USB A/D ボード

FAD-24

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

1.1 概要

パソコンの USB ポートに接続して、**FAD-24** 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 **FAD-24** の A/D 機能を制御することができます。
下図は全体の構成です。

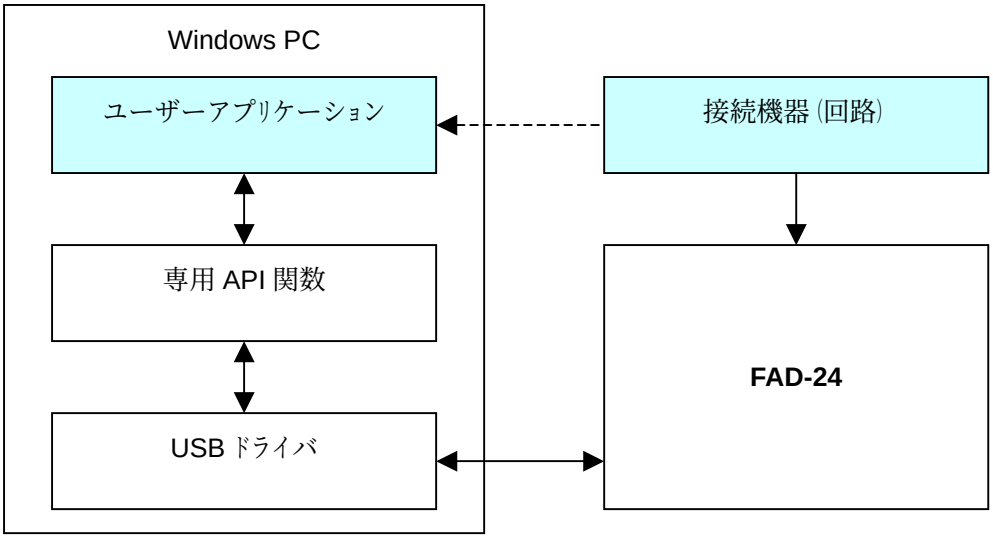

図 1.1 構成

1.2 関数について

FAD-24 専用 API 関数は関数群をモジュール化した「FAD24.dll」で提供されます。
「FAD24.dll」ファイルは **FAD-24** をインストールするときにシステムフォルダに入ります。

関数は**デバイス関数**、**A/D 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。
A/D 制御関数はアナログ電圧値を取得する関数です。

1.3 プログラミングの準備

・Visual C++の場合

「FAD24.h」、「FAD24.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

「FAD24.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

「FAD24.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

「FAD24.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
FAD24_OK	0	正常終了
FAD24_INVALID_ID	1	デバイスの ID が無効
FAD24_DEVICE_NOT_FOUND	2	デバイスが見つからない
FAD24_DEVICE_NOT_OPENED	3	デバイスがオープンできない
FAD24_OTHER_ERROR	4	その他のエラーが発生した
FAD24_COMMUNICATION_ERROR	5	通信エラーが発生した
FAD24_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同じ基板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

FAD24_GetNumberOfDevices

現在接続されているデバイスの数を取得します。

int **FAD24_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 = FAD24_GetNumberOfDevices(&number);
if (result == FAD24_OK) {
	// FAD24_GetNumberOfDevices 成功
}
else {
	// FAD24_GetNumberOfDevices 失敗
}
```

FAD24_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 **FAD24_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 **FAD24_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 = FAD24_GetSerialNumber(index, &serialNumber);
if (result == FAD24_OK) {
	// FAD24_GetSerialNumber 成功
}
else {
	// FAD24_GetSerialNumber 失敗
}
```

FAD24_Open

デバイスをオープンし、デバイスの ID を取得します。

int **FAD24_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、A/D 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 = FAD24_Open(&id);
if (result == FAD24_OK) {
	// FAD24_Open 成功
}
else {
	// FAD24_Open 失敗
}
```

FAD24_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 **FAD24_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 **FAD24_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、A/D 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 = FAD24_OpenBySerialNumber(serialNumber, &id);
if (result == FAD24_OK) {
       // FAD24_OpenBySerialNumber 成功
}
else {
       // FAD24_OpenBySerialNumber 失敗
}
```

FAD24_Close

デバイスをクローズします。

int **FAD24_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 = FAD24_Close(id);
if (result == FAD24_OK) {
       // FAD24_Close 成功
}
else {
       // FAD24_Close 失敗
}
```

2.2 A/D 制御関数

FAD24_Acquisition

指定したサンプリングでアナログ電圧値を取得します。

int **FAD24_Acquisition** (BYTE *ID*, double **Value*, int *SamplingRate*, int *SamplingCount*,
BYTE *TriggerType*, double *TriggerThreshold*, int *TriggerWaitCount,* int **AcquisitionCount*)

Parameters

ID	デバイスの ID
Value	アナログ電圧値の格納先配列へのポインタ
SamplingRate	取得するアナログ電圧値のサンプリングレート
SamplingCount	取得するアナログ電圧値のサンプリング数
TriggerType	トリガータイプ
TriggerThreshold	しきい値
TriggerWaitCount	トリガー待ちカウント数
AcquisitionCount	取得したアナログ電圧値のカウント数のポインタ

Remarks

本関数が成功すると *Value* の配列にアナログ電圧値が格納されます。単位は[V]です。
SamplingRate、*SamplingCount* にサンプリング設定をします。
SamplingRate×20 がサンプリング周期（usec）となります。
SamplingCount の範囲は 1～1000000 です。
TriggerType、*TriggerThreshold*、*TriggerWaitCount* にトリガー条件を設定します。
TriggerType にトリガーなしを設定すると *TriggerThreshold*、*TriggerWaitCount* は無効になります。

TriggerType	説明
0	トリガーなし
1	設定したしきい値より電圧が高い（レベル）
2	設定したしきい値より電圧が低い（レベル）
3	設定したしきい値を低い方から横切る（立上り）
4	設定したしきい値を高い方から横切る（立下り）
5	外部同期信号が”Off”から”On”への変化点（On エッジ）
6	外部同期信号が”On”から”Off”への変化点（Off エッジ）

TriggerThreshold はトリガーするしきい値を電圧値[V]で設定します。
TriggerWaitCount×20 がトリガー条件が満たされるまでの待ち時間（usec）となります。
トリガー条件が満たされない時のタイムアウトに使用します。
トリガー条件が満たされなかった場合は、データを取得しないで関数から戻ってきます。
このとき、*AcquisitionCount* は 0 になります。

Example

```
int result;
BYTE id;      // オープンしたデバイスの ID
double val[10000];
int samplingRate, samplingCount;
BYTE triggerType;
double triggerThreshold;
int triggerWaitCount;
int acquisitionCount;

samplingRate = 50;   // 1kSPS
samplingCount = 10000;
triggerType = 3;     // 立上り
triggerThreshold = 3.5;     // 3.5V
triggerWaitCount = 500000;  // 500000×20usec＝10sec（タイムアウト）
acquisitionCount = 0;
result = FAD24_Aquisition(id, val, samplingClock, samplingCount,
       triggerType, triggerThreshold, triggerWaitCount, &acquisitionCount);

if (result == FAD24_OK) {
       // FAD24_Acquisition 成功
       if (acquisitionCount == 0) {
              // トリガー条件満たされず
       }
}
else {
       // FAD24_Acquisition 失敗
}
```

FAD24_AcquisitionExt

外部同期信号をサンプリングクロックとしてアナログ電圧値を取得します。

int **FAD24_AcquisitionExt** (BYTE *ID*, double **Value*,
BYTE *SamplingEdge*, int *SamplingCount*, int *SamplingEdgeWaitCount*,
BYTE *TriggerType*, double *TriggerThreshold*, int *TriggerWaitCount,*
int **AcquisitionCount*)

Parameters

ID	デバイスの ID
Value	アナログ電圧値の格納先配列へのポインタ
SamplingEdge	サンプリングクロック（外部同期信号）
SamplingCount	取得するアナログ電圧値のサンプリング数
SamplingEdgeWaitCount	サンプリングクロック（外部同期信号変化点）待ちカウント数
TriggerType	トリガータイプ
TriggerThreshold	しきい値
TriggerWaitCount	トリガー待ちカウント数
AcquisitionCount	取得したアナログ電圧値のカウント数のポインタ

Remarks

本関数が成功すると *Value* の配列にアナログ電圧値が格納されます。単位は[V]です。
SamplingEdge、*SamplingCount* にサンプリング設定をします。

SamplingEdge	説明
0	外部同期信号が”Off”から”On”への変化点 （On エッジ）
1	外部同期信号が”On”から”Off”への変化点 （Off エッジ）

SamplingCount の範囲は 1～1000000 です。
SamplingEdgeWaitCount×20 がサンプリングクロック（外部同期信号変化点）の待ち時間（usec）となります。外部同期信号に変化がない時のタイムアウトに使用します。
TriggerType、*TriggerThreshold*、*TriggerWaitCount* にトリガー条件を設定します。
TriggerType にトリガーなしを設定すると *TriggerThreshold*、*TriggerWaitCount* は無効になります。

TriggerType	説明
0	トリガーなし
1	設定したしきい値より電圧が高い （レベル）
2	設定したしきい値より電圧が低い （レベル）
3	設定したしきい値を低い方から横切る （立上り）
4	設定したしきい値を高い方から横切る （立下り）

TriggerThreshold はﾄﾘｶﾞｰするしきい値を電圧値[V]で設定します。
TriggerWaitCount×20 がﾄﾘｶﾞｰ条件が満たされるまでの待ち時間（usec）となります。
ﾄﾘｶﾞｰ条件が満たされない時のタイムアウトに使用します。
ﾄﾘｶﾞｰ条件が満たされなかった場合は、データを取得しないで関数から戻ってきます。
このとき、*AcquisitionCount* は 0 になります。

Example

```
int result;
BYTE id;      // オープンしたデバイスの ID
double val[10000];
BYTE samplingEdge;
int samplingCount, samplingEdgeWaitCount;
BYTE triggerType;
double triggerThreshold;
int triggerWaitCount;
int acquisitionCount;

samplingEdge = 0;    // 外部同期信号 ”Off”→”On”
samplingCount = 10000;
samplingEdgeWaitCount = 5000; // 5000×20usec＝100msec（タイムアウト）
triggerType = 0;     // なし
triggerThreshold = 0.0;
triggerWaitCount = 0;
acquisitionCount = 0;
result = FAD24_AcquisitionExt(id, val, samplingClock, samplingCount,
       samplingEdgeWaitCount, triggerType, triggerThreshold, triggerWaitCount,
       &acquisitionCount);

if (result == FAD24_OK) {
       // FAD24_AcquisitionExt 成功
}
else {
       // FAD24_AcquisitionExt 失敗
}
```

FAD24_AcquisitionSingle

アナログ電圧値を取得します。(単発取得)

int **FAD24_AcquisitionSingle** (BYTE *ID*, double **Value*,)

Parameters

ID	デバイスの ID
Value	アナログ電圧値の格納先配列へのポインタ

Remarks

本関数が成功すると *Value* にアナログ電圧値が格納されます。単位は[V]です。

Example

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

result = FAD24_AcquisitionSingle(id, &val);

if (result == FAD24_OK) {
       // FAD24_AcquisitionSingle 成功
}
else {
       // FAD24_AcquisitionSingle 失敗
}
```