Cooler Master peripheral owners at http://makerhub.coolermaster.com

SDK Package

Example Folder:

(I) Example directory content

1. **Sample execution file**
2. **Sample source code;**
3. **Sample description**

(II) SDK directory content

1. **CoolerMaster LED Table.xls: LED Matrix (7 row x 24 Column)**
2. **x86 directory: dynamic link libraries, Lib file and header file**

Requirements

Windows 7 (32-bit and 64-bit);

Windows 8.1 (32-bit and 64-bit);

Windows 10 (32-bit and 64-bit).

Supported Devices

Device	Keyboard	Mouse	Headset	Accessory
	MasterKeys Pro L MasterKeys Pro M MasterKeys Pro S MasterKeys Pro L White MasterKeys Pro M White MasterKeys Pro S White MK750	MasterMouse Pro L MasterMouse Pro S MM520 MM530		

Reference

SDK Related Definitions

LED Matrix SIZE

MAX_LED_RO:7

MAX_LED_COLUMN:24

struct KEY_COLOR {

BYTE r;

BYTE g;

BYTE b;

};

struct COLOR_MATRIX {

Description: set/store entire LED Color structure

KEY_COLOR KeyColor[MAX_LED_ROW][MAX_LED_COLUMN];

};

enum EFF_INDEX {

Description: set/store entire LED Color structure Special effects list

EFF_FULL_ON = 0, EFF_BREATH = 1,

EFF_BREATH_CYCLE = 2 , EFF_SINGLE = 3,

EFF_WAVE = 4, EFF_RIPPLE = 5,

EFF_CROSS = 6, EFF_RAIN = 7,

EFF_STAR = 8, EFF_SNAKE = 9,

EFF_REC = 10,

EFF_INDICATOR = 13

EFF_MULTI_1 = 0xE0,

EFF_MULTI_2 = 0xE1, EFF_MULTI_3 = 0xE2,

EFF_MULTI_4 = 0xE3, EFF_OFF = 0xFE};

System data related function

TCHAR * GetNowTime()

Description: Obtain current system time

Function name : GetNowTime

variable:

return : TCHAR : string index format is %Y %m/%d %H:%M %S

note:

LONG GetNowCPUUsage(DWORD * pErrorCode = NULL)

Description: obtain current CPU usuage ratio

Function name: GetNowCPUUsage

variable : DWORD * pErrorCode : return the error code

returns : LONG : 0 ~ 100 integer

note :

DWORD GetRamUsage()

Description: Obtain current RAM usuage ratio

Function name: GetRamUsage

variable :

returns : DWORD : 0 ~ 100 integer

note :

float GetNowVolumePeekValue()

Description: Obtain current volume

Function name : GetNowVolumePeekValue

variable :

returns : float : 0 ~ 1 float number

note :

Device operation function

Void SetControlDevice(DEVICE_INDEX devIndex)

Description: set default operating device

Function name: SetControlDevice

variable : DEVICE_INDEX: device list

DEV_MKeys_L,

DEV_MKeys_S,

DEV_MKeys_L_White

DEV_MKeys_M_White

DEV_MMouse_L

DEV_MMouse_S

DEV_MKeys_M

DEV_MKeys_S_White

DEV_MM520

DEV_MM530

DEV_MK750

returns :

note :

bool IsDevicePlug(DEVICE_INDEX devIndex = DEV_DEFAULT**)**

Description: verify if the deviced is plugged in

Function name : IsDevicePlug

variable : DEVICE_INDEX devIndex : target device

returns: bool : true plugged in，false not plugged in

note :

LAYOUT_KEYOBARD GetDeviceLayout(DEVICE_INDEX devIndex = DEV_DEFAULT**)**

Description: Obtain current device layout

Function name: GetDeviceLayout

variable : DEVICE_INDEX devIndex : target device

returns: LAYOUT_KEYOBARD List:

currently 3 LAYOUT_UNINIT ,

LAYOUT_US , LAYOUT_EU

note:

bool EnableLedControl(bool bEnable , DEVICE_INDEX devIndex = DEV_DEFAULT**)**

Description: set control over device's LED

Function name: EnableLedControl

variable : bool bEnable:

true Controlled by SW,

false Controlled by FW,

DEVICE_INDEX devIndex : target device

returns : bool : true Success,false Fail

note : **User could switch the effects when the function send the false value.**

bool SwitchLedEffect(EFF_INDEX iEffectIndex , DEVICE_INDEX devIndex =

DEV_DEFAULT**)**

Description: switch device current effect

Function name: SwitchLedEffect

variable : EFF_INDEX iEffectIndex: index value of the effect

DEVICE_INDEX devIndex : target device

returns : bool : true Success，false Fail

note :

bool SetFullLedColor(BYTE r, BYTE g, BYTE b ,DEVICE_INDEX devIndex =

DEV_DEFAULT **)**

Description: set entire keyboard LED **one** color

Function name : SetFullLedColor

variable : BYTE r :red, BYTE g :green, BYTE b :blue

DEVICE_INDEX devIndex : target device

returns : bool : true Success，false Fail

note :

bool SetAllLedColor(COLOR_MATRIX colorMatrix,DEVICE_INDEX devIndex =

DEV_DEFAULT **)**

Description: Set Keyboard "every LED" color

Function name: SetAllLedColor

variable : COLOR_MATRIX colorMatrix:structure,

fill up RGB value according to LED Table

DEVICE_INDEX devIndex : target device

returns : bool : true Success,false Fail

note :

bool SetLedColor(int iRow, int iColumn, BYTE r, BYTE g, BYTE b,DEVICE_INDEX

devIndex = DEV_DEFAULT **)**

Description: Set single Key LED color

Function name: SetLedColor

variable: int iRow: row,

int iColumn:column BYTE r :red,

BYTE g :green, BYTE b :blue

DEVICE_INDEX devIndex : target device

returns : bool : true Success，false Fail

note :

bool EnableKeyInterrupt (bool bEnable,DEVICE_INDEX devIndex = DEV_DEFAULT **)**

Description: To enable the call back function

Function name: EnableKeyInterrupt

variable: bool bEnable: true enable ，false disable

DEVICE_INDEX devIndex : target device

returns : bool : true Success，false Fail

note : will call the call back function of SetKeyCallBack()

void SetKeyCallBack(KEY_CALLBACK callback,DEVICE_INDEX devIndex =

DEV_DEFAULT**);**

Description: Setup the call back function of button

Function name: SetKeyCallBack

variable: KEY_CALLBACK callback call back setup，

please reference the def of KEY_CALLBACK

DEVICE_INDEX devIndex : target device

note :

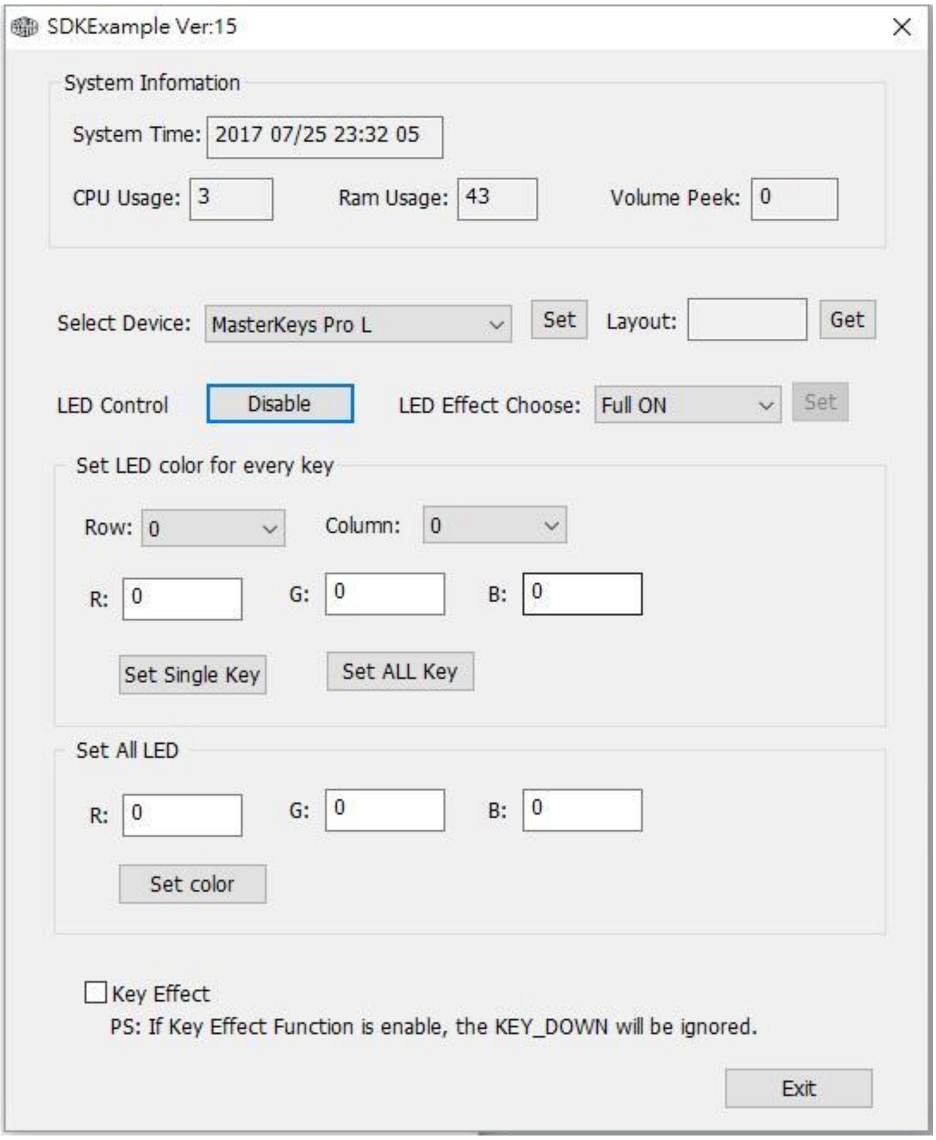
typedef void (CALLBACK * KEY_CALLBACK)(int iRow, int iColumn, bool bPressed);

Description: User could setup the callback function. It will return the status of button's position when the button status change.

SDK example

SDK examples illustrate

Examples UI screen



SDKExample Ver:15

System Infomation

System Time: 2017 07/25 23:32 05

CPU Usage: 3

Ram Usage: 43

Volume Peek: 0

Select Device: MasterKeys Pro L

Layout:

LED Control

LED Effect Choose: Full ON

Set LED color for every key

Row: 0

Column: 0

R: 0

G: 0

B: 0

Set All LED

R: 0

G: 0

B: 0

□ Key Effect

PS: If Key Effect Function is enable, the KEY_DOWN will be ignored.

Development environment: VC ++ 2008

1. System Information: for the user's computer to fetch the local system time, CPU usage, memory usage percentage, the current playback volume percentage.
2. Select Device: to select the device that you want to control, the default option is MasterKeys Pro L.
3. LED Control: can choose from enable and disable, in the disable state can switch effects; in the enable state can setup the keyboard LED color
4. Set LED Color for every Key: is allowed to set different colors of each key, there are two ways to set up. One is to set a single Key; the other one is to set all keys on the keyboard to specified / different color. Please use the drop-down menu and select Row Column with "CoolerMaster LED Table.xls" table to determine the location specified color.
5. Set All Led: set the whole keyboard as a single color quickly.
6. Set the Key effect : if it enable and the button status change, the Led of key will light.