

PVG-16B 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.....	3
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
	PVG16B_GetNumberOfDevices.....	5
	PVG16B_GetSerialNumber.....	6
	PVG16B_Open.....	7
	PVG16B_OpenBySerialNumber.....	8
	PVG16B_Close	9
2.2	制御関数.....	10
	PVG16B_GetStatus.....	10
	PVG16B_GetOutputMode.....	12
	PVG16B_SetOutputMode.....	13
	PVG16B_DirectOutput.....	14
	PVG16B_GetStepTime.....	15
	PVG16B_SetStepTime.....	16
	PVG16B_GetStepCount.....	17
	PVG16B_WriteStepMemory.....	18
	PVG16B_ReadStepCounter.....	19
	PVG16B_Start.....	20
	PVG16B_Cancel.....	21

1 はじめに

1.1 概要

パソコンの USB ポートに接続して、**PVG-16B** 専用の 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 **PVG-16B** の D/A 機能(任意の電圧出力、ステップシーケンス)を制御することができます。

1.2 関数について

「PVG-16B API 関数」は関数群をモジュール化した「PVG16B.dll」で提供されます。
「PVG16B.dll」ファイルは **PVG-16B** 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**、**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。

制御関数は電圧出力制御、ステップデータ転送、パラメータ設定等の関数です。

1.3 プログラミングの準備

「PVG16B.dll」ファイルをシステムフォルダまたは実行フォルダにコピーしてください。

- Visual C++の場合

「PVG16B.h」、「PVG16B.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

「PVG16B.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

「PVG16B.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

「PVG16B.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を下表に示します。

定数	値	意味
PVG16B_OK	0	正常終了
PVG16B_INVALID_ID	1	デバイスの ID が無効
PVG16B_DEVICE_NOT_FOUND	2	デバイスが見つからない
PVG16B_DEVICE_NOT_OPENED	3	デバイスがオープンできない
PVG16B_OTHER_ERROR	4	その他のエラーが発生した
PVG16B_COMMUNICATION_ERROR	5	通信エラーが発生した
PVG16B_INVALID_PARAMETER	6	パラメータが無効
PVG16B_EXECUTION_ERROR	7	実行エラーが発生した
PVG16B_MEMORY_ERROR	8	メモリが不足している

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(製品付属の GUI ソフトウェア含)、またはマルチスレッドによる **PVG-16B** 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

PVG16B_GetNumberOfDevices

現在接続されているデバイスの数を取得します。

```
int PVG16B_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 = PVG16B_GetNumberOfDevices(&number);  
if (result == PVG16B_OK) {  
    // PVG16B_GetNumberOfDevices 成功  
}  
else {  
    // PVG16B_GetNumberOfDevices 失敗  
}
```

PVG16B_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 **PVG16B_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 **PVG16B_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本体記載のシリアル番号と同一です。

Example

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

index = 0;
result = PVG16B_GetSerialNumber(index, &serialNumber);
if (result == PVG16B_OK) {
    // PVG16B_GetSerialNumber 成功
}
else {
    // PVG16B_GetSerialNumber 失敗
}
```

PVG16B_Open

デバイスをオープンし、デバイスの ID を取得します。

int PVG16B_Open (BYTE *ID)

Parameters

ID デバイスの ID 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 ID を取得します。
取得した ID は以後、制御関数等に引数として渡すことになります。

Example

```
int result;
BYTE id;

result = PVG16B_Open(&id);
if (result == PVG16B_OK) {
    // PVG16B_Open 成功
}
else {
    // PVG16B_Open 失敗
}
```

PVG16B_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 ID を取得します。

int PVG16B_OpenBySerialNumber (int *SerialNumber*, BYTE **ID*)

Parameters

SerialNumber デバイスのシリアル番号

ID デバイスの ID の格納先へのポインタ

Remarks

シリアル番号は **PVG16B_GetSerialNumber** で取得することができます。

尚、このシリアル番号は本体記載のシリアル番号と同一です。

取得した ID は以後、制御関数等に引数として渡すことになります。

Example

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

result = PVG16B_OpenBySerialNumber(serialNumber, &id);
if (result == PVG16B_OK) {
    // PVG16B_OpenBySerialNumber 成功
}
else {
    // PVG16B_OpenBySerialNumber 失敗
}
```

PVG16B_Close

デバイスをクローズします。

int PVG16B_Close (BYTE *ID*)

Parameters

ID デバイスの ID

Example

```
int result;
BYTE id;            // オープンしたデバイスの ID

result = PVG16B_Close(id);
if (result == PVG16B_OK) {
    // PVG16B_Close 成功
}
else {
    // PVG16B_Close 失敗
}
```

2.2 制御関数

PVG16B_GetStatus

PVG-16B のステータスを取得します。

```
int PVG16B_GetStatus (BYTE ID, BYTE *Status, BYTE *ExtState)
```

Parameters

ID デバイスの ID
Status 動作ステータスの格納先へのポインタ
ExtState 外部入力信号の格納先へのポインタ

Remarks

本関数が成功すると、*Status* には動作ステータスが、*ExtState* には外部入力信号の状態が、下記の値で格納されます。

また、動作ステータスは、出力モードが「プログラムモード」設定時に使用します。

出力モードについては「**PVG-16B 取扱説明書**」をご参照ください。

Status

定数	値	説明
PVG16B_STATUS_NO_DATA	0	出力停止中(ステップデータ未設定)
PVG16B_STATUS_READY	1	出力停止中(ステップデータ設定済)
PVG16B_STATUS_WAITING	2	出力待機中
PVG16B_STATUS_RUNNING	3	出力中

ExtState

値	説明
0	外部入力信号が L
1	外部入力信号が H

Example

```
int result;
BYTE id;      // オープンしたデバイスの ID
BYTE status, extState;

result = PVG16B_GetStatus(id, &status, &extState);
if (result == PVG16B_OK) {
    // PVG16B_GetStatus 成功
}
else {
    // PVG16B_GetStatus 失敗
}
```

PVG16B_GetOutputMode

出力モードを取得します。

int PVG16B_GetOutputMode(BYTE ID, BYTE*Mode)

Parameters

ID デバイスの ID
Mode 出力モードの格納先へのポインタ

Remarks

本関数が成功すると *Mode* には下記の値が格納されます。
出力モードについては「**PVG-16B 取扱説明書**」をご参照ください。

値	説明
0	ダイレクトモード
1	プログラムモード【PC スタート】
2	プログラムモード【トリガスタート(立ち上がり)】
3	プログラムモード【トリガスタート(立ち下がり)】
4	プログラムモード【トリガステップ(立ち上がり)】
5	プログラムモード【トリガステップ(立ち下がり)】

Example

```
int result;  
BYTE id;            // オープンしたデバイスの ID  
BYTE mode;  
  
result = PVG16B_GetOutputMode(id, &mode);  
if (result == PVG16B_OK) {  
    // PVG16B_GetOutputMode 成功  
}  
else {  
    // PVG16B_GetOutputMode 失敗  
}
```

PVG16B_SetOutputMode

出力モードを設定します。

int PVG16B_SetOutputMode (BYTE ID, BYTE Mode)

Parameters

ID デバイスの ID
Mode 出力モード

Remarks

出力モードを設定します。

出力モードについては「**PVG-16B 取扱説明書**」をご参照ください。

値	説明
0	ダイレクトモード
1	プログラムモード【PC スタート】
2	プログラムモード【トリガスタート(立ち上がり)】
3	プログラムモード【トリガスタート(立ち下がり)】
4	プログラムモード【トリガステップ(立ち上がり)】
5	プログラムモード【トリガステップ(立ち下がり)】

Example

```
int result;  
BYTE id;            // オープンしたデバイスの ID  
BYTE mode;  
  
mode = 1;           // プログラムモード(PC スタート)  
result = PVG16B_SetOutputMode(id, mode);  
if (result == PVG16B_OK) {  
    // PVG16B_SetOutputMode 成功  
}  
else {  
    // PVG16B_SetOutputMode 失敗  
}
```

PVG16B_DirectOutput

指定した電圧を出力します。

int PVG16B_DirectOutput (BYTE *ID*, double **Voltage*)

Parameters

ID デバイスの ID
Voltage デバイス出力する電圧値

Remarks

指定した電圧を出力します。単位は[V]です。
出力モードが「ダイレクトモード」に設定されている時のみ有効です。
出力モードについては「**PVG-16B 取扱説明書**」をご参照ください。

Example

```
int result;
BYTE id;            // オープンしたデバイスの ID
double voltage;

voltage = 7.5;            // +7.5V

result = PVG16B_DirectOutput(id, voltage);
if (result == PVG16B_OK) {
    // PVG16B_DirectOutput 成功
}
else {
    // PVG16B_DirectOutput 失敗
}
```

PVG16B_GetStepTime

ステップ時間を取得します。

int PVG16B_GetStepTime (BYTE *ID*, int **Time*)

Parameters

ID デバイスの ID
Time ステップ時間の格納先へのポインタ

Remarks

本関数が成功すると *Time* にステップ時間が格納されます。単位は[ms]です。

Example

```
int result;
BYTE id;            // オープンしたデバイスの ID
int time;

result = PVG16B_GetStepTime(id, &time);
if (result == PVG16B_OK) {
    // PVG16B_GetStepTime 成功
}
else {
    // PVG16B_GetStepTime 失敗
}
```

PVG16B_SetStepTime

ステップ時間を設定します。

int PVG16B_SetStepTime (BYTE *ID*, int *Time*)

Parameters

ID デバイスの ID
Time ステップ時間

Remarks

1ステップの時間を設定します。単位は[ms]です。範囲は1～65536です。
ステップ時間設定は出力中にはできません。出力停止時に設定してください。
出力モードが「プログラムモード(PCスタート、トリガスタート)」設定時に使用します。
出力モードについては「PVG-16B 取扱説明書」をご参照ください。

Example

```
int result;  
BYTE id;            // オープンしたデバイスの ID  
int time;  
  
time= 100;        // 100ms  
result = PVG16B_SetStepTime(id, time);  
if (result == PVG16B_OK) {  
    // PVG16B_SetStepTime 成功  
}  
else {  
    // PVG16B_SetStepTime 失敗  
}
```

PVG16B_GetStepCount

ステップデータのステップ数を取得します。

int PVG16B_GetStepCount (BYTE *ID*, int **Count*)

Parameters

ID デバイスの ID
Count ステップ数の格納先へのポインタ

Remarks

本関数が成功すると *Count* にステップデータのステップ数が格納されます。

Example

```
int result;  
BYTE id;            // オープンしたデバイスの ID  
int count;  
  
result = PVG16B_GetStepCount(id, &count);  
if (result == PVG16B_OK) {  
    // PVG16B_GetStepCount 成功  
}  
else {  
    // PVG16B_GetStepCount 失敗  
}
```

PVG16B_WriteStepMemory

ステップデータを転送します。

int **PVG16B_WriteStepMemory** (BYTE *ID*, double **Data*, int *Count*)

Parameters

ID デバイスの ID
Data 電圧データ配列へのポインタ
Count ステップ数

Remarks

ステップデータを **PVG-16B** 本体のメモリに転送します。

Data にステップデータが格納された配列のポインタを指定します。ステップデータの単位は[V]です。

Count にステップデータのステップ数を設定します。範囲は 1～16384 です。

転送はプログラム出力中にはできません。出力停止時に転送してください。

Example

```
int result;
BYTE id;            // オープンしたデバイスの ID
double stepData[1000];    // ステップデータが格納されたバッファ
int stepCount;

stepCount = 1000;

result = PVG16B_WiteStepMemory(id, stepData, stepCount);
if (result == PVG16B_OK) {
    // PVG16B_WiteStepMemory 成功
}
else {
    // PVG16B_WiteStepMemory 失敗
}
```

PVG16B_ReadStepCounter

ステップカウンタを取得します。

```
int PVG16B_ReadStepCounter(BYTE ID, int *Counter)
```

Parameters

ID デバイスの ID
Counter ステップカウンタの格納先へのポインタ

Remarks

本関数が成功すると *Counter* に現在のステップカウンタが格納されます。

Example

```
int result;  
BYTE id;                // オープンしたデバイスの ID  
int counter;  
  
result = PVG16B_ReadStepCounter(id, &counter);  
if (result == PVG16B_OK) {  
    // PVG16B_ReadStepCounter 成功  
}  
else {  
    // PVG16B_ReadStepCounter 失敗  
}
```

PVG16B_Start

プログラム出力を開始します。

int PVG16B_Start (BYTE ID)

Parameters

ID デバイスの ID

Remarks

ステップデータが本体に設定されていない(データ転送していない)場合、プログラム出力は開始できません。

また、本体の出力モードが「ダイレクトモード」に設定されている場合、本関数ではプログラム出力は開始できません。本体の出力モードが「プログラムモード」の時のみ有効です。

出力モードについては「**PVG-16B 取扱説明書**」をご参照ください。

Example

```
int result;
BYTE id;            // オープンしたデバイスの ID

result = PVG16B_Start(id);
if (result == PVG16B_OK) {
    // PVG16B_Start 成功
}
else {
    // PVG16B_Start 失敗
}
```

PVG16B_Cancel

電圧出力をキャンセルします。

int PVG16B_Cancel (BYTE ID)

Parameters

ID デバイスの ID

Remarks

本体の出力モードが「ダイレクトモード」に設定されている場合、出力電圧は 0V になります。

本体の出力モードが「プログラムモード」に設定されている場合、プログラム出力シーケンスを停止し、出力電圧は 0V になります。ステップカウンタの値は保持されます。

出力モードについては「**PVG-16B 取扱説明書**」をご参照ください。

Example

```
int result;
BYTE id;            // オープンしたデバイスの ID

result = PVG16B_Cancel(id);
if (result == PVG16B_OK) {
    // PVG16B_Cancel 成功
}
else {
    // PVG16B_Cancel 失敗
}
```