

EPX-64S プログラミングガイド

Rev 1.2

株式会社エルモス

目次

1 はじめに.....	3
1.1 概要.....	3
1.2 関数について.....	3
1.3 プログラミングの準備.....	4
1.4 関数の戻り値について.....	4
1.5 注意事項.....	4
2 関数リファレンス.....	5
2.1 デバイス関数.....	5
EPX64S_GetNumberOfDevices.....	5
EPX64S_GetSerialNumber.....	6
EPX64S_Open.....	7
EPX64S_OpenBySerialNumber.....	8
EPX64S_Close	9
2.2 I/O 制御関数.....	10
EPX64S_SetDirection.....	10
EPX64S_GetDirection.....	11
EPX64S_OutputPort.....	12
EPX64S_InputPort.....	13

1 はじめに

1.1 概要

パソコンのUSBポートに接続して、デジタル入出力信号を制御する「EPX-64S API関数」をユーザーアプリケーションから呼び出すことで簡単に **EPX-64S** の I/O 機能を制御することができます。

下図は全体の構成です。

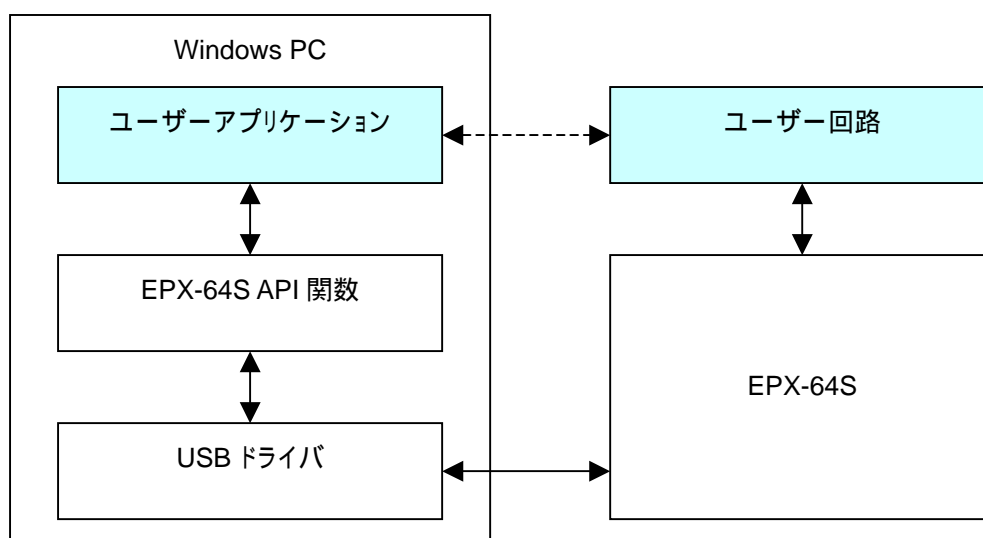


図 1.1 構成

1.2 関数について

「EPX-64S API関数」は関数群をモジュール化した「EPX64S.dll」で提供されます。

「EPX64S.dll」ファイルは **EPX-64S** をインストールする時にシステムフォルダに入ります。

関数は**デバイス関数**、**I/O 制御関数**に分類されます。

デバイス関数はデバイスの検索、デバイスとの接続、切断等の関数です。

I/O 制御関数は I/O 機能を制御する関数です。

1.3 プログラミングの準備

・Visual C++の場合

「EPX64S.h」、「EPX64S.lib」ファイルをプロジェクトに追加してください。

・Visual Basic 6.0 の場合

「EPX64S.bas」ファイルをプロジェクトの標準モジュールに追加してください。

・Visual Basic.NET の場合

「EPX64S.vb」ファイルをプロジェクトに追加してください。

・Visual C#.NET の場合

「EPX64S.cs」ファイルをプロジェクトに追加してください。

これらのファイルは本製品に付属の CD-ROM「¥library」フォルダにあります。

1.4 関数の戻り値について

関数の戻り値の説明は下表に示します。

定数	値	意味
EPX64S_OK	0	正常終了
EPX64S_INVALID_HANDLE	1	デバイスのハンドルが無効
EPX64S_DEVICE_NOT_FOUND	2	デバイスが見つからない
EPX64S_DEVICE_NOT_OPENED	3	デバイスがオープンできない
EPX64S_OTHER_ERROR	4	その他のエラーが発生した
EPX64S_COMMUNICATION_ERROR	5	通信エラーが発生した
EPX64S_INVALID_PARAMETER	6	パラメータが無効

表 1.3 関数の戻り値

1.5 注意事項

複数のアプリケーション、またはマルチスレッドによる **EPX-64S** への同時アクセスはできませんので注意してください。

2 関数リファレンス

2.1 デバイス関数

EPX64S_GetNumberOfDevices

現在接続されているデバイスの数を取得します。

```
int EPX64S_GetNumberOfDevices (int *Number)
```

Parameters

Number 接続デバイス数の格納先へのポインタ

Remarks

デバイスが接続されていない場合は *Number* には 0 が格納されます。

Example

```
int result;  
int number;  
  
result = EPX64S_GetNumberOfDevices(&number);  
if (result == EPX64S_OK) {  
    // EPX64S_GetNumberOfDevices 成功  
}  
else {  
    // EPX64S_GetNumberOfDevices 失敗  
}
```

EPX64S_GetSerialNumber

現在接続されているデバイスのシリアル番号を取得します。

```
int EPX64S_GetSerialNumber (int Index, int *SerialNumber)
```

Parameters

Index 0 から始まる接続デバイスのインデックス

SerialNumber シリアル番号の格納先へのポインタ

Remarks

取得したシリアル番号を引数として **EPX64S_OpenBySerialNumber** でオープンすることができます。尚、このシリアル番号は基板裏面のシリアル番号と同一です。

Example

```
int result;  
int index;  
int serialNumber;  
  
index = 0;  
result = EPX64S_GetSerialNumber(index, &serialNumber);  
if (result == EPX64S_OK) {  
    // EPX64S_GetSerialNumber 成功  
}  
else {  
    // EPX64S_GetSerialNumber 失敗  
}
```

EPX64S_Open

デバイスをオープンし、デバイスのハンドルを取得します。

```
int EPX64S_Open (EPX64S_HANDLE *Handle)
```

Parameters

Handle デバイスのハンドルの格納先へのポインタ

Remarks

デバイスが複数接続されている場合は接続できた最初のデバイスのハンドルを取得します。
取得したハンドルは以後、I/O 制御関数等に引数として渡すことになります。

Example

```
int result;  
EPX64S_HANDLE handle;  
  
result = EPX64S_Open(&handle);  
if (result == EPX64S_OK) {  
    // EPX64S_Open 成功  
}  
else {  
    // EPX64S_Open 失敗  
}
```

EPX64S_OpenBySerialNumber

指定したシリアル番号と一致するデバイスをオープンし、デバイスのハンドルを取得します。

```
int EPX64S_OpenBySerialNumber (int SerialNumber, EPX64S_HANDLE *Handle)
```

Parameters

SerialNumber デバイスのシリアル番号

Handle デバイスのハンドルの格納先へのポインタ

Remarks

シリアル番号は **EPX64S_GetSerialNumber** で取得することができます。

尚、このシリアル番号は基板裏面のシリアル番号と同一です。

取得したハンドルは以後、I/O 制御関数等に引数として渡すことになります。

Example

```
int result;
EPX64S_HANDLE handle;
int serialNumber;    // 取得したデバイスのシリアル番号

result = EPX64S_OpenBySerialNumber(serialNumber, &handle);
if (result == EPX64S_OK) {
    // EPX64S_OpenBySerialNumber 成功
}
else {
    // EPX64S_OpenBySerialNumber 失敗
}
```


EPX64S_Close

デバイスをクローズします。

```
int EPX64S_Close (EPX64S_HANDLE Handle)
```

Parameters

Handle デバイスのハンドル

Example

```
int result;
EPX64S_HANDLE handle;            // オープンしたデバイスのハンドル

result = EPX64S_Close(handle);
if (result == EPX64S_OK) {
    // EPX64S_Close 成功
}
else {
    // EPX64S_Close 失敗
}
```

2.2 I/O 制御関数

EPX64S_SetDirection

ポートの入出力方向を設定します。

```
int EPX64S_SetDirection (EPX64S_HANDLE Handle, BYTE Direction)
```

Parameters

Handle デバイスのハンドル
Direction ポートの入出力方向

Remarks

Direction の各ビットが各ポートの入出力方向に対応しています。
ビットの値が **0** の場合は入力、**1** の場合は出力に設定されます。
初期値 (電源投入時) は全て入力 (**0x00**) に設定されています。

例) *Direction* = **0xD1**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
<i>Direction</i>	1	1	0	1	0	0	0	1
PORT	P7	P6	P5	P4	P3	P2	P1	P0
I/O	OUT	OUT	IN	OUT	IN	IN	IN	OUT

Example

```
int result;  
EPX64S_HANDLE handle;            // オープンしたデバイスのハンドル  
BYTE dir;  
  
dir = 0xD1;    // P7,6,4,0 出力 P5,3,2,1 入力  
result = EPX64S_SetDirection(handle, dir);  
if (result == EPX64S_OK) {  
    // EPX64S_SetDirection 成功  
}  
else {  
    // EPX64S_SetDirection 失敗  
}
```

EPX64S_GetDirection

現在設定されているポートの入出力方向を取得します。

```
int EPX64S_GetDirection (EPX64S_HANDLE Handle, BYTE *Direction)
```

Parameters

Handle デバイスのハンドル
Direction ポートの入出力方向の格納先へのポインタ

Remarks

Direction の各ビットが各ポートの入出力方向に対応しています。
ビットの値が **0** の場合は入力、**1** の場合は出力に設定されています。
初期値 (電源投入時) は全て入力 (**0x00**) に設定されています。

例) *Direction* = **0x5A**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
<i>Direction</i>	0	1	0	1	1	0	1	0
PORT	P7	P6	P5	P4	P3	P2	P1	P0
I/O	IN	OUT	IN	OUT	OUT	IN	OUT	IN

Example

```
int result;  
EPX64S_HANDLE handle;            // オープンしたデバイスのハンドル  
BYTE dir;  
  
result = EPX64S_GetDirection(handle, &dir);  
if (result == EPX64S_OK) {  
    // EPX64S_GetDirection 成功  
}  
else {  
    // EPX64S_GetDirection 失敗  
}
```

EPX64S_OutputPort

ポートに出力します。(8ビットデータ同時出力)

int EPX64S_OutputPort (EPX64S_HANDLE Handle, BYTE Port, BYTE Value)

Parameters

Handle デバイスのハンドル
Port ポートの指定
Value ポート出力値

Remarks

ポートが入力に設定されている場合は出力されません。

Port の各ビットが各ポートの指定に対応しています。複数ポートの指定はできません。

	P7	P6	P5	P4	P3	P2	P1	P0
<i>Port</i>	0x80	0x40	0x20	0x10	0x08	0x04	0x02	0x01

Value の各ビットがポートの各ピンに対応しています。

ビットの値が **0** の場合は **LOW** レベル、**1** の場合は **HIGH** レベルになります。

例) *Port* = **0x04**, *Value* = **0xA5**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
<i>Value</i>	1	0	1	0	0	1	0	1
PIN	P27	P26	P25	P24	P23	P22	P21	P20
LEVEL	H	L	H	L	L	H	L	H

Example

```
int result;
EPX64S_HANDLE handle;            // オープンしたデバイスのハンドル
BYTE port, value;

port = 0x04; // Port2
value = 0xA5; // P27,25,22,20 "HIGH" P26,24,23,21 "LOW"
result = EPX64S_OutputPort(handle, port, value);
if (result == EPX64S_OK) {
    // EPX64S_OutputPort 成功
}
else {
    // EPX64S_OutputPort 失敗
}
```

EPX64S_InputPort

ポートから入力します。(8 ビットデータ同時入力)

int EPX64S_InputPort (EPX64S_HANDLE Handle, BYTE Port, BYTE *Value)

Parameters

Handle デバイスのハンドル
Port ポートの指定
Value ポート入力値の格納先へのポインタ

Remarks

Port の各ビットが各ポートの指定に対応しています。複数ポートの指定はできません。

	P7	P6	P5	P4	P3	P2	P1	P0
<i>Port</i>	0x80	0x40	0x20	0x10	0x08	0x04	0x02	0x01

Value の各ビットがポートの各ピンに対応しています。

ビットの値が **0** の場合は **LOW** レベル、**1** の場合は **HIGH** レベルになります。

例) *Port* = **0x20**, *Value* = **0x3C**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
<i>Value</i>	0	0	1	1	1	1	0	0
PIN	P57	P56	P55	P54	P53	P52	P51	P50
LEVEL	L	L	H	H	H	H	L	L

Example

```
int result;  
EPX64S_HANDLE handle;            // オープンしたデバイスのハンドル  
BYTE port, value;  
  
port = 0x20; // Port5  
result = EPX64S_InputPort(handle, port, &value);  
if (result == EPX64S_OK) {  
    // EPX64S_InputPort 成功  
}  
else {  
    // EPX64S_InputPort 失敗  
}
```