

## USB汎用I/Oインターフェース GPIO-USB-01

本ボードはUSB接続の汎用入出力インターフェースです（以後、単にインターフェースと呼びます。）ユーザーは専用DLLによるAPI関数を用いて簡単にデジタル信号の入出力が行えます。

インターフェース1枚につき16入力、16出力の合計32入出力を備えており、複数のインターフェースを併用することにより、より多くの入出力を行うことができます。

### 1. API関数

インターフェースには専用DLL(gpio.dll)が添付され、ユーザーはダイナミックロードあるいは静的リンクすることによってAPI関数を使用することができます。DLLの構成ファイルとして以下の3ファイルが供給されます。

- ・ gpio.dll （DLL本体）
- ・ gpio.h （ヘッダーファイル）
- ・ gpio.lib （静的リンク用ライブラリ）

API関数を用いて複数のプログラムから同一デバイスにアクセスした場合、片方の実行が終了するまでは、もう片方のプログラムに対してはデバイス使用中である旨のステータスが返ります。したがってユーザーはこのステータスをハンドルしてプロセス間の同期をとる必要があります。

最初にユーザーが行うことはインターフェースをオープンすることです。正常にオープンされれば、以後、そのインターフェースに対して、制御を行うことができるようになります。

## 2 - 1 . 関数リファレンス

### 2 - 1 - 1 . GPIO\_ScanInterface

実装されているインターフェースをスキャンし、その台数を返します。

#### 宣言

```
GPIO_STATUS GPIO_ScanInterface (unsigned int* puiNumber)
```

#### 引数

puiNumber インターフェース数を受け取る変数へのポインタ

#### 戻り値

状況に応じたステータスコードが返ります。(関数ステータスの項を参照)

#### 説明

本関数が正常に終了した場合、GPIO\_OK が返り、puiNumber で示される変数にはインターフェース数が代入されています。

## 2 - 1 - 2 . GPIO\_OpenInterface

インターフェースをオープンし、使用可能な状態にします。

### 宣言

```
GPIO_STATUS GPIO_OpenInterface (unsigned int uiInterface, GPIO_HANDLE *gpioHandle)
```

### 引数

uiInterface    インターフェース番号  
gpioHandle    インターフェースハンドルへのポインタ

### 戻り値

状況に応じたステータスコードが返ります。(関数ステータスの項を参照)

### 説明

uiInterface はオープンすべきインターフェースの番号を指定します。0 から始まりインターフェース台数未満の整数です。

gpioHandle は GPIO\_HANDLE 型変数へのポインタで、本関数が正常に終了すると GPIO\_OK が返り、gpioHandle で示されるインターフェースハンドルにハンドル値が代入されています。

### 2 - 1 - 3 . GPIO\_CloseInterface

インターフェースをクローズします。

#### 宣言

```
GPIO_STATUS GPIO_CloseInterface (GPIO_HANDLE gpioHandle)
```

#### 引数

gpioHandle    インターフェースハンドル

#### 戻り値

状況に応じたステータスコードが返ります。(関数ステータスの項を参照)

#### 説明

gpioHandle で指定されたインターフェースをクローズします。

本関数が正常に終了すると GPIO\_OK が返ってきます。

## 2 - 1 - 4 . GPIO\_Input

指定したポートをバイト単位で読み取ります。

### 宣言

```
GPIO_STATUS GPIO_Input (GPIO_HANDLE gpioHandle, char cPort, unsigned char* pucData)
```

### 引数

gpioHandle	インターフェースハンドル
cPort	ポート名
pucData	入力データを受取る変数へのポインタ

### 戻り値

状況に応じたステータスコードが返ります。(関数ステータスの項を参照)

### 説明

cPort は読み取るポート名をアスキー文字 1 バイトで指定します。( A 、 B 等)

pucData は読み取ったデータを格納する変数へのポインタです。関数が正常終了したとき、ここで指定した変数にデータが格納されています。

## 2 - 1 - 5 . GPIO\_Output

指定したポートにバイト単位でデータを出力します。

### 宣言

```
GPIO_STATUS GPIO_Output (GPIO_HANDLE gpioHandle, char cPort, unsigned char ucData)
```

### 引数

gpioHandle	インターフェースハンドル
cPort	ポート名
ucData	出力データ

### 戻り値

状況に応じたステータスコードが返ります。(関数ステータスの項を参照)

### 説明

cPort は出力するポート名をアスキー文字 1 バイトで指定します。( C 、 D 等)

ucData は出力するデータを指定します。

## 2 - 1 - 6 . GPIO\_BitTest

入力ポートの特定ビットの状態を調べます。

### 宣言

```
GPIO_STATUS GPIO_BitTest (GPIO_HANDLE gpioHandle, char cPort, unsigned char ucBit)
```

### 引数

gpioHandle	インターフェースハンドル
cPort	ポート名
ucBit	ビット番号

### 戻り値

状況に応じたステータスコードが返ります。(関数ステータスの項を参照)

### 説明

cPort は読み取るポート名をアスキー文字 1 バイトで指定します。( A 、 B 等)

ucBit はビット番号を 0 ~ 7 のバイナリ値で指定します。

## 2 - 1 - 7 . GPIO\_BitSet

出力ポートの特定ビットをセットします。

### 宣言

```
GPIO_STATUS GPIO_BitSet (GPIO_HANDLE gpioHandle, char cPort, unsigned char ucBit)
```

### 引数

gpioHandle	インターフェースハンドル
cPort	ポート名
ucBit	ビット番号

### 戻り値

状況に応じたステータスコードが返ります。(関数ステータスの項を参照)

### 説明

cPort は出力するポート名をアスキー文字 1 バイトで指定します。( C 、 D 等)

ucBit はビット番号を 0 ~ 7 のバイナリ値で指定します。



## 2 - 1 - 7 . GPIO\_BitReset

出力ポートの特定ビットをクリアします。

### 宣言

```
GPIO_STATUS GPIO_BitSet (GPIO_HANDLE gpioHandle, char cPort, unsigned char ucBit)
```

### 引数

gpioHandle	インターフェースハンドル
cPort	ポート名
ucBit	ビット番号

### 戻り値

状況に応じたステータスコードが返ります。(関数ステータスの項を参照)

### 説明

cPort は出力するポート名をアスキー文字 1 バイトで指定します。( C 、 D 等)

ucBit はビット番号を 0 ~ 7 のバイナリ値で指定します。&"

## 2 - 2 . 関数ステータス

A P I関数の戻り値である GPIO\_STSTATUS は以下の通り定義されています。

### **GPIO\_OK**

正常に関数が終了するとこの値が返ってきます。

### **GPIO\_INVALID\_HANDLE**

インターフェースハンドルが間違っています。

### **GPIO\_IO\_ERROR**

インターフェースとの通信でエラーが発生しました。

### **GPIO\_INVALID\_PARAMETER**

引数が間違っています。

### **GPIO\_OTHER\_ERROR**

その他のエラー

### **GPIO\_BUSY**

デバイスは現在、コマンドを受けられません。

### **GPIO\_H\_LEVEL**

ビットテストの結果、指定ポートはHレベルです。

### **GPIO\_L\_LEVEL**

ビットテストの結果、指定ポートはLレベルです。