AWG-10K プログラミングガイド

Rev 1.1

株式会社エルモス

目次

1　はじめに

1.1 概要

製品付属の波形編集ソフトウェアで作成した波形ファイルをユーザーアプリケーションから「AWG-10K API 関数」を介して本体に転送することができます。
本体の動作モードが「リモート（PC）」時は、波形出力制御（スタート／ストップ）ができます。
また、汎用 I/O ポートを制御することもできます。
動作モード、出力パラメータ設定の詳細は「**AWG-10K 取扱説明書**」をご参照ください。

1.2 動作環境

対応 OS	Windows 10／8（8.1）／7／Vista／XP
対応開発言語	Visual C++／Visual Basic.NET／Visual C#.NET
必要メモリ空き容量	10M バイト以上

1.3 関数について

「AWG-10K API 関数」は関数群をモジュール化した「AWG10K.dll」で提供されます。
「AWG10K.dll」ファイルは **AWG-10K** 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。
制御関数は波形ファイル転送、波形出力、汎用 I/O 操作等の関数です。

1.4 プログラミングの準備

「AWG10K.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
「AWG10K.h」、「AWG10K.lib」ファイルをプロジェクトに追加してください。

・Visual Basic.NET の場合
「AWG10K.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
「AWG10K.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。

1.5 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
AWG10K_OK	0	正常終了
AWG10K_INVALID_ID	1	デバイスの ID が無効
AWG10K_DEVICE_NOT_FOUND	2	デバイスが見つからない
AWG10K_DEVICE_NOT_OPENED	3	デバイスがオープンできない
AWG10K_MEMORY_ERROR	4	メモリが不足している
AWG10K_INVALID_PARAMETER	5	パラメータが無効
AWG10K_COMMUNICATION_ERROR	6	通信エラーが発生した
AWG10K_EXECUTION_ERROR	7	実行エラーが発生した
AWG10K_OTHER_ERROR	8	その他のエラーが発生した

表 1.5 関数の戻り値

1.6 注意事項

複数のアプリケーション（製品付属の波形編集ソフトウェア含）、またはマルチスレッドによる **AWG-10K** 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

AWG10K_GetNumberOfDevices

現在接続されているデバイスの数を取得します。

int **AWG10K_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 = AWG10K_GetNumberOfDevices(&number);
if (result == AWG10K_OK) {
	// AWG10K_GetNumberOfDevices 成功
}
else {
	// AWG10K_GetNumberOfDevices 失敗
}
```

AWG10K_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 **AWG10K_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 **AWG10K_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 = AWG10K_GetSerialNumber(index, &serialNumber);
if (result == AWG10K_OK) {
	// AWG10K_GetSerialNumber 成功
}
else {
	// AWG10K_GetSerialNumber 失敗
}
```

AWG10K_Open

デバイスをオープンし、デバイスの ID を取得します。

int **AWG10K_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 = AWG10K_Open(&id);
if (result == AWG10K_OK) {
	// AWG10K_Open 成功
}
else {
	// AWG10K_Open 失敗
}
```

AWG10K_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 **AWG10K_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 **AWG10K_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 = AWG10K_OpenBySerialNumber(serialNumber, &id);
if (result == AWG10K_OK) {
       // AWG10K_OpenBySerialNumber 成功
}
else {
       // AWG10K_OpenBySerialNumber 失敗
}
```

AWG10K_Close

デバイスをクローズします。

int **AWG10K_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 = AWG10K_Close(id);
if (result == AWG10K_OK) {
       // AWG10K_Close 成功
}
else {
       // AWG10K_Close 失敗
}
```

2.2 制御関数

AWG10K_SetFile

波形ファイルを転送します。

int **AWG10K_SetFile** (BYTE *ID*, const char **FilePath*)

Parameters

ID	デバイスの ID
FilePath	ファイルパスを保持している文字列へのポインタ

Remarks

ファイルパスは終端が NULL で終わる絶対パス（フルパス）をマルチバイト文字列で指定してください。
尚、波形編集ソフトウェアで作成したファイル（拡張子*.kwg）以外は指定できません。
波形ファイルの転送は波形出力中にはできません。波形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
char filePath[] = “C:¥¥temp¥¥test.kwg”;

result = AWG10K_SetFile(id, filePath);
if (result == AWG10K_OK) {
       // AWG10K_SetFile 成功
}
else {
       // AWG10K_SetFile 失敗
}
```

AWG10K_Start

波形出力をスタートします。

int **AWG10K_Start** (BYTE *ID*)

Parameters

ID　　デバイスの ID

Remarks

波形データが本体に設定されていない（ファイル転送していない）場合、波形出力は開始できません。
本体の動作モードが「外部トリガー」に設定されている場合、本関数では波形出力は開始できません。
本体の動作モードが「リモート(PC)」の時のみ有効です。
「外部トリガー」時の波形出力制御については「**AWG-10K 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 = AWG10K_Start(id);
if (result == AWG10K_OK) {
       // AWG10K_Start 成功
}
else {
       // AWG10K_Start 失敗
}
```

AWG10K_Stop

波形出力をｽﾄｯﾌﾟします。

int **AWG10K_Stop** (BYTE *ID*)

Parameters

ID　　デバイスの ID

Remarks

本体の動作モードが「外部ﾄﾘｶﾞｰ」に設定されている場合、本関数では波形出力は停止できません。
本体の動作モードが「ﾘﾓｰﾄ(PC)」の時のみ有効です。
「外部ﾄﾘｶﾞｰ」時の波形出力制御については「**AWG-10K 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 = AWG10K_Stop(id);
if (result == AWG10K_OK) {
       // AWG10K_Stop 成功
}
else {
       // AWG10K_Stop 失敗
}
```

AWG10K_GetStatus

動作ステータスを取得します。

int **AWG10K_GetStatus** (BYTE *ID*, BYTE **Status*)

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

Remarks

本関数が成功すると *Status* には動作ステータスが下記の値で格納されます。

定数	値	意味
AWG10K_STATUS_NO_DATA	0	波形出力停止中（波形ファイル未設定）
AWG10K_STATUS_READY	1	波形出力停止中（波形ファイル設定済）
AWG10K_STATUS_RUNNING	2	波形出力中

Example

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

result = AWG10K_GetStatus(id, &status);
if (result == AWG10K_OK) {
      // AWG10K_GetStatus 成功
}
else {
      // AWG10K_GetStatus 失敗
}
```

AWG10K_OutputPort

汎用出力端子に出力します。

int **AWG10K_OutputPort** (BYTE *ID*, BYTE *Port*)

Parameters

ID　　デバイスの ID

Port　　汎用出力端子に出力する値

Remarks

本関数が成功すると汎用出力端子に出力されます。

値	状態
0	汎用出力端子が Low レベル
1	汎用出力端子が High レベル

Example

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

port = 1;     // High レベル出力
result = AWG10K_OutputPort(id, port);
if (result == AWG10K_OK) {
       // AWG10K_OutputPort 成功
}
else {
       // AWG10K_OutputPort 失敗
}
```

AWG10K_InputPort

汎用入力端子の状態を取得します。

int **AWG10K_InputPort** (BYTE *ID*, BYTE **Port*)

Parameters

ID	デバイスの ID
Port	汎用入力端子の状態の格納先へのポインタ

Remarks

本関数が成功すると *Port* には汎用入力端子の状態が下記の値が格納されます。

値	状態
0	汎用入力端子が Low レベル
1	汎用入力端子が High レベル

Example

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

result = AWG10K_InputPort(id, &port);
if (result == AWG10K_OK) {
       // AWG10K_InputPort 成功
}
else {
       // AWG10K_InputPort 失敗
}
```