

**USB A/D ボード**  
**SAI シリーズ**  
**SAI-12 / SAI-16**  
**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.....	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
SAI_GetNumberOfDevices.....	5
SAI_GetSerialNumber.....	6
SAI_Open.....	7
SAI_OpenBySerialNumber.....	8
SAI_Close .....	9
2.2 A/D 制御関数.....	10
SAI_GetAnalogVoltage.....	10

# 1 はじめに

## 1.1 概要

パソコンのUSBポートに接続して、**SAIシリーズ**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**SAIシリーズ**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。

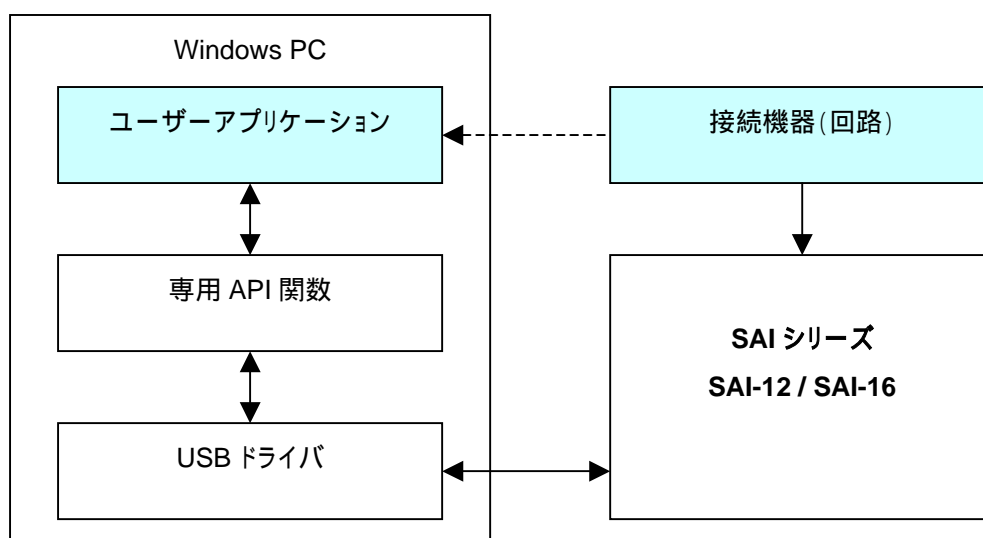


図 1.1 構成

## 1.2 関数について

**SAIシリーズ**専用API関数は関数群をモジュール化した「SAIxx.dll」で提供されます。「SAIxx.dll」ファイルは**SAIシリーズ**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

「SAIxx.h」、「SAIxx.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

「SAIxx.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

「SAIxx.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

「SAIxx.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
SAI_OK	0	正常終了
SAI_INVALID_ID	1	デバイスの ID が無効
SAI_DEVICE_NOT_FOUND	2	デバイスが見つからない
SAI_DEVICE_NOT_OPENED	3	デバイスがオープンできない
SAI_OTHER_ERROR	4	その他のエラーが発生した
SAI_COMMUNICATION_ERROR	5	通信エラーが発生した

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

#### SAI\_GetNumberOfDevices

現在接続されているデバイスの数を取得します。

```
int SAI_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 = SAI_GetNumberOfDevices(&number);  
if (result == SAI_OK) {  
    // SAI_GetNumberOfDevices 成功  
}  
else {  
    // SAI_GetNumberOfDevices 失敗  
}
```

## SAI\_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 SAI_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 **SAI\_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 = SAI_GetSerialNumber(index, &serialNumber);
if (result == SAI_OK) {
    // SAI_GetSerialNumber 成功
}
else {
    // SAI_GetSerialNumber 失敗
}
```

## SAI\_Open

デバイスをオープンし、デバイスの ID を取得します。

```
int SAI_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 = SAI_Open(&id);  
if (result == SAI_OK) {  
    // SAI_Open 成功  
}  
else {  
    // SAI_Open 失敗  
}
```

## SAI\_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 SAI_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 **SAI\_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 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 = SAI_OpenBySerialNumber(serialNumber, &id);  
if (result == SAI_OK) {  
    // SAI_OpenBySerialNumber 成功  
}  
else {  
    // SAI_OpenBySerialNumber 失敗  
}
```



## SAI\_Close

デバイスをクローズします。

int **SAI\_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 = SAI_Close(id);
if (result == SAI_OK) {
    // SAI_Close 成功
}
else {
    // SAI_Close 失敗
}
```

## 2.2 A/D 制御関数

### SAI\_GetAnalogVoltage

アナログ電圧値を取得します。

```
int SAI_GetAnalogVoltage (BYTE ID,  
                           double *CH0Value, double *CH1Value, double *CH2Value, double *CH3Value)
```

#### Parameters

<i>ID</i>	デバイスの ID
<i>CH0Value</i>	CH0 アナログ電圧値の格納先へのポインタ
<i>CH1Value</i>	CH1 アナログ電圧値の格納先へのポインタ
<i>CH2Value</i>	CH2 アナログ電圧値の格納先へのポインタ
<i>CH3Value</i>	CH3 アナログ電圧値の格納先へのポインタ

#### Remarks

本関数が成功すると *CH0~3Value* にアナログ電圧値が格納されます。単位は[V]です。  
アナログ電圧値が入力測定レンジ外の場合のアナログ電圧値は保証できません。

#### Example

```
int result;  
  
BYTE id;          // オープンしたデバイスの ID  
double ch0val, ch1val, ch2val, ch3val;  
  
result = SAI_GetAnalogVoltage(id, &ch0val, &ch1val, &ch2val, &ch3val);  
if (result == SAI_OK) {  
    // SAI_GetAnalogVoltage 成功  
}  
else {  
    // SAI_GetAnalogVoltage 失敗  
}
```