

リファレンスマニュアル

INtime
汎用ライブラリ

目次

第1章 関数一覧

1-1 汎用ライブラリAPIについて	1-1
--------------------------	-----

第2章 関数仕様

2-1 汎用入出力関数	2-1
2-2 初期化スイッチ・汎用LEDライブラリ関数	2-6
2-3 汎用SRAMライブラリ関数	2-11

第 1 章 関数一覧

本章では、INtime アプリケーションからデバイス制御を行うために用意した汎用ライブラリの使用方法について説明します。

1-1 汎用ライブラリAPIについて

汎用ライブラリ API は、INtime アプリケーションから各デバイスの操作を行うためのインターフェースを提供します。

産業用 PC/産業用パネル PC シリーズで使用できる汎用ライブラリ API 関数の一覧を以下に示します。

各機能の詳細は、産業用 PC/産業用パネル PC のマニュアルを参照してください。

1) 汎用入出力

RSL 名称	関 数	機 能
GenIo. RSL	GenIo_Open ()	汎用入出力ライブラリをオープンします
	GenIo_Close ()	汎用入出力ライブラリをクローズします
	GenIo_ReadIn ()	INPUT データを取得します
	GenIo_ReadOut ()	OUTPUT データを取得します
	GenIo_WriteOut ()	OUTPUT データを出力します

2) 初期化スイッチ・汎用 LED

RSL 名称	関 数	機 能
SwLed. RSL	SwLed_Open ()	初期化スイッチ・汎用 LED ライブラリをオープンします
	SwLed_Close ()	初期化スイッチ・汎用 LED ライブラリをクローズします
	SwLed_ReadSw ()	スイッチ状態を取得します
	SwLed_ReadLed ()	LED 状態を取得します
	SwLed_WriteLed ()	LED を出力します

3) 汎用 SRAM

RSL 名称	関 数	機 能
Sram. RSL	Sram_Open ()	汎用 SRAM ライブラリをオープンします
	Sram_Close ()	汎用 SRAM ライブラリをクローズします
	Sram_ReadByte ()	SRAM 領域からバイト単位で読み出します
	Sram_ReadWord ()	SRAM 領域からワード単位で読み出します
	Sram_WriteByte ()	SRAM 領域をバイト単位で書き込みます
	Sram_WriteWord ()	SRAM 領域をワード単位で書き込みます

※ ソフト PLC 機能使用時はソフト PLC が全領域を使用しますので、ユーザーは SRAM を使用できません。

第2章 関数仕様

2-1 汎用入出力関数

GenIo_Open 関数

機能 汎用入出力ライブラリをオープンします

書式 int GenIo_Open(void)

引数 なし

戻り値 0 : 正常
-1 : 異常

説明 ライブラリをオープンします。
汎用入出力ライブラリを使用の際には、必ずコールする必要があります。

GenIo_Close 関数

機能 汎用入出力ライブラリをクローズします

書式 int GenIo_Close(void)

引数 なし

戻り値 0 : 正常
-1 : 異常

説明 ライブラリをクローズします。
この関数コール後は、汎用入出力ライブラリにアクセス不可になります。

GenIo_ReadIn 関数

- 機能** 汎用入力データを取得します
- 書式** int GenIo_ReadIn(ULONG* data)
- 引数** data : 32 ビットデータを格納するポインタ
- 戻り値** 0 : 正常
-1 : 異常
- 説明** 汎用入力データを取得します。
汎用入力のサイズは各端末によって異なります。

GenIo_ReadOut 関数

- 機能** 汎用出力データを取得します
- 書式** int GenIo_ReadOut (ULONG* data)
- 引数** data : 32 ビットデータを格納するポインタ
- 戻り値** 0 : 正常
-1 : 異常
- 説明** 現在出力している汎用出力データを取得します。
汎用出力のサイズは各端末によって異なります。

GenIo_WriteOut 関数

機能 汎用出力データを出力します

書式 int GenIo_WriteOut(ULONG data)

引数 data : 32 ビット出力データ

戻り値 0 : 正常
-1 : 異常

説明 汎用出力データを出力します。
汎用出力のサイズは各端末によって異なります。

2-2 初期化スイッチ・汎用LEDライブラリ関数

SwLed_Open 関数

機能 初期化スイッチ・汎用LEDライブラリをオープンします

書式 int SwLed_Open(void)

引数 なし

戻り値 0 : 正常
-1 : 異常

説明 ライブラリをオープンします。
初期化スイッチ・汎用LEDライブラリを使用の際には、必ずコールする必要があります。

SwLed_Close 関数

機能 初期化スイッチ・汎用 LED ライブラリをクローズします

書式 int SwLed_Close(void)

引数 なし

戻り値 0 : 正常
-1 : 異常

説明 ライブラリをクローズします。
この関数コール後は、初期化スイッチ・汎用 LED ライブラリにアクセス不可になります。

SwLed_ReadSw 関数

機能 初期化スイッチの入力状態を取得します

書式 int SwLed_ReadSw(UCHAR* data)

引数 data : スイッチ状態を格納するポインタ

戻り値 0 : 正常
-1 : 異常

説明 初期化スイッチの入力状態を取得します。

SwLed_ReadLed 関数

- 機能** 汎用 LED の出力状態を取得します
- 書式** int SwLed_ReadLed (UCHAR* data)
- 引数** data : LED の状態を格納するポインタ
- 戻り値** 0 : 正常
-1 : 異常
- 説明** 汎用 LED の出力状態を取得します。

SwLed_WriteLed 関数

- 機能** 汎用 LED を出力します
- 書式** int SwLed_WriteLed(UCHAR data)
- 引数** data : LED 出力データ
- 戻り値** 0 : 正常
-1 : 異常
- 説明** LED 出力データで指定した LED に出力します。

2-3 汎用SRAMライブラリ関数

Sram_Open 関数

機能 汎用 SRAM ライブラリをオープンします

書式 int Sram_Open(void)

引数 なし

戻り値 0 : 正常
-1 : 異常

説明 ライブラリをオープンします。
汎用 SRAM ライブラリを使用の際には、必ずコールする必要があります。

Sram_Close 関数

機能 汎用 SRAM ライブラリをクローズします

書式 int Sram_Close(void)

引数 なし

戻り値 0 : 正常
-1 : 異常

説明 ライブラリをクローズします。
この関数コール後は、汎用 SRAM ライブラリにアクセス不可になります。

Sram_ReadByte 関数

機能

SRAM 領域からバイト単位で読み出します

書式

```
int Sram_ReadByte(int Address, UCHAR* Data, int Size)
```

引数

Address : アドレス [0x00000~0x3FF000]
Data : 読み出すデータを格納するポインタ
Size : 読み出すデータサイズ

戻り値

0 : 正常
-1 : 異常

説明

SRAM 領域の指定したアドレスからデータを読み出します。
サイズは読み出すデータサイズです。
読み出すデータを格納するポインタは読み出すサイズより大きいバッファを指定してください。

※ ソフト PLC 機能使用時はソフト PLC が全領域を使用しますので、ユーザーは SRAM を使用できません。

Sram_ReadWord 関数

機能

SRAM 領域からワード単位で読み出します

書式

```
int Sram_ReadWord(int Address, USHORT* Data, int Size)
```

引数

Address : アドレス [0x00000~0x3FF000]
Data : 読み出すデータを格納するポインタ
Size : 読み出すデータサイズ

戻り値

0 : 正常
-1 : 異常

説明

SRAM 領域の指定したアドレスからデータを読み出します。
サイズは読み出すデータサイズです。
読み出すデータを格納するポインタは読み出すサイズより大きいバッファを指定してください。

※ ソフト PLC 機能使用時はソフト PLC が全領域を使用しますので、ユーザーは SRAM を使用できません。

Sram_WriteByte 関数

機能

SRAM 領域をバイト単位で書き込みます

書式

```
int Sram_WriteByte(int Address, UCHAR* Data, int Size)
```

引数

Address : アドレス [0x00000~0x3FF000]
Data : 書き込むデータを格納したポインタ
Size : 書き込むデータサイズ

戻り値

0 : 正常
-1 : 異常

説明

SRAM 領域の指定したアドレスに指定した書き込みデータを書き込みます。
サイズは書き込むデータサイズです。
書き込むデータを格納したポインタは書き込むデータサイズより大きいバッファを指定してください。

※ ソフト PLC 機能使用時はソフト PLC が全領域を使用しますので、ユーザーは SRAM を使用できません。

Sram_WriteWord 関数

機能

SRAM 領域をワード単位で書き込みます

書式

```
int Sram_WriteWord(int Address, USHORT* Data, int Size)
```

引数

Address : アドレス [0x00000~0x3FF000]
Data : 書き込むデータを格納したポインタ
Size : 書き込むデータサイズ

戻り値

0 : 正常
-1 : 異常

説明

SRAM 領域の指定したアドレスに指定した書き込みデータを書き込みます。
サイズは書き込むデータサイズ×2バイトです。
書き込むデータを格納したポインタは書き込むデータサイズより大きいバッファを指定してください。

※ ソフト PLC 機能使用時はソフト PLC が全領域を使用しますので、ユーザーは SRAM を使用できません。

このマニュアルについて

- (1)本書の内容の一部または全部を当社からの事前の承諾を得ることなく、無断で複写、複製、掲載することは固くお断りします。
- (2)本書の内容に関しては、製品改良のためお断りなく、仕様などを変更することがありますのでご了承下さい。
- (3)本書の内容に関しては万全を期しておりますが、万一ご不審な点や誤りなどお気づきのことがございましたらお手数ですが巻末記載の弊社もしくは、営業所までご連絡下さい。その際、巻末記載の書籍番号も併せてお知らせ下さい。

77IT10021B
77IT10021A

2015年 5月 第2版
2014年 9月 初版

 株式会社アルゴシステム

本社
〒587-0021 大阪府堺市美原区小平尾656番地

TEL (072) 362-5067
FAX (072) 362-4856

ホームページ <http://www.algosystem.co.jp>