

**ユーザーズマニュアル**

**EtherCAT コントロールボード  
Linux ライブラリ**

# 目次

## 概要

## 第1章 アプリケーション開発

- 1-1 ライブラリの作成 ..... 1-1
- 1-2 ライブラリ動作環境 ..... 1-1
- 1-3 アプリケーション開発の準備 ..... 1-2

## 第2章 ライブラリ関数

- 2-1 ライブラリ関数概要 ..... 2-1
- 2-2 ライブラリ使用方法 ..... 2-1
  - 2-2-1 ライブラリパスの設定 ..... 2-1
  - 2-2-2 アプリケーション開始 ..... 2-1
  - 2-2-3 アプリケーション終了 ..... 2-2

## 第3章 付録

- 3-1 サンプルソース ..... 3-1

## 概要

本ライブラリ(以下ライブラリとする)「LibIMM.so」は、EtherCAT コントロールボードとやり取りを行うことができるインターフェイスです。

Linux 側アプリケーションから EtherCAT コントロールボードへアクセスするために提供されます。

ユーザーは、QtCreator 等の開発言語からライブラリ関数をコールすることによって、EtherCAT コントロールボードにアクセスするアプリケーションを作成することができます。

# 第1章 アプリケーション開発

## 1-1 ライブラリの作成

ライブラリは、arm64/x64 のバイナリ形式で提供しています。  
 他のLinux環境で動作するためには、ソースから再コンパイルしてください。

```

/src/IfMotionMasterIf   ライブラリソース
/src/arm64              arm64 版ライブラリ
/src/x64                x64 版ライブラリ
    
```

提供されたライブラリをコンパイルしなおす必要がある場合は、/src/x64のMakefileを参照の上  
 CC=g++  
 AR=ar  
 を開発環境のコンパイラに合わせて変更後、makeを実行することでlibIMM.soが作成されます。

## 1-2 ライブラリ動作環境

ユーザーは作成するアプリケーション内でLibIMM.soの関数をコールすることにより、共有メモリへのデータのRead/Writeを処理します。  
 libIMM.soは任意のフォルダに格納することができますが、標準ライブラリフォルダ/libや/usr/lib以外へ保存する場合はLD\_LIBRARY\_PATH環境変数を設定する必要があります。

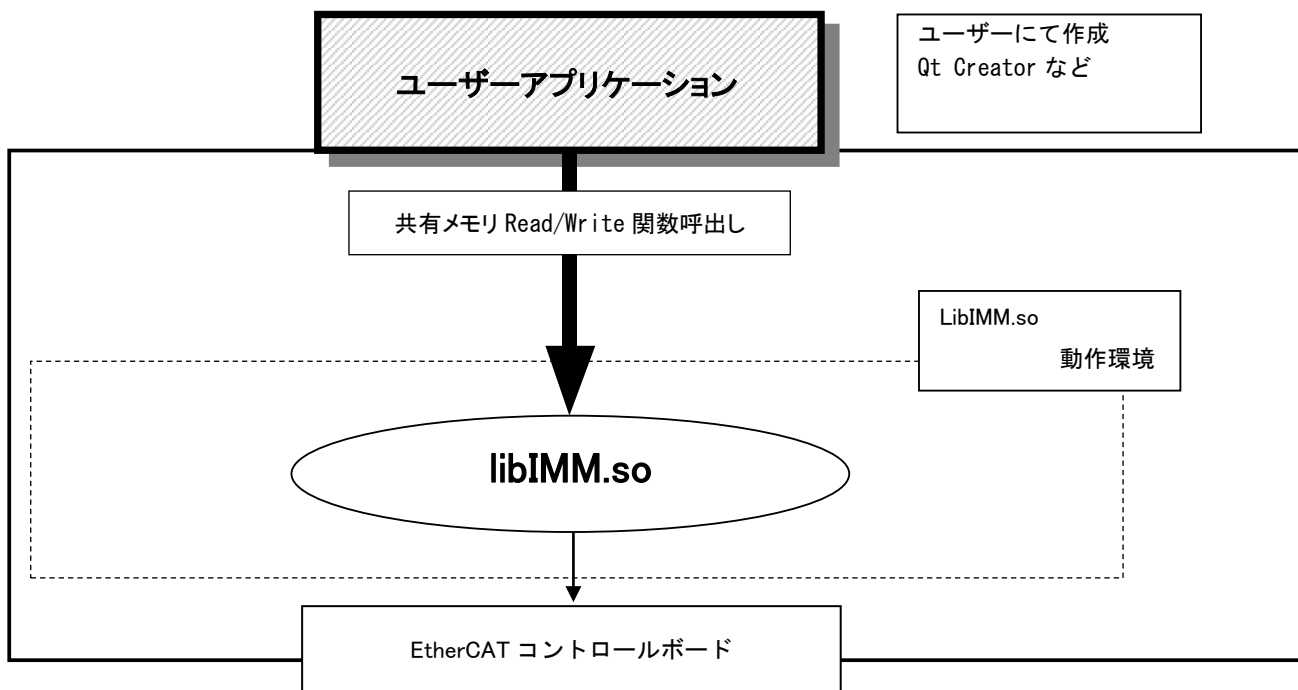


図1-2-1. ライブラリ動作環境

### 1-3 アプリケーション開発の準備

開発アプリケーションからライブラリ関数をコールできるようにする為に、開発ユーザーは下記の手順を実行します。

- 1) プロジェクトのソースファイルがあるフォルダに、以下のファイルをコピーします。

表1-3-1. SDK ディレクトリファイル一覧

種類	ファイル名	内容
ヘッダ	libIMM.h	ライブラリリンク用定義ファイル
ライブラリ	libIMM.so	ライブラリファイル

共有メモリライブラリの関数をコールするソースファイルへ、libIMM.h をインクルードします。

- 2) リンカオプションとして、`-L./ -lIMM` を追加します。  
ライブラリをソースと別ディレクトリ (例 `/home/asdusr/lib` の場合)  
リンカオプションを `-L/home/asdusr/lib -lIMM` としてください、

※ 上記で使用されるヘッダファイル等は、開発環境 CD-ROM に含まれています。

## 第2章 ライブラリ関数

### 2-1 ライブラリ関数概要

ライブラリには、共有メモリの名前を指定し、共有メモリのハンドルを取得します。取得したハンドルを使用して共有メモリにアクセスするための関数が用意されています。各関数の詳細は「EtherCAT コントロールボード ライブラリ リファレンスマニュアル」を参照して下さい。

### 2-2 ライブラリ使用方法

#### 2-2-1 ライブラリパスの設定

動的ライブラリを使用する際は、ライブラリの（環境変数 LD\_LIBRARY\_PATH）を設定する必要があります。

例) ライブラリを/home/asdusr/libにおく場合

```
export LD_LIBRARY_PATH=/home/asdusr/lib
```

#### 2-2-2 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。

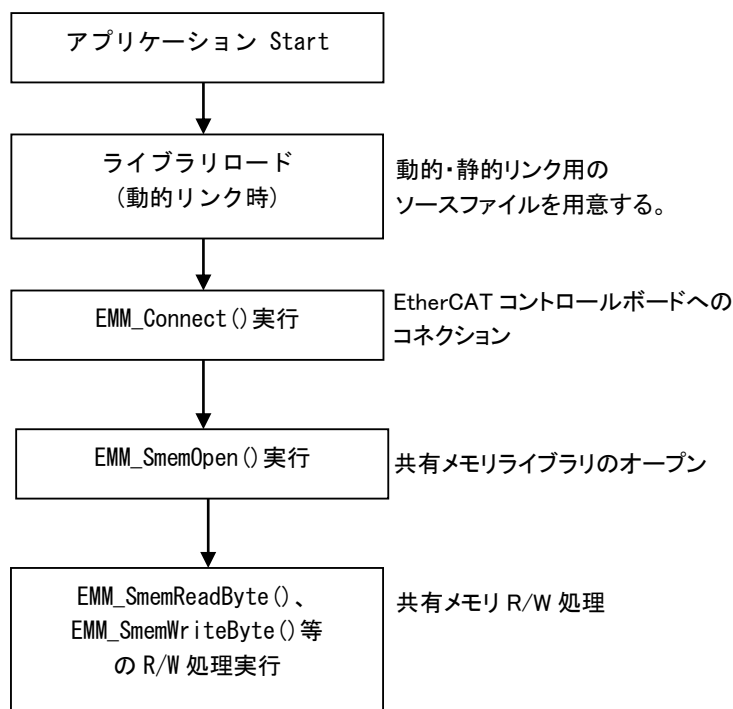


図2-2-2-1. アプリケーション開始フローチャート

## 2-2-3 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。

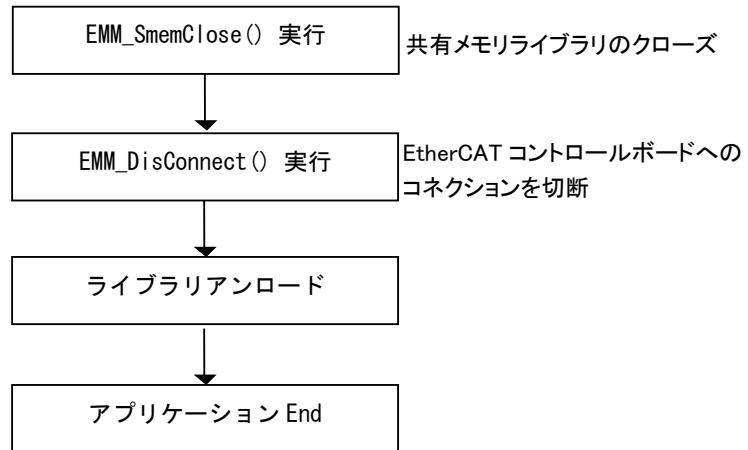


図2-2-3-1. アプリケーション終了フローチャート

## 第3章 付録

### 3-1 サンプルソース

C++ 用 ライブラリアクセスサンプル

ライブラリとのリンク、コネクションと共有メモリの Read/Write を行うサンプルを次に示します。

#### 1) ライブラリリンク、コネクション

```
int error
// EtherCAT コントロールボードのコネクション
error = EMM_Connect(0, "192.168.0.10", 10001);
if ( error!=0 ) {
    printf("open error!!¥n");
    goto LIB_UNLOAD ;
}

// EtherCAT コントロールボードのコネクション切断
EMM_DisConnect(0);
```



## 2) 共有メモリ 読み込み/書き込み

```
int error;
unsigned int handle = -1;
unsigned int address = 100000;
DWORD dat = 0;

// 共有メモリのオープン
handle = EMM_SmemOpen(0, "HMH1");
if (handle<0 ) {
    printf("open error!!\n");
    return ;
}

// 読み込み
error = EMM_SmemReadDWord(0, handle, address, &dat);
if ( error!=0 ) {
    printf("read error!!\n");
    goto_LIB_CLOSE ;
}

// 書き込み
dat = dat + 1;
error = EMM_SmemWriteDWord(0, handle, address, dat);
if ( error!=0 ) {
    printf("write error!!\n");
    goto_LIB_CLOSE ;
}

LIB_CLOSE:
// 共有メモリ ライブラリのクローズ
EMM_SmemClose(0, handle);
```

## このユーザーズ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。

77A820004B  
77A820004A

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/>