

EPX-30SC プログラミングガイド

Rev 1.0

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
EPX30SC_GetNumberOfDevices.....	5
EPX30SC_GetSerialNumber.....	6
EPX30SC_Open.....	7
EPX30SC_OpenBySerialNumber.....	8
EPX30SC_Close.....	9
2.2 カウンタ制御関数.....	10
EPX30SC_SetCounterMode.....	10
EPX30SC_GetCounterMode.....	11
EPX30SC_SetCounterDirection.....	12
EPX30SC_GetCounterDirection.....	13
EPX30SC_SetCounterControl.....	14
EPX30SC_GetCounterControl.....	15
EPX30SC_ResetCounter.....	16
EPX30SC_GetCounterValue.....	17
EPX30SC_GetCounterStatus.....	18
EPX30SC_GetCounterExControl.....	19
EPX30SC_GetCounterLatchFlag.....	20
EPX30SC_GetCounterLatchValue.....	21
2.3 I/O 制御関数.....	22
EPX30SC_SetPortDirection.....	22
EPX30SC_GetPortDirection.....	23
EPX30SC_OutputPort.....	24
EPX30SC_InputPort.....	25

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-30SC 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-30SC** のカウンタ機能、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。

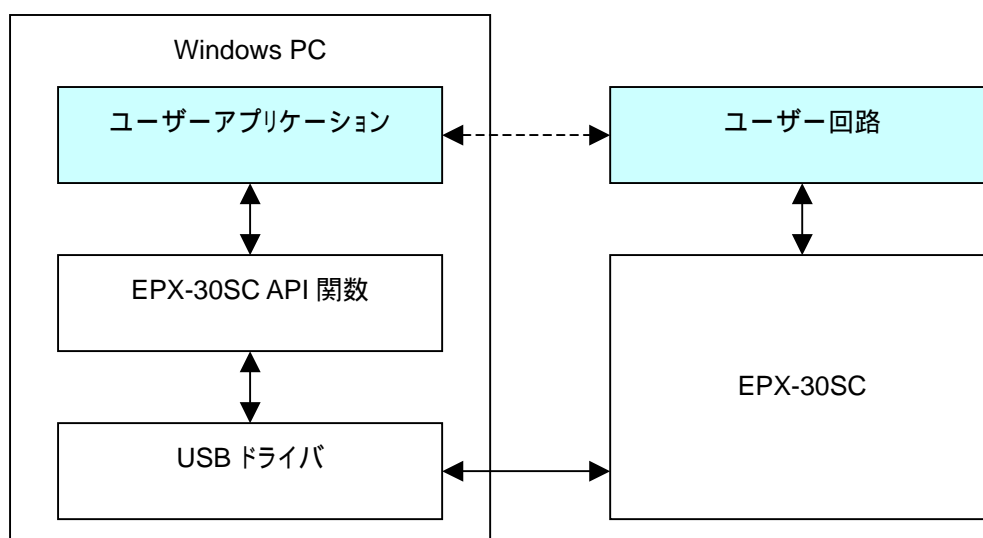


図 1.1 構成

1.2 関数について

「EPX-30SC API 関数」は関数群をモジュール化した「EPX30SC.dll」で提供されます。
「EPX30SC.dll」ファイルは **EPX-30SC** 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**、**カウンタ制御関数**、**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。
カウンタ制御関数はカウンタ機能を制御する関数です。
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

「EPX30SC.h」、「EPX30SC.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

「EPX30SC.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

「EPX30SC.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

「EPX30SC.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「¥EPX-30SC¥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
EPX30SC_OK	0	正常終了
EPX30SC_INVALID_HANDLE	1	デバイスのハンドルが無効
EPX30SC_DEVICE_NOT_FOUND	2	デバイスが見つからない
EPX30SC_DEVICE_NOT_OPENED	3	デバイスがオープンできない
EPX30SC_OTHER_ERROR	4	その他のエラーが発生した
EPX30SC_COMMUNICATION_ERROR	5	通信エラーが発生した
EPX30SC_INVALID_PARAMETER	6	パラメータが無効

表 1.4 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-30SC** 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

EPX30SC_GetNumberOfDevices

現在接続されているデバイスの数を取得します。

```
int EPX30SC_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 = EPX30SC_GetNumberOfDevices(&number);  
if (result == EPX30SC_OK) {  
    // EPX30SC_GetNumberOfDevices 成功  
}  
else {  
    // EPX30SC_GetNumberOfDevices 失敗  
}
```

EPX30SC_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 **EPX30SC_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 **EPX30SC_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記載のシリアル番号と同一です。

Example

```
int result;
int index;
int serialNumber;

index = 0;
result = EPX30SC_GetSerialNumber(index, &serialNumber);
if (result == EPX30SC_OK) {
    // EPX30SC_GetSerialNumber 成功
}
else {
    // EPX30SC_GetSerialNumber 失敗
}
```

EPX-30SC_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 EPX30SC_Open (EPX30SC_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、カウンタ及び 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;
EPX30SC_HANDLE handle;

result = EPX30SC_Open(&handle);
if (result == EPX30SC_OK) {
    // EPX30SC_Open 成功
}
else {
    // EPX30SC_Open 失敗
}
```

EPX30SC_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 EPX30SC_OpenBySerialNumber (int SerialNumber, EPX30SC_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 **EPX30SC_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記載のシリアル番号と同一です。

取得したハンドルは以後、カウンタ及び 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;
EPX30SC_HANDLE handle;
int serialNumber;    // 取得したデバイスのシリアル番号

result = EPX30SC_OpenBySerialNumber(serialNumber, &handle);
if (result == EPX30SC_OK) {
    // EPX30SC_OpenBySerialNumber 成功
}
else {
    // EPX30SC_OpenBySerialNumber 失敗
}
```


EPX30SC_Close

デバイスをクローズします。

Int **EPX30SC_Close** (EPX30SC_HANDLE *Handle*)

Parameters

Handle デバイスのハンドル

Example

```
int result;
EPX30SC_HANDLE handle;            // オープンしたデバイスのハンドル

result = EPX30SC_Close(handle);
if (result == EPX30SC_OK) {
    // EPX30SC_Close 成功
}
else {
    // EPX30SC_Close 失敗
}
```

2.2 カウンタ制御関数

EPX30SC_SetCounterMode

カウンタの動作モードを設定します。

Int EPX30SC_SetCounterMode (EPX30SC_HANDLE Handle, BYTE Mode)

Parameters

Handle デバイスのハンドル
Mode カウンタ動作モード

Remarks

Mode	説明
0	単相パルス入力モード (初期値)
1	2相パルス入力(位相差パルス入力)モード

単相パルス入力、2相パルス入力については「**EPX-30SC 取扱説明書**」をご参照ください。

Example

```
int result;  
EPX30SC_HANDLE handle;            // オープンしたデバイスのハンドル  
BYTE mode;  
  
mode = 0;            // 単相パルス入力モード  
result = EPX30SC_SetCounterMode(handle, mode);  
if (result == EPX30SC_OK) {  
    // EPX30SC_SetCounterMode 成功  
}  
else {  
    // EPX30SC_SetCounterMode 失敗  
}
```

EPX30SC_GetCounterMode

カウンタの動作モードを取得します。

Int EPX30SC_GetCounterMode (EPX30SC_HANDLE *Handle*, BYTE **Mode*)

Parameters

Handle デバイスのハンドル

Mode カウンタ動作モードの格納先へのポインタ

Example

```
int result;
EPX30SC_HANDLE handle;            // オープンしたデバイスのハンドル
BYTE mode;

result = EPX30SC_GetCounterMode(handle, &mode);
if (result == EPX30SC_OK) {
    // EPX30SC_GetCounterMode 成功
}
else {
    // EPX30SC_GetCounterMode 失敗
}
```

EPX30SC_SetCounterDirection

カウンタのカウント方向を設定します。

Int EPX30SC_SetCounterDirection (EPX30SC_HANDLE Handle, BYTE Direction)

Parameters

Handle デバイスのハンドル
Direction カウント方向

Remarks

<i>Direction</i>	説明
0	カウント方向はアップカウント (初期値)
1	カウント方向はダウンカウント

2相パルス入力時のカウント方向については「**EPX-30SC 取扱説明書**」をご参照ください。

Example

```
int result;
EPX30SC_HANDLE handle;        // オープンしたデバイスのハンドル
BYTE dir;

dir = 0;            // カウント方向はアップカウント
result = EPX30SC_SetCounterDirection(handle, dir);
if (result == EPX30SC_OK) {
    // EPX30SC_SetCounterDirection 成功
}
else {
    // EPX30SC_SetCounterDirection 失敗
}
```

EPX30SC_GetCounterDirection

カウンタのカウント方向を取得します。

Int **EPX30SC_GetCounterDirection** (EPX30SC_HANDLE *Handle*, BYTE **Direction*)

Parameters

Handle デバイスのハンドル
Direction カウント方向の格納先へのポインタ

Example

```
int result;
EPX30SC_HANDLE handle;        // オープンしたデバイスのハンドル
BYTE dir;

result = EPX30SC_GetCounterDirection(handle, &dir);
if (result == EPX30SC_OK) {
    // EPX30SC_GetCounterDirection 成功
}
else {
    // EPX30SC_GetCounterDirection 失敗
}
```

EPX30SC_SetCounterControl

カウンタの動作を設定します。

int EPX30SC_SetCounterControl (EPX30SC_HANDLE Handle, BYTE Control)

Parameters

Handle デバイスのハンドル
Control カウンタ動作

Remarks

<i>Control</i>	説明
0	カウンタのカウンタ動作は停止 (初期値)
1	カウンタはカウンタ動作

外部からのカウンタ制御入力信号 ($\overline{\text{EX_CTEN}}$) でも制御が可能です。
カウンタ制御入力信号については「**EPX-30SC 取扱説明書**」をご参照ください。

Example

```
int result;  
EPX30SC_HANDLE handle;            // オープンしたデバイスのハンドル  
BYTE control;  
  
control = 1; // カウンタ動作  
result = EPX30SC_SetCounterControl(handle, control);  
if (result == EPX30SC_OK) {  
    // EPX30SC_SetCounterControl 成功  
}  
else {  
    // EPX30SC_SetCounterControl 失敗  
}
```

EPX30SC_GetCounterControl

カウンタの動作状態を取得します。

```
int EPX30SC_GetCounterControl (EPX30SC_HANDLE Handle, BYTE *Control)
```

Parameters

Handle デバイスのハンドル
Control カウンタ動作状態の格納先へのポインタ

Example

```
int result;  
EPX30SC_HANDLE handle;            // オープンしたデバイスのハンドル  
BYTE control;  
  
result = EPX30SC_GetCounterControl(handle, &control);  
if (result == EPX30SC_OK) {  
    // EPX30SC_GetCounterControl 成功  
}  
else {  
    // EPX30SC_GetCounterControl 失敗  
}
```

EPX30SC_ResetCounter

カウンタをリセット(0クリア)します。

Int EPX30SC_ResetCounter (EPX30SC_HANDLE *Handle*)

Parameters

Handle デバイスのハンドル

Example

```
int result;
EPX30SC_HANDLE handle;            // オープンしたデバイスのハンドル

result = EPX30SC_ResetCounter(handle);
if (result == EPX30SC_OK) {
    // EPX30SC_ResetCounter 成功
}
else {
    // EPX30SC_ResetCounter 失敗
}
```


EPX30SC_GetCounterValue

カウンタの値を取得します。

int EPX30SC_GetCounterValue (EPX30SC_HANDLE *Handle*, int **Value*)

Parameters

Handle デバイスのハンドル
Value カウンタ値の格納先へのポインタ

Remarks

カウンタ値は符号付き 24 ビット (-8,388,608 ~ 8,388,607) です。

Example

```
int result;
EPX30SC_HANDLE handle;            // オープンしたデバイスのハンドル
int value;

result = EPX30SC_GetCounterValue(handle, &value);
if (result == EPX30SC_OK) {
    // EPX30SC_GetCounterValue 成功
}
else {
    // EPX30SC_GetCounterValue 失敗
}
```

EPX30SC_GetCounterStatus

カウンタの状態を取得します。

int EPX30SC_GetCounterStatus (EPX30SC_HANDLE Handle, BYTE *Status)

Parameters

Handle デバイスのハンドル
Status カウンタ動作状態の格納先へのポインタ

Remarks

<i>Status</i>	説明
0	カウンタは正常状態
1	カウンタはオーバーフロー(カウンタ停止状態)
2	カウンタはアンダーフロー(カウンタ停止状態)

オーバーフロー(カウンタ値: 0x7FFFFFFF ~ 0x800000)またはアンダーフロー(カウンタ値: 0x800000 ~ 0x7FFFFFFF)が発生すると、カウンタは自動的に停止します。

再度、カウンタを動作させるには **EPX30SC_ResetCounter** でカウンタをリセットしてください。

Example

```
int result;
EPX30SC_HANDLE handle;            // オープンしたデバイスのハンドル
BYTE status;

result = EPX30SC_GetCounterStatus(handle, &status);
if (result == EPX30SC_OK) {
    // EPX30SC_GetCounterStatus 成功
}
else {
    // EPX30SC_GetCounterStatus 失敗
}
```

EPX30SC_GetCounterExControl

外部からのカウンタ制御入力信号 ($\overline{\text{EX_CTEN}}$) の状態を取得します。

int EPX30SC_GetCounterExControl (EPX30SC_HANDLE Handle, BYTE * Control)

Parameters

Handle デバイスのハンドル
Control カウンタ動作状態の格納先へのポインタ

Remarks

<i>Control</i>	説明
0	カウンタのカウンタ動作有効
1	カウンタはカウンタ動作無効

外部からのカウンタ制御入力信号 ($\overline{\text{EX_CTEN}}$) の状態を取得します。

カウンタ制御入力信号については「**EPX-30SC 取扱説明書**」をご参照ください。

Example

```
int result;
EPX30SC_HANDLE handle;            // オープンしたデバイスのハンドル
BYTE control;

result = EPX30SC_GetCounterExControl(handle, &control);
if (result == EPX30SC_OK) {
    // EPX30SC_GetCounterExControl 成功
}
else {
    // EPX30SC_GetCounterExControl 失敗
}
```

EPX30SC_GetCounterLatchFlag

カウンタラッチ有効データの状態を取得します。

```
int EPX30SC_GetCounterLatchFlag (EPX30SC_HANDLE Handle, BYTE *Flag)
```

Parameters

Handle デバイスのハンドル

Flag カウンタラッチ状態(有効データ有無)の格納先へのポインタ

Remarks

<i>Flag</i>	説明
0	カウンタラッチ有効データ無
1	カウンタラッチ有効データ有

外部からのカウンタラッチ入力信号($\overline{\text{EX_CTEN}}$)による有効データの有無状態を取得します。
カウンタ制御入力信号については「**EPX-30SC 取扱説明書**」をご参照ください。

Example

```
int result;  
EPX30SC_HANDLE handle;            // オープンしたデバイスのハンドル  
BYTE flag;  
  
result = EPX30SC_GetCounterLatchFlag(handle, &flag);  
if (result == EPX30SC_OK) {  
    // EPX30SC_GetCounterLatchFlag 成功  
}  
else {  
    // EPX30SC_GetCounterLatchFlag 失敗  
}
```

EPX30SC_GetCounterLatchValue

カウンタラッチの値を取得します。

int EPX30SC_GetCounterLatchValue (EPX30SC_HANDLE *Handle*, int **Value*)

Parameters

Handle デバイスのハンドル
Value カウンタラッチ値の格納先へのポインタ

Remarks

カウンタラッチ値は符号付き 24 ビット (-8,388,608 ~ 8,388,607) です。

Example

```
int result;
EPX30SC_HANDLE handle;            // オープンしたデバイスのハンドル
int value;

result = EPX30SC_GetCounterLatchValue(handle, &value);
if (result == EPX30SC_OK) {
    // EPX30SC_GetCounterLatchValue 成功
}
else {
    // EPX30SC_GetCounterLatchValue 失敗
}
```

2.3 I/O 制御関数

EPX30SC_SetPortDirection

ポートの入出力方向を設定します。

Int **EPX30SC_SetPortDirection** (EPX30SC_HANDLE *Handle*, BYTE *Port*, BYTE *Direction*)

Parameters

Handle デバイスのハンドル
Port ポートの指定
Direction ポートの入出力方向

Remarks

Port に入出力設定するポート番号を指定します。

<i>Direction</i>	説明
0	入力 (初期値)
1	出力

初期値(電源投入時)は全てのポートが入力に設定されています。

Example

```
int result;
EPX30SC_HANDLE handle;        // オープンしたデバイスのハンドル
BYTE port, dir;

port = 0;        // Port0
dir = 1;        // 出力設定
result = EPX30SC_SetPortDirection(handle, port, dir);
if (result == EPX30SC_OK) {
    // EPX30SC_SetPortDirection 成功
}
else {
    // EPX30SC_SetPortDirection 失敗
}
```

EPX30SC_GetPortDirection

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 **EPX30SC_GetPortDirection** (EPX30SC_HANDLE *Handle*, BYTE *Port*, BYTE **Direction*)

Parameters

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

Example

```
int result;
EPX30SC_HANDLE handle;        // オープンしたデバイスのハンドル
BYTE port, dir;

port = 2;        // Port2
result = EPX30SC_GetPortDirection(handle, port, &dir);
if (result == EPX30SC_OK) {
    // EPX30SC_GetPortDirection 成功
}
else {
    // EPX30SC_GetPortDirection 失敗
}
```

EPX30SC_OutputPort

ポートに出力します。

Int **EPX30SC_OutputPort** (EPX30SC_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

Port に出力するポート番号を指定します。ポートが入力に設定されている場合は出力されません。

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* = 1, *Value* = **0xA5**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
<i>Value</i>	1	0	1	0	0	1	0	1
PIN	P17	P16	P15	P14	P13	P12	P11	P10
LEVEL	H	L	H	L	L	H	L	H

Example

```
int result;
EPX30SC_HANDLE handle;            // オープンしたデバイスのハンドル
BYTE port, value;

port = 1;            // Port1
value = 0xA5; // P17,15,12,10 "HIGH"    P16,14,13,11 "LOW"
result = EPX30SC_OutputPort(handle, port, value);
if (result == EPX30SC_OK) {
    // EPX30SC_OutputPort 成功
}
else {
    // EPX30SC_OutputPort 失敗
}
```


EPX30SC_InputPort

ポートから入力します。

int EPX30SC_InputPort (EPX30SC_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 に入力するポート番号を指定します。

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* = 0, *Value* = 0x3C

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
<i>Value</i>	0	0	1	1	1	1	0	0
PIN	P07	P06	P05	P04	P03	P02	P01	P00
LEVEL	L	L	H	H	H	H	L	L

Example

```
int result;
EPX30SC_HANDLE handle;            // オープンしたデバイスのハンドル
BYTE port, value;

port = 0;            // Port0
result = EPX30SC_InputPort(handle, port, &value);
if (result == EPX30SC_OK) {
    // EPX30SC_InputPort 成功
}
else {
    // EPX30SC_InputPort 失敗
}
}
```