$USB_IR_Remote_Controller_Advance_Library$

x64 64bit 版 取扱説明書

2022/11/21 x64 版 R01 Assembly Desk

USB_IR_Remote_Controller_Advance_Library(x64 64bit 版)を使用すると、弊社製 USB IR Remote controller Advance から赤外線コードを送信する Windows アプリケーション(x64 64bit 版)を簡単に作成することができます。

ライブラリの使用方法は、サンプルコード(C#)を参照してください。ライブラリ関数の仕様は以下の通りです。

関数名	openUSBIR	
概要	USB IR Remot	e Controller Advance と接続をします。
宜言	SafeFileHandle	e openUSBIR(IntPtr hRecipient)
戻り値の型	意味	
SafeFileHandle	USB DEVICE	のハンドルを返します。失敗したら NULL を返します。
引数の型	引数の名称	説明
IntPtr	hRecipient	ウィンドウハンドルを指定します。
※この関数は、USB	赤外線リモコンカ	が接続しているかの確認にも使用できます。接続している場合は、USB
DEVICE のハンドル	~が返り、未接続	の場合は NULL が返ります。

関数名 closeUSBIR
概要 USB IR Remote Controller Advance との接続を切断します。

宣言 int closeUSBIR(SafeFileHandle HandleToUSBDevice)

戻り値の型 意味
int 関数が成功すると 0 が返ります。失敗すると-1 が返ります。
引数の型 引数の名称 説明
SafeFileHandle HandleToUSBDevice USB DEVICE のハンドルを指定します。

	1	
関数名	writeUSBIRData2	
概要	USB IR Remote Contr	oller Advance から赤外線コードを送信します。
	※受信設定または送信	設定コンフィグレーションツールのクリップボードにコピー
	ボタンでコピーしたデ	ータは、この関数の data パラメータで使用可能です。
	※赤外線コード取得関	数で取得したデータを、この関数の data パラメータで使用可
	能です。	
宣言	int writeUSBIRData	2(SafeFileHandle HandleToUSBDevice, uint freq, byte[]
	data, uint bit_len)	
戻り値の型	意味	
int	関数が成功すると0が	返ります。失敗すると-1 が返ります。
引数の型	引数の名称	説明
SafeFileHandle	HandleToUSBDevice	USB DEVICE のハンドルを指定します。
uint	freq	赤外線コード送信周波数を指定します。
		一般的なリモコンの周波数は、38kHz なので 38000 を指定
		します。
byte[]	data	赤外線送信コードをバイト配列で指定します。
		最大 9600 配列長まで指定可。
		4 データで、赤外線送信コードの 1 ビット分となります。
uint	bit_len	赤外線送信コードのビット長を指定します。
		data 配列長の 1/4 の値を指定します。
		最大 2400 ビットまで指定可。

○各種電化製品のリモコンコードの確認方法

USB IR Remote Controller Advance 送信設定 Configuration Tool(以下 CT)を使用します。

USB IR Remote Controller Advance を PC に接続して、CT を起動します。CT の受信開始ボタンをクリックし、コードを調べたいリモコンのボタンを USB IR Remote Controller Advance に向けて押します。受信停止ボタンをクリックすると赤外線コードのデータが表示されます。

上記ライブラリ関数で使用するデータをコピーするには、受信停止ボタンをクリック後、保存ボタンをクリックしてからクリップボードへコピーボタンをクリックします。クリップボードにバイト単位のデータ形式でコピーされます。

下図に赤外線コードとパラメータに設定するデータの具体例を示します。

		ON						ON		ÓN			ÓN		ON		ON		ON		ON	H
						OFF.			OFF			OFF		OFF.		OFF		OFF		OFF		01
時間[ms]		3.2				1.5		0.4	0.4	0.4		1.2	0.4	0.4	0.4	0.4	0.4	1.2	0.4	1.2	0.4	0.
kHz時 のカウント	値	0x7B			c	n3D		0xF	0xF	0xF		0×2E	0xF	0×F	0xF	0xF	0xF	0×2E	0xF	0x2E	0xF	0>
パラメータ																						
freq=	38000																					
data[32] =	0x00,0	7B,0x00,0x3E	,0x00,0x0F	0x0,00x0,	F,0x00,0x	0F,0x00,0	x2E,0x0	0,0×0,0	,0x00	,0x0F,0	0,000	0F,0x00,0	x0F,0x0	0,0×0F	F,0x00,	0x2E,0	0x0,00x	F,0x00,0x2E,0	x00,0x0F,0x	00,0x0F		
bitJen =	8																					
bitJen =	8																					

赤外線コード取得用関数

関数名	recUSBIRData_Start	
概要	USB IR Remote Contr	roller Advance に赤外線コードの記録を開始します。
宜言	int recUSBIRData_Sta	art(SafeFileHandle HandleToUSBDevice, uint freq)
戻り値の型	意味	
int	関数が成功すると 0 が	返ります。失敗すると-1が返ります。
引数の型	引数の名称	説明
SafeFileHandle	HandleToUSBDevice	USB DEVICE のハンドルを指定します。
uint	freq	赤外線コード送信周波数を指定します。
		一般的なリモコンの周波数は、38kHz なので 38000 を指定
		します。

関数名	recUSBIRData_Stop	
概要	USB IR Remote Contr	oller Advance に赤外線コードの記録を停止します。
宜言	int recUSBIRData_Sto	pp(SafeFileHandle HandleToUSBDevice)
戻り値の型	意味	
int	関数が成功すると0が	返ります。失敗すると-1 が返ります。
引数の型	引数の名称	説明
SafeFileHandle	HandleToUSBDevice	USB DEVICE のハンドルを指定します。

関数名	readUSBIRData	
概要	USB IR Remote Contr	roller Advance から記録した赤外線コードを取得します。
宣言	int readUSBIRData(S	SafeFileHandle HandleToUSBDevice, ref IntPtr data, ref
	uint data_len, ref uint	bit_len)
戻り値の型	意味	
int	関数が成功すると0が	返ります。失敗すると-1が返ります。
引数の型	引数の名称	説明
SafeFileHandle	HandleToUSBDevice	USB DEVICE のハンドルを指定します。
IntPtr	data	取得した赤外線送信コードを格納するためのInt型ポインタ
		を指定します。
		最大 9600 配列長まで指定可。
		4データで、赤外線送信コードの1ビット分となります。
uint	datab_len	取得した data 配列長を格納するための uint 型変数を指定し
		ます。
uint	bit_len	取得した赤外線送信コードのビット長を格納するための
		uint 型変数を指定します。
		data 配列に格納した赤外線コードの 1/4 の値を返します。
		最大 2400 ビット。

○赤外線コード取得手順

- 1. recUSBIRData_Start 関数により、赤外線コードの記録を開始します。
- 2. USB IR Remote Controller Advance の受信部にリモコンを向け、記憶したいコードのボタンを押します。
- 3. recUSBIRData_Stop 関数により、赤外線コードの記録を停止します。
- 4. readUSBIRData 関数により、記録した赤外線コードを取得します。

※取得した赤外線コードを、write USBIRData 関数のパラメータに指定することで赤外線コードを送信できます。

赤外線コード出力各種関数

以下の関数でも、赤外線コードを送信することが出来ます。

関数名	writeUSBIRData1	
概要	USB IR Remote Contr	roller Advance から赤外線コードを送信します。
宣言	int writeUSBIRData	1(SafeFileHandle HandleToUSBDevice, uint freq, uint[]
	data, uint bit_len)	
戻り値の型	意味	
概要 USB IR Remote Controller Advance から赤外線コードを送信します。 int writeUSBIRData1(SafeFileHandle HandleToUSBDevice, uint freq, uindata, uint bit_len) 戻り値の型 意味 int 関数が成功すると 0 が返ります。失敗すると-1 が返ります。 引数の型 引数の名称 説明 SafeFileHandle HandleToUSBDevice USB DEVICE のハンドルを指定します。 ー般的なリモコンの周波数は、38kHz なので 38000 を指します。 します。 uint		
引数の型	引数の名称	説明
SafeFileHandle	HandleToUSBDevice	USB DEVICE のハンドルを指定します。
uint	freq	赤外線コード送信周波数を指定します。
		一般的なリモコンの周波数は、38kHz なので 38000 を指定
		します。
uint[]	data	赤外線送信コードを uint 配列で指定します。
		最大 4800 配列長まで指定可。
		2 データで、赤外線送信コードの 1 ビット分となります。
uint	bit_len	赤外線送信コードのビット長を指定します。
		data 配列長の 1/2 の値を指定します。
		最大 2400 ビットまで指定可。

下図に赤外線コードと本関数のパラメータに設定するデータの具体例を示します。

					_		_		_			$\overline{}$	1	$\overline{}$				1			$\overline{}$					1
	_									_		_									L				_	╙
					ON							ON		ON			ON		ON		ON		ÓN		ON	
										ôÆ			0FF			OFF		o#=		o ≡		òr∓		ÓFF		
時間[m	ms]			3	1.2					1.6		0.4	0.4	0.4		1.2	0.4	0.4	0.4	0.4	0.4	1.2	0.4	1.2	0.4	0
Hz時の力	ללנ	ト値		0:	∂B					0x3D		0xF	0xF	0xF		0x2E	0×F	0xF	0xF	0xF	0xF	0×2E	0xF	0x2E	0xF	0
パラ.	у-	Ż																								
freq=	=	;	8000																							
data[[16]	= (>:007B,0	×003	D,0x00	0F,0x0	00F,0	×000F,	0×002E	,0x000F	,0x000F,0x	000F,0	.000F,	0×000F	F,0x00	2E,0x000F	,0x002E	000x0,	F,0x00	00F						
bitJe	en =		3																							

関数名	writeUSBIRCode1	
概要		oller Advance から赤外線コードを送信します。
宣言		1 (SafeFileHandle HandleToUSBDevice, uint freq, uint
		0, uint bit_1, uint stop_code, byte[] code, uint bit_len)
戻り値の型	意味	, , , ,
int	関数が成功すると0が	返ります。失敗すると-1 が返ります。
引数の型	引数の名称	説明
SafeFileHandle	HandleToUSBDevice	USB DEVICE のハンドルを指定します。
uint	freq	赤外線コード送信周波数を指定します。
		一般的なリモコンの周波数は、38kHz なので 38000 を指定
		します。
uint	reader_code	リーダーコードの、赤外線送信コード出力時間を指定しま
		す。
		上位 31-16 ビットに ON 出力するカウント数を設定。
		下位 15-0 ビットに OFF 出力するカウント数を設定。
		38kHz 周波数の場合は、1 カウント約 26 μ 秒となり、ON
		3.2ms、OFF 1.6ms の出力時間を設定する場合は、
		0x007B003D 値を設定します。
uint	bit_0	bit が 0 の場合の、赤外線送信コード出力時間を指定します。
		上位 31-16 ビットに ON 出力するカウント数を設定。
		下位 15·0 ビットに OFF 出力するカウント数を設定。
		38kHz 周波数の場合は、1 カウント約 26μ秒となり、ON
		0.4ms、OFF 0.4ms の出力時間を設定する場合は、
		0x000F000F 値を設定します。
uint	bit_1	bit が 1 の場合の、赤外線送信コード出力時間を指定します。
		上位 31-16 ビットに ON 出力するカウント数を設定。
		下位 15-0 ビットに OFF 出力するカウント数を設定。
		38kHz 周波数の場合は、1 カウント約 26μ秒となり、ON
		0.4ms、OFF 1.2ms の出力時間を設定する場合は、
uint	stop_code	0x000F002E値を設定します。 ストップコードの、赤外線送信コード出力時間を指定しま
uint	stop_code	ストックュートの、
		,。 上位 31-16 ビットに ON 出力するカウント数を設定。
		下位 15-0 ビットに OFF 出力するカウント数を設定。
		$38kHz$ 周波数の場合は、 1 カウント約 26μ 秒となり、 ON
		0.6ms 、OFF 200ms の出力時間を設定する場合は、
		0x00171E0C 値を設定します。
byte[]	code	赤外線送信コードを byte 配列で指定します。
		最大 300 配列長まで指定可。
uint	bit_len	赤外線送信コードのビット長を指定します。
	_	最大 2400 ビットまで指定可。

下図に赤外線コードと本関数のパラメータに設定するデータの具体例を示します。

					data0	bit0		data0 b	it1	data	0 Ыt2	datal	D bit3			data3 bi	t7		
		リーダーコ	− F		0)		1			0		0	all 0		1		7	ホップビット
														省略					
		ON			ON		ON			ON		ON			ON			ON	
				OFF		o r =)FF		0FF		0FF			c	F		OFF
時間(ms)		3.2		1.6	0.4	0.4	0.4		.2	0.4	0.4	0.4	0.4		0.4	1	2	0.4	200
SkHz時のカウ	ント値	0x7B		0x3D	0×F	0xF	0×F	0	×2E	0xF	0×F	0xF	0xF		0xF	0:	QΕ	0×F	0×1E0C
パラメーク	5																		
	freq= 38000)																	
reader_c	ode= 0x007	'B003D																	
	bit_0= 0x000	F000F																	
	bit_1= 0x000	0F002E																	
stop_c	ode= 0x000	F1E0C																	
code	[4] = 0×02,	0x00,0x00,0x0																	
6.14	Jen = 32																		

関数名	writeUSBIRCode2	
		ler Advance から赤外線コードを送信します。
宣言		(SafeFileHandle HandleToUSBDevice, uint freq, uint
型 口		uint bit_1, uint stop_code, byte[] code, uint bit_len, uint[]
		code_len, uint repeat_code_send_num)
戻り値の型	意味	code_ien, unit repeat_code_send_num/
int		ります。失敗すると-1 が返ります。
引数の型	引数の名称	説明
SafeFileHandle	HandleToUSBDevice	USB DEVICE のハンドルを指定します。
uint	freq	赤外線コード送信周波数を指定します。
umi	ireq	一般的なリモコンの周波数は、38kHz なので 38000 を指
		定します。
uint	reader_code	リーダーコードの、赤外線送信コード出力時間を指定しま
ann	reader_code	す。
)。 上位 31-16 ビットに ON 出力するカウント数を設定。
		下位 15-0 ビットに OFF 出力するカウント数を設定。
		38kHz 周波数の場合は、 1 カウント約 26μ 秒となり、ON
		3.2ms、OFF 1.6ms の出力時間を設定する場合は、
		0x007B003D 値を設定します。
uint	bit_0	bit が 0 の場合の、赤外線送信コード出力時間を指定しま
alliv	D10_0	す。
		上位 31-16 ビットに ON 出力するカウント数を設定。
		下位 15-0 ビットに OFF 出力するカウント数を設定。
		38kHz 周波数の場合は、1 カウント約 26 μ 秒となり、ON
		0.4ms、OFF 0.4ms の出力時間を設定する場合は、
		0x000F000F値を設定します。
uint	bit_1	bit が 1 の場合の、赤外線送信コード出力時間を指定しま
		す。
		上位 31-16 ビットに ON 出力するカウント数を設定。
		下位 15-0 ビットに OFF 出力するカウント数を設定。
		38kHz 周波数の場合は、1 カウント約 26 μ 秒となり、ON
		0.4ms、OFF 1.2ms の出力時間を設定する場合は、
		0x000F002E 値を設定します。
uint	stop_code	ストップコードの、赤外線送信コード出力時間を指定しま
		す。
		上位 31-16 ビットに ON 出力するカウント数を設定。
		下位 15-0 ビットに OFF 出力するカウント数を設定。
		$38kHz$ 周波数の場合は、 1 カウント約 26μ 秒となり、ON
		0.6ms、OFF 200ms の出力時間を設定する場合は、
		0x00171E0C 値を設定します。
byte[]	code	赤外線送信コードを byte 配列で指定します。
		最大 300 配列長まで指定可。
uint	bit_len	赤外線送信コードのビット長を指定します。
	. 1	最大 2400 ビットまで指定可。
uint[]	repeat_code	リピートコードの、赤外線送信コード出力時間を指定しま
		す。 1 たの1 1 a 1 x 1 x 0 x 111 + ナマトト、1 米ナニルウ
		上位 31-16 ビットに ON 出力するカウント数を設定。
		下位 15-0 ビットに OFF 出力するカウント数を設定。
		38kHz 周波数の場合は、1 カウント約 26 μ 秒となり、ON
		0.6ms、OFF 200ms の出力時間を設定する場合は、
•		0x00171E0C 値を設定します。
uint	repeat_code_len	リピートコードのビット長を指定します。
uint	repeat_code_send_num	リピートコードの繰り返し回数を指定します

下図に赤外線コードと本関数のパラメータに設定するデータの具体例を示します。

											data0	bit0		data0 bit1		data	D bit2	data() bit3			data3 bit7		
				IJ	-ダ-	- I - F					0	1		1			0	- 1)	all 0		1	7	トップビット
																				省略				
			ON								ON		ON			ON		ON			ON		ON	
									OFF.			0FF		OFF			OFF		OFF			0FF		OFF
時間[ms]			3.2						1.5		0.4	0.4	0.4	1.2		0.4	0.4	0.4	0.4		0.4	1.2	0.4	200
38kHz時のカウ	ント値		0×7B					0	×3D		0xF	0×F	0xF	0×2	E	0xF	0xF	0×F	0xF		0×F	0×2E	0xF	0×1E0C
パラメーク	2																							
	freq= 3	38000																						
reader_c	code= (0x007B00	3D																					
	bit_0= (>000F000)F																					
	bit_1= (>000F00	ΣE																					
stop_c	code= (0x000F1E0	ос																					
code	[4] = 0	0x02,0x00,	0%0,00%0)																				
bit	Jen = 3	32																	リピート	コード				
repeat_code	[1] = 0	0x007B00	7B	٦																				
repeat_code	Jen =	1		-	右のよ	うなり	ピートコ	ードを	5回送信	する場合							ON			OFF				
eat_code_send_r	num = 5	5		JI										時間[ns]		3.2			3.2				
ear_code_send)														8kHz時のが			0x7B			0×7E				

●更新履歴●

Revision	Version	日付	内容
R01	6.0.0.0	2022/11/21	初版