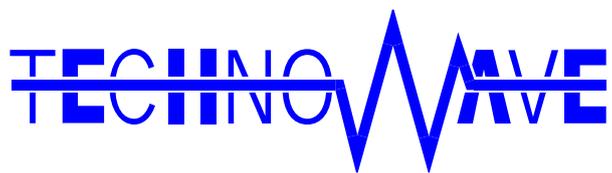


# KB-IT4 ユーザーズマニュアル



テクノウェーブ株式会社

---

## 目次

1. はじめに.....	5
1.1. 安全にご使用いただくために .....	5
1.2. その他の注意事項.....	5
1.3. マニュアル内の表記について .....	6
1.3.1. デジタル入力端子の状態.....	6
1.3.2. 関数.....	7
1.3.3. Null 値.....	7
2. 製品概要.....	8
2.1. 特徴.....	8
3. 製品仕様.....	9
3.1. 製品仕様.....	9
3.2. 外形寸法.....	9
3.3. 各部の名称 .....	10
3.4. 端子説明.....	10
4. ハードウェア .....	11
4.1. デジタル入力端子.....	11
4.1.1. 入力回路.....	11
4.1.2. 接続例.....	11
5. 使用準備.....	12
5.1. 設定ツールのインストール.....	12
5.2. 設定ツールについて .....	12
6. 使用方法.....	13
6.1. 製品の使用方法 .....	13
6.2. レジューム機能の使用方法.....	13
7. キーコード設定ソフトウェア .....	15

---

7.1. 概要 .....	15
7.2. 起動・終了 .....	15
7.2.1. 起動 .....	15
7.2.2. 終了 .....	15
7.3. 使い方.....	16
7.3.1. メイン画面の説明.....	16
7.3.2. 基本的な操作の流れ .....	17
7.3.3. 手動でキーデータを入力する .....	18
7.3.4. ファイルからキーデータを読み出す .....	24
7.3.5. デバイスからキーデータを読み出す .....	24
7.3.6. ファイルにキーデータを保存する .....	25
7.3.7. デバイスにキーデータを書き込む.....	25
7.4. その他の機能.....	26
7.4.1. キャプチャモードの ON/OFF.....	26
7.4.2. 未設定データに初期化.....	26
7.4.3. ソフトウェア・バージョン情報の表示.....	26
8. プログラミング .....	27
8.1. プログラミングの準備 .....	27
8.1.1. C/C++ での開発に必要なファイル.....	27
8.1.2. Visual Basic、C# での開発に必要なファイル .....	28
8.1.3. Visual Basic for Applications での開発に必要なファイル.....	28
8.2. 実行時に必要なファイル .....	28
8.3. 関数リファレンス .....	30
8.3.1. 関数の戻り値の意味 .....	30
TWKB_OK (H' 00000000) .....	30
TWKB_INVALID_HANDLE (H' 00000001) .....	30
TWKB_DEVICE_NOT_FOUND (H' 00000002) .....	30
TWKB_IO_ERROR (H' 00000004) .....	30
TWKB_INVALID_ARGS (H' 00000010) .....	30
TWKB_NOT_SUPPORTED (H' 00000011) .....	30
TW_OTHER_ERROR (H' 00000012) .....	30
8.4. マルチスレッドプログラムからの呼び出しについて .....	31
8.5. 関数一覧 .....	31

---

---

TWKB_OpenEx.....	31
TWKB_Open().....	32
TWKB_Close().....	32
TWKB_GetSerial().....	32
TWKB_KeyEnable().....	33
TWKB_ReadSw().....	33
<b>9. 保証期間.....</b>	<b>34</b>
<b>10. サポート情報.....</b>	<b>34</b>

---

# 1. はじめに

このたびは弊社製品『KB-IT4』をご購入頂き、まことにありがとうございます。以下をよくお読みになり、安全にご使用いただけますようお願い申し上げます。

## 1.1. 安全にご使用いただくために

製品を安全にご利用いただくために、以下の事項をお守りください。

 <b>危険</b>	これらの注意事項を無視して誤った取り扱いをすると人が死亡または重傷を負う危険が差し迫って生じる可能性があります。
<ul style="list-style-type: none"><li>引火性のガスがある場所では使用しないでください。爆発、火災、故障の原因となります。</li></ul>	

 <b>警告</b>	これらの注意事項を無視して誤った取り扱いをすると人が死亡または重傷を負う可能性があります。
<ul style="list-style-type: none"><li>水や薬品のかかる可能性がある場所では使用しないでください。火災、感電の原因となります。</li><li>結露の発生する環境では使用しないでください。火災、感電の原因となります。</li><li>定格の範囲内でご使用ください。火災の原因となります。</li></ul>	

 <b>注意</b>	これらの注意事項を無視して誤った取り扱いをすると人が傷害を負う可能性があります。また物的損害の発生が想定されます。
<ul style="list-style-type: none"><li>濡れた手で製品を扱わないでください。故障の原因となります。</li><li>異臭、過熱、発煙に気がついた場合は、ただちに USB ケーブルを抜き、電源を切断してください。</li><li>製品を改造しないでください。</li></ul>	

## 1.2. その他の注意事項

<ul style="list-style-type: none"><li>本製品は一般民製品です。特別に高い品質・信頼性が要求され、その故障や誤動作が直接人命を脅かしたり、人体に危害を及ぼす恐れのある機器に使用することを前提としていません。本製品をこれらの用途に使用される場合は、お客様の責任においてなされることとなります。当該使用に基づいて発生した損害につきましては、弊社は一切の責任を負えませんのでご了承ください。</li><li>お客様の不注意、誤操作により発生した製品、パソコン、その他の故障、及び事故につきましては弊社は一切の責任を負いませんのでご了承ください。</li><li>本製品または、付属のソフトウェアの使用による要因で生じた損害、逸失利益または第三者からのいかなる請求についても、当社は一切その責任を負えませんのでご了承ください。</li></ul>	
---	--

### 1.3. マニュアル内の表記について

本マニュアル内では対応製品『KB-IT4』を単に「製品」、または、「デバイス」と表記する場合があります。

本マニュアル内ではハードウェアの電気的状態について記述する必要がある場合には、下記のように表記します。

表 1 電気的状態の表記方法

表記	状態
"ON"	電流が流れている状態、スイッチが閉じている状態。
"OFF"	電流が流れていない状態、スイッチが開いている状態。

また、数値について「0x」、「&H」、「H」はいずれもそれに続く数値が 16 進数であることを表します。「0x10」、「&H1F」、「H' 20」などはいずれも 16 進数です。

#### 1.3.1. デジタル入力端子の状態

デジタル入力端子は十分な入力電流が流れている状態を"ON"、入力電流が流れていないか十分でない場合を"OFF"とします。

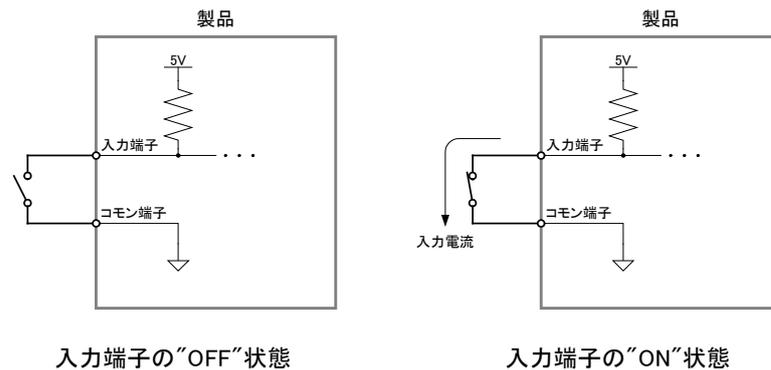


図 1 デジタル入力端子の"OFF"状態と"ON"状態

また、デジタル入力端子の入力内容に応じて、シングルクリック、ダブルクリック、ロングクリックを下記のように表記します。

表 2 デジタル入力端子の状態とクリック表記

表記	状態
シングルクリック	監視カウント時間内に、入力端子へ 1 度だけ"ON"状態が入力された。
ダブルクリック	監視カウント時間内に、入力端子へ 2 回"ON"状態が入力された。
ロングクリック	監視カウント時間以上、入力端子へ"ON"状態が入力された。

### 1.3.2. 関数

本文で関数名を表記する場合、C/C++、Visual Basic®、Visual Basic for Applications の名称に従い“*TWKB\_Open()*”のように表記します。C#の場合、これと対応する関数は *Techw. IO* 名前空間の *TWKB* クラスのスタティックメンバ関数で“*Techw. IO. TWKB. Open()*”となります。

関数の宣言を示す場合、C/C++、Visual Basic (.NET 以後)、Visual Basic for Applications (以下 VBA)、C# の順で、それぞれの言語における関数宣言が記載されます(表 3)。C# の場合は、名前空間とクラス名は省略して記述しています。

表 3 関数宣言の表記例

言語	関数宣言
C/C++	TW_STATUS TWKB_Open(HANDLE *phDev, LPCTSTR *pSerial)
VB	Function TWKB_Open(ByRef phDev As System.IntPtr, ByVal pID As String) As Integer
VBA	Function TWKB_Open(ByRef phDev As Long, ByVal pID As String) As Long
C#	STATUS Open(out System.IntPtr phDev, string pSerial)

### 1.3.3. Null 値

関数の引数の中には Null 値(空値)を要求するものがあります。本文中で Null 値と表記した場合、各言語での対応する記述方法は表 4 のようになります。

表 4 Null 値

言語	記述方法
C/C++	NULL
VB	Nothing
VBA	vbNullString
C#	null

---

## 2. 製品概要

### 2.1. 特徴

- USB 接続の絶縁デジタル入力ユニットです。
- パソコンからは USB キーボードとして認識され、デジタル入力端子を“ON”の状態にすると、キーボードのキー押下と同様に、予め設定されたキーコードがパソコンへ送出されます。
- 各デジタル入力には任意のキーコード、または、キーボードマクロを登録可能です。
- キーボードマクロは、シングルクリック、ダブルクリック、ロングクリック(長押し)の3つの入力状態に応じて3種類を登録することができます。
- 2ピースの端子台を採用していますので、配線作業や製品本体の交換を楽に行えます。
- フットスイッチなどを接続可能です。
- Windows® 用のキーコード設定ソフトウェアが付属します。

## 3. 製品仕様

### 3.1. 製品仕様

表 5 仕様概略

項目	仕様	備考	
寸法	90.2 (W) × 50.3 (D) × 17 (H) [mm]		
重量	45 [g]	付属品含まず	
電源電圧	4.75~5.25 [VDC]	USB バスパワー	
消費電流	最大 100 [mA]		
動作温度範囲	0~50 [°C]		
インタフェース	USB 2.0		
対応 OS	Windows XP, Vista, 7, 8, 8.1, 10	32 ビット/64 ビット	
絶縁デジタル入力点数	4 点		
入力方式	無電圧接点入力		
絶縁方式	フォトカプラ	入力回路図参照	
入力電圧範囲	0~5.25 [V]		
入力抵抗	2 [kΩ]	入力回路図参照	
入力オン電圧	1.3 [V] 以下	KEY±端子間	
入力端子スキャン周期	10 [msec]		
絶縁抵抗	対 SG (内部回路)	10 [GΩ]	測定条件: 500VDC
	対電源端子	10 [GΩ]	
絶縁耐圧	対 SG (内部回路)	2,000 [VDC]	
	対電源端子	2,000 [VDC]	
1 キーに登録可能なマクロステップ数	12 以上		

### 3.2. 外形寸法

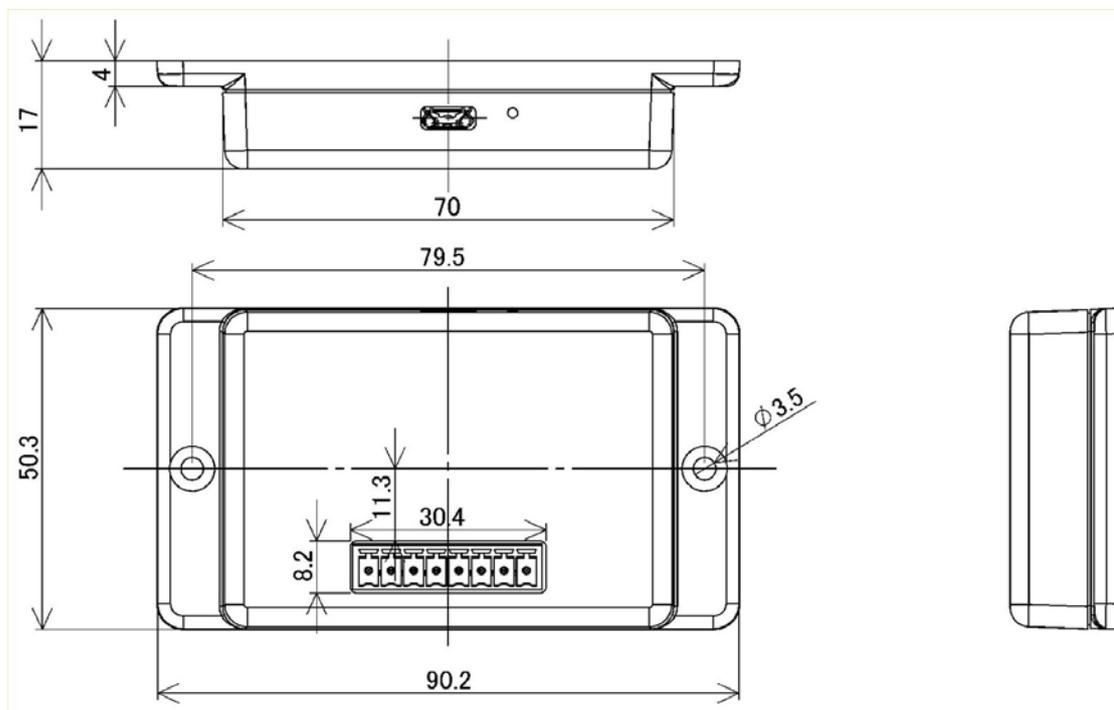


図 2 外形寸法図

### 3.3. 各部の名称



図 3 各部の名称

### 3.4. 端子説明

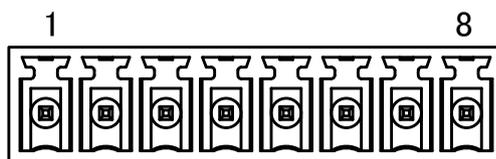


図 4 J1 端子配列<sup>1</sup>

表 6 J1 端子

ピン番	信号名	キー番号	説明	方向
J1-1	KEY1+	1	絶縁デジタル入力	I
J1-2	KEY1-		絶縁デジタル入力	I
J1-3	KEY2+	2	絶縁デジタル入力	I
J1-4	KEY2-		絶縁デジタル入力	I
J1-5	KEY3+	3	絶縁デジタル入力	I
J1-6	KEY3-		絶縁デジタル入力	I
J1-7	KEY4+	4	絶縁デジタル入力	I
J1-8	KEY4-		絶縁デジタル入力	I

<sup>1</sup> 適合コネクタは「EC350R-08P」(DINKLE)、「EC350RL-08P」(DINKLE)、「EC350V-08P」(DINKLE)

## 4. ハードウェア

### 4.1. デジタル入力端子

#### 4.1.1. 入力回路

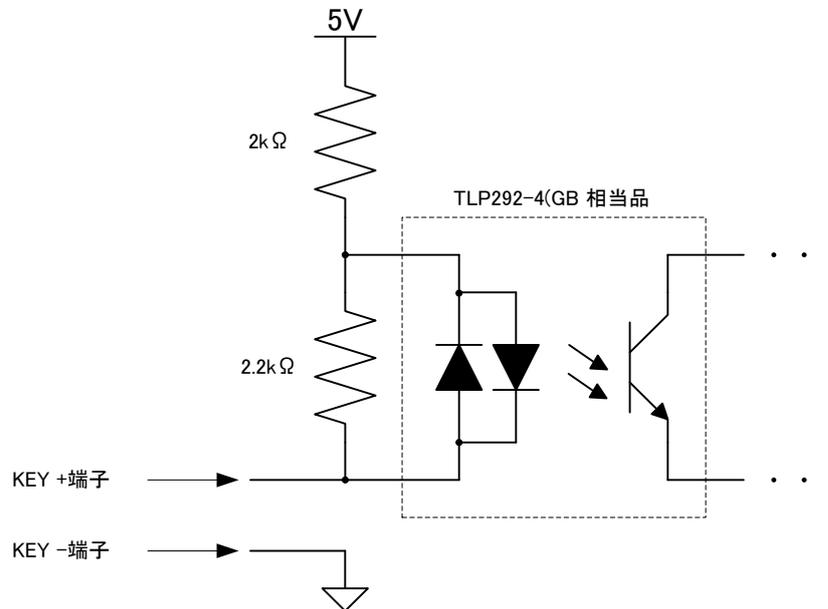


図 5 デジタル入力回路

#### 4.1.2. 接続例

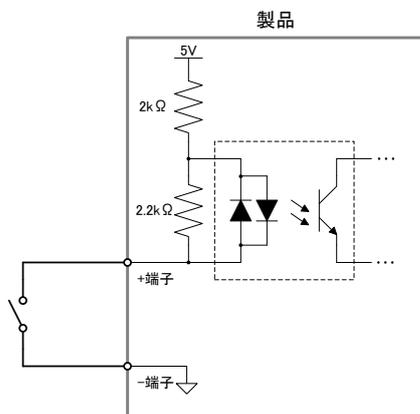


図 6 スイッチの接続例

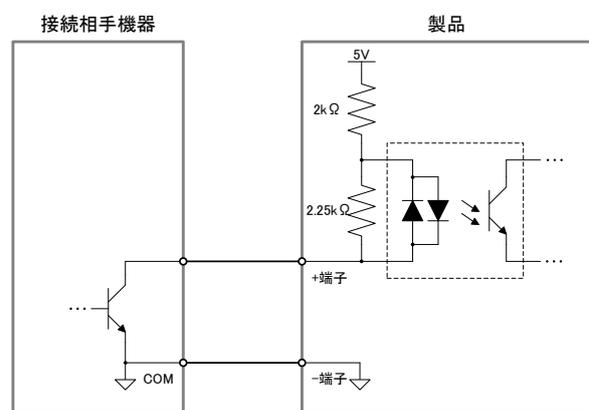


図 7 オープンコレクタ出力の機器との接続例

- 製品や接続相手機器の故障の原因となりますので、+/-端子間に仕様を超える電圧を入力しないでください。

## 5. 使用準備

### 5.1. 設定ツールのインストール

弊社ホームページ「<http://www.techw.co.jp/SupportFrm.html?pid=KB-IT4>」から「ユーティリティ」をダウンロードいただき、「¥KB-IT4\_TOOLS¥setup.exe」を実行し、画面の指示に従ってインストールを行ってください。

設定ツールの動作環境を以下に示します。

表 7 動作環境

対応 OS	日本語版 Windows 7, 8, 8.1, 10
-------	----------------------------

### 5.2. 設定ツールについて

「5.1」の内容に従って設定ツールをインストールすると、[スタート]メニューの中に設定ツールの起動メニューが追加されます。デフォルトのインストールオプションでは[スタート]→[すべてのプログラム]（または、[プログラム]）→[テクノウェーブ]→[KB-IT4\_Tools]から起動することができます。

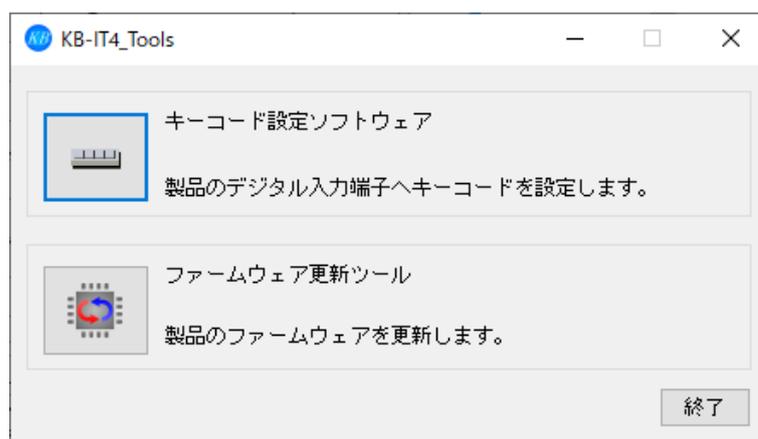


図 8 設定ツールのメニュー画面

表 8 設定ツールの機能説明

プログラム名	機能説明
キーコード設定ソフトウェア	製品のデジタル入力端子へキーコードを設定します。
ファームウェア更新ツール	製品のファームウェアを更新します。

キーコード設定ソフトウェアの使用方法は、本マニュアルの「7. キーコード設定ソフトウェア」を、ファームウェア更新ツールの使用方法は、ツールのオンラインヘルプを参照してください。

- ファームウェアはバグの修正や、機能追加のために不定期に新しいバージョンのものが公開されます<sup>2</sup>。ファームウェアの更新ファイルは設定ツールの中に含まれていますので、更新する場合には、まず新しい設定ツールをご利用のパソコンにインストールしてください。

<sup>2</sup> 弊社ホームページにて随時公開します。

---

## 6. 使用方法

### 6.1. 製品の使用方法

入力端子が“ON”状態となると、予め製品に設定されたキーコードがコンピュータに送出されます。キーコードは付属の「キーコード設定ソフトウェア」で製品に設定します。設定方法に関しては、「7 キーコード設定ソフトウェア」をご覧ください。

表 9 は出荷時のキーコード割り当て状態です。キー番号は表 6 と対応しています。

表 9 出荷時のキーコード割り当て

キー番号	1	2	3	4
キーコード	“Y”	“N”	Tab	Enter

### 6.2. レジューム機能の使用方法

コンピュータがスタンバイ状態にある時、キー番号 4 の入力端子を“ON”状態にすることにより、スタンバイを解除することができます。

レジューム機能を利用するには、ドライバが正しく設定されている必要があります。レジュームがうまく機能しない場合、以下に従い、レジューム機能が有効となっているかご確認下さい。

1. 本製品をコンピュータに接続し、管理者権限のアカウントでログオンします。
2. デバイスマネージャを開きます。

#### Windows 8.1 / Windows 10 の場合

「スタート」メニューを右クリックし、表示された一覧から「デバイスマネージャ」をクリックしてください。

#### Windows 7 の場合

「スタート」メニューの「コンピュータ」を右クリックし、「プロパティ」を選択します。「システム」画面が表示されますので、「コントロールパネルホーム」中の「デバイスマネージャ」をクリックしてください。

3. 本製品のドライバのプロパティを開きます。  
プロパティを開くには、デバイスマネージャのメニューバーの「表示」をクリックし、「デバイス（種類別）」を選択します。デバイスツリーの中にある「キーボード」をダブルクリックして、「キーボード」の下に接続されているデバイスを表示させます。その中の「HID キーボードデバイス」をダブルクリックするか、もしくは右クリックして表示されたポップアップメニューの中の「プロパティ」をクリックします。
4. 開いたプロパティが本製品のものか確認します。  
本製品のものか確認するには、プロパティウィンドウの「詳細」タブをクリックし、プロパティで「ハードウェア ID」を選択し、図 9 の赤丸の中と同様に“HID\VID\_1237&PID\_F101”であることを確認します。  
違う場合は他の「HID キーボードデバイス」を探します。

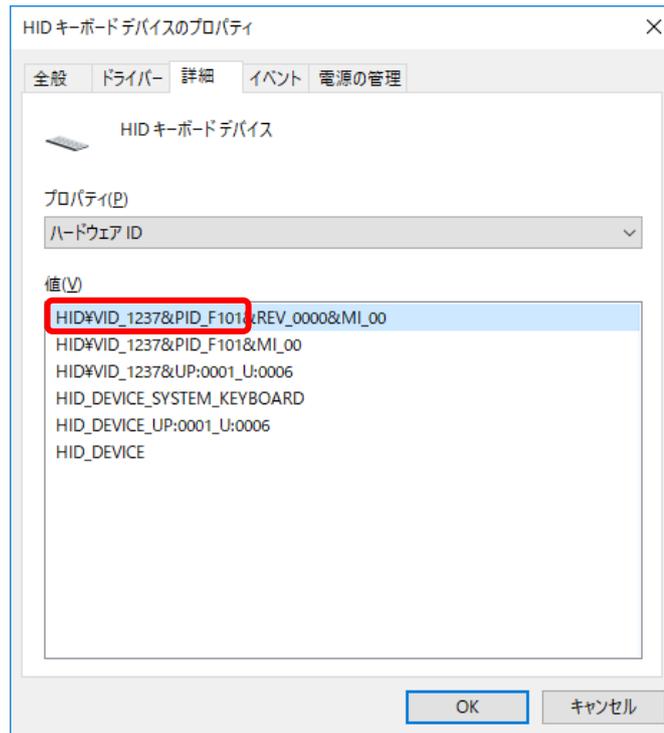


図 9 製品のハードウェア ID

5. レジューム機能が有効となっているか確認します。  
プロパティウィンドウの中の「電源の管理」タブをクリックします。  
「このデバイスで、コンピュータのスタンバイ状態を解除できるようにする」がチェックされているならば、レジュームが許可されています（図は Windows 10 の画面です）。

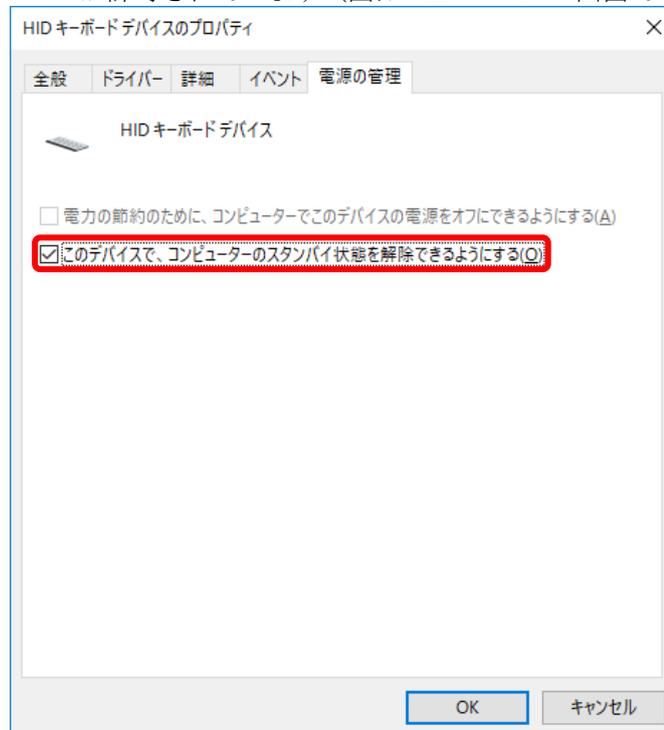


図 10 レジューム機能の確認

---

## 7. キーコード設定ソフトウェア

### 7.1. 概要

- 対象デバイスに設定するキーコードは、メニューから選択することができます。また、キャプチャ機能を使用すると、キーボード上のキー入力により、そのキーに相当するキーコードを入力することができます。
- 入力したデータはファイルに保存することができます。また、後で読み出して使用することもできます。
- ダブルクリック、ロングクリックの判定時間を画面上の表示を見ながら調整をすることができます。

### 7.2. 起動・終了

#### 7.2.1. 起動

デフォルトの設定では[スタート]→[すべてのプログラム]（または、[プログラム]）→[テクノロジー]→[KB-IT4\_Tools]→[キーコード設定ソフトウェア]から起動することができます。

#### 7.2.2. 終了

画面の右上の×ボタンをクリックして終了してください。

## 7.3. 使い方

### 7.3.1. メイン画面の説明

本ソフトウェアのメイン画面を以下に示します。

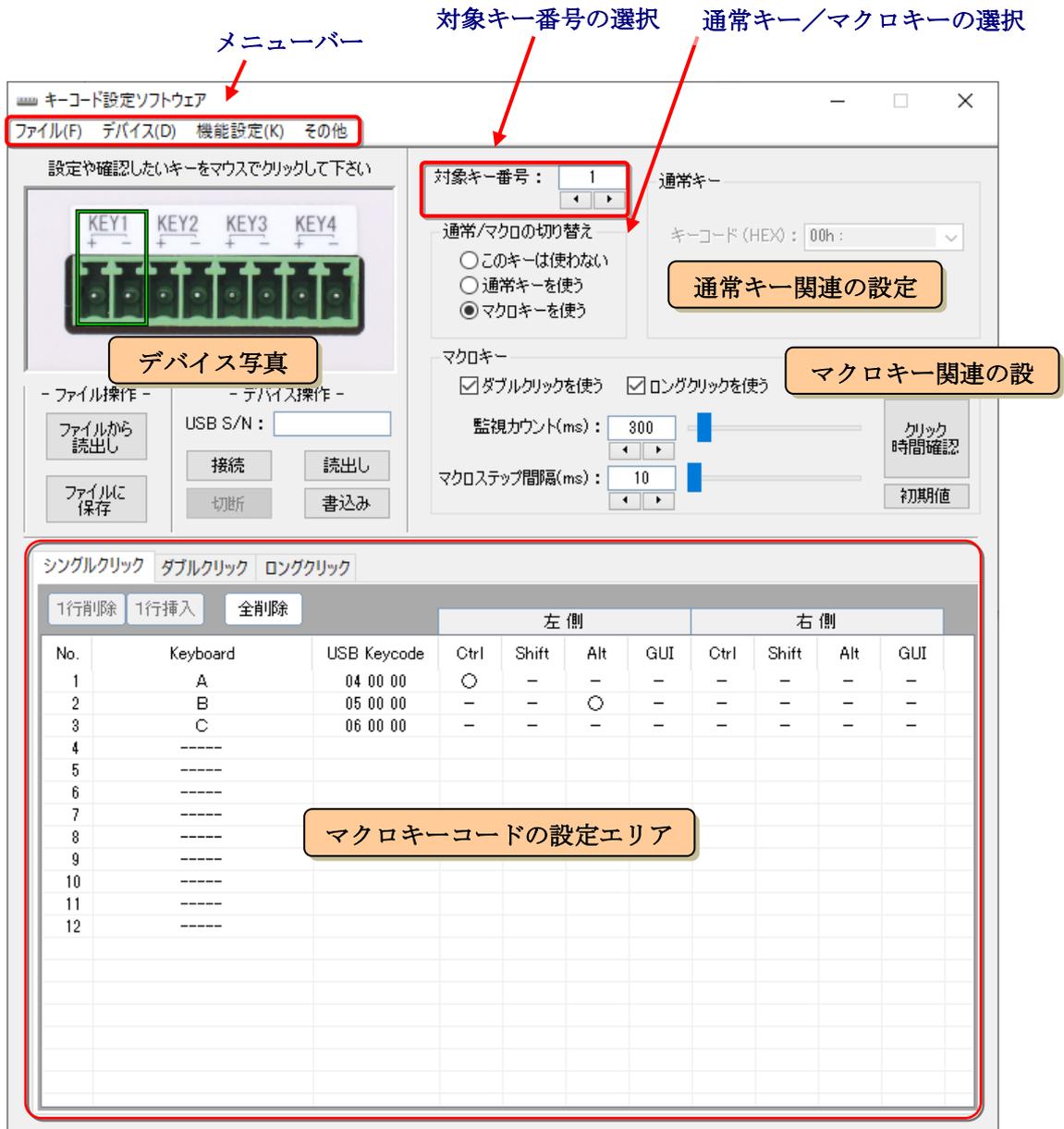


図 11 メイン画面

- 対象キー番号 : 数値の下のボタンでも変更できますが、デバイス写真の端子や端子名の場所をマウスでクリックしても変更できます。
- 通常／マクロの切り替え : 本ソフトウェアを起動した直後は、「このキーは使わない」になっています。「7.3.2 基本的な操作の流れ」に従って設定して下さい。
- 通常キーとマクロ関連設定 : 通常キーとマクロキーに関する設定を行います。通常／マクロの切り替えにより、入力可否の状況が変わります。

### 7.3.2. 基本的な操作の流れ

基本的な操作の流れを下図に示します。

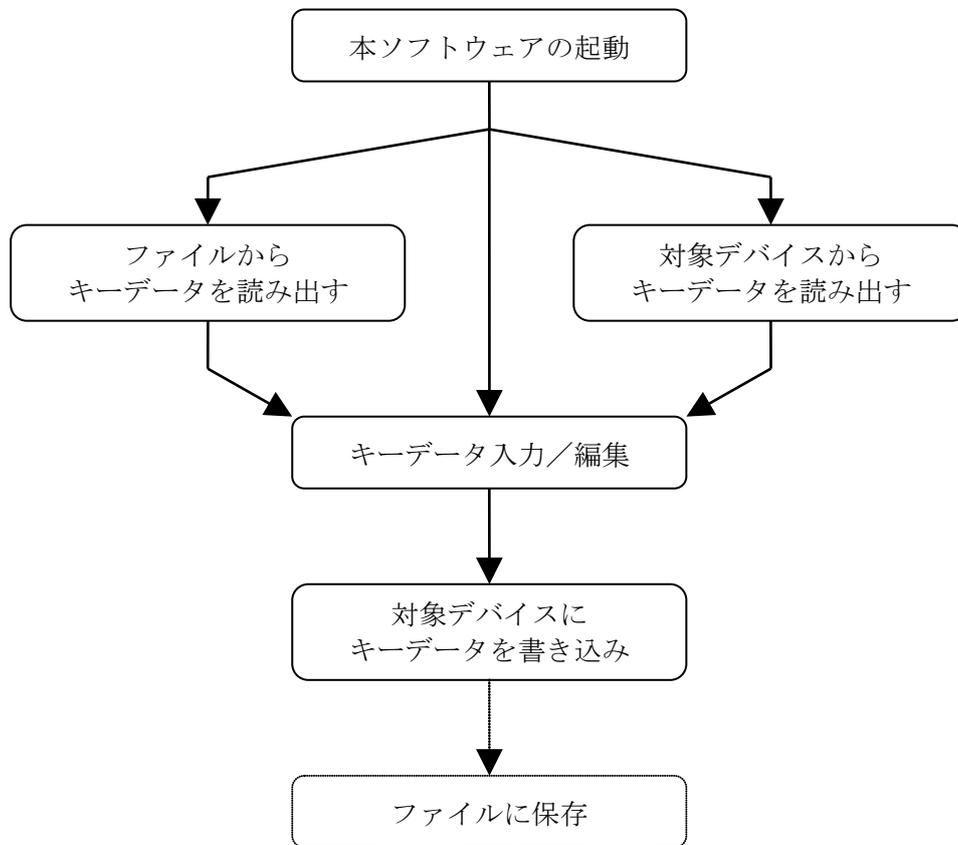


図 12 基本的な操作の流れ

対象デバイスに書き込むキーコードを作成する方法として、①全く新しくキーデータを作成する、②以前に保存しておいたキーデータファイルを元にして編集して作成する、③対象デバイスが記憶しているキーデータを元にして作成する、の3つの方法が考えられます。 それぞれに関する説明は以下で示す段落を参照して下さい。

- ・ 全く新しくキーデータを作成する ⇒ 「7.3.3 手動でキーデータを入力する」を参照。
- ・ キーデータファイルを元にする ⇒ 「7.3.4 ファイルからキーデータを読み出す」を参照。
- ・ デバイスのキーデータを元にする ⇒ 「7.3.5 デバイスからキーデータを読み出す」を参照。

### 7.3.3. 手動でキーデータを入力する

手動でキーデータを入力するには次の手順でデータを設定します。

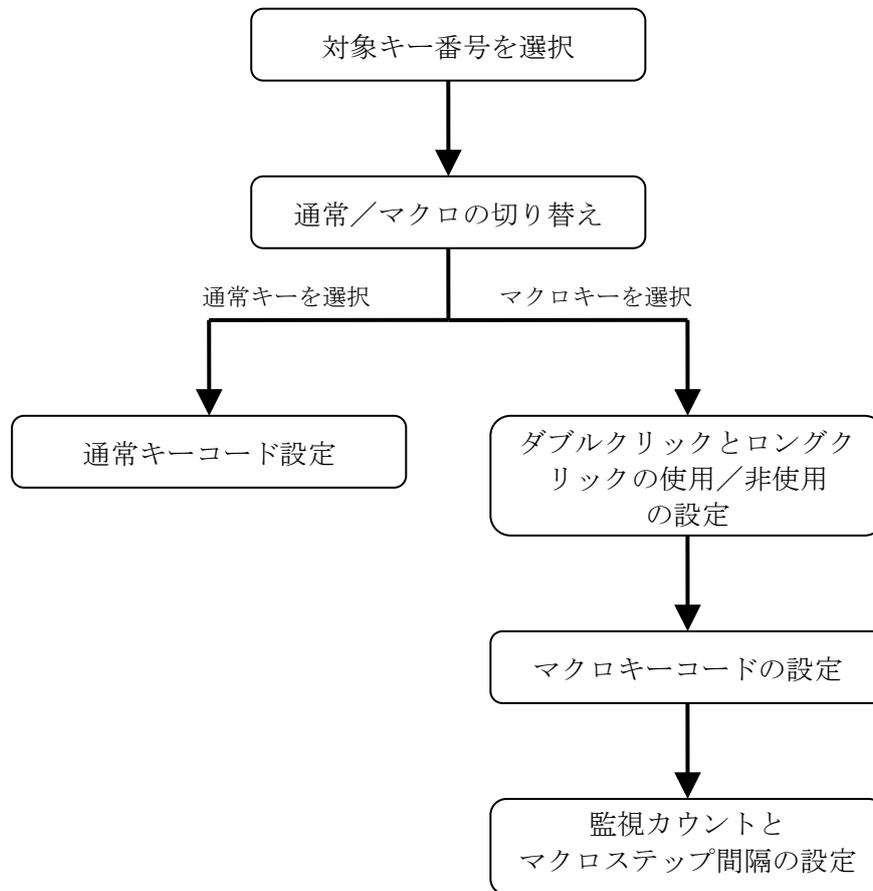


図 13 手動でのキーデータの設定手順

#### (1) 対象キー番号を選択

対象キー番号：

画面上の中央上にある「対象キー番号」を、設定したいキー番号に変更します。

変更するには、数値が表示されているボックスの下にあるスピンドタン（◀ ▶）を使用するか、デバイス写真内の端子や端子名の部分をマウスでクリックします。

#### (2) 通常／マクロの切り替え

通常／マクロの切り替え

このキーは使わない

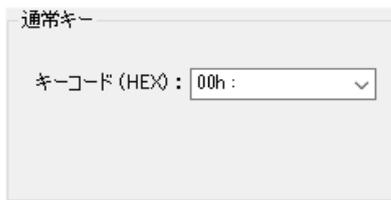
通常キーを使う

マクロキーを使う

「通常／マクロの切り替え」と表示された枠の中から、「通常キーを使う」、「マクロキーを使う」または「このキーは使わない」のどれかを選びます。「通常キーを使う」を選ぶと通常キーに関する設定項目のみが入力可能になり、「マクロキーを使う」を選ぶとマクロキーに関する設定項目のみが入力可能になります。

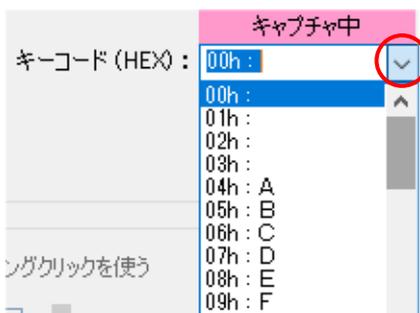
このキーからの入力を行わない場合は「このキーは使わない」を選んで下さい。

### (3) 通常キーコード設定



「通常キー」の枠の中にあるキーコードを設定します。キーコードは、リストによる入力や、キャプチャモードによる入力が可能です。

#### ・ リストによるキーコードの入力



赤丸の部分をクリックすると、左図のようにリストが現れるので、その中から希望のコードを選びます。

リストの左側の数値 (xxh と書かれた部分) は、キーコードを 16 進数で表したものです。: (コロン) の右側は、対応するキーボード上のキーです。

たとえば、キーコード 04h を選ぶと、キーボード上の A の刻印のキーを押したのと同様の動作を対象デバイスに設定できます。

#### ・ キャプチャモードによるキーコードの入力



キャプチャモードが設定されている場合、キーコードをマウスでクリックして入力状態にすると、キーコード表示の上に「キャプチャ中」の文字が表示されます。

この状態で、キーボード上のキーを押すと、そのキーに相当するキーコードが自動的に選択されます。たとえば、キーボードの 'A' を押すと、上図のように、キーコード 04h が選ばれます。

### (4) ダブルクリックとロングクリックの使用／非使用の設定



ダブルクリックを使いたい時は「ダブルクリックを使う」をマウスでクリックしてチェックを入れます。

また、ロングクリックを使いたい時は「ロングクリックを使う」にチェックを入れます。

1つのキーに対するマクロ数は決まっているので、ダブルクリックとロングクリックの使用／非使用により、シングルクリックやダブルクリックに設定できるマクロ数が変わってきます。(画面下側に表示されているリストの行数が適切な数に自動的に変わります)

## (5) マクロキーコードの設定

マクロキーコードは画面の下半分に表示されているタブ付きのリスト画面に設定します。

No.	Keyboard	USB Keycode	左側				右側				
			Ctrl	Shift	Alt	GUI	Ctrl	Shift	Alt	GUI	
1	-----										
2	-----										
3	-----										
4	-----										
5	-----										
6	-----										
7	-----										
8	-----										
9	-----										
10	-----										
11	-----										
12	-----										

図 14 マクロキーコード入力画面

上図は、「ダブルクリックを使う」、「ロングクリックを使う」の両方をチェックした場合の画面です。

右上のタブをクリックすることにより、「シングルクリック」「ダブルクリック」「ロングクリック」の各入力のマクロを設定できます。

リストの1行が1つのマクロステップに相当し、対象デバイス上では1から順番に処理されます。

1つのマクロステップには、キーコードが3つと、修飾キー（左右のCtrlキー、左右のShiftキー、左右のAltキー、および左右のGUIキー）が同時に設定できます。

※ GUIキーはWindow OSではWindowsロゴキーに該当します。

### ・ 入力行の選択

リストの行をマウスでクリックしてください。

たとえば1行目をマウスでクリックすると、下図のようになります。

No.	Keyboard	USB Keycode	左側				右側				
			Ctrl	Shift	Alt	GUI	Ctrl	Shift	Alt	GUI	
1	---	00 00 00	-	-	-	-	-	-	-	-	-
2	-----										
3	-----										

1行目が青色の表示になり、現在1行目の入力状態であることを示しています。また、キャプチャモードであると「USB Keycode」と書かれたヘッダーの上には、「キャプチャ中」と表示されます。

## ・キーコードの入力

キーコードの入力は通常キーコードと同様に、キャプチャモードによるキーコード入力と、リストによるキーコード入力ができます。

### ・キャプチャモードによるキーコードの入力

キャプチャモードによる入力は、「キャプチャ中」の表示がされている状態で、キーボード上のキーを押します。

たとえば、キーボードの「A」を押すと、下図のようになります。キャプチャモードでは、押されたキーに該当するキーコードが自動的に選ばれ、USB Keycode の欄にそのキーコードが、Keyboard の欄にキーボード上の相当するキーが表示されます。

No.	Keyboard	USB Keycode
1	A	04 00 00
2	-----	
3	-----	

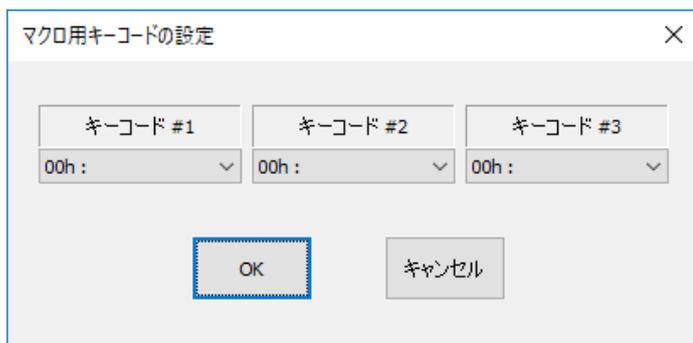
キーコードは1ステップに対して3つまで入力することが可能です。

すでに入力済みのキーコードをもう一度入力すると、そのキーコードを取り消すことができます。上図の例では、もう一度キーボードの「A」を押すと、設定が取り消され、そのキーコードが消えます。

また、すでに3つのキーコードが入力されている状態で、新たなキーコードを入力すると、一番左側のキーコードは削除され、新たなキーコードは一番右に追加されます。

### ・リストによるキーコードの入力

リストによる入力の場合は、入力したい行の USB KeyCode 欄をダブルクリックして下さい。すると、下図のウィンドウが開きます。



コードが表示されたボックスの右側のボタンをクリックしてリストを出し、希望のキーコードを選択して下さい。

設定が終わったら、「OK」ボタンをクリックしてウィンドウを閉じます。

設定をせずに途中で終了するには「キャンセル」ボタンをクリックして下さい。

## (6) 修飾キーの入力

修飾キーの入力は、キャプチャモードによる入力とマウスクリックによる直接入力との2つの方法があります。

キャプチャモードによる入力の場合、入力した行を選択後、「キャプチャ中」の表示がされている状態キーボード上の修飾キーを押して下さい。

たとえば、キーボード上の左 Alt を押すと下図のような表示になります。

キャプチャ中	左側				右側			
USB Keycode	Ctrl	Shift	Alt	GUI	Ctrl	Shift	Alt	GUI
00 00 00	-	-	○	-	-	-	-	-

左側の Alt の欄に「○」が表示され、キーが設定されたことが示されます。

なお、もう一度、左 Alt を押すと「○」の表示が消え、キーが解除されます。

マウスクリックによる直接入力は、マウスのダブルクリックで行います。

設定したい行の希望のキーの欄をマウスでダブルクリックして下さい。その欄に「○」が表示され、キーが設定されたことが示されます。解除するには、もう一度同じ欄をダブルクリックします。

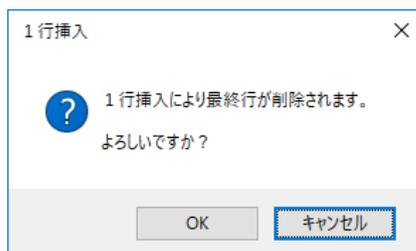
## (7) マクロステップの編集（削除・挿入）

マクロステップ入力画面の左上には、「1行削除」「1行挿入」「全削除」のボタンがあり、これによりマクロステップを編集することができます。「全削除」はいつでも使用可能ですが、「1行削除」「1行挿入」ボタンは、選択されている行が未設定でない場合にだけ使用可能です。

「全削除」ボタンをクリックすると、表示されているマクロステップをクリアして、「図 14 マクロキーコード入力画面」のような未入力状態に戻ります。全削除は、他のキー番号や他のクリック種別には影響を与えません

「1行削除」ボタンをクリックすると、現在の選択行が削除され、それ以降の行が上に詰められます。

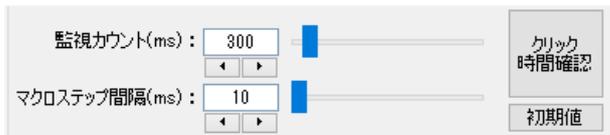
「1行挿入」ボタンをクリックすると、現在の選択行の前に1行、空の行が入力され、選択行以降の行は1行下に移動します。最終行が設定状態の場合に1行挿入を行うと、左図の



ように、問い合わせのウィンドウが表示されますので、状況に合わせて「OK」か「キャンセル」をクリックして下さい。

## (8) 監視カウントとマクロステップ間隔の設定

監視カウントとマクロステップ間隔を画面上で調整が可能です（下図）。

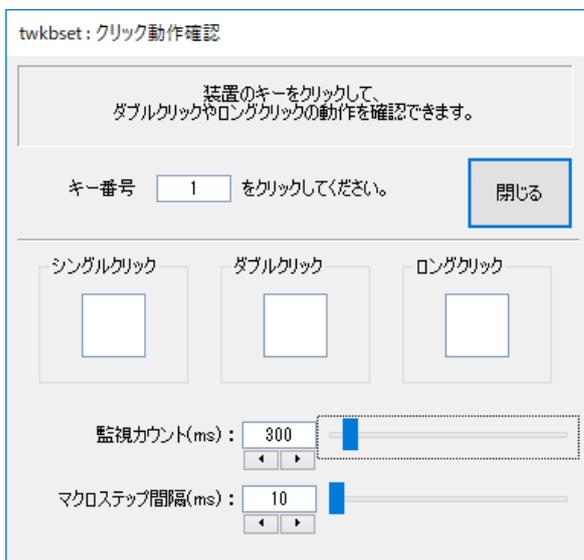


監視カウント(ms) : 300  
マクロステップ間隔(ms) : 10  
クリック時間確認  
初期値

監視カウントは、ダブルクリックやロングクリックであるかどうかを判定するための監視時間です。

マクロステップ間隔は、マクロステップのステップ間に入れられるインターバル時間です。

「クリック時間確認」ボタンをクリックすると、下図のウィンドウが開き、動作状況を見ながら時間設定をする事が可能となります。



twkbsset: クリック動作確認

装置のキーをクリックして、  
ダブルクリックやロングクリックの動作を確認できます。

キー番号 1 をクリックしてください。 閉じる

シングルクリック    ダブルクリック    ロングクリック

監視カウント(ms) : 300  
マクロステップ間隔(ms) : 10



数値が表示された状態

シングルクリック    ダブルクリック    ロングクリック

12    12    12

キー番号に表示されている対象デバイス上の端子へ“ON”状態を入力して下さい。

上図を例とした場合、入力端子 1-2 番間へシングルクリックを入力とすると「シングルクリック」と書かれた枠に、1 から 12 まで数値が順番に表示されます。 1 から 12 の数値はマクロステップの番号を意味しています。

マクロステップの進行速度は、「マクロステップ間隔」の数値で調整します。

ダブルクリックを入力すると、「ダブルクリック」の枠に 1 から 12 までの数値が表示されます。

ロングクリックの場合も同様に、「ロングクリック」の枠に 1 から 12 までの数値が表示されます。

ダブルクリックやロングクリックと判定される時間は、「監視カウント」の数値で調整します。

調整が済んだら、「閉じる」ボタンをクリックします。ここで設定した値が、元の画面に反

映されます。

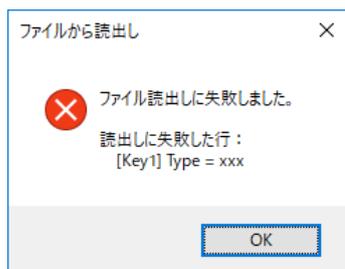
※ 調整中の数値は一時的な値として対象デバイスに書かれているだけなので、最後には「デバイスへの書込み」を行う必要があります。

#### 7.3.4. ファイルからキーデータを読み出す

以前に本ソフトウェアにより作成したキーデータのファイルがあれば、それを読み出すことができます。 ファイルからの読み出しは以下の手順で行います。

- 1、画面上にある「ファイルから読出し」ボタンをクリックするか、メニューバーの「ファイル」→「ファイルから読出し」を選択します。⇒「ファイルを開く」ダイアログが表示されます。
- 2、「ファイルを開く」ダイアログで、読み出したいファイルを選んで「開く」ボタンをクリックします。

ファイルからの読み出しが失敗すると、下図のようなメッセージが表示されます。



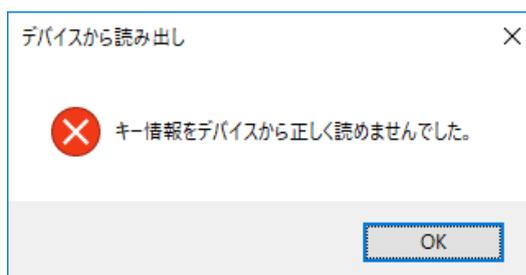
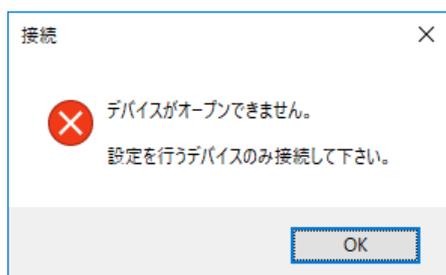
この場合、読み出しに失敗した行に関する情報が表示されません。左図の例では、[Key1] と書かれた行に続く、Type = x x x と書かれた行で問題が発生したことを示しています。(x x x は実際のファイル上では数値などが書かれています) この情報を元に、必要であればファイル内容が正しいか確認して下さい。

#### 7.3.5. デバイスからキーデータを読み出す

対象デバイスが接続されていれば、対象デバイス内に記録しているキーデータを読み出すことが可能です。対象デバイスからの読み出しは以下の手順で行います。

- 1、画面上にある「読出し」ボタンをクリックするか、メニューバーの「デバイス」→「デバイスから読出し」を選択します。
- 2、「デバイスから読み出して良いか？」の問い合わせのウィンドウが表示されるので、良ければ「OK」をクリックします。

対象デバイスからの読み出しが失敗すると、下図のようなメッセージが表示されます。



---

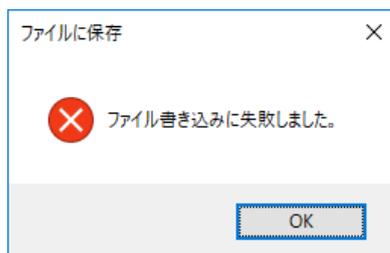
このようなメッセージが表示された場合は、対象デバイスが正しくPCに接続されているかを確認して下さい。正しく接続されているにも関わらず上図のようなメッセージが表示される場合は故障も考えられます。

### 7.3.6. ファイルにキーデータを保存する

キーデータをファイルに保存することができます。ファイルに保存するには以下の手順で行います。

- 1、画面上にある「ファイルに保存」ボタンをクリックするか、メニューバーの「ファイル」→「ファイルに保存」を選択します。
- 2、「名前を付けて保存」ダイアログで、保存したい場所を選び、さらにファイル名を入力して、「保存」ボタンをクリックします。  
拡張子が無いファイル名を指定した場合、自動的に「.twk」の拡張子が付けられます。たとえば、「test」というファイル名を入力して保存すると、「test.twk」というファイル名で保存されます。

保存に失敗すると、下図のようなメッセージが表示されます。

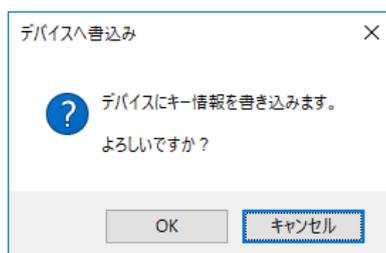


このようなメッセージが表示された場合は、書き込み先のフォルダへの書き込みが可能でない、または、指定したファイルが存在しており書き換えが不可能である、などの原因が考えられます。

### 7.3.7. デバイスにキーデータを書き込む

入力したキーデータの内容をデバイスに反映させるには、対象デバイスにキーデータを書き込む必要があります。対象デバイスへの書き込みは以下の手順で行います。

- 1、画面上にある「書込み」ボタンをクリックするか、メニューバーの「デバイス」→「デバイスへの書込み」を選択します。
- 2、下図のように「デバイスに書き込んでよいですか？」の問い合わせのウィンドウが表示されるので、良ければ「OK」をクリックします。



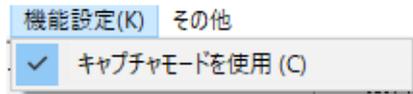
---

## 7.4. その他の機能

### 7.4.1. キャプチャモードの ON/OFF

通常キーコードやマクロキーコードを入力する際のキャプチャモードを ON/OFF することが可能です。本ソフトウェアを起動した直後は、ON になっています。

切り替えは、メニューバーの「機能設定」 → 「キャプチャモードを使用」 を選びます。選ぶ度に ON/OFF を切り替えます。



左図のように「キャプチャモードを使用」の前にチェックがある場合は、現在 ON になっています。

### 7.4.2. 未設定データに初期化

すべてのデータを未設定の状態に戻すことができます。未設定データになると、通常キーもマクロキーも選択されていない状態となり、画面は、本ソフトウェアの起動直後と同等になります。

実行するには、メニューバーの「その他」 → 「未設定データに初期化」 を選びます。

「初期化してよろしいか？」のメッセージが表示されるので、「OK」をクリックすると初期化されます。

### 7.4.3. ソフトウェア・バージョン情報の表示

本ソフトウェアのバージョン情報を表示可能です。

メニューバーの「ファイル」 → 「バージョン情報」 で表示されます。

## 8. プログラミング

サンプルプログラムは、弊社のホームページ「<http://www.techw.co.jp/SupportFrm.html?pid=KB-IT4>」の「サンプルプログラム」からダウンロードいただけます。

表 10 言語別のサンプルファイル

言語	ダウンロードファイル内の格納フォルダ	ソリューションファイル
Visual C++ (MFC) <sup>3</sup>	SAMPLES¥it4_sample	it4_sample.sln
Visual Basic <sup>3</sup>	SAMPLES¥it4_sample_vb	it4_sample_vb.sln
Visual C# <sup>3</sup>	SAMPLES¥it4_sample_cs	it4_sample_cs.sln
Visual Basic for Application	SAMPLES¥it4_sample_vba	-

### 8.1. プログラミングの準備

プログラミングに必要なファイルは、製品の設定ツール「KB-IT4\_Tools」をインストールした場合、ローカルドライブにコピーが作られ、デフォルトの設定では、Windows 10 の場合[スタート]メニュー→[アプリの一覧]→[テクノウェーブ]→[ライブラリ]を、Windows 7 の場合[スタート]メニュー→[すべてのプログラム]→[テクノウェーブ]→[ライブラリ]を選択して表示することができます。

また、これらのファイルは、弊社ホームページ「<http://www.techw.co.jp/SupportFrm.html?pid=KB-IT4>」の「KB-IT4 用 TWKB ライブラリ」からダウンロードすることもできます。

#### 8.1.1. C/C++ での開発に必要なファイル

表 11 は C/C++ で開発を行うために必要なファイルです。

表 11 C/C++での開発に必要なファイル

ファイル名	説明	ダウンロードファイル内のフォルダ
usb_kb.h	TWKB ライブラリを使用するためのヘッダーファイル	¥kb-io_dll
usb_kb.lib (32bit 用)	TWKB ライブラリを静的にリンクするための lib ファイル	¥kb-io_dll
usb_kb.lib (64bit 用)		¥kb-io_dll¥x64

「usb\_kb.h」は、TWKB ライブラリの関数や定数を使用するソースファイルでインクルードしてください。

「usb\_kb.lib」はプロジェクトをビルドする際のリンクファイルに含める必要があります。Visual Studio では、リスト 1 のように `#pragma` を使用してソースファイル中でリンク指定することもできます。

#### リスト 1 インクルードとリンク指定

```
#include "usb_kb.h"
#pragma comment(lib, "usb_kb.lib")
```

これらのファイルはコンパイラがビルド時に検索できるフォルダにコピーしておく必要が

<sup>3</sup> Visual Studio 2010 で作成されています。ご利用のバージョンによっては変換作業が必要になります(ソリューションファイルを開くと自動的に変換ウィザードが起動します)。

---

あります。最も簡単な方法は、ビルドするプロジェクトと同一フォルダにコピーすることです。

複数のプロジェクトを開発する場合は、これらのファイルを格納したフォルダを、開発環境の標準のインクルードパスや標準のリンクパスに追加すると便利です。追加の方法は開発環境によって異なりますので、それぞれのオンラインヘルプなどを参照してください。

### 8.1.2. Visual Basic、C# での開発に必要なファイル

表 12 は Visual Basic、または、C# で開発を行うために必要なファイルです。

表 12 Visual Basic、C#での開発に必要なファイル

開発環境	ファイル名	説明	ダウンロードファイル内のフォルダ
Visual Basic	usb_kb.vb	TWKB ライブラリを使用するための定義ファイル	¥kb-io_dll
Visual C#	usb_kb.cs		

どちらの開発環境の場合も、Visual Studio の「ソリューション エクスプローラ」を開き、対応するファイルを開発プロジェクトの中にドラッグ・アンド・ドロップで追加することで、TWKB ライブラリの呼び出しが可能になります。これらのファイルは 32 ビット、64 ビットのどちらのプログラムを作成する場合にも共通で利用可能です。

### 8.1.3. Visual Basic for Applications での開発に必要なファイル

表 13 は Microsoft Office 製品の VBA で開発を行うために必要なファイルです。

表 13 Visual Basic for Applications での開発に必要なファイル

ファイル名	説明	ダウンロードファイル内のフォルダ
usb_kb.bas	TWKB ライブラリを使用するための定義ファイル	¥kb-io_dll

開発を行うアプリケーションソフトで [Alt] + [F11] キーを押し、Visual Basic Editor を起動し、上記ファイルをプロジェクトウィンドウにドラッグ・アンド・ドロップで追加することで、TWKB ライブラリの呼び出しが可能になります。

- |  |
|--|
| <ul style="list-style-type: none"><li>プロジェクトに追加したファイルは、ドキュメントファイル内にコピーが作成されます。ファイルを更新する場合は、以前に追加したファイルを一度解放し、新しいファイルを追加してください。</li></ul> |
|--|

## 8.2. 実行時に必要なファイル

表 14 は開発したアプリケーションプログラムを実行する場合に必要なファイルです。ファイルは設定ツール（「キーコード設定ソフトウェア」）のインストール時にシステムコピーされます。設定ツールがインストールされていないパソコンで、開発したアプリケーションプログラムを実行する場合は、表 14 を参考にファイルをシステムフォルダ（通常、「¥Windows¥System32」）、もしくはアプリケーションの実行ファイル（.exe）と同一フォルダにコピーしてください。

64 ビット OS に対してファイルを手動でコピーする場合は、作成したアプリケーションプログラムが 32 ビットの命令で動作するものか、64 ビットの命令で動作するものかによって、使用される DLL とシステムフォルダの位置が変わるので注意が必要です（表 15）。

表 14 プログラムの実行に必要なファイル

ファイル名	説明	ダウンロードファイル内のフォルダ	
usb_kb.dll	TWKB ライブラリ本体。製品の設定ツールをインストールすることでシステムフォルダにコピーされます。	32 ビット版	¥kb-io_dll
		64 ビット版	¥kb-io_dll¥x64

表 15 必要となる DLL と対応するシステムフォルダ

OS	実行するプログラム	必要な DLL	DLL のコピー先システムフォルダ
64 ビット OS	32 ビットプログラム	32 ビット版 DLL	「Windows¥SysWOW64」フォルダ
	64 ビットプログラム	64 ビット版 DLL	「Windows¥System32」フォルダ
32 ビット OS	32 ビットプログラム	32 ビット版 DLL	「Windows¥System32」フォルダ

### 32 ビットプログラムか 64 ビットプログラムかを調べる方法

64 ビット OS 上で動作しているプログラムが、32 ビットプログラムか 64 ビットプログラムかは「タスク マネージャー」で調べることができます。[Ctrl] + [Alt] + [Del] キーを押して「タスク マネージャー」を起動し、[プロセス] タブを開きます。実行ファイルの右に「32 ビット」、または、「\*32」という表示があれば 32 ビットプログラムです。

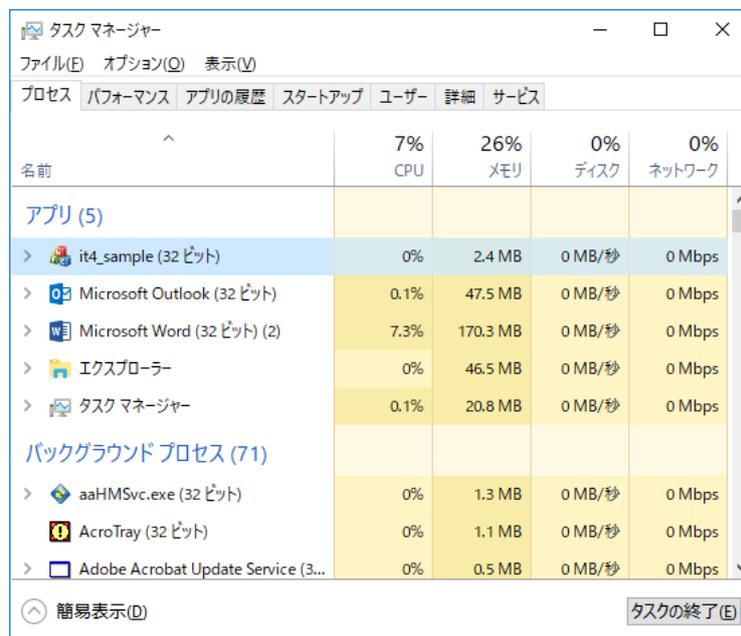


図 15 32 ビットプログラムの「タスク マネージャー」での表示

---

### 8.3. 関数リファレンス

製品の制御は、以下で説明するライブラリ関数を使用して行います。各関数の説明は、C/C++、Visual Basic、Visual Basic for Applications、Visual C#、それぞれにおけるプロトタイプ、変数の説明、動作説明の順になっています。

関数の戻り値は 32 ビットの整数で関数の実行結果を表します（以下参照）。

#### 8.3.1. 関数の戻り値の意味

以下に主な戻り値と、予想される原因、対処方法を示します。括弧内の数値が関数から返される数値です。尚、戻り値を示す各定数は各言語用の定義ファイル（拡張子が「.h」、「.vb」、「.bas」、「.cs」のファイル）中で定義されています。

##### **TWKB\_OK (H' 00000000)**

正常終了。

##### **TWKB\_INVALID\_HANDLE (H' 00000001)**

デバイスのハンドルが無効であることを示します。既に切断されたハンドルや、接続されていないハンドルを関数に渡した可能性があります。プログラムの内容を確認してください。

##### **TWKB\_DEVICE\_NOT\_FOUND (H' 00000002)**

デバイスが見つからなかった場合に返ります。デバイスの電源が入っていない、他のプログラムがデバイスを使用しているなどの理由が考えられます。デバイスの状態を確認してください。

##### **TWKB\_IO\_ERROR (H' 00000004)**

デバイスとの通信中にエラーが発生したことを示します。ケーブルが抜かれた場合などに発生します。接続を確認してください。

##### **TWKB\_INVALID\_ARGS (H' 00000010)**

関数に与えられた引数が無効であることを示します。引数の値が誤っている可能性がありますので、プログラムの内容を確認してください。

##### **TWKB\_NOT\_SUPPORTED (H' 00000011)**

サポートされない機能が呼び出されたことを示します。接続中のデバイスがその機能をサポートしない場合などに発生します。デバイスの仕様や状態を確認してください。

##### **TW\_OTHER\_ERROR (H' 00000012)**

予期しないエラーを示します。お手数ですが製品サポートにお問い合わせください。

## 8.4. マルチスレッドプログラムからの呼び出しについて

ライブラリでは複数のスレッドからの関数呼び出しをサポートしていますが、デバイスとの通信仕様により、1つのデバイスを複数のスレッドから同時に制御することはできません。何らかの理由により、複数のスレッドから1つのデバイスにアクセスする必要がある場合には、クリティカルセクションなどを使用することにより、ライブラリ関数の呼び出しをシリアル化し、複数のスレッドが同時に1つのデバイスにアクセスしないようにプログラムしてください。

1つのスレッドが1つのデバイスのみを制御する場合は、複数のスレッドから同時にライブラリ関数を呼び出しても問題ありません。

## 8.5. 関数一覧

### TWKB\_OpenEx

言語	関数宣言
C/C++	TWKB_STATUS TWKB_OpenEx (HANDLE *phDev, LPCTSTR *pSerial, long Opt)
VB	Function TWKB_OpenEx (ByRef phDev As System.IntPtr, ByVal pID As String, ByVal Opt As OPEN_OPT) As Integer
VBA	Function TWKB_OpenEx (ByRef phDev As Long, ByVal pID As String, ByVal Opt As USB_KB_OPEN_OPT) As Long
C#	STATUS OpenEx(out System.IntPtr phDev, string pSerial, OPEN_OPT) STATUS OpenEx(out System.IntPtr phDev, string pSerial) STATUS OpenEx(out System.IntPtr phDev)

phDev : [出力]取得したハンドルの格納先  
pSerial : [入力]USBのシリアル番号  
Opt : [入力]接続オプション。以下をORで結合  
TWKB\_TYPE\_IOPAD4 (0x00010000) : KB-IOPAD4に接続する  
TWKB\_TYPE\_IT4 (0x00040000) : KB-IT4に接続する  
TWKB\_TYPE\_ANY (0x00FF0000) : 製品タイプを問わない

デバイスに接続しハンドルを取得します。以降、接続したデバイスを制御するためには取得したハンドルが必要になります。

pSerialにUSBのシリアル番号を指定すると、その番号の製品へ接続を行います(指定が無い場合NULLまたは""としてください)。USBのシリアル番号は8文字の文字列で"KB000001"のような形式です。下位4桁は製品のシリアル番号と一致します。

Optには接続する製品タイプを指定します。デバイスを限定しない場合はTWKB\_TYPE\_ANY(相当の)オプションを指定してください。

### TWKB\_Open()

言語	関数宣言
C/C++	TWKB_STATUS TWKB_Open(HANDLE *phDev, LPCTSTR *pSerial)
VB	Function TWKB_Open(ByRef phDev As System.IntPtr, ByVal pID As String) As Integer
VBA	Function TWKB_Open(ByRef phDev As Long, ByVal pID As String) As Long
C#	STATUS Open(out System.IntPtr phDev, string pSerial) STATUS Open(out System.IntPtr phDev)

phDev : [出力]取得したハンドルの格納先  
pSerial : [入力]USBのシリアル番号

デバイスに接続しハンドルを取得します。以降、接続したデバイスを制御するためには取得したハンドルが必要になります。

pSerial に USB のシリアル番号を指定すると、その番号の製品へ接続を行います (指定が無い場合、NULL、または、""としてください)。USB のシリアル番号は 8 文字の文字列で "KT001234" のような形式です。下位 4 桁は製品のシリアル番号と一致します。

### TWKB\_Close()

言語	関数宣言
C/C++	TWKB_STATUS TWKB_Close(HANDLE hDev)
VB	Function TWKB_Close(ByVal hDev As System.IntPtr) As Integer
VBA	Function TWKB_Close(ByVal hDev As Long) As Long
C#	STATUS Close(System.IntPtr hDev)

hDev : [入力]デバイスのハンドル

デバイスの操作を終了します。

### TWKB\_GetSerial()

言語	関数宣言
C/C++	TWKB_STATUS TWKB_GetSerial(HANDLE hDev, LPTSTR *pSerial)
VB	Function TWKB_GetSerial(ByVal hDev As System.IntPtr, ByVal pSerial As System.Text.StringBuilder) As Integer
VBA	Function TWKB_GetSerial(ByVal hDev As Long, ByVal pSerial As String) As Long
C#	STATUS GetSerial(System.IntPtr hDev, System.Text.StringBuilder pSerial)

hDev : [入力]デバイスのハンドル  
pSerial : [出力]USB シリアル番号の格納先

接続中のデバイスの USB シリアル番号を取得します。

### TWKB\_KeyEnable()

言語	関数宣言
C/C++	TWKB_STATUS TWKB_KeyEnable(HANDLE hDev, BOOL flgEnable)
VB	Function TWKB_KeyEnable(ByVal hDev As System.IntPtr, ByVal flgEnable As Integer) As Integer
VBA	Function TWKB_KeyEnable(ByVal hDev As Long, ByVal flgEnable As Long) As Long
C#	STATUS KeyEnable(System.IntPtr hDev, bool flgEnable)

hDev : [入力]デバイスのハンドル  
 flgEnable : [入力]キーボード動作の許可/禁止を指定  
           0 : キーコード通知動作を禁止  
           0以外 : キーコード通知動作を許可

キーコード通知動作の許可/禁止を設定します。キーコード通知動作を禁止すると、入力端子が“ON”になってもキーコードの通知が行われません。

### TWKB\_ReadSw()

言語	関数宣言
C/C++	TWKB_STATUS TWKB_ReadSw(HANDLE hDev, BYTE *pSw, BYTE *pSwHyst)
VB	Function TWKB_ReadSw(ByVal hDev As System.IntPtr, ByRef pSw As Byte, ByRef pSwHyst As Byte) As Integer
VBA	Function TWKB_ReadSw(ByVal hDev As Long, ByRef pSw As Byte, ByRef pSwHyst As Byte) As Long
C#	STATUS ReadSw(System.IntPtr hDev, out byte pSw, out byte pSwHyst) STATUS ReadSw(System.IntPtr hDev, out byte pSw)

hDev : [入力]デバイスのハンドル  
 pSw : [出力]現在の入力状態  
 pSwHyst : [出力]前回、読み出してから1度でも押されたスイッチを示す

キーコード通知動作とは無関係に入力端子の情報を返します。1から4番までの各入力、\*pSw、\*pSwHystの0から3ビットと対応しています。\*pSwは現在の入力状態を示し、“ON”となっている入力に対応するビットが1となります。\*pSwHystは前回の読み出し以降に“ON”となった入力を示します。不要な引数はNULLにできます。

---

## 9. 保証期間

本製品の保証期間は、お買い上げ日より1年間です。保証期間中の故障につきましては、無償修理または代品と交換で対応させていただきます。ただし、以下の場合は保証期間内であっても有償での対応とさせていただきますのでご了承ください。

- 1) 本マニュアルに記載外の誤った使用方法による故障。
- 2) 火災、震災、風水害、落雷などの天災地変および公害、塩害、ガス害などによる故障。
- 3) お買い上げ後の輸送、落下などによる故障。

製品に関する情報、最新のバージョンなどは弊社ホームページにてご案内しております。また、お問い合わせ、ご質問などは下記までご連絡ください。(弊社ホームページの「お問い合わせフォーム」もご利用ください)

## 10. サポート情報

製品に関する情報、最新のユーティリティなどは弊社ホームページにてご案内しております。また、お問い合わせ、ご質問などは下記までご連絡ください。

テクノウェーブ(株)

URL : <https://www.techw.co.jp>

E-mail : [support@techw.co.jp](mailto:support@techw.co.jp)

- (1) 本書、および本製品のホームページに掲載されている応用回路、プログラム、使用方法などは、製品の代表的動作・応用例を説明するための参考資料です。これらに起因する第三者の権利（工業所有権を含む）侵害、損害に対し、弊社はいかなる責任も負いません。
- (2) 本書の内容の一部または全部を無断転載することをお断りします。
- (3) 本書の内容については、将来予告なしに変更することがあります。
- (4) 本書の内容については、万全を期して作成いたしました。が、万一ご不審な点や誤り、記載もれなど、お気づきの点がございましたらご連絡ください。

#### 改訂記録

年月	版	改訂内容
2017年12月	初	
2017年12月	2	・誤記を修正
2018年1月	3	・誤記を修正
2020年8月	4	・設定ツール追加に対応 ・関数追加に対応 ・誤記を修正
2021年4月	5	・コネクタの型番を追加 ・誤記を修正