

NLA50-USB(C#)ユーザーガイド

UM-110 Ver.01

DLLソフトウェアライブラリを使用した操作

一、概要

DLLライブラリを介したUSB積層信号灯の制御

二、開発環境

項目	内容		概要
プログラミング言語	C#		—
開発用ソフトウェア	Visual Studio 2017® 以上		.NET Framework 4.0 以降をインストールする必要があります。
必要な外部ファイル	ソフトウェアライブラリ (DLL)	USB_NIKKI_Tower.dll	本製品を制御するために使用されるメインライブラリ。
		USB_NIKKI_Tower.lib	静的ライブラリとして使用する場合に必要なファイル（動的ライブラリの場合は不要）。
		USB_NIKKI_Tower_DLL.h	ライブラリの関数を宣言するヘッダファイル。
		USB_NIKKI_Definition.h	パラメータを定義するヘッダファイル。
	Windows標準	HID.dll	Windows に標準で搭載されているファイル。
		setupapi.dll	
	その他	vcruntime140.dll	DLL を使用して開発する場合、Microsoft Visual C++® 2015 再頒布可能パッケージのインストールが必要です。

三、API一覧

番号	関数	概要
1	Start_HID	USB通信の開始
2	Stop_HID	USB通信の終了
3	LED_Control	LEDユニットの色による制御（点灯モード）
4	Led_All_Off	すべてのLEDユニットを消灯
5	Buzzer_Control	選択したモードによるブザー制御（モード／周波数／音量）
6	Buzzer_Off	ブザー停止
7	Reset_Device	リセット：すべてのLED消灯およびブザー停止
8	Get_Firmware_Version	ファームウェアバージョンの取得
9	Get_Dll_Version	DLLバージョンの取得

四、API 詳細

(一) Start_HID

項目	説明
名称	int Start_HID()
機能概要	USB通信 : USB積層信号灯への接続
戻り値	成功時は0を返します。 エラー時は負の値を返します。 詳細は「エラーコード」表を参照してください。
注意事項	この関数は、内部でUSB通信用のデバイスハンドルを取得します。 デバイスハンドルを解放するには、Stop_HID 関数を呼び出す必要があります。 この関数は、1つ以上のデバイスハンドルを取得することはできません。

(二) Stop_HID

項目	説明
名称	int Stop_HID ()
機能概要	USB積層信号灯とのUSB通信を終了
戻り値	成功時は0を返します。 エラー時は負の値を返します。 詳細は「エラーコード」表を参照してください。

(三)LED_Control

項目	説明
名称	int LED_Control (int R, int Y, int G, int B, int W)
機能概要	各LEDカラーの点灯モードを設定し、USB積層信号灯のLEDを点灯させます。 ブザーは現在の状態を維持します。
パラメータ	R（赤）、Y（黄）、G（緑）、B（青）、W（白）：制御するLEDモードを指定します。
戻り値	エラー時は負の値を返します。 詳細は「エラーコード」表を参照してください。
注意事項	この関数を呼び出す前に、次の関数「Start_HID」を呼び出す必要があります。
プログラム例	<pre> int result = DLL_Function.Start_HID(); if(result==0) { DLL_Function.LED_Control(Parameter.RLED_Mode0, Parameter.YLED_Mode1,Parameter.GLED_Mode2, Parameter.BLED_Mode3, Parameter.WLED_Mode5); //RedMode0 , YellowMode3, GreenMode3, BlueMode0, WhiteMode0 } DLL_Function.Stop_HID(); </pre>

(四) Led_All_Off

項目	説明
名称	int Led_All_Off ()
機能概要	USB積層信号灯のすべてのLEDを消灯します。
パラメータ	この関数を呼び出す前に、次の関数「Start_HID」を呼び出す必要があります。
プログラム例	<pre>int result = DLL_Function.Start_HID(); if(result==0) { DLL_Function.Led_All_Off(); } DLL_Function.Stop_HID();</pre>

(五) Buzzer_Control

項目	説明
名称	int Buzzer_Control(int mode, int f1, int f2, int volume);
機能概要	ブザーモード、周波数1、周波数2を指定して、USB積層信号灯のブザーを鳴らします。 LEDユニットは現在の状態を維持します。
パラメータ	mode : ブザーモードを選択します。 f1 : 周波数1を選択します。 f2 : 周波数2を選択します。 volume : 音量を選択します。 ブザーの動作方式を指定します。
戻り値	エラー時は負の値を返します。 詳細は「エラーコード」表を参照してください。
注意事項	この関数を呼び出す前に、次の関数「Start_HID」を呼び出す必要があります。
プログラム例	<pre>int result = DLL_Function.Start_HID(); if(result==0) { DLL_Function.Buzzer_Control(Parameter.Buz_Mode3, Parameter.Buz_F1_1864_7,Parameter.Buz_F2_2793_8, Parameter.Volume1); } DLL_Function.Stop_HID();</pre>

(六) Buzzer_Off

項目	説明
名称	int Buzzer_Off()
機能概要	USB積層信号灯のブザーを停止します。
注意事項	この関数を呼び出す前に、次の関数「Start_HID」を呼び出す必要があります。
プログラム例	<pre>int result = DLL_Function.Start_HID(); if(result==0) { DLL_Function.Buzzer_Off(); } DLL_Function.Stop_HID();</pre>

(七)Reset_Device

項目	説明
名称	int Reset_Device()
機能概要	すべてのLEDユニットを消灯し、ブザーを停止します。
注意事項	この関数を呼び出す前に、次の関数「Start_HID」を呼び出す必要があります。
プログラム例	<pre>int result = DLL_Function.Start_HID(); if(result==0) { DLL_Function.Reset_Device(); } DLL_Function.Stop_HID();</pre>

(八)Get_Firmware_Version

項目	説明
名称	DLL_Function.Get_Firmware_Version(bufferPtr, bufferSize);
機能概要	現在接続されているUSB積層信号灯のファームウェアバージョンを取得します。
戻り値	<p>先頭8ビット = メジャーバージョン 中間4ビット = マイナーバージョン 末尾4ビット = リビジョンバージョン</p> <p>すべての値は16進数 (HEX) で表記されます。 (例 : 00 00 00 01 00 00 00 00=Ver1.0, 00 00 01 02 00 03 00 04=Ver12.3.4), エラー時は負の値を返します。 詳細は「エラーコード」表を参照してください。</p>
注意事項	<p>この関数を呼び出す前に、次の関数「Start_HID」を呼び出す必要があります。</p> <p>BufferPtr は8より大きくする必要があります。bufferSize は Buffer と等しくする必要があります。LEDおよびブザーはリセットされます。</p>

プログラム例	<pre>GCHandle handle = GCHandle.Alloc(buffer, GCHandleType.Pinned); IntPtr bufferPtr = handle.AddrOfPinnedObject(); try { IntPtr retPtr = DLL_Function.Get_Firmware_Version(bufferPtr, bufferSize); string resultStr = Marshal.PtrToStringAnsi(retPtr); Console.WriteLine("Firmware version (hex): "); for (int i = 0; i < 8; i++) { Console.WriteLine(\$"{buffer[i]:X2} "); } Console.WriteLine(); } finally { handle.Free(); } //Get_Firmware_Version</pre>
--------	---

(九)Get_Dll_Version

項目	説明
名称	const char* Get_Dll_Version()
機能概要	DLLバージョンの取得。
戻り値	例：1.0 → 文字列 "1.0" を返します。
プログラム例	<pre>IntPtr ptr = DLL_Function.Get_Dll_Version(); string DLLVer = Marshal.PtrToStringAnsi(ptr); Console.WriteLine("DLL Version: " + DLLVer); // Get DLL Version</pre>

五、パラメータ

(一) 制御するLEDユニットの色

以下のパラメータを使用して、制御するLEDの色とモードを指定します。

制御項目	入力	
制御するLEDの色	パラメータ	値
赤モード0(OFF) 黄モード0(OFF) 緑モード0(OFF) 青モード0(OFF) 白モード0(OFF)	RLED_Mode0 YLED_Mode0 GLED_Mode0 BLED_Mode0 WLED_Mode0	0
赤モード1 黄モード1 緑モード1 青モード1 白モード1	RLED_Mode1 YLED_Mode1 GLED_Mode1 BLED_Mode1 WLED_Mode1	1
赤モード2 黄モード2 緑モード2 青モード2 白モード2	RLED_Mode2 YLED_Mode2 GLED_Mode2 BLED_Mode2 WLED_Mode2	2
赤モード3 黄モード3 緑モード3 青モード3 白モード3	RLED_Mode3 YLED_Mode3 GLED_Mode3 BLED_Mode3 WLED_Mode3	3

制御項目	入力	
制御するLEDの色	パラメータ	値
赤モード4 黄モード4 緑モード4 青モード4 白モード4	RLED_Mode4 YLED_Mode4 GLED_Mode4 BLED_Mode4 WLED_Mode4	4
赤モード5 黄モード5 緑モード5 青モード5 白モード5	RLED_Mode5 YLED_Mode5 GLED_Mode5 BLED_Mode5 WLED_Mode5	5
赤モード6 黄モード6 緑モード6 青モード6 白モード6	RLED_Mode6 YLED_Mode6 GLED_Mode6 BLED_Mode6 WLED_Mode6	6
赤モード保持 黄モード保持 緑モード保持 青モード保持 白モード保持	RLED_Keep YLED_Keep GLED_Keep BLED_Keep WLED_Keep	7
関連API		
LED_Control		

(二) ブザーモード

以下のパラメータを使用して、制御するブザーモードを指定します。

制御項目	入力	
制御するブザー	パラメータ	値
ブザーモード 0	Buz_Mode0	0
ブザーモード 1	Buz_Mode1	1
ブザーモード 2	Buz_Mode2	2
ブザーモード 3	Buz_Mode3	3
ブザーモード 4	Buz_Mode4	4
ブザーモード 5	Buz_Mode5	5
ブザーモード 6	Buz_Mode6	6
ブザーモード保持	Buz_Keep	7
関連API		
Buzzer_Control		

(三) ブザートーン（音階）

以下のパラメータを使用して、サウンド1およびサウンド2の音階を指定します。

サウンド 1／サウンド 2		入力	
音階	周波数（参照値）	パラメータ	値
(Stop)	—	Buz_F1_OFF, Buz_F2_OFF	1
A6	1760.0 Hz	Buz_F1_1760、Buz_F2_1760	2
B♭6	1864.7 Hz	Buz_F1_1864_7、Buz_F2_1864_7	3
B6	1975.5 Hz	Buz_F1_1975_5、Buz_F2_1975_5	4
C7	2093.0 Hz	Buz_F1_2093、Buz_F2_2093	5
D♭7	2217.5 Hz	Buz_F1_2217_5、Buz_F2_2217_5	6
D7	2349.3 Hz	Buz_F1_2349_3、Buz_F2_2349_3	7
E♭7	2489.0 Hz	Buz_F1_2480、Buz_F2_2480	8
E7	2637.0 Hz	Buz_F1_2637、Buz_F2_2637	9
F7	2793.8 Hz	Buz_F1_2793_8、Buz_F2_2793_8	10
G♭7	2960.0 Hz	Buz_F1_2960、Buz_F2_2960	11
G7	3136.0 Hz	Buz_F1_3136、Buz_F2_3136	12
A♭7	3322.4 Hz	Buz_F1_3322_4、Buz_F2_3322_4	13
A7	3520.0 Hz	Buz_F1_3520、Buz_F2_3520	14
関連 API			
Buzzer_Control			

(四) ブザー音量

以下のパラメータを使用して、音量を指定します。

制御項目	入力	
音量を選択	パラメータ	値
音量OFF	Volume1	1
音量60%	Volume2	2
音量80%	Volume3	3
音量100%	Volume4	4
音量保持	Vol_Keep	5
関連API		
Buzzer_Control		

六、エラー

(一)エラー一覧

マクロ文字列	説明	値
ERR_NOEXIST	USB信号灯を検出できませんでした。接続を確認する必要があります。	-1
ERR_LOCKED	USB信号灯は検出されましたが、別のプログラムで使用中のため、通信を確立できませんでした。	-2
ERR_CONNECTION	接続が確立されていません。「Start_HID」を呼び出す必要があります。	-3
ERR_PARAMETER	パラメータに範囲外の値が指定されました。パラメータを確認する必要があります。	-4
ERR_SEND_FAIL	送信できません（例：送信中の接続断など）。	-5
ERR_FIRMRECV_FAIL	送受信できません（例：送信中の接続断など）。	-6
ERR_DLL_LINK	setupapi.dll または HID.DLL がインストールされていないため、これらを取得する必要があります。	-7

(二)エラーを返すAPI一覧

API	返される可能性のあるエラー
Start_HID Stop_HID	ERR_NOEXIST、 ERR_LOCKED、 ERR_CONNECTION、 ERR_DLL_LINK
LED_Control Buzzer_Control	ERR_SEND_FAIL、 ERR_PARAMETER
Reset_Device	ERR_SEND_FAIL
Get_Firmware_Version	ERR_CONNECTION、 ERR_FIRMRECV_FAIL、 ERR_PARAMETER

七、モード間隔時間

(一) LED 間隔周波数

MODE0	OFF							
MODE1	ON	OFF	ON	OFF	ON	OFF	ON	OFF
MODE2	ON	ON	OFF	OFF	ON	ON	OFF	OFF
MODE3	ON	ON	ON	ON	OFF	OFF	OFF	OFF
MODE4	ON	ON	ON	ON	ON	ON	ON	ON
時間	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
	1 秒							

MODE5	0.08 秒 LED On	0.17 秒 LED OFF	0.08 秒 LED On	0.67 秒 LED OFF
MODE6	0.1 秒 LED On	0.4 秒 LED OFF	0.1 秒 LED On	0.4 秒 LED OFF
時間	1 秒			

(二) ブザー間隔周波数

MODE0	OFF							
MODE1	ON	OFF	ON	OFF	ON	OFF	ON	OFF
MODE2	ON	ON	OFF	OFF	ON	ON	OFF	OFF
MODE3	ON	ON	ON	ON	OFF	OFF	OFF	OFF
MODE4	ON	ON	ON	ON	ON	ON	ON	ON
時間	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
	1 秒							

MODE5	0.08 秒 On	0.17 秒 OFF	0.08 秒 On	0.67 秒 OFF
MODE6	0.1 秒 On	0.4 秒 OFF	0.1 秒 On	0.4 秒 OFF
時間	1 秒			

八、USB通信プロトコル

(一)通信プロトコルデータ形式

1st (0xFF)	モード	赤	黄	緑	青	白	音量	ブザーモ ード	ブザートーン (音階)	CHECK SUM
1bytes	2bytes L	2bytes H	3bytes L	3bytes H	4bytes L	4bytes H	5bytes	6bytes	7bytes	8bytes

(二)チェックサム計算：

チェックサムは、data[1] から data[6] までを加算し、&= 0xFF を適用して計算されます。

算出された値は data[7] にチェックサムとして格納されます。

日機株式会社

〒550-0003 大阪市西区京町堀二丁目 6-28

TEL:06-6225-2855 FAX:06-6225-2856

<https://www.nikki-tr.co.jp> e-mail:contactus@nikki-tr.co.jp