

リファレンスマニュアル

ML3 RSL

# 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
2-2 マスタアクセス関数 .....	2-2
2-3 デジタル入出力ユニット関数 .....	2-16
2-4 アナログ入出力ユニット関数 .....	2-22
2-5 エンコーダユニット関数 .....	2-36
2-6 付録 .....	2-44
2-6-1 エラーコード .....	2-44

## 第 1 章 関数一覧

### 1) マスタアクセス関数

関 数	機 能
ML3_Mst_Open( )	マスタをオープンします
ML3_Mst_Close( )	マスタをクローズします
ML3_Mst_GetCommSetting( )	マスタの通信設定を取得します
ML3_Mst_StartComm( )	マスタの通信を開始します
ML3_Mst_ChkStartComm( )	マスタの通信開始を確認します
ML3_Mst_GetSystemError( )	マスタのシステムエラーを取得します
ML3_Mst_GetMasterStatus( )	マスタの通信状態を取得します
ML3_Mst_ResetComm( )	マスタの通信を停止します
ML3_Mst_ChkResetComm( )	マスタの通信停止を確認します
ML3_Slv_GetSlaveStatus( )	スレーブの通信状態を取得します
ML3_Slv_GetSlaveCommInfo( )	スレーブの通信設定を取得します

### 2) デジタル入出力ユニット

関 数	機 能
ML3_Dio_Open( )	ユニットをオープンします
ML3_Dio_Close( )	ユニットをクローズします
ML3_Dio_Read( )	ユニットの Input データを取得します
ML3_Dio_Write( )	ユニットに Output データを出力します
ML3_Dio_GetParam( )	ユニットの設定パラメータを取得します
ML3_Dio_SetParam( )	ユニットに設定パラメータを設定します

### 3) アナログ入出力ユニット

関 数	機 能
ML3_Aio_Open( )	ユニットをオープンします
ML3_Aio_Close( )	ユニットをクローズします
ML3_Aio_Read( )	ユニットの Input データを取得します
ML3_Aio_Write( )	ユニットに Output データを出力します
ML3_Aio_AD_GetParam( )	ユニットの設定パラメータを取得します
ML3_Aio_AD_SetMode( )	ユニットのモードを設定します
ML3_Aio_AD_SetParam( )	ユニットに設定パラメータを設定します
ML3_Aio_AD_SetCalib( )	ユニットのキャリブレーションを行います
ML3_Aio_DA_GetParam( )	ユニットの設定パラメータを取得します
ML3_Aio_DA_SetMode( )	ユニットのモードを設定します
ML3_Aio_DA_SetParam( )	ユニットに設定パラメータを設定します
ML3_Aio_DA_SetCalib( )	ユニットのキャリブレーションを行います

## 4) エンコーダユニット

関 数	機 能
ML3_Enc_Open( )	ユニットをオープンします
ML3_Enc_Close( )	ユニットをクローズします
ML3_Enc_Read( )	ユニットの Input データを取得します
ML3_Enc_Write( )	ユニットに Output データを出力します
ML3_Enc_GetParam( )	ユニットの設定パラメータを取得します
ML3_Enc_SetParam( )	ユニットの設定パラメータを設定します

## 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のフローチャートを以下に示します。

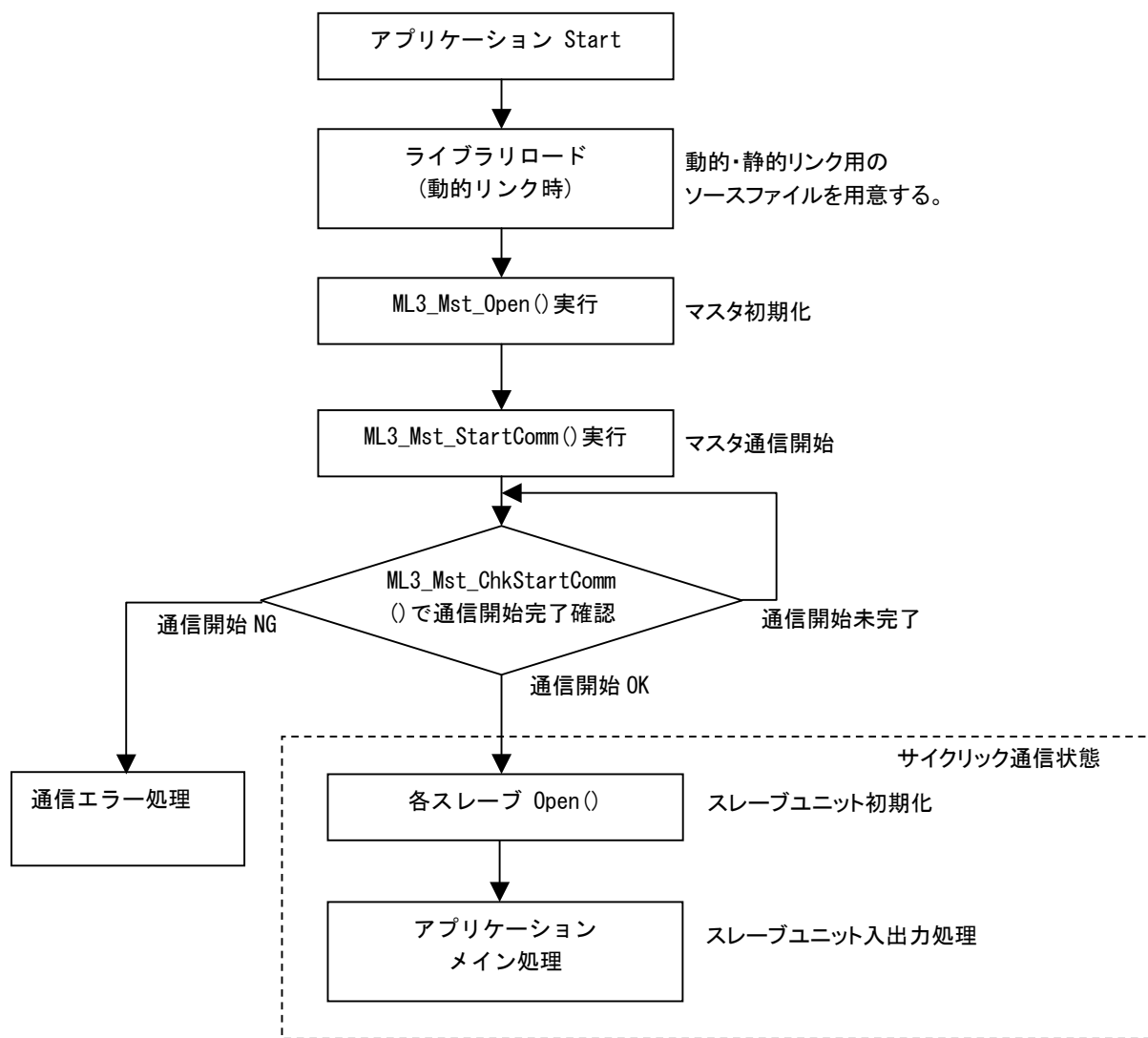


図 2-1-1. フローチャート

ライブラリロード後、マスタ初期化、マスタ通信開始を行うことでスレーブユニットへアクセス可能となります。スレーブユニットの Open をコール後、スレーブユニットの入出力処理を行うことができます。

## 2-2 マスタアクセス関数

### ML3\_Mst\_Open 関数

---

**機能** MECHATROLINK マスタをオープンします。

**書式** `int ML3_Mst_Open (void);`

**引数** なし

**戻り値** ML3\_ER\_OK : 正常  
ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

**説明** MECHATROLINK マスタをオープンします。  
本関数をコール後、MECHATROLINK マスタにアクセス可能となります。本ライブラリを使用してマスタ制御、スレーブユニット制御を行う前に、必ずコールする必要があります。

---

## ML3\_Mst\_Close 関数

---

**機能** MECHATROLINK マスタをクローズします。

**書式** `int ML3_Mst_Close (void);`

**引数** なし

**戻り値**

ML3_ER_OK	: 正常
ML3_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

**説明** MECHATROLINK マスタをクローズします。  
本関数をコール後、MECHATROLINK マスタにアクセス不可となります。本ライブラリの使用を終了する場合は、必ずコールする必要があります。

## ML3\_Mst\_GetCommSetting 関数

**機能** MECHATROLINK マスタの通信設定を取得します。

**書式**

```
int ML3_Mst_GetCommSetting (
    TM3_SYSTEM *pSystem
);
```

**引数** `pSystem` : マスタ通信設定を格納するポインタ

## マスタ通信設定

```
typedef struct {
    unsigned long   CommMode;
    unsigned short  MaxSlave;
    unsigned long   CycleTime;
    unsigned long   IntOffset;
    unsigned long   C2MstDly;
    unsigned short  CommProtocol;
    unsigned short  MaxRetry;
    unsigned short  HostWdt;
} TM3_SYSTEM;
```

`CommMode` : 通信モード設定 (初期値: 0x8002)

CommMode (論理和)	内容
SYS_MOD_TYPE_C1MST *1	0x00000002 : 動作タイプ C1MST
SYS_MOD_INTLV_PLS	0x00000800 : INTOL 信号をパルス出力設定
SYS_MOD_ESYNC	0x00001000 : RTCIL 信号入力に同期
SYS_MOD_INT_FR *1	0x00008000 : ハード同期有効

\*1 必ず指定する必要があります。

`MaxSlave` : 最大接続スレーブ数 (初期値: 1)  
[ 1 ~ 62 ] 単位: 【局】

`CycleTime` : 伝送周期 (初期値: 50000)  
[ 3125 (31.25us) ~ 640000 (64ms) ] 単位: 【10ns】

`IntOffset` : 割込遅延時間 (初期値: 25000)  
[ 0 ~ 伝送周期設定値-500 (5us) ] 単位: 【10ns】

`C2MstDly` : C2 マスタ送信開始時間 (初期値: 0)  
[ 0 : C2 マスタ使用しない ]  
[ 1 ~ 伝送周期設定値-500 (5us) ] 単位: 【10ns】

`CommProtocol` : 通信プロトコル選択 (初期値: 0)  
[ 0 : サイクリック通信 ]  
[ 1 : イベントドリブン通信 ]

`MaxRetry` : 最大リトライ回数 (初期値: 1)  
[ 0 ~ 62 ] 単位: 【回】



---

**HostWdt** : ホスト監視用 WDT 設定 (初期値 : 16384)  
[ 0 : 機能無効 ]  
[ 1(8us) ~ 16384(131072us) ] 単位 : 【8us】

---

**戻り値** ML3\_ER\_OK : 正常  
ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

**説明** MECHATROLINK マスタの通信設定を取得します。

---

## ML3\_Mst\_StartComm 関数

---

<b>機能</b>	MECHATROLINK 通信を開始します。
<b>書式</b>	<code>int ML3_Mst_StartComm (void);</code>
<b>引数</b>	なし
<b>戻り値</b>	ML3_ER_OK : 正常 ML3_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
<b>説明</b>	MECHATROLINK 通信を開始します。 通信が正常に開始されたかどうかは、「ML3_Mst_ChkStartComm」関数で確認してください。

---

### ML3\_Mst\_ChkStartComm 関数

---

**機能** MECHATROLINK 通信開始を確認します。

**書式** int ML3\_Mst\_ChkStartComm (void);

**引数** なし

**戻り値**

ML_ER_OK	: 正常
ML_ER_NOTOPEN	: オープンされていない
ML_ER_NOTCOMM	: 通信開始未完了
ML_ER_COMMINITERR	: 通信開始失敗

**説明** MECHATROLINK 通信が開始されたかどうかを確認します。  
戻り値が「ML\_ER\_NOTCOMM」の間は、ポーリング処理してください。  
設定している全スレーブと正常に通信が確立された場合は、「ML\_ER\_OK」が返ります。  
戻り値で「ML\_ER\_COMMINITERR」が返ってきた場合は、「ML3\_Mst\_GetSystemError」を使ってエラー情報を取得してください。

## ML3\_Mst\_GetSystemError 関数

**機能** MECHATROLINK マスタのシステムエラーを取得します。

**書式**

```
int ML3_Mst_GetSystemError (
    unsigned short *pErrorCode,
    unsigned short *pChipErrorCode
);
```

**引数**

`pErrorCode` : システムエラーを格納するポインタ  
`pChipErrorCode` : チップエラーを格納するポインタ

システムエラーコード	内容
0	0 : エラーなし
MSTPROC_ERR_NOTWAIT	1 : マスタプロセス通信待機状態でない
MSTPROC_ERR_INIT	2 : マスタプロセス通信初期化エラー

チップエラーコード	内容
0	エラーなし
0x10XX	ユーザエラー
0x20XX	通信エラー
0x30XX	システムエラー

**戻り値**

ML3\_ER\_OK : 正常  
 ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

**説明**

MECHATROLINK マスタのシステムエラーを取得します。  
 「ML3\_Mst\_ChkStartComm」関数の戻り値で「ML\_ER\_COMMINITERR」が返ってきた場合のみ、エラー情報が設定されます。  
 「pChipErrorCode」の詳細はチップエラーコード一覧を参照してください。

## ML3\_Mst\_GetMasterStatus 関数

**機能** MECHATROLINK マスタのステータスを取得します。

**書式**

```
int ML3_Mst_GetMasterStatus (
    unsigned short *pMasterStatus
);
```

**引数** `pMasterStatus` : マスタステータスを格納するポインタ

マスタステータス	内容
MSTPROC_STS_INIT	0 : マスタプロセス初期状態
MSTPROC_STS_WAIT	1 : マスタプロセス通信待機状態
MSTPROC_STS_ACT	2 : マスタプロセス通信開始処理中
MSTPROC_STS_CYCL	3 : マスタプロセスサイクリック通信状態
MSTPROC_STS_EVNT	4 : マスタプロセスイベントドリブン通信状態
MSTPROC_STS_RACT	5 : マスタプロセス通信リセット処理中

**戻り値**

ML3\_ER\_OK : 正常

ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

**説明** MECHATROLINK マスタのステータスを取得します。

---

### ML3\_Mst\_ResetComm 関数

---

<b>機能</b>	MECHATROLINK 通信を停止します。
<b>書式</b>	<code>int ML3_Mst_ResetComm (void);</code>
<b>引数</b>	なし
<b>戻り値</b>	ML3_ER_OK : 正常 ML3_ER_OK 以外 : エラーコード(エラーコード一覧を参照)
<b>説明</b>	MECHATROLINK 通信を停止します。

---

### ML3\_Mst\_ChkResetComm 関数

---

**機能** MECHATROLINK 通信停止を確認します。

**書式** int ML3\_Mst\_ChkResetComm (void);

**引数** なし

**戻り値**

ML_ER_OK	: 正常
ML_ER_NOTOPEN	: オープンされていない
ML_ER_NOTRESET	: 通信停止処理中

**説明** MECHATROLINK 通信が開始されたかどうかを確認します。  
戻り値が「ML\_ER\_NOTRESET」の間は、ポーリング処理してください。  
設定している全スレーブと正常に通信が切断された場合は、「ML\_ER\_OK」が返ります。

## ML3\_Slv\_GetSlaveStatus 関数

**機能** スレーブユニットの通信状態を取得します。

**書式**

```
int ML3_Slv_GetSlaveStatus (
    WORD SlaveNo,
    TM3_SLVSTAT *pSlaveStatus
);
```

**引数**

**SlaveNo** : スレーブ番号 [ 1 ~ 62 ]

**pSlaveStatus** : スレーブ通信状態を格納するポインタ

## スレーブ通信状態

```
typedef struct {
    unsigned short SlaveStatus;
    unsigned long ErrorSts;
    unsigned char CommStatus;
} TM3_SLVSTAT;
```

**SlaveStatus** : スレーブ状態

SlaveStatus	内容
SLV_STSNUM_PON	0x0000 : 電源未投入/未接続状態
SLV_STSNUM_WAIT_PARS_MOD	0x0001 : ユーザパラメータとモード設定待ち
SLV_STSNUM_WAIT_PAR	0x0002 : ユーザパラメータ設定待ち
SLV_STSNUM_WAIT_DTCT	0x0021 : 接続局検出要求待ち
SLV_STSNUM_DETECTED	0x0022 : 接続局検出完了状態
SLV_STSNUM_AXISNUM_ERR	0x0023 : 重複アドレス検出状態
SLV_STSNUM_WAIT_MEASURE_DLY	0x0024 : 伝送遅延計測待ち
SLV_STSNUM_WAIT_DLY_APPFRM	0x0025 : 伝送遅延計測要求待ち
SLV_STSNUM_WAIT_DLY_MFRM	0x0026 : 伝送遅延計測フレーム待ち
SLV_STSNUM_WAIT_TMCFRM	0x0027 : 伝送周期通知フレーム受信待ち
SLV_STSNUM_WAIT_SYNC_START	0x0040 : サイクリック通信開始待ち
SLV_STSNUM_SYNC_COM	0x0050 : サイクリック通信中
SLV_STSNUM_ASYNC_COM	0x0060 : イベントドリブ通信中



**ErrorSts** : JL100M エラーステータス

ErrorSts	内容
SLV_ESTS_FCS_ERR	0x00000001 : データ受信中 FCS エラー発生
SLV_ESTS_SCNF_ERR	0x00000002 : 指定した通信状態に遷移していないスレーブがある
SLV_ESTS_PAR_ERR	0x00000004 : パラメータ上限値オーバーエラー
SLV_ESTS_TMCYCOVR	0x00000008 : 伝送周期内にデータ送受信未完了
SLV_ESTS_AXIS_NO_ERR	0x00000010 : 同じスレーブ局アドレスが設定されている
SLV_ESTS_ADDSTN_ERR	0x00000020 : 途中接続局検出エラー
SLV_ESTS_RXLEN_ERR	0x00000100 : 受信フレームデータ長不一致
SLV_ESTS_RFO_ERR	0x00000200 : 受信 FIFO オーバランエラー
SLV_ESTS_RUR_ERR	0x00000400 : 受信 FIFO アンダーランエラー
SLV_ESTS_RXSM_ERR	0x00000800 : 受信シーケンス異常
SLV_ESTS_WDTOVR	0x00008000 : チップ内蔵 WDT タイムオーバ
SLV_ESTS_TXTIM_OVR	0x00010000 : PHY 転送データサイズオーバ
SLV_ESTS_RXTX_ERR	0x00020000 : 受信動作中に送信発生
SLV_ESTS_TXRX_ERR	0x00040000 : 送信動作中に受信発生
SLV_ESTS_RRX_ERR	0x00080000 : 受信動作中に別系統から受信発生
SLV_ESTS_TXSM_UNDER	0x00100000 : 送信データのアンダーランエラー
SLV_ESTS_TXSM_OVER	0x00200000 : 送信データのオーバランエラー
SLV_ESTS_TFO_ERR	0x02000000 : 送信 FIFO オーバランエラー
SLV_ESTS_TUR_ERR	0x04000000 : 送信 FIFO アンダーランエラー
SLV_ESTS_TXSM_ERR	0x08000000 : 送信シーケンス異常
SLV_ESTS_TX_ERROR	0x0E3F0000 : 送信エラーマスク
SLV_ESTS_RX_ERROR	0x00000F01 : 受信エラーマスク

**CommStatus** : 通信状態

CommStatus	内容
SLV_COMMSTS_PHASE1	0 : 通信未確立
SLV_COMMSTS_PHASE2	1 : 非同期通信中
SLV_COMMSTS_PHASE3	2 : 同期通信中

**戻り値**

ML3\_ER\_OK : 正常  
 ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

**説明**

スレーブユニットの通信状態を取得します。

## ML3\_Slv\_GetSlaveCommInfo 関数

**機能** スレーブの通信設定を取得します。

**書式**

```
int ML3_Slv_GetSlaveCommInfo (
    WORD SlaveNo,
    TM3_SLVINFO *pSlaveInfo
);
```

**引数**

**SlaveNo** : スレーブ番号 [ 1 ~ 62 ]

**pSlaveInfo** : スレーブ通信設定を格納するポインタ

## スレーブ通信設定

```
typedef struct {
    unsigned short SlaveID;
    unsigned short Cd_Rd_Len;
    unsigned long ResponseTime;
    unsigned char Com_Mod;
    unsigned char Com_Tim;
    unsigned char Profile_Type;
} TM3_SLVINFO;
```

**SlaveID** : 局アドレス (初期値 : 0)

[ 0 : スレーブなし ]

拡張アドレス (上位バイト) [ 0x00 ~ 0x3D ]

局アドレス (下位バイト) [ 0x03 ~ 0xEF ]

**Cd\_Rd\_Len** : 送受信データ長 (初期値 : 48)

[ 8/16/32/48/64 ] 単位【Byte】

**ResponseTime** : 応答監視時間 (初期値 : 5000)

[ 500(5us) ~ 伝送周期設定値 ] 単位【10ns】

**Com\_Mod** : 通信モード (初期値 : 0x82)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SUBCMD	0	0	0	0	0	SYNCMODE	0

SYNCMODE:同期設定 [ 0:非同期通信 1:同期通信 ]

SUBCMD:サブコマンド設定 [ 0:サブコマンド無効 1:サブコマンド有効 ]

**Com\_Tim** : 通信周期 (通信周期 = 設定値 × 伝送周期) (初期値 : 1)

[ 1 ~ 255 ] 単位【倍】

**Profile\_Type** : プロファイルタイプ

Profile_Type	内容
SLV_PROFILETYPE_SRV	0x10 : サーボパックプロファイルタイプ (初期値)
SLV_PROFILETYPE_IO	0x30 : I/O プロファイルタイプ

**戻り値**

ML3\_ER\_OK : 正常  
ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

**説明**

スレーブの通信設定を取得します。

## 2-3 デジタル入出力ユニット関数

### ML3\_Dio\_Open 関数

---

**機能** デジタル入出力ユニットをオープンします。

**書式** int ML3\_Dio\_Open (  
    WORD SlaveNo  
);

**引数** SlaveNo : スレーブ番号 [ 1 ~ 62 ]

**戻り値** ML3\_ER\_OK : 正常  
ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

**説明** スレーブ番号で指定したデジタル入出力ユニットをオープンします。  
スレーブ番号はマスタプロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。  
この関数コール後、指定したデジタル入出力ユニットにアクセス可能となります。  
デジタル入出力ユニットを使用する際には、必ずコールする必要があります。

---

## ML3\_Dio\_Close 関数

---

**機能** デジタル入出力ユニットをクローズします。

**書式**

```
int ML3_Dio_Close (  
    WORD SlaveNo  
);
```

**引数** `SlaveNo` : スレーブ番号 [ 1 ~ 62 ]

**戻り値**

ML3_ER_OK	: 正常
ML3_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

**説明**

スレーブ番号で指定したデジタル入出力ユニットをクローズします。  
スレーブ番号はマスタプロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。  
この関数コール後は、指定したデジタル入出力ユニットにアクセス不可となります。  
アプリケーションの終了時に必ずコールする必要があります。

---

## ML3\_Dio\_Read 関数

---

**機能**

デジタル入出力ユニットから入力データを取得します。

**書式**

```
int ML3_Dio_Read(  
    WORD SlaveNo,  
    DWORD *pData  
);
```

**引数**

**SlaveNo** : スレーブ番号 [ 1 ~ 62 ]  
**pData** : 読込データ

**戻り値**

**ML3\_ER\_OK** : 正常  
**ML3\_ER\_OK 以外** : エラーコード(エラーコード一覧を参照)

**説明**

スレーブ番号で指定したデジタル入出力ユニットから入力データを取得します。  
スレーブ番号はマスタープロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。

---

## ML3\_Dio\_Write 関数

---

**機能**

デジタル入出力ユニットの出力にデータを出力します。

**書式**

```
int ML3_Dio_Write (  
    WORD SlaveNo,  
    DWORD Data  
);
```

**引数**

**SlaveNo** : スレーブ番号 [ 1 ~ 62 ]  
**Data** : 書込データ

**戻り値**

ML3\_ER\_OK : 正常  
ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

**説明**

スレーブ番号で指定したデジタル入出力ユニットへデータを出力します。  
スレーブ番号はマスタプロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。。

## ML3\_Dio\_GetParam 関数

**機能** デジタル入出力ユニットの設定パラメータを取得します。

**書式**

```
int ML3_Dio_GetParam(
    WORD SlaveNo,
    TML3_DIO_PARAM *pParam
);
```

**引数**

**SlaveNo** : スレーブ番号 [ 1 ~ 62 ]  
**pParam** : DIO 設定パラメータを格納するポインタ

## DIO 設定パラメータ

```
typedef struct {
    WORD HoldClear;
    WORD Filter;
} TML3_DIO_PARAM, *PTML3_DIO_PARAM;
```

**HoldClear** : 異常時出力設定

HoldClear	内容
DIO_DO_ERRROUTPUT_HOLD	0: 出力保持
DIO_DO_ERRROUTPUT_CLEAR	1: 出力クリア

**Filter** : 入力フィルタ設定

Filter	内容
DIO_DI_FILTER_TYPE_0MS	0: フィルタなし
DIO_DI_FILTER_TYPE_05MS	1: 0.5ms
DIO_DI_FILTER_TYPE_1MS	2: 1ms
DIO_DI_FILTER_TYPE_2MS	3: 2ms
DIO_DI_FILTER_TYPE_4MS	4: 4ms
DIO_DI_FILTER_TYPE_8MS	5: 8ms
DIO_DI_FILTER_TYPE_16MS	6: 16ms
DIO_DI_FILTER_TYPE_32MS	7: 32ms

**戻り値**

ML3\_ER\_OK : 正常  
 ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

**説明**

スレーブ番号で指定したデジタル入出力ユニットのDIO設定パラメータを取得します。スレーブ番号はマスタプロセス設定ファイル(MLMstSetting.INI)で設定したスレーブ番号を指定する必要があります。入力フィルタはDI、DIOユニット、異常時出力はDO、DIOユニットの時のみ、それぞれ有効となります。



## ML3\_Dio\_SetParam 関数

**機能** デジタル入出力ユニットの設定パラメータを設定します。

**書式**

```
int ML3_Dio_SetParam (
    WORD SlaveNo,
    TML3_DIO_PARAM *pParam
);
```

**引数**

**SlaveNo** : スレーブ番号 [ 1 ~ 62 ]

**pParam** : DIO 設定パラメータを格納するポインタ

## DIO 設定パラメータ

```
typedef struct {
    WORD HoldClear;
    WORD Filter;
} TML3_DIO_PARAM, *PTML3_DIO_PARAM;
```

**HoldClear** : 異常時出力設定

HoldClear	内容
DIO_DO_ERRROUTPUT_HOLD	0: 出力保持
DIO_DO_ERRROUTPUT_CLEAR	1: 出力クリア

**Filter** : 入力フィルタ設定

Filter	内容
DIO_DI_FILTER_TYPE_0MS	0: フィルタなし
DIO_DI_FILTER_TYPE_05MS	1: 0.5ms
DIO_DI_FILTER_TYPE_1MS	2: 1ms
DIO_DI_FILTER_TYPE_2MS	3: 2ms
DIO_DI_FILTER_TYPE_4MS	4: 4ms
DIO_DI_FILTER_TYPE_8MS	5: 8ms
DIO_DI_FILTER_TYPE_16MS	6: 16ms
DIO_DI_FILTER_TYPE_32MS	7: 32ms

**戻り値**

ML3\_ER\_OK : 正常

ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

**説明**

スレーブ番号で指定したデジタル入出力ユニットのDIO設定パラメータを設定します。スレーブ番号はマスタプロセス設定ファイル(MLMstSetting.INI)で設定したスレーブ番号を指定する必要があります。入力フィルタはDI、DIOユニット、異常時出力はDO、DIOユニットの時のみ、それぞれ有効となります。

## 2-4 アナログ入出力ユニット関数

### ML3\_Aio\_Open 関数

---

**機能**

アナログ入出力ユニットをオープンします。

**書式**

```
int ML3_Aio_Open (  
    WORD SlaveNo  
);
```

**引数**

**SlaveNo** : スレーブ番号 [ 1 ~ 62 ]

**戻り値**

ML3\_ER\_OK : 正常  
ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

**説明**

スレーブ番号で指定したアナログ入出力ユニットをオープンします。  
スレーブ番号はマスタプロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。  
この関数コール後、指定したアナログ入出力ユニットにアクセス可能となります。  
アナログ入出力ユニットを使用する際には、必ずコールする必要があります。

---

## ML3\_Aio\_Close 関数

---

**機能** アナログ入出力ユニットをクローズします。

**書式**

```
int ML3_Aio_Close (  
    WORD SlaveNo  
);
```

**引数** `SlaveNo` : スレーブ番号 [ 1 ~ 62 ]

**戻り値**

ML3_ER_OK	: 正常
ML3_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

**説明**

スレーブ番号で指定したアナログ入出力ユニットをクローズします。  
スレーブ番号はマスタプロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。  
この関数コール後は、指定したアナログ入出力ユニットにアクセス不可になります。  
アプリケーションの終了時に必ずコールする必要があります。

---

## ML3\_Aio\_Read 関数

---

**機能** アナログ入出力ユニットから 16 ビットの AD データを取得します。

**書式**

```
int ML3_Aio_Read (  
    WORD SlaveNo,  
    WORD Ch,  
    TML3_AIO_READ *pRead  
);
```

**引数**

SlaveNo	: スレーブ番号 [ 1 ~ 62 ]
Ch	: チャンネル番号 [ 1 ~ 4 ]
pRead	: AD データを格納するポインタ

---

### AD データ

---

```
typedef struct {  
    WORD Data;  
    WORD Detect;  
} TML3_AIO_READ, *PTML3_AIO_READ;
```

Data : アナログ入力データ  
Detect : 断線検出 [0: 断線なし、1: 断線あり]

---

**戻り値**

ML3_ER_OK	: 正常
ML3_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

**説明** スレーブ番号で指定したアナログ入出力ユニットの指定した AD チャンネルから 16 ビットデータと断線検出情報を取得します。  
スレーブ番号はマスタープロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。  
本関数はモードが「サンプリングモード」時のみ有効です。  
モードに関しては ML3\_Aio\_AD\_SetMode() を参照してください。

---

## ML3\_Aio\_Write 関数

---

**機能** アナログ入出力ユニットへ 16 ビットの DA データを出力します。

**書式**

```
int ML3_Aio_Write (  
    WORD SlaveNo,  
    DWORD Ch,  
    TML3_AIO_WRITE *pWrite  
);
```

**引数**

SlaveNo	: スレーブ番号 [ 1 ~ 62 ]
Ch	: チャンネル番号 [ 1 ~ 4 ]
pWrite	: DA データを格納するポインタ

### DA データ

---

```
typedef struct {  
    WORD Data;  
} TML3_AIO_WRITE, *PTML3_AIO_WRITE;
```

**Data** : アナログ出力データ

---

**戻り値**

ML3_ER_OK	: 正常
ML3_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

**説明**

スレーブ番号で指定したアナログ入出力ユニットの指定した DA チャンネルに 16 ビットデータを出力します。

スレーブ番号はマスタプロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。

本関数はモードが「サンプリングモード」時のみ有効です。

モードに関しては ML3\_Aio\_AD\_SetMode() を参照してください。

## ML3\_Aio\_AD\_GetParam 関数

**機能**

アナログ入出力ユニットの AD チャンネルの設定値を取得します。

**書式**

```
int ML3_Aio_AD_GetParam(
    WORD SlaveNo,
    DWORD Ch,
    TML3_AIO_ADGETPARAM *pParam
);
```

**引数**

**SlaveNo** : スレーブ番号 [ 1 ~ 62 ]  
**Ch** : チャンネル番号 [ 1 ~ 4 ]  
**pParam** : AD 設定パラメータを格納するポインタ

## AD 設定パラメータ

```
typedef struct {
    WORD Mode;
    WORD Filter;
    WORD MinMaxSw;
    WORD SamplingNum;
    WORD CalibState;
} TML3_AIO_ADGETPARAM, *PTML3_AIO_ADGETPARAM;
```

**Mode** : モード

Mode	内容
AIO_AD_MODE_SAMPLING	0 : サンプリングモード (初期値)
AIO_AD_MODE_CONFIG	1 : 設定モード
AIO_AD_MODE_CALIBRATION	2 : キャリブレーションモード
AIO_AD_MODE_DISABLE_CH	3 : チャンネル無効

**Filter** : フィルタタイプ

Filter	内容
AIO_AD_FILTYPE_DISABLE	0 : フィルタ なし (初期値)
AIO_AD_FILTYPE_SIMPLE	1 : フィルタ 単純平均
AIO_AD_FILTYPE_MOVEAVG	2 : フィルタ 移動平均

**MinMaxSw** : 最大/最小値除外

MinMaxSw	内容
AIO_AD_ENABLE_MINMAX	0 : 最大/最小値除外しない (初期値)
AIO_AD_DISABLE_MINMAX	1 : 最大/最小値除外する

**SamplingNum** : サンプル回数 [ 0 ~ 6 ]  
2 の (SamplingNum + 1) で設定される (初期値=0)

**CalibState** : キャリブレーション

CalibState	内容
AIO_AD_END_CALIBRATION	0 : キャリブレーション終了 (初期値)
AIO_AD_MIN_CALIBRATION	1 : 最小値登録
AIO_AD_MAX_CALIBRATION	2 : 最大値登録

#### 戻り値

ML3\_ER\_OK : 正常  
ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

#### 説明

スレーブ番号で指定したアナログ入出力ユニットの指定した AD チャンネルから現在の AD 設定パラメータを取得します。  
スレーブ番号はマスタプロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。

## ML3\_Aio\_AD\_SetMode 関数

**機能** アナログ入出力ユニットの AD チャンネルのモードを変更します。

**書式**

```
int ML3_Aio_AD_SetMode (
    WORD SlaveNo,
    WORD Ch,
    WORD Mode
);
```

**引数**

**SlaveNo** : スレーブ番号 [ 1 ~ 62 ]  
**Ch** : チャンネル番号 [ 1 ~ 4 ]  
**Mode** : モード

## モード

Mode	内容
AIO_AD_MODE_SAMPLING	0 : サンプリングモード (初期値)
AIO_AD_MODE_CONFIG	1 : 設定モード
AIO_AD_MODE_CALIBRATION	2 : キャリブレーションモード
AIO_AD_MODE_DISABLE_CH	3 : チャンネル無効

**戻り値**

ML3\_ER\_OK : 正常  
 ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

**説明**

スレーブ番号で指定したアナログ入出力ユニットの指定した AD チャンネルの現在のモードを変更します。  
 スレーブ番号はマスタプロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。



## ML3\_Aio\_AD\_SetParam 関数

**機能** アナログ入出力ユニットの AD に関するパラメータ設定を行います。

**書式**

```
int ML3_Aio_AD_SetParam (
    WORD SlaveNo,
    WORD Ch,
    TML3_AIO_ADSETPARAM *pParam
);
```

**引数**

**SlaveNo** : スレーブ番号 [ 1 ~ 62 ]  
**Ch** : チャンネル番号 [ 1 ~ 4 ]  
**pParam** : AD 設定パラメータを格納するポインタ

## AD 設定パラメータ

```
typedef struct {
    WORD Filter;
    WORD MinMaxSw;
    WORD SamplingNum;
} TML3_AIO_ADSETPARAM, *PTML3_AIO_ADSETPARAM;
```

**Filter** : フィルタタイプ

Filter	内容
AIO_AD_FILTYPE_DISABLE	0 : フィルタ なし (初期値)
AIO_AD_FILTYPE_SIMPLE	1 : フィルタ 単純平均
AIO_AD_FILTYPE_MOVEAVG	2 : フィルタ 移動平均

**MinMaxSw** : 最大/最小値除外

MinMaxSw	内容
AIO_AD_ENABLE_MINMAX	0 : 最大/最小値除外しない (初期値)
AIO_AD_DISABLE_MINMAX	1 : 最大/最小値除外する

**SamplingNum** : サンプル回数 [ 0 ~ 6 ]  
 2 の (SamplingNum + 1) 乗で設定される (初期値=0)

**戻り値**

ML3\_ER\_OK : 正常  
 ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

**説明**

スレーブ番号で指定したアナログ入出力ユニットの AD 設定パラメータの設定を行います。スレーブ番号はマスタプロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。

本関数はモードが「設定モード」時のみ有効です。  
 モードに関しては ML3\_Aio\_AD\_SetMode() を参照してください。

## ML3\_Aio\_AD\_SetCalib 関数

**機能** アナログ入出力ユニットの AD のキャリブレーションを設定します。

**書式**

```
int ML3_Aio_AD_SetCalib (
    WORD SlaveNo,
    WORD Ch,
    WORD CalibState
);
```

**引数**

**SlaveNo** : スレーブ番号 [ 1 ~ 62 ]  
**Ch** : チャンネル番号 [ 1 ~ 4 ]  
**CalibState** : キャリブレーション

## キャリブレーション

CalibState	内容
AIO_AD_END_CALIBRATION	0 : キャリブレーション終了 (初期値)
AIO_AD_MIN_CALIBRATION	1 : 最小値登録
AIO_AD_MAX_CALIBRATION	2 : 最大値登録

**戻り値**

ML3\_ER\_OK : 正常  
 ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

**説明**

スレーブ番号で指定したアナログ入出力ユニットのキャリブレーション設定を行います。スレーブ番号はマスタプロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。  
 本関数はモードが「キャリブレーションモード」時のみ有効です。  
 モードに関しては ML3\_Aio\_AD\_SetMode() を参照してください。

本関数を呼び出した際、その時に入力されている AD 値を引数で指定された最小値/最大値の値として登録します。  
 キャリブレーション終了後は引数を「キャリブレーション終了」として本関数を実行してください。

## ML3\_Aio\_DA\_GetParam 関数

**機能** アナログ入出力ユニットの DA チャンネルの設定値を取得します。

**書式**

```
int ML3_Aio_DA_GetParam (
    WORD SlaveNo,
    WORD Ch,
    TML3_AIO_DAGETPARAM *pParam
);
```

**引数**

**SlaveNo** : スレーブ番号 [ 1 ~ 62 ]  
**Ch** : チャンネル番号 [ 1 ~ 4 ]  
**pParam** : DA 設定パラメータを格納するポインタ

## DA 設定パラメータ

```
typedef struct {
    WORD Mode,
    WORD ErrorOutput,
    WORD ErrorOutputData,
    WORD CalibState
} TML3_AIO_DAGETPARAM, *PTML3_AIO_DAGETPARAM;
```

**Mode** : モード

Mode	内容
AIO_DA_MODE_OUTPUT	0 : D/A 出力モード (初期値)
AIO_DA_MODE_CONFIG	1 : 設定モード
AIO_DA_MODE_CALIBRATION	2 : キャリブレーションモード
AIO_DA_MODE_DISABLE_CH	3 : チャンネル無効

**ErrorOutput** : 異常時出力

ErrorOutput	内容
AIO_DA_ERROUTPUT_0	0 : 0V 出力 (初期値)
AIO_DA_ERROUTPUT_KEEP	1 : 出力データ保持
AIO_DA_ERROUTPUT_LOWLIMIT	2 : LowLimit データ出力
AIO_DA_ERROUTPUT_HIGHLIMIT	3 : HighLimit データ出力
AIO_DA_ERROUTPUT_USERNUM	4 : ユーザー設定データ出力

**ErrorOutputData** : 異常時出力データ

ErrorOutputData	内容
AIO_DA_ERROUTPUT_RANGE_MIN	0x0000 : 最小設定 (初期値)
AIO_DA_ERROUTPUT_RANGE_MAX	0xFFFF : 最大設定

**CalibState** : キャリブレーション

CalibState	内容
AIO_DA_END_CALIBRATION	0 : キャリブレーション終了 (初期値)
AIO_DA_MIN_CALIBRATION	1 : 最小値登録
AIO_DA_MAX_CALIBRATION	2 : 最大値登録

**戻り値**

ML3\_ER\_OK : 正常  
ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

**説明**

スレーブ番号で指定したアナログ入出力ユニットの指定した DA チャンネルから現在の DA 設定パラメータを取得します。  
スレーブ番号はマスタプロセス設定ファイル (MLMstSetting. INI) で設定したスレーブ番号を指定する必要があります。

## ML3\_Aio\_DA\_SetMode 関数

**機能** アナログ入出力ユニットの DA チャンネルのモードを変更します。

**書式**

```
int ML3_Aio_DA_SetMode (
    WORD SlaveNo,
    WORD Ch,
    WORD Mode
);
```

**引数**

**SlaveNo** : スレーブ番号 [ 1 ~ 62 ]  
**Ch** : チャンネル番号 [ 1 ~ 4 ]  
**Mode** : モード

## モード

Mode	内容
AIO_DA_MODE_OUTPUT	0 : D/A 出力モード (初期値)
AIO_DA_MODE_CONFIG	1 : 設定モード
AIO_DA_MODE_CALIBRATION	2 : キャリブレーションモード
AIO_DA_MODE_DISABLE_CH	3 : チャンネル無効

**戻り値**

ML3\_ER\_OK : 正常  
 ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

**説明**

スレーブ番号で指定したアナログ入出力ユニットの指定した DA チャンネルの現在のモードを変更します。  
 スレーブ番号はマスタプロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。

## ML3\_Aio\_DA\_SetParam 関数

**機能** アナログ入出力ユニットの DA に関するパラメータ設定を行います。

**書式**

```
int ML3_Aio_DA_SetParam (
    WORD SlaveNo,
    WORD Ch,
    TML3_AIO_DASETPARAM *pParam
);
```

**引数**

**SlaveNo** : スレーブ番号 [ 1 ~ 62 ]  
**Ch** : チャンネル番号 [ 1 ~ 4 ]  
**pParam** : DA 設定パラメータを格納するポインタ

## DA 設定パラメータ

```
typedef struct {
    WORD ErrorOutput,
    WORD ErrorOutputData
} TML3_AIO_DASETPARAM, *PTML3_AIO_DASETPARAM;
```

**ErrorOutput** : 異常時出力

ErrorOutput	内容
AIO_DA_ERROUTPUT_0	0 : 0V 出力 (初期値)
AIO_DA_ERROUTPUT_KEEP	1 : 出力データ保持
AIO_DA_ERROUTPUT_LOWLIMIT	2 : LowLimit データ出力
AIO_DA_ERROUTPUT_HIGHLIMIT	3 : HighLimit データ出力
AIO_DA_ERROUTPUT_USERNUM	4 : ユーザー設定データ出力

**ErrorOutputData** : 異常時出力データ

ErrorOutputData	内容
AIO_DA_ERROUTPUT_RANGE_MIN	0x0000 : 最小設定 (初期値)
AIO_DA_ERROUTPUT_RANGE_MAX	0xFFFF : 最大設定

**戻り値**

ML3\_ER\_OK : 正常  
 ML3\_ER\_OK 以外 : エラーコード (エラーコード一覧を参照)

**説明**

スレーブ番号で指定したアナログ入出力ユニットの DA に関する DA 設定パラメータの設定を行います。スレーブ番号はマスタプロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。

本関数はモードが「設定モード」時のみ有効です。

モードに関しては ML3\_Aio\_DA\_SetMode() を参照してください。

異常時出力データは異常時出力の設定がユーザー設定データ出力の時のみ有効です。

## ML3\_Aio\_DA\_SetCalib 関数

**機能** アナログ入出力ユニットの DA のキャリブレーションを設定します。

**書式**

```
int ML3_Aio_DA_SetCalib (
    WORD SlaveNo,
    WORD Ch,
    WORD CalibState
);
```

**引数**

**SlaveNo** : スレーブ番号 [ 1 ~ 62 ]  
**Ch** : チャンネル番号 [ 1 ~ 4 ]  
**CalibState** : キャリブレーション

## キャリブレーション

CalibState	内容
AIO_DA_END_CALIBRATION	0 : キャリブレーション終了 (初期値)
AIO_DA_MIN_CALIBRATION	1 : 最小値登録
AIO_DA_MAX_CALIBRATION	2 : 最大値登録

**戻り値**

ML3\_ER\_OK : 正常  
 ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

**説明**

スレーブ番号で指定したアナログ入出力ユニットのキャリブレーション設定を行います。スレーブ番号はマスタプロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。  
 本関数はモードが「キャリブレーションモード」時のみ有効です。  
 モードに関しては ML3\_Aio\_DA\_SetMode() を参照してください。

本関数を呼び出した際、その時に出力している DA 値を引数で指定された最小値/最大値の値として登録します。  
 キャリブレーション終了後は引数を「キャリブレーション終了」として本関数を実行してください。

## 2-5 エンコーダユニット関数

### ML3\_Enc\_Open 関数

---

**機能** エンコーダユニットをオープンします。

**書式** int ML3\_Enc\_Open (  
    WORD SlaveNo  
);

**引数** SlaveNo : スレーブ番号 [ 1 ~ 62 ]

**戻り値** ML3\_ER\_OK : 正常  
ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

**説明** スレーブ番号で指定したエンコーダユニットをオープンします。  
スレーブ番号はマスタプロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。  
この関数コール後、指定したエンコーダユニットにアクセス可能となります。  
エンコーダユニットを使用する際には、必ずコールする必要があります。



---

## ML3\_Enc\_Close 関数

---

**機能** エンコーダユニットをクローズします

**書式**

```
int ML3_Enc_Close (  
    WORD SlaveNo  
);
```

**引数** `SlaveNo` : スレーブ番号 [ 1 ~ 62 ]

**戻り値**

ML3_ER_OK	: 正常
ML3_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

**説明**

スレーブ番号で指定したエンコーダユニットをクローズします。  
スレーブ番号はマスタプロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。  
この関数コール後は、指定したエンコーダユニットにアクセス不可になります。  
アプリケーションの終了時に必ずコールする必要があります。

---

## ML3\_Enc\_Read 関数

---

**機能** エンコーダユニットから入力データを取得します。

**書式**

```
int ML3_Enc_Read (  
    WORD SlaveNo,  
    WORD Ch,  
    TML3_ENC_READ *pRead  
);
```

**引数**

SlaveNo	: スレーブ番号 [ 1 ~ 62 ]
Ch	: チャンネル番号 [ 1 ~ 2 ]
pRead	: ENC 入力データを格納するポインタ

---

### ENC 入力データ

---

```
typedef struct {  
    WORD    Status;  
    DWORD   CurCount;  
    DWORD   Latch_A;  
    DWORD   Latch_B;  
} TML3_ENC_READ, *PTML3_ENC_READ;
```

Status : ステータス (詳細は次ページ参照)  
CurCount : カウンタ現在値  
Latch\_A : ラッチ A 値  
Latch\_B : ラッチ B 値

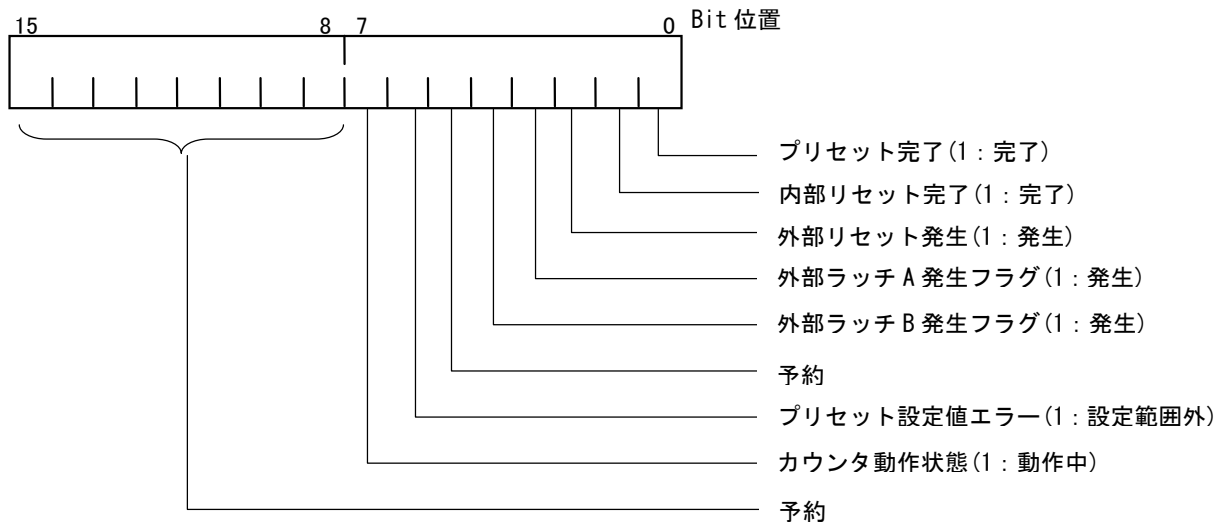
---

**戻り値**

ML3_ER_OK	: 正常
ML3_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

**説明** スレーブ番号で指定したエンコーダユニットから入力データを取得します。  
スレーブ番号はマスタプロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。

ステータスの意味(下図参照)



---

## ML3\_Enc\_Write 関数

---

**機能** エンコーダユニットヘデータを出力します。

**書式**

```
int ML3_Enc_Write (  
    WORD SlaveNo,  
    DWORD Ch,  
    TML3_ENC_WRITE *pWrite  
);
```

**引数**

SlaveNo	: スレーブ番号 [ 1 ~ 62 ]
Ch	: チャンネル番号 [ 1 ~ 2 ]
pWrite	: ENC 出力データを格納するポインタ

---

### ENC 出力データ

---

```
typedef struct {  
    WORD SoftSw;  
    DWORD Preset;  
} TML3_ENC_WRITE, *PTML3_ENC_WRITE;
```

SoftSw : ソフト SW (詳細は次ページ参照)  
Preset : プリセット値

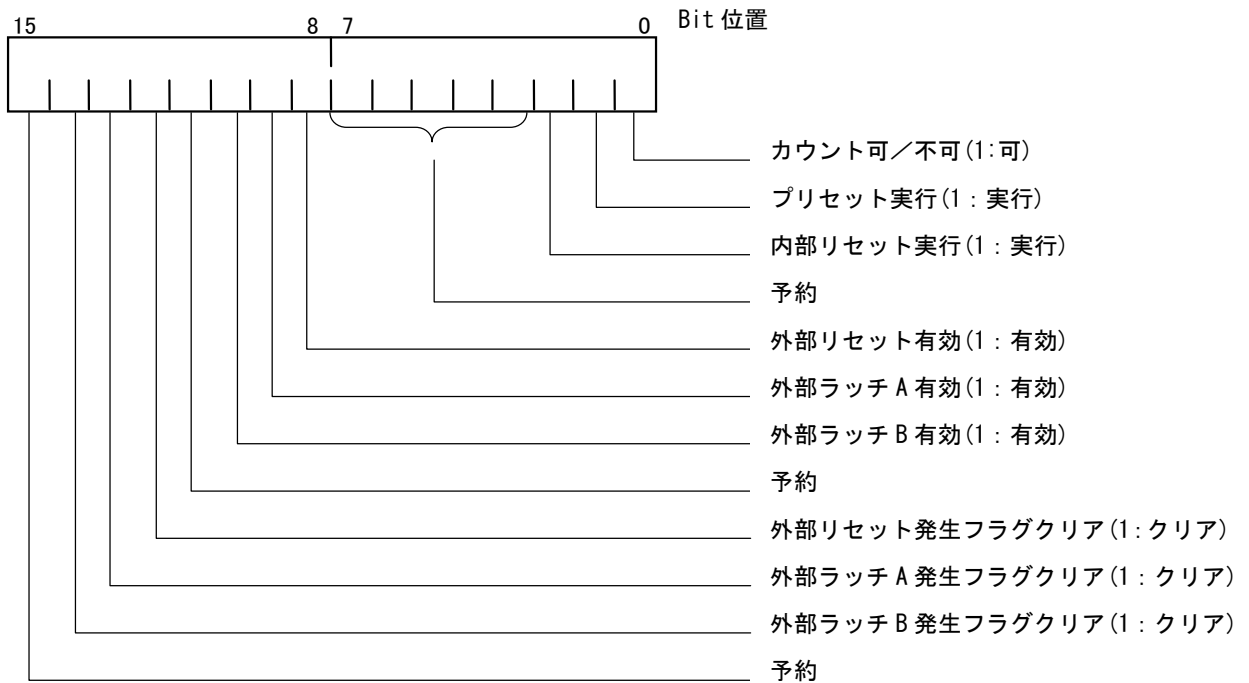
---

**戻り値**

ML3_ER_OK	: 正常
ML3_ER_OK 以外	: エラーコード(エラーコード一覧を参照)

**説明** スレーブ番号で指定したエンコーダユニットヘデータを出力します。  
スレーブ番号はマスタプロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。

ソフト SW の意味(下図参照)



## ML3\_Enc\_GetParam 関数

**機能** エンコーダユニットの設定パラメータを取得します。

**書式**

```
int ML3_Enc_GetParam (
    WORD SlaveNo,
    WORD Ch,
    TML3_ENC_PARAM *pParam
);
```

**引数**

**SlaveNo** : スレーブ番号 [ 1 ~ 62 ]  
**Ch** : チャンネル番号 [ 1 ~ 2 ]  
**pParam** : ENC 設定パラメータを格納するポインタ

## ENC 設定パラメータ

```
typedef struct {
    WORD Direction;
    WORD InputType;
    DWORD MaxCount;
} TML3_ENC_PARAM, *PTML3_ENC_PARAM;
```

**Direction** : 回転方向

Direction	内容
ENC_RND_CW	0 : CW (初期値)
ENC_RND_CCW	1 : CCW

**InputType** : 入力方式 (逡倍)

InputType	内容
IN_TYPE_1TMS	0 : A-B 相位相パルス 1 逡倍 (初期値)
IN_TYPE_2TMS	1 : A-B 相位相パルス 2 逡倍
IN_TYPE_4TMS	2 : A-B 相位相パルス 4 逡倍
IN_TYPE_PLS	3 : 符号+パルス
IN_TYPE_PLMN	4 : 加算・減算

**MaxCount** : リングカウンタ最大値 [ 0 ~ 4294967295 ]

**戻り値**

ML3\_ER\_OK : 正常  
 ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

**説明**

スレーブ番号で指定したエンコーダユニットから設定パラメータを取得します。  
 スレーブ番号はマスタプロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。

## ML3\_Enc\_SetParam 関数

**機能** エンコーダユニットの設定パラメータを設定します。

**書式**

```
int ML3_Enc_SetParam (
    WORD SlaveNo,
    WORD Ch,
    TML3_ENC_PARAM *pParam
);
```

**引数**

**SlaveNo** : スレーブ番号 [ 1 ~ 62 ]  
**Ch** : チャンネル番号 [ 1 ~ 2 ]  
**pParam** : ENC 設定パラメータを格納するポインタ

## ENC 設定パラメータ

```
typedef struct {
    WORD Direction;
    WORD InputType;
    DWORD MaxCount;
} TML3_ENC_PARAM, *PTML3_ENC_PARAM;
```

**Direction** : 回転方向

Direction	内容
ENC_RND_CW	0 : CW (初期値)
ENC_RND_CCW	1 : CCW

**InputType** : 入力方式 (逡倍)

InputType	内容
IN_TYPE_1TMS	0 : A-B 相位相パルス 1 逡倍 (初期値)
IN_TYPE_2TMS	1 : A-B 相位相パルス 2 逡倍
IN_TYPE_4TMS	2 : A-B 相位相パルス 4 逡倍
IN_TYPE_PLS	3 : 符号+パルス
IN_TYPE_PLMN	4 : 加算・減算

**MaxCount** : リングカウンタ最大値 [ 0 ~ 4294967295 ]

**戻り値**

ML3\_ER\_OK : 正常  
 ML3\_ER\_OK 以外 : エラーコード(エラーコード一覧を参照)

**説明**

スレーブ番号で指定したエンコーダユニットの設定パラメータを設定します。  
 スレーブ番号はマスタプロセス設定ファイル (MLMstSetting.INI) で設定したスレーブ番号を指定する必要があります。

## 2-6 付録

### 2-6-1 エラーコード

#### 関数戻り値エラーコード一覧

表 2-6-1-1. 関数戻り値エラーコード一覧

エラーコード定義名	エラーコード	内容
ML3_ER_OK	0x0000	正常です。
ML3_ER_ALREADYOPEN	0x0001	すでにオープンしています。
ML3_ER_NOMSTPROC	0x0002	MECHATROLINK-III マスタプロセスが起動していません。
ML3_ER_INVALIDPARAM	0x0003	無効な引数です。
ML3_ER_NOTOPEN	0x0004	オープンしていません。
ML3_ER_ALREADYCOMM	0x0005	すでに通信開始されています。
ML3_ER_NOTCOMM	0x0006	通信していません。
ML3_ER_NOTRESET	0x0007	リセットしていません。
ML3_ER_NOTSYNC	0x0008	同期通信していません。
ML3_ER_COMMINITERR	0x0009	通信初期化エラーです。
ML3_ER_NOSETCMD	0x000A	コマンドセットしていません。
ML3_ER_NORESPONSE	0x000B	レスポンスなしです。
ML3_ER_ERROR	0x0101	内部エラーです。
ML3_ER_CREATE	0x0102	各種デバイス生成失敗です。
ML3_ER_TIMEOUT	0x0103	タイムアウトエラーです。
ML3_ER_LOADDEVICE	0x0201	デバイスドライバロードエラーです。
ML3_ER_PROFILETYPE	0x0202	プロファイルタイプが異なります。
ML3_ER_UNITTYPE	0x0203	ユニットタイプが異なります。



## チップエラーコード一覧

表 2-6-1-2. チップエラーコード一覧

エラー種別 (上位 4bit)	エラーコード (下位 12bit)	内容	エラー種別
0×1	0×00	ユーザーパラメータの設定が誤っています。	ユーザー
0×1	0×04	指定された offset, size が RAM の領域を越えています。	ユーザー
0×1	0×05	ホストインタフェースのアドレス割り付けが設定範囲外です。	ユーザー
0×1	0×06	RAM 容量が不足しています。	ユーザー
0×1	0×07	指定された伝送周期が設定範囲外です。	ユーザー
0×1	0×08	RAM 容量が不足しています。	ユーザー
0×1	0×0B	本関数を実行する前に必要な処理が行われていません。	ユーザー
0×1	0×0C	スレーブの軸番号設定が誤っています。	ユーザー
0×1	0×1E	設定範囲外の局番号がセットされています。	ユーザー
0×1	0×20	設定範囲外のサイズが指定されています。	ユーザー
0×2	0×11	処理中に送信エラーが発生しました。	通信
0×3	0×02	通信パラメータの設定処理が正常に完了しませんでした。	システム
0×3	0×03	通信モードの設定処理が正常に完了しませんでした。	システム
0×3	0×04	通信パラメータの設定処理が正常に完了しませんでした。	システム
0×3	0×08	同期通信開始処理が正常に完了しませんでした。	システム
0×3	0×09	非同期通信開始処理が正常に完了しませんでした。	システム
0×3	0×0A	マスタより指定した通信モードで通信可能な状態に遷移できないスレーブがあります。	システム
0×3	0×0E	マイクロコードのダウンロードが正常に完了しませんでした。	システム
0×3	0×1A	応答データバッファの切り替え処理中にタイムアウトが発生しました。	システム
0×3	0×1C	指令データバッファの切り替え処理中にタイムアウトが発生しました。	システム

## 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。

76DLH0051B  
76DLH0051A

2014年 10月 第2版  
2012年 6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/>