LPU23X F/W Update API

user manual

V8.0

table of contents

This document describes how to use the Application Programming Interface (API) to update the firmware of the LPU23X card reader (MSR).

If you use version 3.x, NDM(The Next Device Manager) must be running on your system.

From 4.0, you can select a device-io mode by "lpu230_fw.ini".(dual mode)

API Baisc information.

	value	etc
folder	-	User definition item.
File name	tg_lpu237_fw.dll	Version 4.2.
type	win32 regular dynamic linked library(dll)	
Sub component	tg_rom.dll is sub component of tg_lpu237_fw.dll.	tg_rom.dll and tg_lpu237_fw.dll exist in the same folder.

The exported functions of API(tg_lpu237_fw.dll)

The exported name	prototype	Description.
LPU237_fw_on	DWORD WINAPI LPU237_fw_on()	Initializes dll.
LPU237_fw_off	DWORD WINAPI LPU237_fw_off()	Terminates dll inner worker.
LPU237_fw_get_list_w	DWORD WINAPI LPU237_fw_get_list_w (WCHAR *ssDevPaths)	gets the connected MSR list. unicode version.
LPU237_fw_get_list_a	DWORD WINAPI LPU237_fw_get_list_a (CHAR *ssDevPaths)	gets the connected MSR list. unicode type. Multi Byte Code Set(MBCS) version.
LPU237_fw_open_w	HANDLE WINAPI LPU237_fw_open_w (CONST WCHAR *sDevPath)	open the channel of MSR. Unicode version.
LPU237_fw_open_a	HANDLE WINAPI LPU237_fw_open_a (CONST CHAR sDevPath)	open the channel of MSR. MBCS version.
LPU237_fw_close	DWORD WINAPI LPU237_fw_close (HANDLE hDev)	close the channel of MSR.
LPU237_fw_msr_save_setting	DWORD WINAPI LPU237_fw_msr_save_setting(HANDLE hDev)	Save MSR setting to memory. From V3.2 or later, **LPU237_fw_msr_update_x() recovers a MSR setting automatically.**
LPU237_fw_msr_recover_setting	DWORD WINAPI LPU237_fw_msr_recover_setting(HANDLE hDev)	**From v3.2 or later, This function dosen't anything. Only for compatiblitiy exist.**
LPU237_fw_msr_get_id	DWORD WINAPI LPU237_fw_msr_get_id (HANDLE hDev, BYTE *sId)	Gets a device ID(16 bytes).
LPU237_fw_msr_get_name	DWORD WINAPI LPU237_fw_msr_get_name (HANDLE hDev, BYTE *sName)	Get MSR name.

LPU237_fw_msr_get_version	DWORD WINAPI LPU237_fw_msr_get_version (HANDLE hDev, BYTE *sVersion)	Get a device 4 bytes FW verion. bytes)
LPU237_fw_msr_get_version_major	DWORD WINAPI LPU237_fw_msr_get_version_major (const BYTE *sVersion)	From 4 bytes FW verion, Get major version.
LPU237_fw_msr_get_version_minor	DWORD WINAPI LPU237_fw_msr_get_version_minor (const BYTE *sVersion)	From 4 bytes FW verion, Get minor version.
LPU237_fw_msr_cancel_update	DWORD WINAPI LPU237_fw_msr_cancel_update()	Stop FW update.
LPU237_fw_msr_update_w	DWORD WINAPI LPU237_fw_msr_update_w (const BYTE *sId, DWORD dwWaitTime, const WCHAR *sRomFileName, DWORD dwIndex);	Update MSR FW by sync-operation. Unicode version.
LPU237_fw_msr_update_a	DWORD WINAPI LPU237_fw_msr_update_a (const BYTE *sId, DWORD dwWaitTime, const CHAR *sRomFileName, DWORD dwIndex);	Update MSR FW by sync-operation. MBCS version.
LPU237_fw_msr_update_callback_w	DWORD WINAPI LPU237_fw_msr_update_callback_w (const BYTE *sId, type_lpu237_fw_callback cbUpdate, void *pUser, const WCHAR *sRomFileName, DWORD dwIndex)	Starts the MSR FW undate by async-callback method. Unicode version.
LPU237_fw_msr_update_callback_a	DWORD WINAPI LPU237_fw_msr_update_callback_a (const BYTE *sId, type_lpu237_fw_callback cbUpdate, void *pUser, const CHAR *sRomFileName, DWORD dwIndex)	Starts the MSR FW undate by async-callback method. MBCS version.
LPU237_fw_msr_update_wnd_w	DWORD WINAPI LPU237_fw_msr_update_wnd_w (const BYTE *sId, HWND hWnd, UINT uMsg, const WCHAR *sRomFileName, DWORD dwIndex)	Starts the MSR FW undate by async-message method. Unicode version.
LPU237_fw_msr_update_wnd_a	DWORD WINAPI LPU237_fw_msr_update_wnd_w (const BYTE *sId, HWND hWnd, UINT uMsg, const CHAR *sRomFileName, DWORD dwIndex)	Starts the MSR FW undate by async-message method. MBCS version.
LPU237_fw_rom_load_w	DWORD WINAPI LPU237_fw_rom_load_w (const WCHAR *sRomFileName)	Read FW from rom-file. Unicode version.
LPU237_fw_rom_load_a	DWORD WINAPI LPU237_fw_rom_load_a (const CHAR *sRomFileName)	Read FW from rom-file. MBCS version.
LPU237_fw_rom_get_index_w	DWORD WINAPI LPU237_fw_rom_get_index_w (const WCHAR *sRomFileName, const BYTE *sName, const BYTE *sVersion)	Get FW index that is updatable of Rom-file. Unicode version.
LPU237_fw_rom_get_index_a	DWORD WINAPI LPU237_fw_rom_get_index_a (const CHAR *sRomFileName, const BYTE *sName, const BYTE *sVersion)	Get FW index that is updatable of Rom-file. MBCS version.

- the exported function with "_w" must be used in the unicode project.
- the exported function with "_a" must be used in the MBCS project.
- In this document description, "_w" and "_a" are removed from the name of function.

(ex – LPU237_fw_rom_load_w() and LPU237_fw_rom_load_a() are called LPU237_fw_rom_load())

The defintion of return value.

This values are the return of API function. Or

the second parameter of callback funtion. The callback function is parameter of LPU237_fw_msr_update_callback(). Or

the LPARAM of window handler in LPU237_fw_msr_update_wnd().

Symbol	Hexcimal value(double word)	Description
LPU237_FW_RESULT_SUCCESS	0x00000000	success.
LPU237_FW_RESULT_ERROR	0xFFFFFFFF	error.
LPU237_FW_RESULT_CANCEL	0xFFFFFFFE	Canceled by LPU237_fw_msr_cancel_update()
LPU237_FW_RESULT_TIMEOUT	0xFFFFFFFC	Time-out in sync-update.
LPU237_FW_RESULT_NO_MSR	0xFFFFFFFB	None MSR

The definition of update-status

This value describes the current update status.

In LPU237_fw_msr_update_wnd(), this value is the window handle WPARAM.

In LPU237_fw_msr_update_callback(), the 3'rd parameter of callback function.

In normal processing, LPU237 send LPU237_FW_WPARAM_FOUND_BL one-time, LPU237_FW_WPARAM_SECTOR_ERASE one-time, LPU237_FW_WPARAM_SECTOR_WRITE six-times and LPU237_FW_WPARAM_COMPLETE one-time.

Symbol	Hexcimal value(double word)	Description
LPU237_FW_WPARAM_COMPLETE	0	Success update.
LPU237_FW_WPARAM_FOUND_BL	1	Found bootloader(BL) OK.
LPU237_FW_WPARAM_SECTOR_ERASE	2	Erase all MSR flash memory.
LPU237_FW_WPARAM_SECTOR_WRITE	3	Writes a MSR flash memory sector.

The defintion of callback function.

Callback function is used in the 2'nd parameter of LPU237_fw_msr_update_callback(). Prototype is

*DWORD (WINAPI *type_lpu237_fw_callback)(void*,DWORD,DWORD)*

- 1'st parameter – user memory pointer
- 2'nd parameter – the result of current step.(LPU237_FW_RESULT_SUCCESS, LPU237_FW_RESULT_ERROR or LPU237_FW_RESULT_CANCEL)
- 3'rd parameter – the description of currnet step.(LPU237_FW_WPARAM_COMPLETE, LPU237_FW_WPARAM_FOUND_BL, LPU237_FW_WPARAM_SECTOR_ERASE or LPU237_FW_WPARAM_SECTOR_WRITE).

This callback is executed by the inner worker thread(WTH).

The basic progamming sequence.

1. Call LPU237_fw_on(). - initialize dll.
2. Call LPU237_fw_get_list(). - get the MSR path list.
3. Call LPU237_fw_open(). - open the channel of MSR.
4. Call LPU237_fw_msr_get_name(). - get the current MSR name.
5. Call LPU237_fw_msr_get_version(). - get the current MSR FW verison.
6. Call LPU237_fw_msr_get_id(). - get the current MSR 16 byets ID.
7. Call LPU237_fw_msr_save_setting(). - save the currnet MSR settings.
8. Call LPU237_fw_rom_load(). - load FW rom file.
9. Call LPU237_fw_rom_get_index(). - Get FW index that is updatable of Rom-file.
10. Call LPU237_fw_msr_update() , LPU237_fw_msr_update_callback() or LPU237_fw_msr_update_wnd() . - starts MSR FW update. Waits the end of updateing FW. Or cancel by LPU237_fw_msr_cancel_update().
11. Call LPU237_fw _get_list() . - get the MSR path list.
12. Call LPU237_fw _open() . - open the channel of MSR.
13. Call LPU237_fw_msr_recover_setting() . - ~~recovers the saved settings.~~ From V3.2 or later, No need.
14. Call LPU237_fw_close() . - close channel.
15. Call LPU237_fw_off(). - terminates dll inner worker.

The important component.

- **worker thread** - API use WTH(inner worker thread) for supporting async-IO. LPU237_fw_on() and LPU237_fw_off() are that starts ot terminates WTH. Therefore LPU237_fw_on() must be called before another function. And LPU237_fw_off() have to be called after terminating your work.
- **BoorLoader(BL)** - When BL of MSR is executed by calling one of LPU237_fw_msr_update(), LPU237_fw_msr_update_callback() or LPU237_fw_msr_update_wnd(), MSR is physically disconnected from the computer USB port and a new USB device called BL is connected to the USB port. BL writes a new FW by the request of WTH, and if the update is successful, the new FW is executed, the BL is physically removed from the USB port, and the updated MSR is connected to the USB port. At this time, the MSR setting is changed to the default value. If an error occurs while writing a new FW, it stays in the BL state. BL and MSR have different USB PIDs. Therefore, from the point of view of the PC, it is recognized as completely different hardware. If the PC has never ever recognized the BL, in the process of converting from the MSR to the BL, the BL goes through the USB device installation process of the PC, which can take from a few seconds to 10 minutes. This time delay can be recognized as stopping the program from the user's point of view, so if BL is not found even after about 5 seconds after switching from MSR to BL, cancel the update and have to retry the update with ID value 0 periodically.
- **Rom File** – It contains one more FWs inside, the name of the device to which the FW is

applicable, the version value, and the conditions under which the FW can be applied. The current MSR ROM file basically has two FWs stored. The FW applicable to the name "callisto" is stored in index 0, and the FW applicable to the name "ganymede" is stored in index 1, and the update condition is that the FW version of the ROM file is higher than or equal to the current MSR FW version.The LPU237_fw_rom_get_index() function checks whether there is a FW that satisfies the update condition in the ROM file.

In one MSR connected PC, updates FW by sync-method.

1. Call LPU237_fw_on().
2. Gets MSR path list by LPU237_fw_get_list(), open channel by LPU237_fw_open().
3. Gets MSR name by LPU237_fw_msr_get_name().
4. Gets MSR FW version by LPU237_fw_msr_get_version().
5. Gets MSR ID by LPU237_fw_msr_get_id().
6. Save the currnet MSR setting to memory by LPU237_fw_msr_save_setting().
7. Load rom-file by LPU237_fw_rom_load().
8. Gets FW index by LPU237_fw_rom_get_index().
9. Make user the thread. It calls LPU237_fw_msr_update(). LPU237_fw_msr_update() function does not return until the FW update is normally completed, operation timeout, error occurs, or cancellation.
10. **If FW update is normally completed, MSR setting will be recovered automatically.**
11. In updating , If error , timeout or cancellation occures, since MSR is in BL state, BL path cannot be obtained with LPU237_fw_get_list(), and communication with MSR is not possible. In this case, you can call LPU237_fw_msr_update() again without MSR ID, WTH automatically searches for the connected BL and restarts the update. In timeout case, you call LPU237_fw_msr_cancel_update() and retry.

In one MSR connected PC, updates FW by async-method.

Callback

1. Call LPU237_fw_on().
2. Gets MSR path list by LPU237_fw_get_list(), open channel by LPU237_fw_open().
3. Gets MSR name by LPU237_fw_msr_get_name().
4. Gets MSR FW version by LPU237_fw_msr_get_version().
5. Gets MSR ID by LPU237_fw_msr_get_id().
6. Save the currnet MSR setting to memory by LPU237_fw_msr_save_setting().
7. Load rom-file by LPU237_fw_rom_load().
8. Gets FW index by LPU237_fw_rom_get_index().
9. Call LPU 237_fw_msr_update_callback(). Callback function will be called by WTH periodically for announcing update progress.
10. **If FW update is normally completed, MSR setting will be recovered automatically.**
11. In updating , If error , timeout or cancellation occures, since MSR is in BL state, BL path cannot be obtained with LPU237_fw_get_list(), and communication with MSR is not possible. In this case, you can call LPU237_fw_msr_update_callback() again without MSR ID, WTH automatically searches for the connected BL and restarts the update.

Messageing

1. Call LPU237_fw_on().
2. Gets MSR path list by LPU237_fw_get_list(), open channel by LPU237_fw_open().
3. Gets MSR name by LPU237_fw_msr_get_name().
4. Gets MSR FW version by LPU237_fw_msr_get_version().
5. Gets MSR ID by LPU237_fw_msr_get_id().
6. Save the currnet MSR setting to memory by LPU237_fw_msr_save_setting().
7. Load rom-file by LPU237_fw_rom_load().
8. Gets FW index by LPU237_fw_rom_get_index().
9. Call LPU 237_fw_msr_update_wnd(). WTH will post a message periodically for announcing update progress.
10. **If FW update is normally completed, MSR setting will be recovered automatically.**
11. In updating , If error , timeout or cancellation occures, since MSR is in BL state, BL path cannot be obtained with LPU237_fw_get_list(), and communication with MSR is not possible. In this case, you can call LPU237_fw_msr_update_wnd() again without MSR ID, WTH automatically searches for the connected BL and restarts the update.

LPU237_fw_on

Creates and executes WTH. You must be called before another function.
If this function is called in DllMain(), it may occur the deadlock.

Prototype

DWORD WINAPI LPU237_fw_on()

parameters

none

return

condition	value	etc
success	LPU237_FW_RESULT_SUCCESS	

LPU237_fw_off

Terminates WTH. You must call this function before terminating main program.

Prototype

DWORD WINAPI LPU237_fw_off()

parameters

none

return

condition	value	etc
success	LPU237_FW_RESULT_SUCCESS	Always

LPU237_fw_get_list

Get the path list of MSR (VID 0x134B, PID 0x0206). Path list is multi-zero string type.

Prototype

DWORD WINAPI LPU237_fw_get_list_w(WCHAR *ssDevPaths) - unicode version.
DWORD WINAPI LPU237_fw_get_list_a(CHAR *ssDevPaths) - MBCS version.

parameters

- ssDevPaths – [in/out] the buffer that saves MSR path. If this is zero, return the size of buffer. (byte unit)

return

condition	value	etc
ssDevPaths is 0	the buffer that saves MSR path	Byte unit
ssDevPaths isn't zero and process is success.	The number of MSR.	ssDevPaths is multi-zero strinf type.
error	LPU237_FW_RESULT_ERROR	

- zero string – A way to mark the end of a string as 0 . In Unicode, each character is 2 bytes, so 0 is entered twice. (Standard form of string in Windows API and C language)
- multi-zero string - A method of concatenating multiple zero strings consecutively and adding 0 to the end to indicate the end of the strings. In this method, 0 for the last string and 0 to indicate the end of the string appear consecutively at the end. In Unicode, each character is 2 bytes, so the last zeros goes in 4 times.

ex) if the numbr of MSR is two and each MSR path are "ab" and "12", ssDevPaths value is

unicode version

offset	value		etc
0	0x61	Unicode 'a'	
1	0x00		
2	0x62	Unicode 'b'	
3	0x00		
4	0x00	Unicode NULL	The end of "ab" string
5	0x00		
6	0x31	Unicode '1'	
7	0x00		
8	0x32	Unicode '2'	
9	0x00		
10	0x00	Unicode NULL	The end of "12" string
11	0x00		
12	0x00	Unicode NULL	The end of multi-zero string.
13	0x00		

MBCS version

offset	value		etc
0	0x61	'a'	
1	0x62	'b'	
2	0x00	NULL	The end of "ab" string
3	0x31	'1'	
4	0x32	'2'	
5	0x00	NULL	The end of "12" string
6	0x00	NULL	The end of multi-zero string.

LPU237_fw_open

Open channel.

Prototype

HANDLE WINAPI LPU237_fw_open_w(CONST WCHAR *sDevPath) - unicode version.
HANDLE WINAPI LPU237_fw_open_a(CONST CHAR *sDevPath) - MBCS version.

parameters

- sDevPath – [in] MSR path, zero-string type.

return

condition	value	etc
success	The handle of MSR.	
error	INVALID_HANDLE_VALUE	long type address value(-1. Defined by Microsoft)

LPU237_fw_close

Close channel.

Prototype

DWORD WINAPI LPU237_fw_close(HANDLE hDev)

parameters

- hDev – [in] MSR handle.(from LPU237_FW_open())

return

condition	value	etc
success	LPU237_FW_RESULT_SUCCESS	
error	LPU237_FW_RESULT_ERROR	

LPU237_fw_msr_save_setting

Save the currnet setting to memory.

Prototype

DWORD WINAPI LPU237_fw_msr_save_setting(HANDLE hDev)

parameters

- hDev – [in] MSR handle.(from LPU237_FW_open())

return

condition	value	etc
success	LPU237_FW_RESULT_SUCCESS	
error	LPU237_FW_RESULT_ERROR	

LPU237_fw_msr_recover_setting

From v3.2 or later, the setting is recoverd automatically. Therefore this function dose nothing.

Prototype

DWORD WINAPI LPU237_fw_msr_recover_setting(HANDLE hDev)

parameters

- hDev – [in] MSR handle.(from LPU237_FW_open())

return

condition	value	etc
success	LPU237_FW_RESULT_SUCCESS	

LPU237_fw_msr_get_id

Gets MSR 16 bytes ID.

Prototype

DWORD WINAPI LPU237_fw_msr_get_id(HANDLE hDev, BYTE *sId)

parameters

- hDev – [in] MSR handle.(from LPU237_FW_open())
- sId – [out] the buffer point that saves the 16 byets ID.

return

condition	value	etc
success	LPU237_FW_RESULT_SUCCESS	
success	If sId is zero, return 16.	
error	LPU237_FW_RESULT_ERROR	

Each MSR have a unique ID.

LPU237_fw_msr_get_name

Gets the 16 bytes MSR name.

Prototype

DWORD WINAPI LPU237_fw_msr_get_name(HANDLE hDev, BYTE *sName)

parameters

- hDev – [in] MSR handle.(from LPU237_FW_open())
- sName – [out] the buffer point that saves the 16 byets name.

return

condition	value	etc
success	LPU237_FW_RESULT_SUCCESS	
success	If sName is 0, return 16.	
error	LPU237_FW_RESULT_ERROR	

ex) If name is "callisto" ,

offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
값	c	a	l	l	i	s	t	o	SP	SP	SP	SP	SP	SP	SP	SP
hex	63	61	6c	6c	69	73	74	6f	20	20	20	20	20	20	20	20

LPU237_fw_msr_get_version

Gets 4 byets version.

Prototype

DWORD WINAPI LPU237_fw_msr_get_version(HANDLE hDev, BYTE *sVersion)

parameters

- hDev – [in] MSR handle.(from LPU237_FW_open())
- sVersion – [out] the buffer point that saves the 4 byets version.

return

condition	value	etc
successs	LPU237_FW_RESULT_SUCCESS	
success	If sVersion is zer, return 4.	
error	LPU237_FW_RESULT_ERROR	

ex) If version is 3.14.0.4, sVersion contains

offset	value	description
0	3	major version.
1	14	minor version
2	0	bug fix version.
3	4	build version.

If the name of MSR is “callisto", major version is 3.

If the name of MSR is "ganymede" major version is 5.

From version 3.9.0.0 or later, the bugfix and build version of “callisto" is zero.

From version 5.2.0.0 or later, the bugfix and build version of “ganymede" is zero.

LPU237_fw_msr_get_version_major

Gets major version from 4 bytes version.

Prototype

DWORD WINAPI LPU237_fw_msr_get_version_major(const BYTE *sVersion)

parameters

- sVersion – [in] 4 bytes version buffer.

return

condition	value	etc
success	major version	
error	LPU237_FW_RESULT_ERROR	

LPU237_fw_msr_get_version_minor

Gets minor version from 4 bytes version.

Prototype

DWORD WINAPI LPU237_fw_msr_get_version_minor(const BYTE *sVersion)

parameters

- sVersion – [in] 4 bytes version buffer.

return

condition	value	etc
success	minor version	
error	LPU237_FW_RESULT_ERROR	

LPU237_fw_msr_cancel_update

Cancels the update process initiated by LPU237_fw_msr_update(), LPU237_fw_msr_update_callback(), or LPU237_fw_msr_update_wnd(). Returns LPU237_FW_RESULT_SUCCESS if current state is not update.

Prototype

DWORD WINAPI LPU237_fw_msr_cancel_update()

parameters

none.

return

condition	value	etc
success	LPU237_FW_RESULT_SUCCESS	
error	LPU237_FW_RESULT_ERROR	

LPU237_fw_msr_update

Starts the update processing with sync-method. This function does not return until the FW update is normally completed, operation timeout, error occurs, or cancellation.

Prototype

DWORD WINAPI LPU237_fw_msr_update_w(const BYTE *sId, DWORD dwWaitTime, const WCHAR *sRomFileName, DWORD dwIndex) - unicode version.
DWORD WINAPI LPU237_fw_msr_update_a(const BYTE *sId, DWORD dwWaitTime, const CHAR *sRomFileName, DWORD dwIndex) - MBCS version.

parameters

- sId – [in] 16 bytes ID buffer. If this value is zero, WTH finds BL and starts update automatically. Else LPU237_fw_msr_open() must be called successfully before calls this function.
- dwWaitTime – [in] update time-out. Unit mmsec. This value greater is then zero. or INFINITE(defined by Microsoft. 0xFFFFFFFF)
- sRomFileName – [in] rom file name.
- dwIndex – [in] the FW index value of sRomFileName. This value is greater then or equal to zero.

return

condition	value	etc
success	LPU237_FW_RESULT_SUCCESS	Update success
error	LPU237_FW_RESULT_ERROR	For updating, Error
error	LPU237_FW_RESULT_NO_MSR	None MSR.
error	LPU237_FW_RESULT_TIMEOUT	Timeout

LPU237_fw_msr_update_callback

Starts the update processing with callback async-method. If sId is not zero, this function will return after running BL.

Prototype

DWORD WINAPI LPU237_fw_msr_update_callback_w(const BYTE *sId, type_lpu237_fw_callback cbUpdate, void *pUser, const WCHAR *sRomFileName, DWORD dwIndex) - unicode version
DWORD WINAPI LPU237_fw_msr_update_callback_a(const BYTE *sId, type_lpu237_fw_callback cbUpdate, void *pUser, const CHAR *sRomFileName, DWORD dwIndex) - MBCS version.

parameters

- sId – [in] 16 bytes ID buffer. If this value is zero, WTH finds BL and starts update automatically. Else LPU237_fw_msr_open() must be called successfully before calls this function.
- cbUpdate – [in] For announcing the update progress, WTH will call this callback function periodically.
- pUser – [in] the first parameter of cbUpdate callbak function.
- sRomFileName – [in] rom file name.
- dwIndex – [in] the FW index value of sRomFileName. This value is greater then or equal to zero.

return

condition	value	etc
success	LPU237_FW_RESULT_SUCCESS	the starting update is success.
error	LPU237_FW_RESULT_ERROR	error
error	LPU237_FW_RESULT_NO_MSR	None MSR

LPU237_fw_msr_update_wnd

Starts the update processing with messaging async-method. If sId is not zero, this function will return after running BL.

Prototype

DWORD WINAPI LPU237_fw_msr_update_wnd_w(const BYTE *sId, HWND hWnd, UINT uMsg, const WCHAR *sRomFileName, DWORD dwIndex) - uncode version.
DWORD WINAPI LPU237_fw_msr_update_wnd_a(const BYTE *sId, HWND hWnd, UINT uMsg, const CHAR *sRomFileName, DWORD dwIndex) - MBCS version.

parameters

- sId – [in] 6 bytes ID buffer. If this value is zero, WTH finds BL and starts update automatically. Else LPU237_fw_msr_open() must be called successfully before calls this function.
- hWnd – [in] the target window handle of posting message by WTH.
- uMsg – [in] the posting message by WTH.
- sRomFileName – [in] rom file name.
- dwIndex – [in] the FW index value of sRomFileName. This value is greater then or equal to zero.

return

condition	value	etc
success	LPU237_FW_RESULT_SUCCESS	the starting update is success.
error	LPU237_FW_RESULT_ERROR	error
error	LPU237_FW_RESULT_NO_MSR	None MSR

the WPARM parameter of window handler is LPU237_FW_WPARAM_COMPLETE LPU237_FW_WPARAM_FOUND_BL, LPU237_FW_WPARAM_SECTOR_ERASE or LPU237_FW_WPARAM_SECTOR_WRITE .
The LPARAM parameter of window handler is LPU237_FW_RESULT_SUCCESS, LPU237_FW_RESULT_ERROR or LPU237_FW_RESULT_CANCEL.

LPU237_fw_rom_load

Load rom-file that conatins a FWs. If this function is success, you can do that the 1'st parameter of LPU237_fw_rom_get_index() is zero.

Prototype

DWORD WINAPI LPU237_fw_rom_load_w(const WCHAR *sRomFileName) - unicode version.
DWORD WINAPI LPU237_fw_rom_load_a(const CHAR *sRomFileName) - MBCS version.

parameters

- sRomFileName – [in] rom file name.

return

condition	value	etc
success	LPU237_FW_RESULT_SUCCESS	
error	LPU237_FW_RESULT_ERROR	

LPU237_fw_rom_get_index

Load rom-file that conatins a FWs, and get FW index that is updatable of Rom-file.

Prototype

DWORD WINAPI LPU237_fw_rom_get_index_w(const WCHAR *sRomFileName, const BYTE *sName, const BYTE *sVersion) - 유니코드 버전.
DWORD WINAPI LPU237_fw_rom_get_index_a(const CHAR *sRomFileName, const BYTE *sName, const BYTE *sVersion)- MBCS 버전.

parameters

- sRomFileName – [in] rom file name.
- sName – for updating, the target MSR name(from LPU237_fw_msr_get_name())
- sVersion - the 4 bytes version of target MSR.(from LPU237_fw_msr_get_version())

return

condition	value	etc
success	This value is greater then or equal to zero.	FW Index
error	LPU237_FW_RESULT_ERROR	

Log file

API can generate the log file.
It generates in a special condition. tg_lpu237_tools.dll generates a new log file when is loaded into memory.

Log file generation condition

1. In Logon user document folder, "Easyset₩lpu230" folder exist
2. In Logon user document folder, "Easyset₩lpu230₩log" folder exist.
3. In "Easyset₩lpu230" folder, lpu230_fw.ini exist.
4. In lpu230_fw.ini, [LogSetting] session exist.
5. In [LogSetting] session, the value of logenable key is 1 .

the folder of Log file

A log file is generated at "Easyset₩lpu230₩log". If tg_lpu237_fw.dll is loaded to memory at 2017/12/25, PM3H 45MIN 12SEC, log file name is tg_lpu237_fw_071225154512.txt(at version 3.x api071225154512.txt), log file format is text.

lpu230_fw.ini file

For setting log-file.

[LogSetting] session	Log setting session
logenable key	1 – generation log. Else not be generated.
loglevel key	This value must be 3.
logtimestemp key	1 - [MM-dd hh:mm:ss] is added to each log line.
logtimetick key	1 – the systemtick is added to each log line.
[control] session	etc session.
io ㅋ|	0(default) - If NDM is detected, a device-io operation is processed by NDM. Else, a device-io operation is processed by direct(windows API). 1 – a device-io operation is processed by direct(windows API). 2 - a device-io operation is processed by NDM.

Dual mode

By lpu230_fw.ini, You can select a device-io mode(auto, NDM or direct mode)

NDM mode

On NDM mode, General User applications structure layer.

mapper (lpu230.exe)	user MSR application	user KeyLock application1	user KeyLock application2	...
	tg_lpu237_dll.dll	tg_lpu237_ibutton.dll	tg_lpu237_ibutton.dll	
ng_DDL_hid.dll	NDM			
windows HID Driver				
LPU237 device (Physical device)				

Direct mode

On Direct mode, General User applications structure layer.

mapper (lpu230.exe)	user MSR application	user KeyLock application1	user KeyLock application2	...
ng_DDL_hid.dll	tg_lpu237_dll.dll	tg_lpu237_ibutton.dll	tg_lpu237_ibutton.dll	
windows HID Driver				
LPU237 device (Physical device)				

History

- 2017.07.12 – the first release. V1.0
- 2017.07.18 – add logging information.
- 2017.11.28 – V3.0. Supports NDM(the next device manager, ng_DevManager.exe).
  fix bug : always LPU237_fw_rom_get_index() return 0. need installer package v1.8.55 or later.
- 2019.11.19 – V3.1. Add the magnetic card reading direction parameter. Requested by POSBANK.
- 2022.11.10 – v3.2. Add Combination & i-button remove parameters After updating, setting recover automatically. (No need LPU237_fw_msr_recover_setting())
- 2023.09.21 – V4.0, For supporting lpu238 device and and dual mode.
- 2023.10.24 – V4.1, For supporting i-button range functionality.
- 2024.02.28 – V4.2, Rebuild due to dependency library changes without functional changes.