

EPX-64R 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
EPX64R_GetNumberOfDevices.....	5
EPX64R_GetSerialNumber.....	6
EPX64R_Open.....	7
EPX64R_OpenBySerialNumber.....	8
EPX64R_Close	9
2.2 I/O 制御関数.....	10
EPX64R_SetDirection.....	10
EPX64R_GetDirection.....	11
EPX64R_OutputPort.....	12
EPX64R_InputPort.....	13
EPX64R_SetPortOutputBuffer.....	14
EPX64R_GetPortInputBuffer.....	15
EPX64R_SetOutputLatch.....	16
EPX64R_SetInputLatch.....	17

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「EPX64R 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-64R** 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。

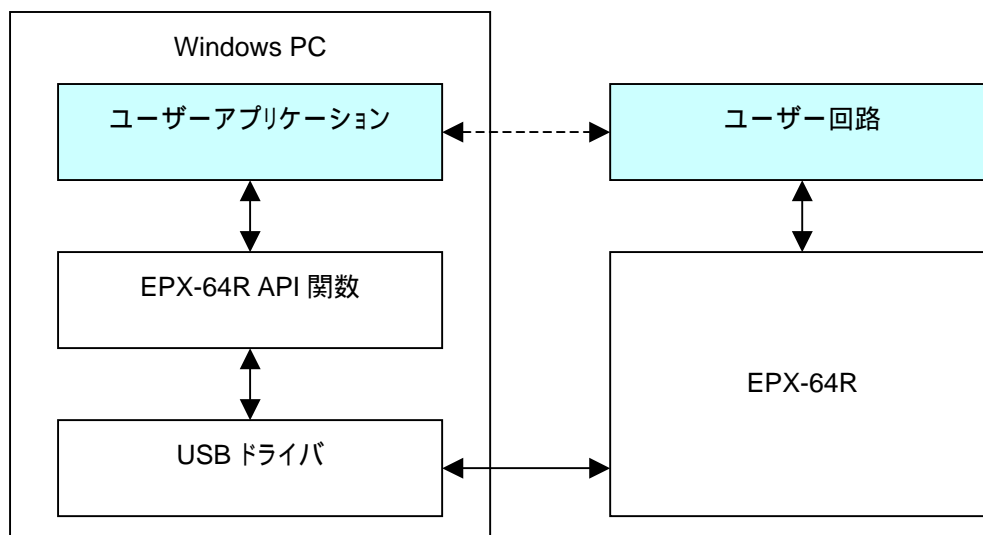


図 1.1 構成

1.2 関数について

「EPX64R API 関数」は関数群をモジュール化した「EPX64R.dll」で提供されます。

「EPX64R.dll」ファイルは **EPX-64R** 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

「EPX64R.h」、「EPX64R.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

「EPX64R.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

「EPX64R.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

「EPX64R.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
EPX64R_OK	0	正常終了
EPX64R_INVALID_HANDLE	1	デバイスのハンドルが無効
EPX64R_DEVICE_NOT_FOUND	2	デバイスが見つからない
EPX64R_DEVICE_NOT_OPENED	3	デバイスがオープンできない
EPX64R_OTHER_ERROR	4	その他のエラーが発生した
EPX64R_COMMUNICATION_ERROR	5	通信エラーが発生した
EPX64R_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-64R** 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

EPX64R_GetNumberOfDevices

現在接続されているデバイスの数を取得します。

```
int EPX64R_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 = EPX64R_GetNumberOfDevices(&number);  
if (result == EPX64R_OK) {  
    // EPX64R_GetNumberOfDevices 成功  
}  
else {  
    // EPX64R_GetNumberOfDevices 失敗  
}
```

EPX64R_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 EPX64R_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 **EPX64R_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 = EPX64R_GetSerialNumber(index, &serialNumber);  
if (result == EPX64R_OK) {  
    // EPX64R_GetSerialNumber 成功  
}  
else {  
    // EPX64R_GetSerialNumber 失敗  
}
```

EPX64R_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 EPX64R_Open (EPX64R_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;  
EPX64R_HANDLE handle;  
  
result = EPX64R_Open(&handle);  
if (result == EPX64R_OK) {  
    // EPX64R_Open 成功  
}  
else {  
    // EPX64R_Open 失敗  
}
```

EPX64R_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 EPX64R_OpenBySerialNumber (int SerialNumber, EPX64R_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 **EPX64R_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;
EPX64R_HANDLE handle;
int serialNumber;    // 取得したデバイスのシリアル番号

result = EPX64R_OpenBySerialNumber(serialNumber, &handle);
if (result == EPX64R_OK) {
    // EPX64R_OpenBySerialNumber 成功
}
else {
    // EPX64R_OpenBySerialNumber 失敗
}
```


EPX64R_Close

デバイスをクローズします。

```
int EPX64R_Close (EPX64R_HANDLE Handle)
```

Parameters

Handle デバイスのハンドル

Example

```
int result;
EPX64R_HANDLE handle;            // オープンしたデバイスのハンドル

result = EPX64R_Close(handle);
if (result == EPX64R_OK) {
    // EPX64R_Close 成功
}
else {
    // EPX64R_Close 失敗
}
```

2.2 I/O 制御関数

EPX64R_SetDirection

ポートの入出力方向を設定します。

int EPX64R_SetDirection (EPX64R_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;
EPX64R_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 = EPX64R_SetDirection(handle, dir);
if (result == EPX64R_OK) {
    // EPX64R_SetDirection 成功
}
else {
    // EPX64R_SetDirection 失敗
}
```

EPX64R_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 EPX64R_GetDirection (EPX64R_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;  
EPX64R_HANDLE handle;            // オープンしたデバイスのハンドル  
BYTE dir;  
  
result = EPX64R_GetDirection(handle, &dir);  
if (result == EPX64R_OK) {  
    // EPX64R_GetDirection 成功  
}  
else {  
    // EPX64R_GetDirection 失敗  
}
```

EPX64R_OutputPort

ポートに出力します。

int EPX64R_OutputPort (EPX64R_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;
EPX64R_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 = EPX64R_OutputPort(handle, port, value);
if (result == EPX64R_OK) {
    // EPX64R_OutputPort 成功
}
else {
    // EPX64R_OutputPort 失敗
}
```

EPX64R_InputPort

ポートから入力します。

int EPX64R_InputPort (EPX64R_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;
EPX64R_HANDLE handle;            // オープンしたデバイスのハンドル
BYTE port, value;

port = 0x20; // Port5
result = EPX64R_InputPort(handle, port, &value);
if (result == EPX64R_OK) {
    // EPX64R_InputPort 成功
}
else {
    // EPX64R_InputPort 失敗
}
```

EPX64R_SetPortOutputBuffer

ポート出力バッファに値を設定します。

int EPX64R_SetPortOutputBuffer (EPX64R_HANDLE *Handle*, BYTE *Port*, BYTE *Value*)

Parameters

<i>Handle</i>	デバイスのハンドル
<i>Port</i>	ポート出力バッファの指定
<i>Value</i>	ポート出力バッファ設定値

Remarks

ポート出力バッファに値を設定してもポートの状態には反映されません。

設定後、EPX64R_SetOutputLatch でポート出力バッファの値をポートに出力します。

ポート出力バッファ、出力ラッチについては「EPX-64R 取扱説明書」をご参照ください。

Example

```
int result;
EPX64R_HANDLE handle;          // オープンしたデバイスのハンドル
BYTE port, value;

port = 0x04; // PORT2 出力バッファ
value = 0xA5; // P27,25,22,20 "HIGH" P26,24,23,21 "LOW"
result = EPX64R_SetPortOutputBuffer(handle, port, value);
if (result == EPX64R_OK) {
    // EPX64R_SetPortOutputBuffer 成功
}
else {
    // EPX64R_SetPortOutputBuffer 失敗
}
```

EPX64R_GetPortInputBuffer

ポート入力バッファの値を取得します。

```
int EPX64R_GetPortInputBuffer (EPX64R_HANDLE Handle, BYTE Port, BYTE *Value)
```

Parameters

<i>Handle</i>	デバイスのハンドル
<i>Port</i>	ポート入力バッファの指定
<i>Value</i>	ポート入力バッファ値の格納先へのポインタ

Remarks

取得前に、**EPX64R_SetInputLatch** でポートの状態をポート入力バッファに転送していないと、ポート入力バッファにはポートの状態が反映されません。

ポート入力バッファ、入力ラッチについては「**EPX-64R 取扱説明書**」をご参照ください。

Example

```
int result;
EPX64R_HANDLE handle;          // オープンしたデバイスのハンドル
BYTE port, value;

buffer = 0x20;                 // PORT5 入力バッファ
result = EPX64R_GetPortInputBuffer(handle, port, &value);
if (result == EPX64R_OK) {
    // EPX64R_GetPortInputBuffer 成功
}
else {
    // EPX64R_GetPortInputBuffer 失敗
}
```

EPX64R_SetOutputLatch

出力ラッチします。

int EPX64R_SetOutputLatch (EPX64R_HANDLE *Handle*)

Parameters

Handle デバイスのハンドル

Remarks

ポートに対応したポート出力バッファの値を、出力に設定されている全てのポートに出力します。
外部からの制御入力信号 (EX_OTLT) での出力ラッチも可能です。
制御入力信号については「**EPX-64R 取扱説明書**」をご参照ください。

Example

```
int result;
EPX64R_HANDLE handle;            // オープンしたデバイスのハンドル

result = EPX64R_SetOutputLatch(handle);
if (result == EPX64R_OK) {
    // EPX64R_SetOutputLatch 成功
}
else {
    // EPX64R_SetOutputLatch 失敗
}
```


EPX64R_SetInputLatch

入力ラッチします。

int EPX64R_SetInputLatch (EPX64R_HANDLE *Handle*)

Parameters

Handle デバイスのハンドル

Remarks

全てのポートの状態を、対応したポート入力バッファに転送(入力)します。
外部からの制御入力信号(EX_INLT)での入力ラッチも可能です。
制御入力信号については「**EPX-64R 取扱説明書**」をご参照ください。

Example

```
int result;
EPX64R_HANDLE handle;            // オープンしたデバイスのハンドル

result = EPX64R_SetInputLatch(handle);
if (result == EPX64R_OK) {
    // EPX64R_SetInputLatch 成功
}
else {
    // EPX64R_SetInputLatch 失敗
}
```