

**リファレンスマニュアル**

**EtherCAT コントロールボード  
Linux ライブラリ**

# 目次

## 第1章 関数一覧

## 第2章 関数仕様

2-1 ライブラリ使用方法	2-1
2-1-1 アプリケーション開始	2-1
2-1-2 アプリケーション終了	2-2
2-2 EtherCATコントロールボード アクセス関数	2-3

## 第 1 章 関数一覧

### 1) EtherCAT コントロールボード アクセス関数

関 数	機 能
EMM_Connect()	コントロールボードと接続します。
EMM_DisConnect()	コントロールボードと切断します。
EMM_SmemOpen()	指定した共有メモリをオープンします。
EMM_SmemClose()	指定した共有メモリをクローズします。
EMM_SmemIsMemSize()	指定した共有メモリのサイズを取得します。
EMM_ShutdownNotify()	シャットダウン発生を通知します。
EMM_SetDateTime()	現在日時を設定します。
EMM_Reset()	コントロールボードにソフトリセットを実行します。
EMM_GetVersion()	ハードバージョン、ソフトバージョンを取得します。
EMM_GetBoardNumber()	ボードナンバーを取得します。
EMM_SmemReadByte()	指定した共有メモリから 1BYTE (uint8_t) データを取得します。
EMM_SmemReadWord()	指定した共有メモリから 1WORD (uint16_t) データを取得します。
EMM_SmemReadDWord()	指定した共有メモリから 1DWORD (uint32_t) データを取得します。
EMM_SmemReadLWord()	指定した共有メモリから 1DWORDLONG (uint64_t) データを取得します。
EMM_SmemReadBytes()	指定した共有メモリから指定した長さの BYTE 列を取得します。
EMM_SmemWriteByte()	指定した共有メモリへ 1BYTE (uint8_t) データを書き込みます。
EMM_SmemWriteWord()	指定した共有メモリへ 1WORD (uint16_t) データを書き込みます。
EMM_SmemWriteDWord()	指定した共有メモリへ 1DWORD (uint32_t) データを書き込みます。
EMM_SmemWriteLWord()	指定した共有メモリへ 1DWORDLONG (uint64_t) データを書き込みます。
EMM_SmemWriteBytes()	指定した共有メモリへ指定した長さの BYTE 列を書き込みます。

## 第2章 関数仕様

### 2-1 ライブラリ使用方法

ライブラリを使用したアプリケーション開始、アプリケーション終了のフローチャートを以下に示します。

#### 2-1-1 アプリケーション開始

ライブラリを使用したアプリケーション開始のフローチャートを以下に示します。

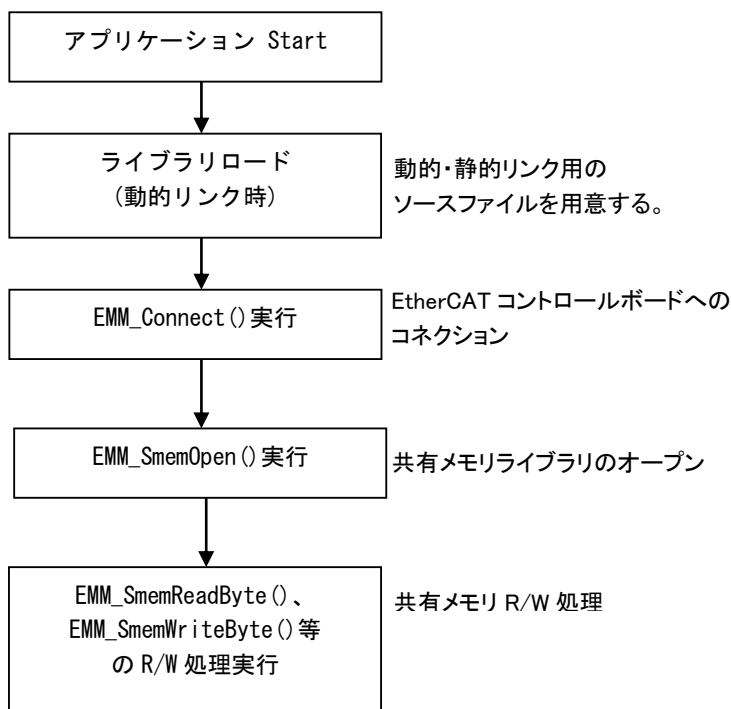


図 2-1-1. アプリケーション開始フローチャート

ライブラリロード後、コネクション、共有メモリのオープンを行うことで共有メモリへアクセス可能となります。

## 2-1-2 アプリケーション終了

ライブラリを使用したアプリケーション終了のフローチャートを以下に示します。

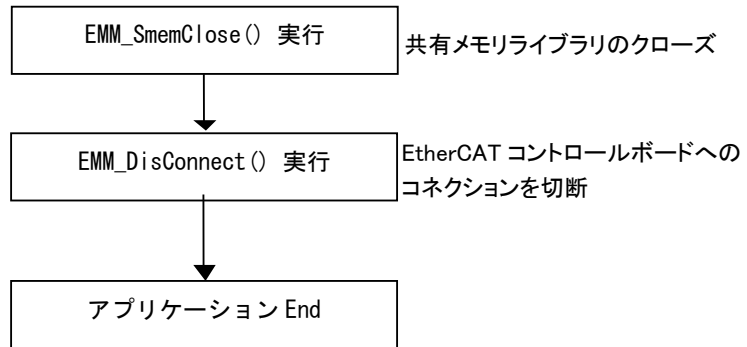


図 2-1-2. アプリケーション終了フローチャート

## 2-2 EtherCAT コントロールボード アクセス関数

### EMM\_Connect 関数

---

**機能**

コントロールボードと接続します。

**書式**

```
int EMM_Connect (int Board, const char *pIpAddress, unsigned int port);
```

**引数**

**Board :**

ボード番号 (0-3)

**pIpAddress :**

コントロールボードの IP アドレスを格納したポインタ

**port :**

コントロールボードのポート番号

**戻り値**

MOTION\_ER\_OK :

成功。

MOTION\_ER\_INVALIDPARAM :

無効な引数。

MOTION\_ER\_ALREADYCONNECT :

既に接続済みです。

MOTION\_ER\_OPENDEVICE :

コントロールボードの接続に失敗しました。

**説明**

本ライブラリを初期化し、指定した IP アドレス、ポート番号のコントロールボードとコネクション接続します。

本関数をコール後、コントロールボードとコマンドのやり取りをすることができます。

---

## EMM\_DisConnect 関数

---

<b>機能</b>	コントロールボードと切断します。
<b>書式</b>	<code>int EMM_DisConnect (int Board);</code>
<b>引数</b>	<b>Board :</b> ボード番号 (0-3)
<b>戻り値</b>	<b>MOTION_ER_OK :</b> 成功。 <b>MOTION_ER_INVALIDPARAM :</b> 無効な引数。 <b>MOTION_ER_NOTOPEN :</b> オープンされていません。
<b>説明</b>	コントロールボードとコネクションを切断します。 本関数をコール後、コントロールボードとコマンドのやり取りを行うことができなくなります。

---

## EMM\_SmemOpen 関数

---

- 機能**            コントロールボードの共有メモリをオープンします。
- 書式**            `unsigned int EMM_SmemOpen (int Board, const char *pShmemName);`
- 引数**            `Board` :  
                  ボード番号 (0-3)  
`pShmemName` :  
                  コントロールボード上の共有メモリ名を格納したポインタ
- 戻り値**            共有メモリのハンドル (値 0 以上) :  
                  成功。  
`MOTION_ER_INVALIDPARAM` :  
                  無効な引数。  
`MOTION_ER_NOTCONNECT` :  
                  コントロールボードと接続されていません。  
`ER_MEMORY_ALREADYOPEN` :  
                  既にオープン済みです。  
`MOTION_ER_OPENDEVICE` :  
                  コントロールボードの接続に失敗しました。
- 説明**            共有メモリ名で指定したコントロールボード上の共有メモリをオープンします。  
                  本ライブラリが成功した場合、共有メモリのハンドルを取得することができます。  
                  本関数をコール後、取得したハンドルを使用し、`Read/Write` 関数で共有メモリを  
                  読み書きすることができます。



---

## EMM\_SmemClose 関数

---

**機能**

コントロールボードの共有メモリをクローズします。

**書式**

```
int EMM_SmemClose (int Board, unsigned int handle);
```

**引数**

**Board :**

ボード番号 (0-3)

**handle :**

共有メモリのハンドル

**戻り値**

MOTION\_ER\_OK :

成功。

MOTION\_ER\_INVALIDPARAM :

無効な引数。

MOTION\_ER\_NOTOPEN :

オープンされていません。

MOTION\_ER\_OPENDEVICE :

コントロールボードの接続に失敗しました。

**説明**

引数の共有メモリハンドルで指定した共有メモリをクローズします。

本関数をコール後、取得した共有メモリのハンドルでは共有メモリの Read/Write は使用できません。

---

## EMM\_SmemIsMemSize 関数

---

<b>機能</b>	共有メモリのサイズを確認します。
<b>書式</b>	<code>int EMM_SmemIsMemSize (int Board, unsigned int handle);</code>
<b>引数</b>	<b>Board :</b> ボード番号 (0-3) <b>handle :</b> 共有メモリのハンドル
<b>戻り値</b>	共有メモリのサイズ (値 0 以上) : 成功。 MOTION_ER_INVALIDPARAM : 無効な引数。 MOTION_ER_NOTCONNECT : コントロールボードと接続されていません。 MOTION_ER_NOTOPEN : オープンされていません。 MOTION_ER_OPENDEVICE : コントロールボードの接続に失敗しました。
<b>説明</b>	共有メモリのサイズを取得します。 本ライブラリが成功した場合、戻り値に引数の共有メモリハンドルで指定した共有メモリのサイズが取得できます。

---

## EMM\_ShutdownNotify 関数

---

<b>機能</b>	コントロールボードにシャットダウンを通知します。
<b>書式</b>	<code>int EMM_ShutdownNotify(int Board);</code>
<b>引数</b>	<b>Board :</b> ボード番号 (0-3)
<b>戻り値</b>	<b>MOTION_ER_OK :</b> 成功。 <b>MOTION_ER_INVALIDPARAM :</b> 無効な引数。 <b>MOTION_ER_NOTOPEN :</b> オープンされていません。 <b>MOTION_ER_OPENDEVICE :</b> コントロールボードの接続に失敗しました。
<b>説明</b>	コントロールボードにシャットダウンを通知し、設定の保存を要求します。

---

## EMM\_SetDateTime 関数

---

- 機能** システム日時を設定します。
- 書式** `int EMM_SetDateTime (int Board, LPSYSTEMTIME lpSystemTime);`
- 引数**
- Board :**  
ボード番号 (0-3)
- lpSystemTime :**  
SYSTEMTIME 構造体を格納するポインタ
- 戻り値**
- MOTION\_ER\_OK :  
成功。
- MOTION\_ER\_INVALIDPARAM :  
無効な引数。
- MOTION\_ER\_NOTOPEN :  
オープンされていません。
- MOTION\_ER\_OPENDEVICE :  
コントロールボードの接続に失敗しました。
- 説明** コントロールボードのシステム日時を設定します。

---

## EMM\_Reset 関数

---

<b>機能</b>	コントロールボードをリセットします。
<b>書式</b>	<code>int EMM_Reset (int Board);</code>
<b>引数</b>	<b>Board :</b> ボード番号 (0-3)
<b>戻り値</b>	<b>MOTION_ER_OK :</b> 成功。 <b>MOTION_ER_INVALIDPARAM :</b> 無効な引数。 <b>MOTION_ER_NOTOPEN :</b> オープンされていません。 <b>MOTION_ER_OPENDEVICE :</b> コントロールボードの接続に失敗しました。
<b>説明</b>	コントロールボードにリセット要求をします。 成功した場合、コントロールボードがリセットされます。

---

## EMM\_GetVersion 関数

---

- 機能** バージョンを取得します。
- 書式** `int EMM_GetVersion (int Board, unsigned short *hver, unsigned short *sver);`
- 引数**
- Board :**  
ボード番号 (0-3)
  - hver :**  
ハードウェアバージョンを格納するポインタ
  - sver :**  
ソフトウェアバージョンを格納するポインタ
- 戻り値**
- MOTION\_ER\_OK :**  
成功。
  - MOTION\_ER\_INVALIDPARAM :**  
無効な引数。
  - MOTION\_ER\_NOTOPEN :**  
オープンされていません。
  - MOTION\_ER\_OPENDEVICE :**  
コントロールボードの接続に失敗しました。
- 説明** コントロールボードのハードウェアバージョンとソフトウェアバージョンを取得します。

---

## EMM\_GetBoardNumber 関数

---

- 機能**      ボードナンバー(基板識別番号)を取得します。
- 書式**      `int EMM_GetBoardNumber (int Board, unsigned short *boardnumber);`
- 引数**      `Board` :  
              ボード番号(0-3)  
`boardnumber` :  
              ボードナンバーを格納するポインタ
- 戻り値**      `MOTION_ER_OK` :  
              成功。  
`MOTION_ER_INVALIDPARAM` :  
              無効な引数。  
`MOTION_ER_NOTOPEN` :  
              オープンされていません。  
`MOTION_ER_OPENDEVICE` :  
              コントロールボードの接続に失敗しました。
- 説明**      コントロールボードのボードナンバーを取得します。

---

## EMM\_SmemReadByte 関数

---

**機能**

共有メモリから 1BYTE (uint8\_t) のデータを読み出します。

**書式**

```
int EMM_SmemReadByte (  
    int Board,  
    unsigned int handle,  
    unsigned int address,  
    BYTE *pdata  
);
```

**引数**

**Board** :  
 ボード番号 (0-3)  
**handle** :  
 共有メモリのハンドル  
**address** :  
 共有メモリのアドレス  
**pdata** :  
 取得データを保存するポインタ

**戻り値**

**MOTION\_ER\_OK** :  
 成功。  
**MOTION\_ER\_INVALIDPARAM** :  
 無効な引数。  
**MOTION\_ER\_NOTOPEN** :  
 オープンされていません。  
**MOTION\_ER\_OPENDEVICE** :  
 コントロールボードの接続に失敗しました。

**説明**

ハンドルで指定した共有メモリの指定した address からデータを 1BYTE (uint8\_t) 読み出します。



---

## EMM\_SmemReadWord 関数

---

- 機能** 共有メモリから 1WORD (uint16\_t) のデータを読み出します。
- 書式**
- ```
int EMM_SmemReadWord (  
    int Board,  
    unsigned int handle,  
    unsigned int address,  
    WORD *pdata  
);
```
- 引数**
- Board :**  
ボード番号 (0-3)
- handle :**  
共有メモリのハンドル
- address :**  
共有メモリのアドレス
- pdata :**  
取得データを保存するポインタ
- 戻り値**
- MOTION\_ER\_OK :  
成功。
- MOTION\_ER\_INVALIDPARAM :  
無効な引数。
- MOTION\_ER\_NOTOPEN :  
オープンされていません。
- MOTION\_ER\_OPENDEVICE :  
コントロールボードの接続に失敗しました。
- 説明** ハンドルで指定した共有メモリの指定した address からデータを 1WORD (uint16\_t) 読み出します。

## EMM\_SmemReadDWord 関数

**機能**

共有メモリから 1DWORD (uint32\_t) のデータを読み出します。

**書式**

```
int EMM_SmemReadDWord (  
    int Board,  
    unsigned int handle,  
    unsigned int address,  
    DWORD *pdata  
);
```

**引数**

**Board** :  
ボード番号 (0-3)  
**handle** :  
共有メモリのハンドル  
**address** :  
共有メモリのアドレス  
**pdata** :  
取得データを保存するポインタ

**戻り値**

**MOTION\_ER\_OK** :  
成功。  
**MOTION\_ER\_INVALIDPARAM** :  
無効な引数。  
**MOTION\_ER\_NOTOPEN** :  
オープンされていません。  
**MOTION\_ER\_OPENDEVICE** :  
コントロールボードの接続に失敗しました。

**説明**

ハンドルで指定した共有メモリの指定した address からデータを 1DWORD (uint32\_t) 読み出します。

---

## EMM\_SmemReadLWord 関数

---

**機能**

共有メモリから 1DWORDLONG (uint64\_t) のデータを読み出します。

**書式**

```
int EMM_SmemReadLWord (  
    int Board,  
    unsigned int handle,  
    unsigned int address,  
    DWORDLONG *pdata  
);
```

**引数**

**Board** :  
 ボード番号 (0-3)  
**handle** :  
 共有メモリのハンドル  
**address** :  
 共有メモリのアドレス  
**pdata** :  
 取得データを保存するポインタ

**戻り値**

**MOTION\_ER\_OK** :  
 成功。  
**MOTION\_ER\_INVALIDPARAM** :  
 無効な引数。  
**MOTION\_ER\_NOTOPEN** :  
 オープンされていません。  
**MOTION\_ER\_OPENDEVICE** :  
 コントロールボードの接続に失敗しました。

**説明**

ハンドルで指定した共有メモリの指定した address からデータを 1DWORDLONG (uint64\_t) 読み出します。

---

## EMM\_SmemReadBytes 関数

---

**機能**

共有メモリから BYTE 列 (uint8\_t\*) のデータを読み出します。

**書式**

```
int EMM_SmemReadBytes (  
    int Board,  
    unsigned int handle,  
    unsigned int address,  
    BYTE *pdata,  
    unsigned int len  
);
```

**引数**

**Board** :  
 ボード番号 (0-3)  
**handle** :  
 共有メモリのハンドル  
**address** :  
 共有メモリのアドレス  
**pdata** :  
 取得データを保存するポインタ  
**len** :  
 取得データ長

**戻り値**

**MOTION\_ER\_OK** :  
 成功。  
**MOTION\_ER\_INVALIDPARAM** :  
 無効な引数。  
**MOTION\_ER\_NOTOPEN** :  
 オープンされていません。  
**MOTION\_ER\_OPENDEVICE** :  
 コントロールボードの接続に失敗しました。

**説明**

ハンドルで指定した共有メモリの指定した address から取得データ長分、取得データを読み出します。

---

## EMM\_SmemWriteByte 関数

---

**機能**

共有メモリへ 1BYTE (uint8\_t) のデータを書き込みます。

**書式**

```
int EMM_SmemWriteByte (  
    int Board,  
    unsigned int handle,  
    unsigned int address,  
    BYTE data  
);
```

**引数**

**Board** :  
 ボード番号 (0-3)  
**handle** :  
 共有メモリのハンドル  
**address** :  
 共有メモリのアドレス  
**data** :  
 書き込みデータ

**戻り値**

**MOTION\_ER\_OK** :  
 成功。  
**MOTION\_ER\_INVALIDPARAM** :  
 無効な引数。  
**MOTION\_ER\_NOTOPEN** :  
 オープンされていません。  
**MOTION\_ER\_OPENDEVICE** :  
 コントロールボードの接続に失敗しました。

**説明**

ハンドルで指定した共有メモリの指定した address へデータを 1BYTE (uint8\_t) 書き込みます。

---

## EMM\_SmemWriteWord 関数

---

- 機能** 共有メモリへ 1WORD(uint16\_t) のデータを書き込みます。
- 書式**
- ```
int EMM_SmemWriteWord (  
    int Board,  
    unsigned int handle,  
    unsigned int address,  
    WORD data  
);
```
- 引数**
- Board :**  
ボード番号 (0-3)
- handle :**  
共有メモリのハンドル
- address :**  
共有メモリのアドレス
- data :**  
書き込みデータ
- 戻り値**
- MOTION\_ER\_OK :  
成功。
- MOTION\_ER\_INVALIDPARAM :  
無効な引数。
- MOTION\_ER\_NOTOPEN :  
オープンされていません。
- MOTION\_ER\_OPENDEVICE :  
コントロールボードの接続に失敗しました。
- 説明** ハンドルで指定した共有メモリの指定した address へデータを 1WORD(uint16\_t) 書き込みます。

---

## EMM\_SmemWriteDWord 関数

---

- 機能** 共有メモリへ 1DWORD (uint32\_t) のデータを書き込みます。
- 書式**
- ```
int EMM_SmemWriteDWord (  
    int Board,  
    unsigned int handle,  
    unsigned int address,  
    DWORD data  
);
```
- 引数**
- Board :**  
ボード番号 (0-3)
- handle :**  
共有メモリのハンドル
- address :**  
共有メモリのアドレス
- data :**  
書き込みデータ
- 戻り値**
- MOTION\_ER\_OK :  
成功。
- MOTION\_ER\_INVALIDPARAM :  
無効な引数。
- MOTION\_ER\_NOTOPEN :  
オープンされていません。
- MOTION\_ER\_OPENDEVICE :  
コントロールボードの接続に失敗しました。
- 説明** ハンドルで指定した共有メモリの指定した address へデータを 1DWORD (uint32\_t) 書き込みます。

---

## EMM\_SmemWriteLWord 関数

---

**機能**

共有メモリへ 1DWORDLONG (uint64\_t) のデータを書き込みます。

**書式**

```
int EMM_SmemWriteLWord (  
    int Board,  
    unsigned int handle,  
    unsigned int address,  
    DWORDLONG data  
);
```

**引数**

**Board** :  
 ボード番号 (0-3)  
**handle** :  
 共有メモリのハンドル  
**address** :  
 共有メモリのアドレス  
**data** :  
 書き込みデータ

**戻り値**

**MOTION\_ER\_OK** :  
 成功。  
**MOTION\_ER\_INVALIDPARAM** :  
 無効な引数。  
**MOTION\_ER\_NOTOPEN** :  
 オープンされていません。  
**MOTION\_ER\_OPENDEVICE** :  
 コントロールボードの接続に失敗しました。

**説明**

ハンドルで指定した共有メモリの指定した address へデータを 1DWORDLONG (uint64\_t) 書き込みます。



## EMM\_SmemWriteBytes 関数

**機能**

共有メモリへ BYTE 列(uint8\_t\*) データを書き込みます。

**書式**

```
int EMM_SmemWriteBytes (  
    int Board,  
    unsigned int handle,  
    unsigned int address,  
    BYTE *pdata,  
    unsigned int len  
);
```

**引数**

**Board** :  
ボード番号(0-3)  
**handle** :  
共有メモリのハンドル  
**address** :  
共有メモリのアドレス  
**pdata** :  
書き込みデータを保存するポインタ  
**len** :  
書き込みデータ長

**戻り値**

**MOTION\_ER\_OK** :  
成功。  
**MOTION\_ER\_INVALIDPARAM** :  
無効な引数。  
**MOTION\_ER\_NOTOPEN** :  
オープンされていません。  
**MOTION\_ER\_OPENDEVICE** :  
コントロールボードの接続に失敗しました。

**説明**

ハンドルで指定した共有メモリの指定した address へ取得データ長分、BYTE 列データを書き込みます。

## このリファレンス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。

77A820005B  
77A820005A

2021年 9月 第2版  
2021年 1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/>