

リファレンスマニュアル

MLMs t 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 スレーブアクセス関数 | 2-14 |
| 2-4 簡易補間移動関数 | 2-42 |
| 2-5 共通コマンド | 2-47 |
| 2-6 標準サーボプロファイルコマンド | 2-64 |
| 2-7 標準I/Oコマンド | 2-88 |
| 2-8 サブコマンド | 2-90 |
| 2-9 モニタ情報 | 2-100 |
| 2-10 アラーム情報 | 2-101 |
| 2-11 エラーコード一覧 | 2-103 |

第 1 章 関数一覧

1) マスタアクセス関数

| 関 数 | 機 能 |
|--------------------------|------------------|
| MLMst_Open() | マスタをオープンします。 |
| MLMst_Close() | マスタをクローズします。 |
| MLMst_SetCommSetting() | マスタの通信設定を設定します。 |
| MLMst_GetCommSetting() | マスタの通信設定を取得します。 |
| MLMst_StartComm() | マスタの通信を開始します。 |
| MLMst_ChkStartComm() | マスタの通信開始を確認します。 |
| MLMst_ResetComm() | マスタの通信を停止します。 |
| MLMst_ChkResetComm() | マスタの通信停止を確認します。 |
| MLMst_GetSystemError() | マスタのエラー情報を取得します。 |
| MLMst_GetMasterStatus() | マスタの通信状態を取得します。 |

2) スレーブアクセス関数

| 関 数 | 機 能 |
|---------------------------|--------------------------------|
| MLSlv_GetSlaveStatus() | スレーブの通信状態を取得します。 |
| MLSlv_SetSlaveCommInfo() | スレーブの通信設定を設定します。 |
| MLSlv_GetSlaveCommInfo() | スレーブの通信設定を取得します。 |
| MLSlv_SetCmd() | スレーブにコマンドとパラメータを一括設定します。 |
| MLSlv_ChgCmd() | スレーブのコマンドのみを変更します。 |
| MLSlv_GetCmd() | 現在実行中のスレーブのコマンドとパラメータを取得します。 |
| MLSlv_GetCmdPara() | スレーブの指定したコマンドのパラメータを取得します。 |
| MLSlv_SetSubCmd() | スレーブにサブコマンドとパラメータを一括設定します。 |
| MLSlv_ChgSubCmd() | スレーブのサブコマンドのみを変更します。 |
| MLSlv_GetSubCmd() | 現在実行中のスレーブのサブコマンドとパラメータを取得します。 |
| MLSlv_GetSubCmdPara() | スレーブの指定したサブコマンドのパラメータを取得します。 |
| MLSlv_ChkRes() | スレーブのレスポンスを確認します。 |
| MLSlv_GetRes() | スレーブのレスポンスとパラメータを取得します。 |
| MLSlv_GetResPara() | スレーブの指定したレスポンスのパラメータを取得します。 |
| MLSlv_ChkSubRes() | スレーブのサブレスポンスを確認します。 |
| MLSlv_GetSubRes() | スレーブのサブレスポンスとパラメータを取得します。 |
| MLSlv_GetSubResPara() | スレーブの指定したサブレスポンスのパラメータを取得します。 |

3) 簡易補間移動用関数

| 関 数 | 機 能 |
|-------------------|----------------------------|
| MLSlv_IpdtSet() | 補間移動を行う各軸のパラメータを設定します。 |
| MLSlv_IpStrt() | 補間送り運転を開始します。 |
| MLSlv_IpStop() | 補間送り運転を停止します。 |
| MLSlv_IpNumGet() | 補間送りバッファの残数を取得します。 |
| MLSlv_IpdrClr() | IpdtSet で登録したパラメータをクリアします。 |

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。

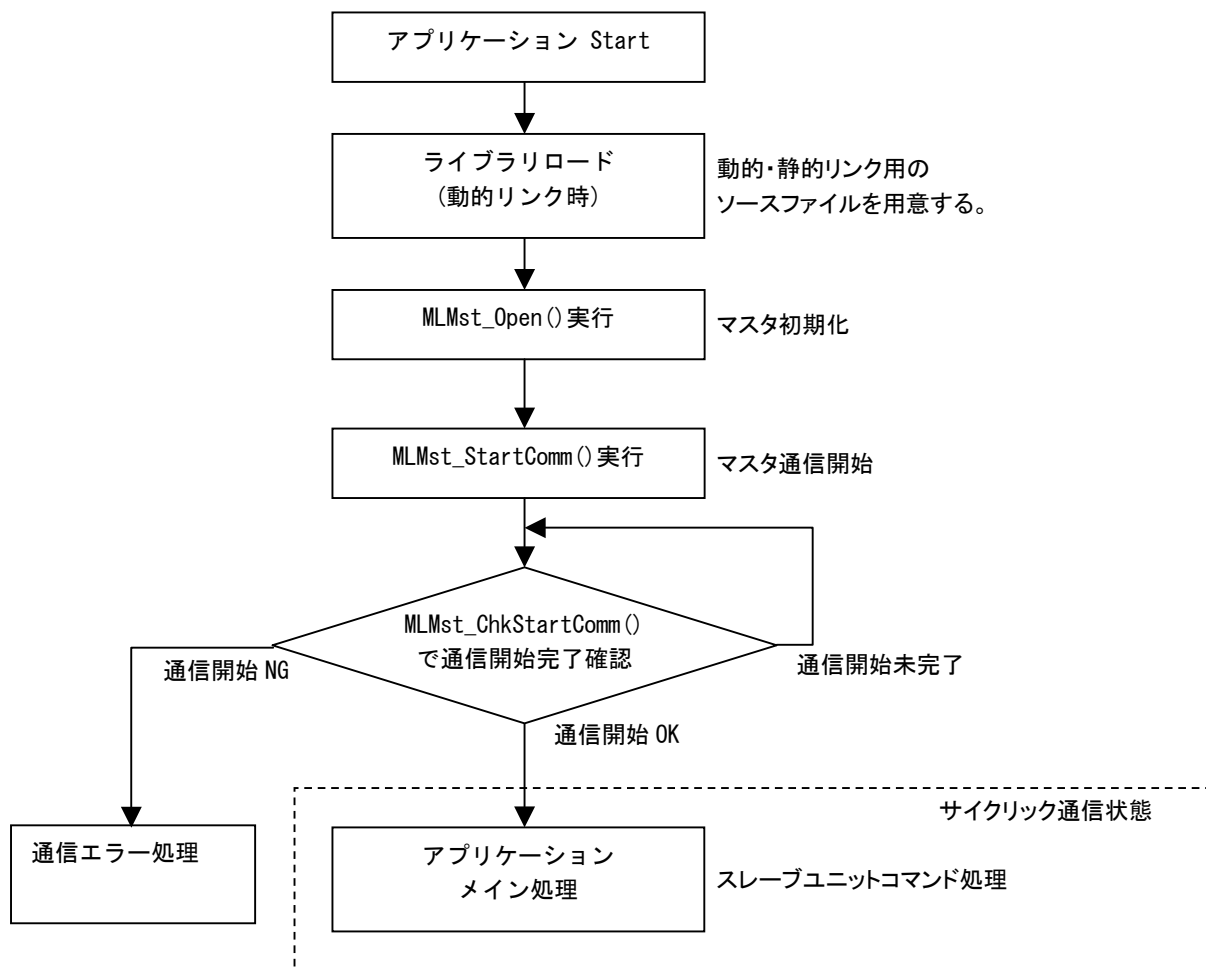


図 2-1-1. フローチャート

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。通信が正常に開始されれば、スレーブユニットへのコマンド実行を行うことができます。

2-2 マスタアクセス関数

MLMst_Open 関数

機能 MECHATROLINK マスタをオープンします。

書式 `int MLMst_Open (void);`

引数 なし

戻り値

| | |
|-------------------|--------------------------------|
| ML_ER_OK | : 正常 |
| ML_ER_NOMSTPROC | : MECHATROLINK マスタプロセスが動作していない |
| ML_ER_ALREADYOPEN | : すでにオープンしています |
| ML_ER_CREATE | : 共有メモリまたはデータメール生成失敗 |

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、スレーブ制御を行う前に、必ずコールする必要があります。

MLMst_Close 関数

機能 MECHATROLINK マスタをクローズします。

書式 `int MLMst_Close (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 |

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。

MLMst_SetCommSetting 関数

機能 MECHATROLINK マスタの通信パラメータを設定します。

書式

```
int ML3_Mst_SetCommSetting (
    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 [0:サイクリック通信] [1:イベントドリブン通信] | (初期値: 0) |
| MaxRetry | : 最大リトライ回数 [0 ~ 62] | (初期値: 1) 単位: 【回】 |
| HostWdt | : ホスト監視用 WDT 設定 [0:機能無効] [1(8us) ~ 16384(131072us)] | (初期値: 16384) 単位: 【8us】 |

戻り値

| | |
|-------------------|----------------|
| ML_ER_OK | : 正常 |
| ML_ER_NOTOPEN | : オープンされていない |
| ML_ER_ALREADYCOMM | : すでに通信開始されている |

説明

MECHATROLINK マスタの通信設定を取得します。

MLMst_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) ~ 6400000(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 [0 ~ 62] | 単位 : 【回】 | (初期値 : 1) |
| HostWdt | : ホスト監視用 WDT 設定 [0 : 機能無効] [1(8us) ~ 16384(131072us)] | 単位 : 【8us】 | (初期値 : 16384) |

| | | |
|------------|---------------|--------------|
| 戻り値 | ML_ER_OK | : 正常 |
| | ML_ER_NOTOPEN | : オープンされていない |

説明 MECHATROLINK マスタの通信設定を取得します。

MLMst_StartComm 関数

機能

MECHATROLINK 通信を開始します。

書式

```
int MLMst_StartComm (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_NOMSTPROC | : マスタプロセスが動作していない |
| ML_ER_NOTRESET | : リセット処理が完了していない |
| ML_ER_ALREADYCOMM | : すでに通信されている |
| ML_ER_ERROR | : マスタプロセスへのアクセスに失敗した |

説明

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、「MLMst_ChkStartComm」関数で確認してください。

MLMst_ChkStartComm 関数

機能 MECHATROLINK 通信開始を確認します。

書式 int MLMst_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、「MLMst_GetSystemError」を使ってエラー情報を取得してください。

MLMst_ResetComm 関数

| | |
|------------|---|
| 機能 | MECHATROLINK 通信を停止します。 |
| 書式 | <code>int MLMst_ResetComm (void);</code> |
| 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 |
| 説明 | MECHATROLINK 通信を停止します。 通信が正常に停止されたかどうかは、「MLMst_ChkResetComm」関数で確認してください。 |

MLMst_ChkResetComm 関数

機能 MECHATROLINK 通信停止を確認します。

書式 int MLMst_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。

MLMst_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 MLMst_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

ML_ER_OK : 正常
 ML_ER_NOTOPEN : オープンされていない

説明 MECHATROLINK マスタのシステムエラーを取得します。
 「MLMst_ChkStartComm」関数の戻り値で「ML_ER_COMMINTERR」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。

MLMst_GetMasterStatus 関数

機能 MECHATROLINK マスタのステータスを取得します。

書式

```
int MLMst_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

ML_ER_OK : 正常

ML_ER_NOTOPEN : オープンされていない

説明 MECHATROLINK マスタのステータスを取得します。

2-3 スレーブアクセス関数

MLSlv_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 MlSlv_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 : イベントドリブン通信中 |

ErrorSts : JL100M エラーステータス

| ErrorSts | 内容 |
|----------------------|--------------------------------------|
| SLV_ESTS_FGS_ERR | 0x00000001 : データ受信中 FGS 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ーランエラー |
| 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 |
| | ML_ER_NOTOPEN | : オープンされていない |
| | ML_ER_INVALIDPARAM | : 引数のパラメータ範囲異常 |

説明 スレーブユニットの通信状態を取得します。

 MLSlv_SetSlaveCommInfo 関数

機能 スレーブの通信設定を設定します。

書式

```
int MLSlv_SetSlaveCommInfo (
    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 | ML_ER_OK | : 正常 |
| | ML_ER_NOTOPEN | : オープンされていない |
| | ML_ER_INVALIDPARAM | : 引数のパラメータ範囲異常 |
| | ML_ER_ALREADYCOMM | : すでに通信開始されている |

説明 スレーブの通信設定を設定します。

 MLSlv_GetSlaveCommInfo 関数

機能 スレーブの通信設定を取得します。

書式

```
int MLSlv_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

ML_ER_OK : 正常
ML_ER_NOTOPEN : オープンされていない
ML_ER_INVALIDPARAM : 引数のパラメータ範囲異常

説明

スレーブの通信設定を取得します。

 MLSlv_SetCmd 関数

機能

スレーブへ MECHATROLINK コマンドとパラメータを設定します。

書式

```
int MLSlv_SetCmd(
    WORD SlaveNo,
    unsigned char cmd,
    TM3_COMMAND_DATA *data
);
```

引数

SlaveNo : スレーブ番号 [1 ~ 62]
cmd : 実行する MECHATROLINK コマンド
data : MECHATROLINK コマンドパラメータを格納するポインタ

 MECHATROLINK コマンドパラメータ

```
typedef union {
    unsigned char data[60];

    TCOMMON_COMMAND cmn;
    TSERVO_COMMAND srv;
    TIO_COMMAND io;
} TM3_COMMAND_DATA;
```

data : コマンドパラメータバッファ (共用体のベースバッファ)
cmn : 共通コマンドパラメータ
 詳細は「2-5 共通コマンド」を参照してください。
srv : 標準サーボコマンドパラメータ
 詳細は「2-6 標準サーボプロファイルコマンド」を参照してください。
io : 標準 I/O コマンドパラメータ
 詳細は「2-7 標準 I/O プロファイルコマンド」を参照してください。

戻り値

ML_ER_OK : 正常
ML_ER_NOTOPEN : オープンされていない
ML_ER_INVALIDPARAM : 引数のパラメータ範囲異常
ML_ER_NOTCOMM : 通信されていない

説明

実行する MECHATROLINK コマンドとパラメータを設定します。

MLSlv_ChgCmd 関数

機能

現在実行中の MECHATROLINK コマンドを変更します。

書式

```
int MLSlv_ChgCmd(  
    WORD SlaveNo,  
    unsigned char cmd  
);
```

引数

SlaveNo : スレーブ番号 [1 ~ 62]
cmd : 実行する MECHATROLINK コマンド

戻り値

ML_ER_OK : 正常
ML_ER_NOTOPEN : オープンされていない
ML_ER_INVALIDPARAM : 引数のパラメータ範囲異常
ML_ER_NOTCOMM : 通信されていない

説明

実行する MECHATROLINK コマンドを変更します。パラメータはすでに設定されている値をそのまま使用します。

MLSlv_GetCmd 関数

機能 現在、実行中の MECHATROLINK コマンドとパラメータを取得します。

書式

```
int MlSlv_GetCmd(
    WORD SlaveNo,
    unsigned char *cmd,
    TM3_COMMAND_DATA *data
);
```

引数

| | |
|---------|-----------------------------------|
| SlaveNo | : スレーブ番号 [1 ~ 62] |
| cmd | : 実行中の MECHATROLINK コマンドを格納するポインタ |
| data | : MECHATROLINK コマンドパラメータを格納するポインタ |

MECHATROLINK コマンドパラメータ

```
typedef union {
    unsigned char data[60];

    TCOMMON_COMMAND cmn;
    TSERVO_COMMAND srv;
    TIO_COMMAND io;
} TM3_COMMAND_DATA;
```

| | |
|------|--|
| data | : コマンドパラメータバッファ (共用体のベースバッファ) |
| cmn | : 共通コマンドパラメータ 詳細は「2-5 共通コマンド」を参照してください。 |
| srv | : 標準サーボコマンドパラメータ 詳細は「2-6 標準サーボプロファイルコマンド」を参照してください。 |
| io | : 標準 I/O コマンドパラメータ 詳細は「2-7 標準 I/O プロファイルコマンド」を参照してください。 |

戻り値

| | |
|--------------------|----------------|
| ML_ER_OK | : 正常 |
| ML_ER_NOTOPEN | : オープンされていない |
| ML_ER_INVALIDPARAM | : 引数のパラメータ範囲異常 |
| ML_ER_NOTCOMM | : 通信されていない |

説明 現在、実行中の MECHATROLINK コマンドとパラメータを取得します。

 MLSlv_GetCmdPara 関数

機能 指定した MECHATROLINK コマンドのパラメータを取得します。

書式

```
int MLSlv_GetCmdPara(
    WORD SlaveNo,
    unsigned char cmd,
    TM3_COMMAND_DATA *data
);
```

引数

| | |
|---------|-----------------------------------|
| SlaveNo | : スレーブ番号 [1 ~ 62] |
| cmd | : パラメータを取得する MECHATROLINK コマンドを指定 |
| data | : MECHATROLINK コマンドパラメータを格納するポインタ |

 MECHATROLINK コマンドパラメータ

```
typedef union {
    unsigned char data[60];

    TCOMMON_COMMAND cmn;
    TSERVO_COMMAND srv;
    TIO_COMMAND io;
} TM3_COMMAND_DATA;
```

| | |
|------|--|
| data | : コマンドパラメータバッファ (共用体のベースバッファ) |
| cmn | : 共通コマンドパラメータ 詳細は「2-5 共通コマンド」を参照してください。 |
| srv | : 標準サーボコマンドパラメータ 詳細は「2-6 標準サーボプロファイルコマンド」を参照してください。 |
| io | : 標準 I/O コマンドパラメータ 詳細は「2-7 標準 I/O プロファイルコマンド」を参照してください。 |

戻り値

| | |
|--------------------|----------------|
| ML_ER_OK | : 正常 |
| ML_ER_NOTOPEN | : オープンされていない |
| ML_ER_INVALIDPARAM | : 引数のパラメータ範囲異常 |
| ML_ER_NOTCOMM | : 通信されていない |

説明 指定した MECHATROLINK コマンドのパラメータを取得します。

MLSlv_SetSubCmd 関数

機能 スレーブへ MECHATROLINK サブコマンドとパラメータを設定します。

書式

```
int MlSlv_SetSubCmd(
    WORD SlaveNo,
    TSUB_COMMAND_HEAD *scmd,
    TSUB_COMMAND_DATA *sdata
);
```

引数

SlaveNo : スレーブ番号 [1 ~ 62]

scmd : 実行する MECHATROLINK サブコマンドヘッダを格納するポインタ

sdata : MECHATROLINK サブコマンドパラメータを格納するポインタ

MECHATROLINK サブコマンドヘッダ

```
typedef struct {
    unsigned char subcmd;
    unsigned char sub_ctrl[3];
} TSUB_COMMAND_HEAD;
```

subcmd : サブコマンド

sub_ctrl : サブコマンドコントロール

| | | | | | | | |
|----------|-------|-------|-------|----------|-------|-------|-------|
| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
| Reserve | | | | | | | |
| bit15 | bit14 | bit13 | bit12 | bit11 | bit10 | bit9 | bit8 |
| SEL_MON4 | | | | Reserve | | | |
| bit23 | bit22 | bit21 | bit20 | bit19 | bit18 | bit17 | bit16 |
| SEL_MON6 | | | | SEL_MON5 | | | |

SEL_MON4: モニタ選択 4 [0 ~ 15] (初期値: 0)

SEL_MON5: モニタ選択 5 [0 ~ 15] (初期値: 0)

SEL_MON6: モニタ選択 6 [0 ~ 15] (初期値: 0)

詳細は「2-9 モニタ情報」を参照してください

MECHATROLINK サブコマンドパラメータ

```
typedef union {  
    unsigned char data[12];  
  
    TSUB_COMMAND_NOP      nop;  
    TSUB_COMMAND_ALM_RD   alm_rd;  
    TSUB_COMMAND_ALM_CLR  alm_clr;  
    TSUB_COMMAND_MEM_RD   mem_rd;  
    TSUB_COMMAND_MEM_WR   mem_wr;  
    TSUB_COMMAND_SMON     smon;  
    TSUB_COMMAND_SVPRM_RD svprm_rd;  
    TSUB_COMMAND_SVPRM_WR svprm_wr;  
} TSUB_COMMAND_DATA;
```

`data` : コマンドパラメータバッファ（共用体のベースバッファ）
`nop` : NOP サブコマンドパラメータ
`alm_rd` : ALM_RD サブコマンドパラメータ
`alm_clr` : ALM_CLR サブコマンドパラメータ
`mem_rd` : MEM_RD サブコマンドパラメータ
`mem_wr` : MEM_WR サブコマンドパラメータ
`smon` : SMON サブコマンドパラメータ
`svprm_rd` : SVPRM_RD サブコマンドパラメータ
`svprm_wr` : SVPRM_WR サブコマンドパラメータ

詳細は「2-8 サブコマンド」を参照してください

| | | |
|------------|--------------------|----------------|
| 戻り値 | ML_ER_OK | : 正常 |
| | ML_ER_NOTOPEN | : オープンされていない |
| | ML_ER_INVALIDPARAM | : 引数のパラメータ範囲異常 |
| | ML_ER_NOTCOMM | : 通信されていない |

説明 実行する MECHATROLINK サブコマンドとパラメータを設定します。

MLSlv_ChgSubCmd 関数

機能

現在実行中の MECHATROLINK サブコマンドを変更します。

書式

```
int MLSlv_ChgCmd(  
    WORD SlaveNo,  
    unsigned char sub_cmd  
);
```

引数

SlaveNo : スレーブ番号 [1 ~ 62]
sub_cmd : 実行する MECHATROLINK サブコマンド

戻り値

ML_ER_OK : 正常
ML_ER_NOTOPEN : オープンされていない
ML_ER_INVALIDPARAM : 引数のパラメータ範囲異常
ML_ER_NOTCOMM : 通信されていない

説明

実行する MECHATROLINK サブコマンドを変更します。パラメータはすでに設定されている値をそのまま使用します。

MLSlv_GetSubCmd 関数

機能

現在、実行中の MECHATROLINK サブコマンドとパラメータを取得します。

書式

```
int MLSlv_GetSubCmd(
    WORD SlaveNo,
    TSUB_COMMAND_HEAD *scmd,
    TSUB_COMMAND_DATA *sdata
);
```

引数

SlaveNo : スレーブ番号 [1 ~ 62]
scmd : 現在実行中の MECHATROLINK サブコマンドヘッダを格納するポインタ
sdata : MECHATROLINK サブコマンドパラメータを格納するポインタ

MECHATROLINK サブコマンドヘッダ

```
typedef struct {
    unsigned char subcmd;
    unsigned char sub_ctrl[3];
} TSUB_COMMAND_HEAD;
```

subcmd : サブコマンド

sub_ctrl : サブコマンドコントロール

| | | | | | | | |
|----------|-------|-------|-------|----------|-------|-------|-------|
| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
| Reserve | | | | | | | |
| bit15 | bit14 | bit13 | bit12 | bit11 | bit10 | bit9 | bit8 |
| SEL_MON4 | | | | Reserve | | | |
| bit23 | bit22 | bit21 | bit20 | bit19 | bit18 | bit17 | bit16 |
| SEL_MON6 | | | | SEL_MON5 | | | |

SEL_MON4 : モニタ選択 4 [0 ~ 15] (初期値 : 0)

SEL_MON5 : モニタ選択 5 [0 ~ 15] (初期値 : 0)

SEL_MON6 : モニタ選択 6 [0 ~ 15] (初期値 : 0)

詳細は「2-9 モニタ情報」を参照してください

MECHATROLINK サブコマンドパラメータ

```
typedef union {
    unsigned char    data[12];

    TSUB_COMMAND_NOP    nop;
    TSUB_COMMAND_ALM_RD    alm_rd;
    TSUB_COMMAND_ALM_CLR    alm_clr;
    TSUB_COMMAND_MEM_RD    mem_rd;
    TSUB_COMMAND_MEM_WR    mem_wr;
    TSUB_COMMAND_SMON    smon;
    TSUB_COMMAND_SVPRM_RD    svprm_rd;
    TSUB_COMMAND_SVPRM_WR    svprm_wr;
} TSUB_COMMAND_DATA;
```

`data` : コマンドパラメータバッファ（共用体のベースバッファ）
`nop` : NOP サブコマンドパラメータ
`alm_rd` : ALM_RD サブコマンドパラメータ
`alm_clr` : ALM_CLR サブコマンドパラメータ
`mem_rd` : MEM_RD サブコマンドパラメータ
`mem_wr` : MEM_WR サブコマンドパラメータ
`smon` : SMON サブコマンドパラメータ
`svprm_rd` : SVPRM_RD サブコマンドパラメータ
`svprm_wr` : SVPRM_WR サブコマンドパラメータ

詳細は「2-8 サブコマンド」を参照してください

| | | |
|------------|--------------------|----------------|
| 戻り値 | ML_ER_OK | : 正常 |
| | ML_ER_NOTOPEN | : オープンされていない |
| | ML_ER_INVALIDPARAM | : 引数のパラメータ範囲異常 |
| | ML_ER_NOTCOMM | : 通信されていない |

説明 現在、実行中の MECHATROLINK サブコマンドとパラメータを取得します。

MLSlv_GetSubCmdPara 関数

機能

指定した MECHATROLINK サブコマンドのパラメータを取得します。

書式

```
int MlSlv_GetSubCmdPara(
    WORD SlaveNo,
    unsigned char sub_cmd,
    TSUB_COMMAND_DATA *sdata
);
```

引数

SlaveNo : スレーブ番号 [1 ~ 62]
sub_cmd : パラメータを取得する MECHATROLINK サブコマンドを指定
sdata : MECHATROLINK サブコマンドパラメータを格納するポインタ

MECHATROLINK サブコマンドパラメータ

```
typedef union {
    unsigned char    data[12];

    TSUB_COMMAND_NOP    nop;
    TSUB_COMMAND_ALM_RD    alm_rd;
    TSUB_COMMAND_ALM_CLR    alm_clr;
    TSUB_COMMAND_MEM_RD    mem_rd;
    TSUB_COMMAND_MEM_WR    mem_wr;
    TSUB_COMMAND_SMON    smon;
    TSUB_COMMAND_SVPRM_RD    svprm_rd;
    TSUB_COMMAND_SVPRM_WR    svprm_wr;
} TSUB_COMMAND_DATA;
```

data : コマンドパラメータバッファ (共用体のベースバッファ)
nop : NOP サブコマンドパラメータ
alm_rd : ALM_RD サブコマンドパラメータ
alm_clr : ALM_CLR サブコマンドパラメータ
mem_rd : MEM_RD サブコマンドパラメータ
mem_wr : MEM_WR サブコマンドパラメータ
smon : SMON サブコマンドパラメータ
svprm_rd : SVPRM_RD サブコマンドパラメータ
svprm_wr : SVPRM_WR サブコマンドパラメータ

詳細は「2-8 サブコマンド」を参照してください

戻り値

ML_ER_OK : 正常
ML_ER_NOTOPEN : オープンされていない
ML_ER_INVALIDPARAM : 引数のパラメータ範囲異常
ML_ER_NOTCOMM : 通信されていない

説明

指定した、MECHATROLINK サブコマンドのパラメータを取得します。

MLSlv_ChkRes 関数

| | |
|------------|---|
| 機能 | MECHATROLINK コマンドのレスポンスを確認します。 |
| 書式 | <pre>int MlSlv_ChkRes(WORD SlaveNo,);</pre> |
| 引数 | SlaveNo : スレーブ番号 [1 ~ 62] |
| 戻り値 | ML_ER_OK : レスポンスあり ML_ER_NOTOPEN : オープンされていない ML_ER_INVALIDPARAM : 引数のパラメータ範囲異常 ML_ER_NOTCOMM : 通信されていない ML_ER_NORESPONSE : レスポンスなし |
| 説明 | MECHATROLINK コマンドのレスポンスを確認します。 |

MLSlv_GetRes 関数

機能

現在、実行中の MECHATROLINK レスポンスとパラメータを取得します。

書式

```
int MlSlv_GetRes(
    WORD SlaveNo,
    unsigned char *res,
    unsigned short *stat,
    TM3_RESPONSE_DATA *rdata
);
```

引数

SlaveNo : スレーブ番号 [1 ~ 62]
res : 現在実行中の MECHATROLINK レスポンスを格納するポインタ
stat : MECHATROLINK レスポンスコマンドステータスを格納するポインタ

| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|----------|-------|---------|---------|-------------|--------|-------|-------|
| RCMD_ID | | Reserve | Reserve | ALM_CLR_CMP | CMDRDY | D_WAR | D_ALM |
| bit15 | bit14 | bit13 | bit12 | bit11 | bit10 | bit9 | bit8 |
| COMM_ALM | | | | CMD_ALM | | | |

D_ALM : アラーム発生中 [1 : 機器アラーム状態 0 : それ以外]
D_WAR : ワーニング発生中 [1 : 機器ワーニング状態中 0 : それ以外]
CMDRDY : コマンド受付可 [1 : コマンド受付可 0 : それ以外]
ALM_CLR_CMP : ALM_CLR の実行完了 [1 : ALM_CLR の実行完了 0 : それ以外]
RCMD_ID : CMD_ID のエコーバック [0 ~ 3 : 同一コマンドの識別に使用]
CMD_ALM : コマンド異常状態通知

詳細は「2-10 アラーム情報」を参照してください

COMM_ALM : 通信異常状態通知

詳細は「2-10 アラーム情報」を参照してください

rdata : MECHATROLINK レスポンスパラメータを格納するポインタ

MECHATROLINK レスポンスパラメータ

```
typedef union {
    unsigned char    data[60];

    TCOMMON_RESPONSE cmn;
    TSERVO_RESPONSE  srv;
    TIO_RESPONSE      io;
} TM3_RESPONSE_DATA;
```

data : コマンドパラメータバッファ（共用体のベースバッファ）

cmn : 共通コマンドレスポンスパラメータ
詳細は「2-5 共通コマンド」を参照してください。

srv : 標準サーボコマンドレスポンスパラメータ
詳細は「2-6 標準サーボプロファイルコマンド」を参照してください。

io : 標準 I/O コマンドレスポンスパラメータ
詳細は「2-7 標準 I/O プロファイルコマンド」を参照してください。

| | | |
|------------|--------------------|----------------|
| 戻り値 | ML_ER_OK | : 正常 |
| | ML_ER_NOTOPEN | : オープンされていない |
| | ML_ER_INVALIDPARAM | : 引数のパラメータ範囲異常 |
| | ML_ER_NOTCOMM | : 通信されていない |

説明 現在、実行中の MECHATROLINK レスポンスとパラメータを取得します。

MLSlv_GetResPara 関数

機能

指定した MECHATROLINK レスポンスのパラメータを取得します。

書式

```
int MlSlv_GetResPara(
    WORD SlaveNo,
    unsigned char res,
    unsigned short *stat,
    TM3_RESPONSE_DATA *rdata
);
```

引数

SlaveNo : スレーブ番号 [1 ~ 62]
res : パラメータを取得する MECHATROLINK レスポンスを指定
stat : MECHATROLINK レスポンスコマンドステータスを格納するポインタ

| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|----------|-------|---------|---------|-------------|--------|-------|-------|
| RCMD_ID | | Reserve | Reserve | ALM_CLR_CMP | CMDRDY | D_WAR | D_ALM |
| bit15 | bit14 | bit13 | bit12 | bit11 | bit10 | bit9 | bit8 |
| COMM_ALM | | | | CMD_ALM | | | |

D_ALM : アラーム発生中 [1 : 機器アラーム状態 0 : それ以外]
D_WAR : ワーニング発生中 [1 : 機器ワーニング状態中 0 : それ以外]
CMDRDY : コマンド受付可 [1 : コマンド受付可 0 : それ以外]
ALM_CLR_CMP : ALM_CLR の実行完了 [1 : ALM_CLR の実行完了 0 : それ以外]
RCMD_ID : CMD_ID のエコーバック [0 ~ 3 : 同一コマンドの識別に使用]
CMD_ALM : コマンド異常状態通知

詳細は「2-10 アラーム情報」を参照してください

COMM_ALM : 通信異常状態通知

詳細は「2-10 アラーム情報」を参照してください

rdata : MECHATROLINK レスポンスパラメータを格納するポインタ

MECHATROLINK レスポンスパラメータ

```
typedef union {
    unsigned char    data[60];

    TCOMMON_RESPONSE cmn;
    TSERVO_RESPONSE  srv;
    TIO_RESPONSE      io;
} TM3_RESPONSE_DATA;
```

data : コマンドパラメータバッファ（共用体のベースバッファ）

cmn : 共通コマンドレスポンスパラメータ
詳細は「2-5 共通コマンド」を参照してください。

srv : 標準サーボコマンドレスポンスパラメータ
詳細は「2-6 標準サーボプロファイルコマンド」を参照してください。

io : 標準 I/O コマンドレスポンスパラメータ
詳細は「2-7 標準 I/O プロファイルコマンド」を参照してください。

| | | |
|------------|--------------------|----------------|
| 戻り値 | ML_ER_OK | : 正常 |
| | ML_ER_NOTOPEN | : オープンされていない |
| | ML_ER_INVALIDPARAM | : 引数のパラメータ範囲異常 |
| | ML_ER_NOTCOMM | : 通信されていない |

説明 指定した MECHATROLINK レスポンスのパラメータを取得します。

MLSlv_ChkSubRes 関数

| | |
|------------|---|
| 機能 | MECHATROLINK サブコマンドのレスポンスを確認します。 |
| 書式 | <pre>int MLSlv_ChkSubRes(WORD SlaveNo);</pre> |
| 引数 | SlaveNo : スレーブ番号 [1 ~ 62] |
| 戻り値 | ML_ER_OK : レスポンスあり ML_ER_NOTOPEN : オープンされていない ML_ER_INVALIDPARAM : 引数のパラメータ範囲異常 ML_ER_NOTCOMM : 通信されていない ML_ER_NORESPONSE : レスポンスなし |
| 説明 | MECHATROLINK サブコマンドのレスポンスを確認します。 |

MLSlv_GetSubRes 関数

機能 現在、実行中の MECHATROLINK サブコマンドレスポンスとパラメータを取得します。

書式

```
int MlSlv_GetSubRes(
    WORD SlaveNo,
    TSUB_RESPONSE_HEAD *sres,
    TSUB_RESPONSE_DATA *srdata
);
```

引数

SlaveNo : スレーブ番号 [1 ~ 62]

sres : 現在実行中の MECHATROLINK サブコマンドレスポンスヘッダを格納するポインタ

srdata : MECHATROLINK サブコマンドレスポンスパラメータを格納するポインタ

MECHATROLINK サブコマンドレスポンスヘッダ

```
typedef struct {
    unsigned char subres;
    unsigned char sub_stat[3];
} TSUB_RESPONSE_HEAD;
```

subres : サブコマンドレスポンス

sub_stat : サブコマンドステータス

| | | | | | | | |
|----------|-------|-------|-------|------------|-----------|---------|-------|
| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
| Reserve | | | | | SUBCMDRDY | Reserve | |
| bit15 | bit14 | bit13 | bit12 | bit11 | bit10 | bit9 | bit8 |
| SEL_MON4 | | | | SUBCMD_ALM | | | |
| bit23 | bit22 | bit21 | bit20 | bit19 | bit18 | bit17 | bit16 |
| SEL_MON6 | | | | SEL_MON5 | | | |

SUBCMDRDY : サブコマンドレディ [1 : コマンド受付可 0 : それ以外]

SUBCMD_ALM : サブコマンドアラーム

詳細は「2-10 アラーム情報」を参照してください

SEL_MON4 : モニタ選択 4 [0 ~ 15]

SEL_MON5 : モニタ選択 5 [0 ~ 15]

SEL_MON6 : モニタ選択 6 [0 ~ 15]

詳細は「2-9 モニタ情報」を参照してください

MECHATROLINK サブコマンドレスポンスパラメータ

```
typedef union {
    unsigned char      data[12];

    TSUB_RESPONSE_NOP      nop;
    TSUB_RESPONSE_ALM_RD   alm_rd;
    TSUB_RESPONSE_ALM_CLR  alm_clr;
    TSUB_RESPONSE_MEM_RD   mem_rd;
    TSUB_RESPONSE_MEM_WR   mem_wr;
    TSUB_RESPONSE_SMON     smon;
    TSUB_RESPONSE_SVPRM_RD svprm_rd;
    TSUB_RESPONSE_SVPRM_WR svprm_wr;
} TSUB_RESPONSE_DATA;
```

`data` : コマンドパラメータバッファ（共用体のベースバッファ）
`nop` : NOP サブコマンドレスポンスパラメータ
`alm_rd` : ALM_RD サブコマンドレスポンスパラメータ
`alm_clr` : ALM_CLR サブコマンドレスポンスパラメータ
`mem_rd` : MEM_RD サブコマンドレスポンスパラメータ
`mem_wr` : MEM_WR サブコマンドレスポンスパラメータ
`smon` : SMON サブコマンドレスポンスパラメータ
`svprm_rd` : SVPRM_RD サブコマンドレスポンスパラメータ
`svprm_wr` : SVPRM_WR サブコマンドレスポンスパラメータ

詳細は「2-8 サブコマンド」を参照してください

戻り値

ML_ER_OK : 正常
ML_ER_NOTOPEN : オープンされていない
ML_ER_INVALIDPARAM : 引数のパラメータ範囲異常
ML_ER_NOTCOMM : 通信されていない

説明

現在、実行中の MECHATROLINK サブコマンドレスポンスとパラメータを取得します。

MLSlv_GetSubRes 関数

機能

現在、実行中の MECHATROLINK サブコマンドレスポンスとパラメータを取得します。

書式

```
int MLSlv_GetSubRes(
    WORD SlaveNo,
    TSUB_RESPONSE_HEAD *sres,
    TSUB_RESPONSE_DATA *srdata
);
```

引数

SlaveNo : スレーブ番号 [1 ~ 62]
sres : 現在実行中の MECHATROLINK サブコマンドレスポンスヘッダを格納するポインタ
srdata : MECHATROLINK サブコマンドレスポンスパラメータを格納するポインタ

MECHATROLINK サブコマンドレスポンスヘッダ

```
typedef struct {
    unsigned char subres;
    unsigned char sub_stat[3];
} TSUB_RESPONSE_HEAD;
```

subres : サブコマンドレスポンス

sub_stat : サブコマンドステータス

| | | | | | | | |
|----------|-------|-------|-------|------------|-----------|---------|-------|
| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
| Reserve | | | | | SUBCMDRDY | Reserve | |
| bit15 | bit14 | bit13 | bit12 | bit11 | bit10 | bit9 | bit8 |
| SEL_MON4 | | | | SUBCMD_ALM | | | |
| bit23 | bit22 | bit21 | bit20 | bit19 | bit18 | bit17 | bit16 |
| SEL_MON6 | | | | SEL_MON5 | | | |

SUBCMDRDY : サブコマンドレディ [1 : コマンド受付可 0 : それ以外]

SUBCMD_ALM : サブコマンドアラーム

詳細は「2-10 アラーム情報」を参照してください

SEL_MON4 : モニタ選択 4 [0 ~ 15]

SEL_MON5 : モニタ選択 5 [0 ~ 15]

SEL_MON6 : モニタ選択 6 [0 ~ 15]

詳細は「2-9 モニタ情報」を参照してください

MECHATROLINK サブコマンドレスポンスパラメータ

```

typedef union {
    unsigned char      data[12];

    TSUB_RESPONSE_NOP      nop;
    TSUB_RESPONSE_ALM_RD   alm_rd;
    TSUB_RESPONSE_ALM_CLR  alm_clr;
    TSUB_RESPONSE_MEM_RD   mem_rd;
    TSUB_RESPONSE_MEM_WR   mem_wr;
    TSUB_RESPONSE_SMON     smon;
    TSUB_RESPONSE_SVPRM_RD svprm_rd;
    TSUB_RESPONSE_SVPRM_WR svprm_wr;
} TSUB_RESPONSE_DATA;

```

data : コマンドパラメータバッファ（共用体のベースバッファ）
nop : NOP サブコマンドレスポンスパラメータ
alm_rd : ALM_RD サブコマンドレスポンスパラメータ
alm_clr : ALM_CLR サブコマンドレスポンスパラメータ
mem_rd : MEM_RD サブコマンドレスポンスパラメータ
mem_wr : MEM_WR サブコマンドレスポンスパラメータ
smon : SMON サブコマンドレスポンスパラメータ
svprm_rd : SVPRM_RD サブコマンドレスポンスパラメータ
svprm_wr : SVPRM_WR サブコマンドレスポンスパラメータ

詳細は「2-8 サブコマンド」を参照してください

戻り値

ML_ER_OK : 正常
 ML_ER_NOTOPEN : オープンされていない
 ML_ER_INVALIDPARAM : 引数のパラメータ範囲異常
 ML_ER_NOTCOMM : 通信されていない

説明

現在、実行中の MECHATROLINK サブコマンドレスポンスとパラメータを取得します。

MLSlv_GetSubResPara 関数

機能

指定した MECHATROLINK サブコマンドレスポンスのパラメータを取得します。

書式

```
int MlSlv_GetSubResPara(
    WORD SlaveNo,
    unsigned char sub_res,
    TSUB_RESPONSE_HEAD *sres,
    TSUB_RESPONSE_DATA *srdata
);
```

引数

SlaveNo : スレーブ番号 [1 ~ 62]
sub_res : パラメータを取得する MECHATROLINK サブコマンドを指定
sres : MECHATROLINK サブコマンドレスポンスヘッダを格納するポインタ
srdata : MECHATROLINK サブコマンドレスポンスパラメータを格納するポインタ

MECHATROLINK サブコマンドレスポンスヘッダ

```
typedef struct {
    unsigned char subres;
    unsigned char sub_stat[3];
} TSUB_RESPONSE_HEAD;
```

subres : サブコマンドレスポンス

sub_stat : サブコマンドステータス

| | | | | | | | |
|----------|-------|-------|-------|------------|-----------|---------|-------|
| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
| Reserve | | | | | SUBCMDRDY | Reserve | |
| bit15 | bit14 | bit13 | bit12 | bit11 | bit10 | bit9 | bit8 |
| SEL_MON4 | | | | SUBCMD_ALM | | | |
| bit23 | bit22 | bit21 | bit20 | bit19 | bit18 | bit17 | bit16 |
| SEL_MON6 | | | | SEL_MON5 | | | |

SUBCMDRDY : サブコマンドレディ [1 : コマンド受付可 0 : それ以外]

SUBCMD_ALM : サブコマンドアラーム

詳細は「2-10 アラーム情報」を参照してください

SEL_MON4 : モニタ選択 4 [0 ~ 15]

SEL_MON5 : モニタ選択 5 [0 ~ 15]

SEL_MON6 : モニタ選択 6 [0 ~ 15]

詳細は「2-9 モニタ情報」を参照してください

MECHATROLINK サブコマンドレスポンスパラメータ

```

typedef union {
    unsigned char      data[12];

    TSUB_RESPONSE_NOP      nop;
    TSUB_RESPONSE_ALM_RD   alm_rd;
    TSUB_RESPONSE_ALM_CLR  alm_clr;
    TSUB_RESPONSE_MEM_RD   mem_rd;
    TSUB_RESPONSE_MEM_WR   mem_wr;
    TSUB_RESPONSE_SMON     smon;
    TSUB_RESPONSE_SVPRM_RD svprm_rd;
    TSUB_RESPONSE_SVPRM_WR svprm_wr;
} TSUB_RESPONSE_DATA;

```

data : コマンドパラメータバッファ（共用体のベースバッファ）
nop : NOP サブコマンドレスポンスパラメータ
alm_rd : ALM_RD サブコマンドレスポンスパラメータ
alm_clr : ALM_CLR サブコマンドレスポンスパラメータ
mem_rd : MEM_RD サブコマンドレスポンスパラメータ
mem_wr : MEM_WR サブコマンドレスポンスパラメータ
smon : SMON サブコマンドレスポンスパラメータ
svprm_rd : SVPRM_RD サブコマンドレスポンスパラメータ
svprm_wr : SVPRM_WR サブコマンドレスポンスパラメータ

詳細は「2-8 サブコマンド」を参照してください

戻り値

ML_ER_OK : 正常
 ML_ER_NOTOPEN : オープンされていない
 ML_ER_INVALIDPARAM : 引数のパラメータ範囲異常
 ML_ER_NOTCOMM : 通信されていない

説明

指定した MECHATROLINK サブコマンドレスポンスとパラメータを取得します。

2-4 簡易補間移動関数

MLSlv_IpdtSet 関数

機能

補間移動を行う各軸のパラメータを設定します。

書式

```
int MlSlv_IpdtSet(  
    int grpno,  
    unsigned short CtrlAxis,  
    long* TPos  
);
```

引数

grpno : グループ番号 [1 ~ 8]
CtrlAxis : 補間制御を行う軸
TPos : 各軸の目標位置を表す long 型配列のポインタ

戻り値

ML_ER_OK : 正常
ML_ER_NOTOPEN : オープンされていない
ML_ER_INVALIDPARAM : 不正な引数
ML_ER_NOTINIT : 初期化されていない
ML_ER_TIMEOUT : タイムアウトエラー

説明

補間制御を行うグループと、制御軸、それぞれの目標位置をバッファメモリに格納します。制御軸を指定するには、引数 CtrlAxis において、対応するビットを 1 に設定します。簡易補間移動で扱うことのできる軸数の最大値は 16 です。引数 TPos には要素数 16 の long 型の配列のポインタを渡す必要があります。目標位置を変更して、複数回実行することで、通信周期ごとに発行される目標位置をバッファリングすることができます。一度登録した目標位置は、「MLSlv_IpdtClr」コマンドでクリアされます。

MLMst_IpStrt 関数

機能 MECHATROLINK マスタの補間送りを開始します。

書式

```
int MLMst_IpStrt (  
    int grpno  
);
```

引数 `grpno` : グループ番号 [1 ~ 8]

戻り値

| | |
|--------------------|--------------|
| ML_ER_OK | : 正常 |
| ML_ER_NOTOPEN | : オープンされていない |
| ML_ER_INVALIDPARAM | : 不正な引数 |
| ML_ER_NOTINIT | : 初期化されていない |
| ML_ER_TIMEOUT | : タイムアウトエラー |

説明 補間送りを開始します。
通信周期ごとに、バッファメモリから読込んだ目標位置をサーボパックに送信します。
読込まれた目標位置はバッファメモリから解放されます。

MLMst_IpStop 関数

機能 MECHATROLINK マスタの補間送りを停止します。

書式

```
int MLMst_IpStop (
    int grpno,
    unsigned short mode
);
```

引数

`grpno` : グループ番号 [1 ~ 8]
`mode` : ストップモード

| mode | 内容 |
|------|---------------|
| 0 | 通常停止 |
| 1 | 即停止 |
| 2 | 即停止 + バッファクリア |

戻り値

ML_ER_OK : 正常
 ML_ER_NOTOPEN : オープンされていない
 ML_ER_INVALIDPARAM : 不正な引数
 ML_ER_NOTINIT : 初期化されていない
 ML_ER_TIMEOUT : タイムアウトエラー

説明 補間送りを停止します。mode に渡す引数によって、停止モードを指定することができます。

mode = 0 : 通常停止

本コマンドを発行した時点で、バッファメモリに格納されている位置までは補間送りを継続します。

本コマンド実行後に、「MLSlv_IpdtSet」コマンドを実行しても、メモリバッファに格納されるだけで、補間送りは実行されません。

mode = 1 : 即停止

本コマンドを発行した時点で、補間送りを停止します。

バッファメモリに格納されているデータは保持されます。バッファメモリをクリアする場合は、「MLSlv_IpdtClr」コマンドを実行します。

mode = 2 : 即停止 + バッファクリア

本コマンドを発行した時点で、補間送りを停止します。

バッファメモリに格納されているデータはクリアされます。

MLMst_IpNumGet 関数

機能 指定したグループの補間送りバッファの残数を取得します。

書式

```
int MLMst_IpNumGet (  
    int grpno,  
    unsigned long *num  
);
```

引数

| | |
|--------------------|----------------------------|
| <code>grpno</code> | : グループ番号 [1 ~ 8] |
| <code>num</code> | : 取得したバッファ残数を格納するバッファのポインタ |

戻り値

| | |
|---------------------------------|--------------|
| <code>ML_ER_OK</code> | : 正常 |
| <code>ML_ER_NOTOPEN</code> | : オープンされていない |
| <code>ML_ER_INVALIDPARAM</code> | : 不正な引数 |
| <code>ML_ER_NOTINIT</code> | : 初期化されていない |
| <code>ML_ER_TIMEOUT</code> | : タイムアウトエラー |

説明 指定したグループの補間送りバッファの残数を取得します。
バッファは1グループあたり256個あります。

MLMst_IpdtClr 関数

機能 指定したグループのバッファメモリに格納されている補間送り運転データをクリアします。。

書式

```
int MLMst_IpdtClr (  
    int grpno,  
);
```

引数 `grpno` : グループ番号 [1 ~ 8]

戻り値

| | |
|--------------------|--------------|
| ML_ER_OK | : 正常 |
| ML_ER_NOTOPEN | : オープンされていない |
| ML_ER_INVALIDPARAM | : 不正な引数 |
| ML_ER_NOTINIT | : 初期化されていない |
| ML_ER_TIMEOUT | : タイムアウトエラー |

説明 指定したグループのバッファメモリに格納されている補間送り運転データをクリアします。

2-5 共通コマンド

MECHATROLINK-Ⅲの共通コマンド一覧を記します。

表 2-5-1. MECHATROLINK-Ⅲ共通コマンド一覧

| コマンドコード | | コマンド | 動作 | 対応 *1 | 通信フェーズ | | |
|----------------------|------|------------|----------------|----------|--------|---|---|
| 定義名 | コード | | | | 1 | 2 | 3 |
| M3CMD_CMN_NOP | 0x00 | NOP | ノーオペレーション | 必須 | — | ○ | ○ |
| M3CMD_CMN_PRM_RD | 0x01 | PRM_RD | パラメータ読み出し | *2 | — | ○ | ○ |
| M3CMD_CMN_PRM_WR | 0x02 | PRM_WR | パラメータ書き込み | *2 | — | ○ | ○ |
| M3CMD_CMN_ID_RD | 0x03 | ID_RD | ID読み出し | 必須 | — | ○ | ○ |
| M3CMD_CMN_CONFIG | 0x04 | CONFIG | 機器セットアップ要求 | 必須 | — | ○ | ○ |
| M3CMD_CMN_ALM_RD | 0x05 | ALM_RD | アラーム/ワーニング読み出し | 必須 | — | ○ | ○ |
| M3CMD_CMN_ALM_CLR | 0x06 | ALM_CLR | アラーム/ワーニングクリア | 必須 | — | ○ | ○ |
| M3CMD_CMN_SYNC_SET | 0x0D | SYNC_SET | 同期確立要求 | 必須 | — | ○ | △ |
| M3CMD_CMN_CONNECT | 0x0E | CONNECT | コネクション確立要求 | 必須 | ○ | △ | △ |
| M3CMD_CMN_DISCONNECT | 0x0F | DISCONNECT | コネクション開放要求 | 必須 | ○ | ○ | ○ |
| M3CMD_CMN_PPRM_RD | 0x1B | PPRM_RD | 不揮発パラメータ読み出し | *2 | — | ○ | ○ |
| M3CMD_CMN_PPRM_WR | 0x1C | PPRM_WR | 不揮発パラメータ書き込み | *2 | — | ○ | ○ |
| M3CMD_CMN_MEM_RD | 0x1D | MEM_RD | メモリ読み出し | 任意 | — | ○ | ○ |
| M3CMD_CMN_MEM_WR | 0x1E | MEM_WR | メモリ書き込み | 任意 | — | ○ | ○ |

*1：必須：MECHATROLINK-Ⅲの通信仕様として、対応が必ず必要なコマンドです。

任意：MECHATROLINK-Ⅲの通信仕様として、対応が必ずしも必要ではなく、対応するかどうかは製品仕様で決定されます。

*2：標準サーボプロファイルでは、PRM_RD、PRM_WR、PPRM_RD、PPRM_WR は使用しません。SVPRM_RD、SVPRM_WR を使用します。標準 I/O プロファイルでは必須となります。

*3：通信フェーズの記号は下記ようになります。

○：実行可能

△：無視

—：不定な応答データ

共通コマンドレスポンスパラメータ

```
typedef union {
    unsigned char          data[60];

    TCOMMON_COMMAND_NOP      nop;
    TCOMMON_COMMAND_PRM_RD   prm_rd;
    TCOMMON_COMMAND_PRM_WR   prm_wr;
    TCOMMON_COMMAND_ID_RD    id_rd;
    TCOMMON_COMMAND_CONFIG   config;
    TCOMMON_COMMAND_ALM_RD   alm_rd;
    TCOMMON_COMMAND_ALM_CLR  alm_clr;
    TCOMMON_COMMAND_SYNC_SET sync_set;
    TCOMMON_COMMAND_CONNECT  connect;
    TCOMMON_COMMAND_DISCONNECT disconnect;
    TCOMMON_COMMAND_PPRM_RD  pprm_rd;
    TCOMMON_COMMAND_PPRM_WR  pprm_wr;
    TCOMMON_COMMAND_MEM_RD   mem_rd;
    TCOMMON_COMMAND_MEM_WR   mem_wr;
} TCOMMON_COMMAND;
```

```
typedef union {
    unsigned char          data[60];

    TCOMMON_RESPONSE_NOP      nop;
    TCOMMON_RESPONSE_PRM_RD   prm_rd;
    TCOMMON_RESPONSE_PRM_WR   prm_wr;
    TCOMMON_RESPONSE_ID_RD    id_rd;
    TCOMMON_RESPONSE_CONFIG   config;
    TCOMMON_RESPONSE_ALM_RD   alm_rd;
    TCOMMON_RESPONSE_ALM_CLR  alm_clr;
    TCOMMON_RESPONSE_SYNC_SET sync_set;
    TCOMMON_RESPONSE_CONNECT  connect;
    TCOMMON_RESPONSE_DISCONNECT disconnect;
    TCOMMON_RESPONSE_PPRM_RD  pprm_rd;
    TCOMMON_RESPONSE_PPRM_WR  pprm_wr;
    TCOMMON_RESPONSE_MEM_RD   mem_rd;
    TCOMMON_RESPONSE_MEM_WR   mem_wr;
} TCOMMON_RESPONSE;
```

data : コマンドレスポンスパラメータバッファ（共用体のベースバッファ）

nop : 無効コマンドレスポンスパラメータ
prm_rd : パラメータ読み出しコマンドレスポンスパラメータ
prm_wr : パラメータ書き込みコマンドレスポンスパラメータ
id_rd : ID読み出しコマンドレスポンスパラメータ
config : 機器セットアップ要求コマンドレスポンスパラメータ
alm_rd : アラーム／ワーニング読み出しコマンドレスポンスパラメータ
alm_clr : アラーム／ワーニングクリアコマンドレスポンスパラメータ
sync_set : 同期確立要求コマンドレスポンスパラメータ

| | |
|-------------------------|------------------------------|
| <code>connect</code> | : コネクション確立要求コマンドレスポンスパラメータ |
| <code>disconnect</code> | : コネクション開放要求コマンドレスポンスパラメータ |
| <code>pprm_rd</code> | : 不揮発パラメータ読み出しコマンドレスポンスパラメータ |
| <code>pprm_wr</code> | : 不揮発パラメータ書き込みコマンドレスポンスパラメータ |
| <code>mem_rd</code> | : メモリ読み出しコマンドレスポンスパラメータ |
| <code>mem_wr</code> | : メモリ書き込みコマンドレスポンスパラメータ |

無効コマンド (NOP : 0x00)

| | |
|-------------------|---------------------------|
| 機能 | 無効コマンドです。スレーブは何も処理を行いません。 |
| コマンドパラメータ | なし |
| レスポンスパラメータ | なし |

パラメータ読み出しコマンド (PRM_RD : 0x01)

機能

パラメータ番号とデータサイズを指定してパラメータの読み出しを行います。

コマンドパラメータ

```
unsigned short no;  
unsigned char size;
```

レスポンスパラメータ

```
unsigned short no;  
unsigned char size;  
unsigned char parameter[56];
```

```
no          : パラメータ番号  
size       : パラメータサイズ 【byte】  
parameter  : パラメータデータ
```

- ※ : パラメータ番号とパラメータデータの内容については、各スレーブのマニュアルを参照してください。
- ※ : 標準サーボプロファイルを採用しているスレーブ（サーボパック等）の場合は、このコマンドは使用できません。標準サーボコマンドの SVPRM_RD を使用してください。

パラメータ書き込みコマンド (PRM_WR : 0x02)

機能

パラメータ番号とデータサイズ、パラメータデータを指定してパラメータの書き込みを行います。

コマンドパラメータ

```
unsigned short no;  
unsigned char size;  
unsigned char parameter[56];
```

レスポンスパラメータ

```
unsigned short no;  
unsigned char size;  
unsigned char parameter[56];
```

```
no          : パラメータ番号  
size       : パラメータサイズ (byte)  
parameter  : パラメータデータ
```

※：パラメータ番号とパラメータデータの内容については、各スレーブのマニュアルを参照してください。

※：標準サーボプロファイルを採用しているスレーブ（サーボパック等）の場合は、このコマンドは使用できません。標準サーボコマンドの SVPRM_WR を使用してください。

ID 読み出しコマンド (ID_RD : 0x03)

機能

機器 ID の読み出しコマンドです。スレーブの製品情報を ID データとして読み出します。

コマンドパラメータ

```
unsigned char id_code;  
unsigned char offset;  
unsigned short size;
```

レスポンスパラメータ

```
unsigned char id_code;  
unsigned char offset;  
unsigned short size;  
unsigned char id[56];
```

```
id_code      : ID データ選択コード  
offset       : ID 読み出しオフセット  
size        : 読み出しデータサイズ (byte)  
id          : ID データ
```

※ : ID コードと ID データの内容については、各スレーブのマニュアルを参照してください。

機器セットアップ要求コマンド (CONFIG : 0x04)

| | |
|-------------------|--|
| 機能 | スレーブ機器のセットアップを行います。実施される処理内容は製品仕様にて規定されます。 |
| コマンドパラメータ | unsigned char <code>mode</code> ; |
| レスポンスパラメータ | unsigned char <code>mode</code> ; |
| | <code>mode</code> : コンフィグモード |
| | 0 : パラメータ再計算およびセットアップ [必須項目] |
| | 1 : 不揮発メモリへの一括書き込み [任意項目] |
| | 2 : 工場出荷時パラメータ設定値に初期化 [任意項目] |

※ : コンフィグモードの対応状況については、各スレーブのマニュアルを参照してください。

 アラーム・ワーニング読み出しコマンド (ALM_RD : 0x05)

機能

アラーム・ワーニング状態の読み出し要求コマンドです。
 現在発生しているアラーム状態、ワーニング状態をアラーム、ワーニングコードで ALM_DATA に読み出します。

コマンドパラメータ

```
unsigned short mode;
unsigned short index;
```

レスポンスパラメータ

```
unsigned short mode;
unsigned short index;
unsigned char data[56];
```

mode : 読み出しモード

- 0 : 現在のアラーム・ワーニング状態読み出し [必須項目]
- 1 : アラーム履歴の読み出し [任意項目]
- 2 : 現在発生中のアラーム・ワーニング詳細読み出し [任意項目]
- 3 : アラーム履歴の詳細読み出し [任意項目]

index : アラームインデックス

0~11 : 発生順位指定 (0 指定で最新の発生アラームが読み出せません)

※注 : mode が 2 もしくは 3 の時のみ有効です。

data : アラーム・ワーニングコード

※ : アラームコードと内容については、各スレーブのマニュアルを参照してください。

アラーム・ワーニングクリアコマンド (ALM_CLR : 0x06)

機能

アラーム状態、ワーニング状態を解除します。

コマンドパラメータ

unsigned short `mode`;

レスポンスパラメータ

unsigned short `mode`;

`mode` : アラームクリアモード

0 : 現在のアラーム・ワーニング状態をクリア

1 : アラーム履歴のクリア

[必須項目]

[任意項目]

同期確立要求コマンド (SYNC_SET : 0x0D)

| | |
|-------------------|--|
| 機能 | 同期確立要求コマンドです。本コマンド完了後は同期通信となります。通信エラーなどによって非同期通信に移行した場合は、このコマンドにより同期通信へ復旧します。通信フェーズはフェーズ3へ移行します。 |
| コマンドパラメータ | なし |
| レスポンスパラメータ | なし |

コネクション確立要求コマンド (CONNECT : 0x0E)

機能

MECHATROLINK コネクション確立要求コマンドです。本コマンドの完了後 MECHATROLINK 通信によるスレーブ局の制御を開始します。同期通信にて通信開始した場合はフェーズ 3 に移行します。非同期通信にて通信開始した場合はフェーズ 2 に移行します。

コマンドパラメータ

```
unsigned char ver;
unsigned char mode;
unsigned char tim;
unsigned char profile;
```

レスポンスパラメータ

```
unsigned char ver;
unsigned char mode;
unsigned char tim;
unsigned char profile;
```

ver : MECHATROLINK アプリケーション層バージョン
[0x30 固定]

mode : 通信モード

| 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]

tim : 通信周期 (通信周期 = 設定値 × 伝送周期)
[1 ~ 255] 単位【倍】

profile : プロファイルタイプ

| profile | 内容 |
|---------------------|------------------------|
| SLV_PROFILETYPE_SRV | 0x10 : サーボパックプロファイルタイプ |
| SLV_PROFILETYPE_IO | 0x30 : I/O プロファイルタイプ |

コネクション開放要求コマンド (DISCONNECT : 0x0F)

| | |
|-------------------|-----------------------------------|
| 機能 | コネクション開放を行います。通信フェーズはフェーズ1に移行します。 |
| コマンドパラメータ | なし |
| レスポンスパラメータ | なし |

不揮発パラメータ読み出しコマンド (PPRM_RD : 0x1B)

機能

パラメータの番号とデータサイズを指定して、不揮発メモリ上のパラメータの読み出しを行います。
存在しないパラメータを指定した場合など正常に読み出しが完了しない場合は、スレーブ局にて検出し、ワーニング状態になります。

コマンドパラメータ

unsigned short **no**;
unsigned char **size**;

レスポンスパラメータ

unsigned short **no**;
unsigned char **size**;
unsigned char **parameter**[56];
no : パラメータ番号
size : パラメータデータサイズ【byte】
parameter : パラメータデータ

- ※ : パラメータ番号とパラメータデータの内容については、各スレーブのマニュアルを参照してください。
- ※ : 標準サーボプロファイルを採用しているスレーブ（サーボパック等）の場合は、このコマンドは使用できません。標準サーボコマンドの SVPRM_RD を使用してください。

不揮発パラメータ書き込みコマンド (PPRM_WR : 0x1C)

機能

パラメータの番号とデータサイズを指定して、不揮発メモリ上のパラメータの書き込みを行います。

設定範囲外の値を書き込んだ場合などで正常に書き込みが完了しない場合は、スレーブ局にて検出し、ワーニング状態になります。

コマンドパラメータ

unsigned short no;
unsigned char size;
unsigned char parameter[56];

レスポンスパラメータ

unsigned short no;
unsigned char size;
unsigned char parameter[56];
no : パラメータ番号
size : パラメータデータサイズ【byte】
parameter : パラメータデータ

※：パラメータ番号とパラメータデータの内容については、各スレーブのマニュアルを参照してください。

※：標準サーボプロファイルを採用しているスレーブ（サーボパック等）の場合は、このコマンドは使用できません。標準サーボコマンドのSVPRM_RDを使用してください。

メモリ読み出しコマンド (MEM_RD : 0x1D)

機能

メモリの読み出し先頭アドレス、読み出しデータサイズを指定して、仮想メモリ上のデータ読み出しを行います。

コマンドパラメータ

```
unsigned char mode_type;
unsigned short size;
unsigned long address;
```

レスポンスパラメータ

```
unsigned char mode_type;
unsigned short size;
unsigned long address;
unsigned char data[52];
```

`mode_type` : モード/データ型

| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|------|------|------|------|-----------|------|------|------|
| MODE | | | | DATA_TYPE | | | |

MODE : 読み出しモード

- [0 : システム予約]
- [1 : 揮発メモリ (必須項目)]
- [2 : 不揮発メモリ (任意項目)]
- [3~15 : システム予約]

DATA_TYPE : データ型

- [0 : システム予約]
- [1 : BYTE 型]
- [2 : SHORT 型]
- [3 : LONG 型]
- [4 : LONGLONG 型]
- [5~15 : システム予約]

```
size      : 読み出し数
address   : 読み出し先頭アドレス
data     : 読み出しデータ
```

※: 読み出しアドレスおよびデータの内容については、各スレーブのマニュアルを参照してください。

メモリ書き込みコマンド (MEM_WR : 0x1E)

機能

メモリの書き込み先頭アドレス、書き込みデータサイズ、書き込みデータを指定して、仮想メモリ上にデータ書き込みを行います。

コマンドパラメータ

```
unsigned char mode_type;
unsigned short size;
unsigned long address;
unsigned char data[52];
```

レスポンスパラメータ

```
unsigned char mode_type;
unsigned short size;
unsigned long address;
unsigned char data[52];
```

`mode_type` : モード/データ型

| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|------|------|------|------|-----------|------|------|------|
| MODE | | | | DATA_TYPE | | | |

MODE : 書き込みモード

- [0 : システム予約]
- [1 : 揮発メモリ (必須項目)]
- [2 : 不揮発メモリ (任意項目)]
- [3~15 : システム予約]

DATA_TYPE : データ型

- [0 : システム予約]
- [1 : BYTE 型]
- [2 : SHORT 型]
- [3 : LONG 型]
- [4 : LONGLONG 型]
- [5~15 : システム予約]

```
size      : 書き込み数
address   : 書き込み先頭アドレス
data      : 書き込みデータ
```

※: 書き込みアドレスおよびデータの内容については、各スレーブのマニュアルを参照してください。

2-6 標準サーボプロファイルコマンド

MECHATROLINK-Ⅲの標準サーボプロファイルコマンド一覧を記します。

表 2-6-1. MECHATROLINK-Ⅲ標準サーボプロファイルコマンド一覧

| コマンドコード | | コマンド | 動作 | 対応 *1 | 通信フェーズ | | |
|-----------------------|------|-------------|--------------|----------|--------|---|---|
| 定義名 | コード | | | | 1 | 2 | 3 |
| M3CMD_SRV_POS_SET | 0x20 | POS_SET | 座標系設定 | 任意 | — | ○ | ○ |
| M3CMD_SRV_BRK_ON | 0x21 | BRK_ON | ブレーキ作動要求 | 任意 | — | ○ | ○ |
| M3CMD_SRV_BRK_OFF | 0x22 | BRK_OFF | ブレーキ作動解除 | 任意 | — | ○ | ○ |
| M3CMD_SRV_SENS_ON | 0x23 | SENS_ON | センサ ON 要求 | 必須 | — | ○ | ○ |
| M3CMD_SRV_SENS_OFF | 0x24 | SENS_OFF | センサ OFF 要求 | 必須 | — | ○ | ○ |
| M3CMD_SRV_SMON | 0x30 | SMON | サーボステータスマニタ | 必須 | — | ○ | ○ |
| M3CMD_SRV_SV_ON | 0x31 | SV_ON | サーボオン | 必須 | — | ○ | ○ |
| M3CMD_SRV_SV_OFF | 0x32 | SV_OFF | サーボオフ | 必須 | — | ○ | ○ |
| M3CMD_SRV_INTERPOLATE | 0x34 | INTERPOLATE | 補間送り | 必須 | — | × | ○ |
| M3CMD_SRV_POSING | 0x35 | POSING | 位置決め | 必須 | — | ○ | ○ |
| M3CMD_SRV_FEED | 0x36 | FEED | 定速送り | 必須 | — | ○ | ○ |
| M3CMD_SRV_EX_FEED | 0x37 | EX_FEED | 外部入力位置決め定速送り | 任意 | — | ○ | ○ |
| M3CMD_SRV_EX_POSING | 0x39 | EX_POSING | 外部入力位置決め | 任意 | — | ○ | ○ |
| M3CMD_SRV_ZRET | 0x3A | ZRET | 原点復帰 | 任意 | — | ○ | ○ |
| M3CMD_SRV_VELCTRL | 0x3C | VELCTRL | 速度制御 | 任意 | — | ○ | ○ |
| M3CMD_SRV_TRQCTRL | 0x3D | TRQCTRL | トルク（推力）制御 | 任意 | — | ○ | ○ |
| M3CMD_SRV_SVPRM_RD | 0x40 | SVPRM_RD | サーボパラメータ読み出し | 必須 | — | ○ | ○ |
| M3CMD_SRV_SVPRM_WR | 0x41 | SVPRM_WR | サーボパラメータ書き込み | 必須 | — | ○ | ○ |

*1：必須：MECHATROLINK-Ⅲの通信仕様として、対応が必ず必要なコマンドです。

任意：MECHATROLINK-Ⅲの通信仕様として、対応が必ずしも必要ではなく対応するかどうかは製品仕様で決定されます。

*2：通信フェーズの記号は下記のようになります。

×：コマンド異常

○：実行可能

△：無視

—：不定な応答データ

標準サーボプロファイル用コマンドレスポンスパラメータ

```

typedef struct {
    TSERVO_COMMAND_HEADER      head;
    union {
        unsigned char          data[20];
        TSERVO_COMMAND_POS_SET pos_set;
        TSERVO_COMMAND_BRK_ON  brk_on;
        TSERVO_COMMAND_BRK_OFF brk_off;
        TSERVO_COMMAND_SENS_ON sens_on;
        TSERVO_COMMAND_SENS_OFF sens_off;
        TSERVO_COMMAND_SMON    smon;
        TSERVO_COMMAND_SV_ON   sv_on;
        TSERVO_COMMAND_SV_OFF  sv_off;
        TSERVO_COMMAND_INTERPOLATE interpolate;
        TSERVO_COMMAND_POSING  posing;
        TSERVO_COMMAND_FEED    feed;
        TSERVO_COMMAND_EX_FEED ex_feed;
        TSERVO_COMMAND_EX_POSING ex_posing;
        TSERVO_COMMAND_ZRET    zret;
        TSERVO_COMMAND_VELCTRL velctrl;
        TSERVO_COMMAND_TRQCTRL trqctrl;
        TSERVO_COMMAND_SVPRM_RD svprm_rd;
        TSERVO_COMMAND_SVPRM_WR svprm_wr;
    } data;
    unsigned char              rsv[22];
} TSERVO_COMMAND;

```

head : 標準サーボプロファイルヘッダ
data : コマンドレスポンスパラメータバッファ (共用体のベースバッファ)
pos_set : 座標系設定コマンドパラメータ
brk_on : ブレーキ作動要求コマンドパラメータ
brk_off : ブレーキ解除要求コマンドパラメータ
sens_on : センサオンコマンドパラメータ
sens_off : センサオフコマンドパラメータ
smon : サーボステータスマニタコマンドパラメータ
sv_on : サーボオンコマンドパラメータ
sv_off : サーボオフコマンドパラメータ
interpolate : 補間送りコマンドパラメータ
posing : 位置決めコマンドパラメータ
feed : 定速送りコマンドパラメータ
ex_feed : 外部入力位置決め定速送りコマンドパラメータ
ex_posing : 外部入力位置決めコマンドパラメータ
zret : 原点復帰コマンドパラメータ
velctrl : 速度制御コマンドパラメータ
trqctrl : トルク (推力) 制御コマンドパラメータ
svprm_rd : サーボパラメータ読み出しコマンドパラメータ
svprm_wr : サーボパラメータ書き込みコマンドパラメータ
rsv : リザーブ (サーボプロファイルは送受信サイズ 48byte 固定なため)

標準サーボプロファイルコマンドヘッダ

標準サーボコマンドには下記の制御フィールドと I/O フィールドを設定する必要があります。

```
typedef struct {
    unsigned long ctrl;
    unsigned long io;
} TSERVO_COMMAND_HEADER;
```

ctrl : サーボコマンド制御フィールド

| | | | | | | | |
|-----------------|-------|---------|-------|-----------|-------|------------|-----------|
| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
| Reserve | | ACCFIL | | STOP_MODE | | CMD_CANSEL | CMD_PAUSE |
| bit15 | bit14 | bit13 | bit12 | bit11 | bit10 | bit9 | bit8 |
| Reserve | | LT_SEL2 | | LT_SEL1 | | LT_REQ2 | LT_REQ1 |
| bit23 | bit22 | bit21 | bit20 | bit19 | bit18 | bit17 | bit16 |
| SEL_MON2 | | | | SEL_MON1 | | | |
| bit31 | bit30 | bit29 | bit28 | bit27 | bit26 | bit25 | bit24 |
| Vendor Specific | | | | SEL_MON3 | | | |

CMD_PAUSE : 移動コマンド一時停止
[1 : 一時停止指令 0 : なし]

CMD_CANSEL : 移動コマンド中断
[1 : 中断指令 0 : なし]

STOP_MODE : 停止モード選択
[0 : 減速停止]
[1 : 急速停止]
[2~3 : Reserve]

ACCFIL : 位置指令フィルタ
[0 : 位置指令フィルタなし]
[1 : 指数関数位置指令フィルタ]
[2 : 移動平均位置指令フィルタ]
[3 : Reserve]

LT_REQ1 : ラッチ要求 1

LT_REQ2 : ラッチ要求 2
[1 : ラッチ要求 0 : なし]

LT_SEL1 : ラッチ信号選択 1

LT_SEL2 : ラッチ信号選択 2
[0 : C相]
[0 : 外部入力信号 1]
[0 : 外部入力信号 2]
[0 : 外部入力信号 3]

SEL_MON1 : モニタ選択 1 [0 ~ 15]

SEL_MON2 : モニタ選択 2 [0 ~ 15]

SEL_MON3 : モニタ選択 3 [0 ~ 15]

詳細は「2-9 モニタ情報」を参照してください

io : サーボコマンド出力信号

| | | | | | | | |
|---------------------|-------|-------|-------|---------|-------|-------|-------|
| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
| N_CL | P_CL | P_PPI | V_PPI | Reserve | | | |
| bit15 | bit14 | bit13 | bit12 | bit11 | bit10 | bit9 | bit8 |
| Reserve | | | | G-SEL | | | |
| bit23 | bit22 | bit21 | bit20 | bit19 | bit18 | bit17 | bit16 |
| Vender Specific I/O | | | | | | | |
| bit31 | bit30 | bit29 | bit28 | bit27 | bit26 | bit25 | bit24 |
| Vender Specific I/O | | | | | | | |

V_PPI : 速度ループ P/PI 制御

P_PPI : 位置ループ P/PI 制御

[0 : PI 制御]

[1 : P 制御]

P_CL : 正転側トルク制限

N_CL : 逆転側トルク制限

[0 : トルク制限しない]

[1 : トルク制限する]

G_SEL : ゲイン切り替え

[0~15 : ゲイン切り替え]

```

typedef struct {
    TSERVO_RESPONSE_HEADER    head;
    union {
        unsigned char         data[20];
        TSERVO_RESPONSE_POS_SET    pos_set;
        TSERVO_RESPONSE_BRK_ON    brk_on;
        TSERVO_RESPONSE_BRK_OFF    brk_off;
        TSERVO_RESPONSE_SENS_ON    sens_on;
        TSERVO_RESPONSE_SENS_OFF    sens_off;
        TSERVO_RESPONSE_SMON    smon;
        TSERVO_RESPONSE_SV_ON    sv_on;
        TSERVO_RESPONSE_SV_OFF    sv_off;
        TSERVO_RESPONSE_INTERPOLATE    interpolate;
        TSERVO_RESPONSE_POSING    posing;
        TSERVO_RESPONSE_FEED    feed;
        TSERVO_RESPONSE_EX_FEED    ex_feed;
        TSERVO_RESPONSE_EX_POSING    ex_posing;
        TSERVO_RESPONSE_ZRET    zret;
        TSERVO_RESPONSE_VELCTRL    velctrl;
        TSERVO_RESPONSE_TRQCTRL    trqctrl;
        TSERVO_RESPONSE_SVPRM_RD    svprm_rd;
        TSERVO_RESPONSE_SVPRM_WR    svprm_wr;
    } data;
    unsigned char             rsv[22];
} TSERVO_RESPONSE;

```

head : 標準サーボプロファイルレスポンスヘッダ
data : コマンドレスポンスパラメータバッファ（共用体のベースバッファ）
pos_set : 座標系設定レスポンスパラメータ
brk_on : ブレーキ作動要求レスポンスパラメータ
brk_off : ブレーキ解除要求レスポンスパラメータ
sens_on : センサオンレスポンスパラメータ
sens_off : センサオフレスポンスパラメータ
smon : サーボステータスマニタレスポンスパラメータ
sv_on : サーボオンレスポンスパラメータ
sv_off : サーボオフレスポンスパラメータ
interpolate : 補間送りレスポンスパラメータ
posing : 位置決めレスポンスパラメータ
feed : 定速送りレスポンスパラメータ
ex_feed : 外部入力位置決め定速送りレスポンスパラメータ
ex_posing : 外部入力位置決めレスポンスパラメータ
zret : 原点復帰レスポンスパラメータ
velctrl : 速度制御レスポンスパラメータ
trqctrl : トルク（推力）制御レスポンスパラメータ
svprm_rd : サーボパラメータ読み出しレスポンスパラメータ
svprm_wr : サーボパラメータ書き込みレスポンスパラメータ
rsv : リザーブ（サーボプロファイルは送受信サイズ 48byte 固定なため）

標準サーボプロファイルレスポンスヘッダ

標準サーボレスポンスには、下記のステータスフィールドと I/O フィールドが返ってきます。

```
typedef struct {
    unsigned long stat;
    unsigned long io;
} TSERVO_RESPONSE_HEADER;
```

stat : サーボコマンドステータスフィールド

| | | | | | | | |
|-----------------|-------|--------|-------|----------|---------|----------------|---------------|
| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
| Reserve | | ACCFIL | | Reserve | | CMD_CANSEL_CMP | CMD_PAUSE_CMP |
| bit15 | bit14 | bit13 | bit12 | bit11 | bit10 | bit9 | bit8 |
| Reserve | | SV_ON | M_RDY | PON | POS_RDY | L_CMP2 | L_CMP1 |
| bit23 | bit22 | bit21 | bit20 | bit19 | bit18 | bit17 | bit16 |
| SEL_MON2 | | | | SEL_MON1 | | | |
| bit31 | bit30 | bit29 | bit28 | bit27 | bit26 | bit25 | bit24 |
| Vendor Specific | | | | SEL_MON3 | | | |

CMD_PAUSE_CMP : 移動コマンド一時停止完了

[1 : 一時停止完了 0 : なし]

CMD_CANSEL_CMP : 移動コマンド中断完了

[1 : 中断完了 0 : なし]

ACCFIL : 実行中の位置指令フィルタ

[0 : 位置指令フィルタなし]

[1 : 指数関数位置指令フィルタ]

[2 : 移動平均位置指令フィルタ]

[3 : Reserve]

L_CMP1 : ラッチ完了 1

L_CMP2 : ラッチ完了 2

[0 : ラッチ未完了]

[1 : ラッチ完了]

POS_RDY : 位置情報有効

[1 : 有効 0 : 無効]

PON : 位置情報有効

[1 : 主電源 ON 0 : 主電源 OFF]

M_RDY : モータ通電準備完了

[1 : 完了 0 : 未完了]

SV_ON : サーボ ON

[1 : サーボ ON 0 : サーボ OFF]

SEL_MON1 : モニタ選択 1 [0 ~ 15]

SEL_MON2 : モニタ選択 2 [0 ~ 15]

SEL_MON3 : モニタ選択 3 [0 ~ 15]

詳細は「2-9 モニタ情報」を参照してください

io : サーボコマンド入力信号

| | | | | | | | |
|---------------------|-------|-------|-------|-------|-------|--------|---------|
| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
| ESTP | EXT3 | EXT2 | EXT1 | N_OT | P_OT | DEC | Reserve |
| bit15 | bit14 | bit13 | bit12 | bit11 | bit10 | bit9 | bit8 |
| ZPOINT | PSET | NEAR | DEN | N_SOT | P_SOT | BRK_ON | Reserve |
| bit23 | bit22 | bit21 | bit20 | bit19 | bit18 | bit17 | bit16 |
| Reserve | | | | ZSPD | V_CMP | V_LIM | T_LIM |
| bit31 | bit30 | bit29 | bit28 | bit27 | bit26 | bit25 | bit24 |
| Vender Specific I/O | | | | | | | |

| | |
|--------|--------------------|
| DEC | : 原点復帰減速リミットスイッチ入力 |
| P_OT | : 正転駆動禁止入力 |
| N_OT | : 逆転駆動禁止入力 |
| EXT1 | : 第1外部ラッチ入力 |
| EXT2 | : 第2外部ラッチ入力 |
| EXT3 | : 第3外部ラッチ入力 |
| ESTP | : 非常停止入力 |
| | [1 : ON 0 : OFF] |
| BRK_ON | : ブレーキ出力 |
| | [0 : ブレーキ解除中] |
| | [1 : ブレーキ作動中] |
| P_SOT | : 正側ソフトリミット |
| N_SOT | : 逆側ソフトリミット |
| | [0 : 正常状態] |
| | [1 : ソフトリミット中] |
| DEN | : 払い出し完了 (位置制御モード) |
| | [0 : 払い出し中] |
| | [1 : 払い出し完了] |
| NEAR | : 位置決め近傍 (位置制御モード) |
| | [0 : 位置決め近傍範囲外] |
| | [1 : 位置決め近傍範囲内] |
| PSET | : 位置決め完了 (位置制御モード) |
| | [0 : 位置決め完了範囲外] |
| | [1 : 位置決め完了範囲内] |
| ZPOINT | : 原点位置 |
| | [0 : 原点位置範囲外] |
| | [1 : 原点位置範囲内] |
| T_LIM | : トルク制限 |
| | [0 : トルク制限中でない] |
| | [1 : トルク制限中] |
| V_LIM | : 速度制限 (トルク制御モード) |
| | [0 : 速度制限中でない] |
| | [1 : 速度制限中] |
| V_CMP | : 速度一致 (速度制御モード) |
| | [0 : 速度不一致] |
| | [1 : 速度一致] |
| ZSPD | : ゼロ速度 (速度制御モード) |
| | [0 : ゼロ速度未検出] |
| | [1 : ゼロ速度検出] |

座標系設定コマンド (POS_SET : 0x20)

機能

スレーブ局の座標系設定を行います。座標値の種類は POS_SEL を使い、モニタ選択コードで指定します。

本コマンドはリファレンス点設定の機能を持ちます。REFE=1 として本コマンドを指令すると、機械原点が座標系設定値に従って設定され、ストロークチェック（ソフトウェアリミット）機能などが有効になります。

コマンドパラメータ

unsigned long mode;
unsigned long data;

レスポンスパラメータ

unsigned long mode;
unsigned long data;
unsigned long monitor1;
unsigned long monitor2;
unsigned long monitor3;

mode : 座標系設定モード

| | | | | | | | |
|---------|---------|-------|-------|---------|-------|-------|-------|
| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
| REFE | Reserve | | | POS_SEL | | | |
| bit15 | bit14 | bit13 | bit12 | bit11 | bit10 | bit9 | bit8 |
| Reserve | | | | | | | |
| bit23 | bit22 | bit21 | bit20 | bit19 | bit18 | bit17 | bit16 |
| Reserve | | | | | | | |
| bit31 | bit30 | bit29 | bit28 | bit27 | bit26 | bit25 | bit24 |
| Reserve | | | | | | | |

POS_SEL : 座標系選択（モニタ選択コード用いて指定します）

[0 ~ 15 : モニタ選択コード]

詳細は「2-9 モニタ情報」を参照してください

REFE : リファレンス点有効/無効設定

[1 : 有効 0 : 無効]

data : 座標系設定値

monitor1 : モニタデータ 1

monitor2 : モニタデータ 2

monitor3 : モニタデータ 3

詳細は「2-9 モニタ情報」を参照してください

ブレーキ作動要求コマンド (BRK_ON : 0x21)

機能

ブレーキ作動信号を出力します。

コマンドパラメータ

なし

レスポンスパラメータ

unsigned long `cprm_mon1`;
unsigned long `cprm_mon2`;
unsigned long `monitor1`;
unsigned long `monitor2`;
unsigned long `monitor3`;

`cprm_mon1` : 共通モニタデータ 1
`cprm_mon2` : 共通モニタデータ 2
`monitor1` : モニタデータ 1
`monitor2` : モニタデータ 2
`monitor3` : モニタデータ 3

詳細は「2-9 モニタ情報」を参照してください

センサ ON コマンド (SENS_ON : 0x23)

機能

センサの情報初期化要求コマンドです。

コマンドパラメータ

なし

レスポンスパラメータ

unsigned long `cprm_mon1`;
unsigned long `cprm_mon2`;
unsigned long `monitor1`;
unsigned long `monitor2`;
unsigned long `monitor3`;

`cprm_mon1` : 共通モニタデータ 1
`cprm_mon2` : 共通モニタデータ 2
`monitor1` : モニタデータ 1
`monitor2` : モニタデータ 2
`monitor3` : モニタデータ 3

詳細は「2-9 モニタ情報」を参照してください

センサ OFF コマンド (SENS_OFF : 0x24)

機能

センサ電源オフ要求コマンドです。

コマンドパラメータ

なし

レスポンスパラメータ

```
unsigned long cprm_mon1;  
unsigned long cprm_mon2;  
unsigned long monitor1;  
unsigned long monitor2;  
unsigned long monitor3;
```

```
cprm_mon1 : 共通モニタデータ 1  
cprm_mon2 : 共通モニタデータ 2  
monitor1  : モニタデータ 1  
monitor2  : モニタデータ 2  
monitor3  : モニタデータ 3
```

詳細は「2-9 モニタ情報」を参照してください

サーボステータスマニタコマンド (SMON : 0x30)

機能

サーボドライブのアラーム、ステータス、モニタ選択で指定したモニタ情報や入出力信号の状態を読み出す際に使用します。

コマンドパラメータ

なし

レスポンスパラメータ

```
unsigned long cprm_mon1;  
unsigned long cprm_mon2;  
unsigned long monitor1;  
unsigned long monitor2;  
unsigned long monitor3;
```

```
cprm_mon1 : 共通モニタデータ 1  
cprm_mon2 : 共通モニタデータ 2  
monitor1  : モニタデータ 1  
monitor2  : モニタデータ 2  
monitor3  : モニタデータ 3
```

詳細は「2-9 モニタ情報」を参照してください

サーボ ON 要求コマンド (SV_ON : 0x31)

機能

サーボ ON (モータ通電) 要求コマンドです。

コマンドパラメータ

なし

レスポンスパラメータ

unsigned long `cprm_mon1`;
unsigned long `cprm_mon2`;
unsigned long `monitor1`;
unsigned long `monitor2`;
unsigned long `monitor3`;

`cprm_mon1` : 共通モニタデータ 1
`cprm_mon2` : 共通モニタデータ 2
`monitor1` : モニタデータ 1
`monitor2` : モニタデータ 2
`monitor3` : モニタデータ 3

詳細は「2-9 モニタ情報」を参照してください

サーボ OFF 要求コマンド (SV_OFF : 0x32)

機能

サーボ OFF (モータ通電解除) 要求コマンドです。

コマンドパラメータ

なし

レスポンスパラメータ

unsigned long `cprm_mon1`;
unsigned long `cprm_mon2`;
unsigned long `monitor1`;
unsigned long `monitor2`;
unsigned long `monitor3`;

`cprm_mon1` : 共通モニタデータ 1
`cprm_mon2` : 共通モニタデータ 2
`monitor1` : モニタデータ 1
`monitor2` : モニタデータ 2
`monitor3` : モニタデータ 3

詳細は「2-9 モニタ情報」を参照してください

補間送りコマンド (INTERPOLATE : 0x34)

機能

本ライブラリでは、単体で実行することはできません。補間制御用の関数を使って補間制御を行います。

位置決めコマンド (POSING : 0x35)

機能

指定した位置への位置決めを行います。位置決め速度で位置決め目標位置に位置決めします。

コマンドパラメータ

unsigned long tpos;
unsigned long tspd;
unsigned long accr;
unsigned long decr;
unsigned long tlim;

tpos : 目標位置
tspd : 目標速度
accr : 加速度
decr : 減速度
tlim : トルク制限

レスポンスパラメータ

unsigned long cprm_mon1;
unsigned long cprm_mon2;
unsigned long monitor1;
unsigned long monitor2;
unsigned long monitor3;

cprm_mon1 : 共通モニタデータ 1
cprm_mon2 : 共通モニタデータ 2
monitor1 : モニタデータ 1
monitor2 : モニタデータ 2
monitor3 : モニタデータ 3

詳細は「2-9 モニタ情報」を参照してください

※ : 各設定値の設定範囲は、各スレーブのマニュアルを参照してください。

定速送りコマンド (FEED : 0x36)

機能

指定した送り速度による、定速送りを行います。速度および方向の変更は、送り速度を変更することによって行います。

コマンドパラメータ

unsigned long `tspd`;
unsigned long `accr`;
unsigned long `decr`;
unsigned long `tlim`;

`tspd` : 目標速度
`accr` : 加速度
`decr` : 減速度
`tlim` : トルク制限

レスポンスパラメータ

unsigned long `cprm_mon1`;
unsigned long `cprm_mon2`;
unsigned long `monitor1`;
unsigned long `monitor2`;
unsigned long `monitor3`;

`cprm_mon1` : 共通モニタデータ 1
`cprm_mon2` : 共通モニタデータ 2
`monitor1` : モニタデータ 1
`monitor2` : モニタデータ 2
`monitor3` : モニタデータ 3

詳細は「2-9 モニタ情報」を参照してください

※ : 各設定値の設定範囲は、各スレーブのマニュアルを参照してください。

外部入力位置決め定速送りコマンド (EX_FEED : 0x37)

機能

指定した送り速度による、定速送り動作中に外部信号位置決め信号の入力によって外部位置決め最終走行距離の位置決めを行います。速度および方向の変更は、送り速度を変更することによって行います。

コマンドパラメータ

unsigned long `tspd`;
unsigned long `accr`;
unsigned long `decr`;
unsigned long `tlim`;

`tspd` : 目標速度
`accr` : 加速度
`decr` : 減速度
`tlim` : トルク制限

レスポンスパラメータ

unsigned long `cprm_mon1`;
unsigned long `cprm_mon2`;
unsigned long `monitor1`;
unsigned long `monitor2`;
unsigned long `monitor3`;

`cprm_mon1` : 共通モニタデータ 1
`cprm_mon2` : 共通モニタデータ 2
`monitor1` : モニタデータ 1
`monitor2` : モニタデータ 2
`monitor3` : モニタデータ 3

詳細は「2-9 モニタ情報」を参照してください

※ : 各設定値の設定範囲は、各スレーブのマニュアルを参照してください。

外部入力位置決めコマンド (EX_POSING : 0x39)

機能

外部位置決め信号の入力によって位置決めを行います。

コマンドパラメータ

unsigned long tpos;
unsigned long tspd;
unsigned long accr;
unsigned long decr;
unsigned long tlim;

tpos : 目標位置
tspd : 目標速度
accr : 加速度
decr : 減速度
tlim : トルク制限

レスポンスパラメータ

unsigned long cprm_mon1;
unsigned long cprm_mon2;
unsigned long monitor1;
unsigned long monitor2;
unsigned long monitor3;

cprm_mon1 : 共通モニタデータ 1
cprm_mon2 : 共通モニタデータ 2
monitor1 : モニタデータ 1
monitor2 : モニタデータ 2
monitor3 : モニタデータ 3

詳細は「2-9 モニタ情報」を参照してください

※ : 各設定値の設定範囲は、各スレーブのマニュアルを参照してください。

原点復帰コマンド (ZRET : 0x3A)

機能

原点LSと位置ラッチ信号を使った原点復帰を行います。位置をラッチする信号は「ラッチ信号選択」で指定します。

コマンドパラメータ

```
unsigned long mode;
unsigned long tspd;
unsigned long accr;
unsigned long decr;
unsigned long tlim;
```

mode : 原点復帰モード

| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|----------|---------|------|------|------|------|------|------|
| HOME_DIR | Reserve | | | TYPE | | | |

TYPE : 原点復帰タイプ

[0 : ラッチ信号] (必須項目)

[1 : 減速LS + ラッチ信号] (任意項目)

HOME_DIR : 原点復帰方向 (必須項目)

[0 : 正方向]

[1 : 負方向]

tspd : 目標速度

accr : 加速度

decr : 減速度

tlim : トルク制限

レスポンスパラメータ

```
unsigned long cprm_mon1;
unsigned long cprm_mon2;
unsigned long monitor1;
unsigned long monitor2;
unsigned long monitor3;
```

cprm_mon1 : 共通モニタデータ 1

cprm_mon2 : 共通モニタデータ 2

monitor1 : モニタデータ 1

monitor2 : モニタデータ 2

monitor3 : モニタデータ 3

詳細は「2-9 モニタ情報」を参照してください

※ : 各設定値の設定範囲や対応状況は、各スレーブのマニュアルを参照してください。

速度制御コマンド (VELCTRL : 0x3C)

機能

スレーブ局に速度指令を送信し、速度制御を行います。スレーブ局は、位置制御無しにダイレクトに速度制御を行います。

コマンドパラメータ

```
unsigned long    tff;
unsigned long    vref;
unsigned long    accr;
unsigned long    decr;
unsigned long    tlim;
```

```
tff      : トルク補正
vref     : 速度指令
accr     : 加速度
decr     : 減速度
tlim     : トルク制限
```

レスポンスパラメータ

```
unsigned long    cprm_mon1;
unsigned long    cprm_mon2;
unsigned long    monitor1;
unsigned long    monitor2;
unsigned long    monitor3;
```

```
cprm_mon1 : 共通モニタデータ 1
cprm_mon2 : 共通モニタデータ 2
monitor1  : モニタデータ 1
monitor2  : モニタデータ 2
monitor3  : モニタデータ 3
```

詳細は「2-9 モニタ情報」を参照してください

※ : 各設定値の設定範囲や対応状況は、各スレーブのマニュアルを参照してください。

トルク（推力）制御コマンド（TRQCTRL : 0x3D）

機能

スレーブ局にトルク（推力）指令を送信し、トルク（推力）制御を行います。スレーブ局は速度制御および位置制御を使用せずにダイレクトトルク制御を行います。

コマンドパラメータ

```
unsigned long      vlim;  
unsigned long      tqref;
```

```
vlim      : 速度制限  
tqref     : トルク（推力）指令
```

レスポンスパラメータ

```
unsigned long      cprm_mon1;  
unsigned long      cprm_mon2;  
unsigned long      monitor1;  
unsigned long      monitor2;  
unsigned long      monitor3;
```

```
cprm_mon1  : 共通モニタデータ 1  
cprm_mon2  : 共通モニタデータ 2  
monitor1   : モニタデータ 1  
monitor2   : モニタデータ 2  
monitor3   : モニタデータ 3
```

詳細は「2-9 モニタ情報」を参照してください

※：各設定値の設定範囲や対応状況は、各スレーブのマニュアルを参照してください。

サーボパラメータ読み出しコマンド (SVPRM_RD : 0x40)

機能

サーボパラメータの番号とデータサイズ、読み出しモードを指定してサーボパラメータの読み出しを行います。

コマンドパラメータ

```
unsigned short no;
unsigned char size;
unsigned char mode;
```

レスポンスパラメータ

```
unsigned short no;
unsigned char size;
unsigned char mode;
unsigned char parameter[16];
```

```
no          : サーボパラメータ番号
size       : サーボパラメータサイズ (byte)
mode       : サーボパラメータ読み出しモード
             [0x00 : 共通パラメータ RAM 領域]          (必須対応)
             [0x01 : 共通パラメータ不揮発メモリ領域] (任意対応)
             [0x10 : 機器パラメータ RAM 領域]         (任意対応)
             [0x11 : 機器パラメータ不揮発メモリ領域] (任意対応)
parameter  : サーボパラメータデータ
```

※ : パラメータ番号とパラメータデータの内容については、各スレーブのマニュアルを参照してください。

サーボパラメータ書き込みコマンド (SVPRM_WR : 0x41)

機能

サーボパラメータの番号とデータサイズ、書き込みモードを指定してサーボパラメータの書き込みを行います。

コマンドパラメータ

unsigned short `no`;
unsigned char `size`;
unsigned char `mode`;
unsigned char `parameter`[16];

レスポンスパラメータ

unsigned short `no`;
unsigned char `size`;
unsigned char `mode`;
unsigned char `parameter`[16];

`no` : サーボパラメータ番号
`size` : サーボパラメータサイズ (byte)
`mode` : サーボパラメータ書き込みモード
[0x00 : 共通パラメータ RAM 領域] (必須対応)
[0x01 : 共通パラメータ不揮発メモリ領域] (任意対応)
[0x10 : 機器パラメータ RAM 領域] (任意対応)
[0x11 : 機器パラメータ不揮発メモリ領域] (任意対応)
`parameter` : サーボパラメータデータ

※ : パラメータ番号とパラメータデータの内容については、各スレーブのマニュアルを参照してください。

2-7 標準I/Oコマンド

MECHATROLINK-Ⅲの標準 I/O プロファイルコマンド一覧を記します。

表 2-7-1. MECHATROLINK-Ⅲ標準I/Oプロファイルコマンド一覧

| コマンドコード | | コマンド | 動作 | 対応 *1 | 通信フェーズ | | |
|-------------------|------|----------|-------------------------|----------|--------|---|---|
| 定義名 | コード | | | | 1 | 2 | 3 |
| M3CMD_IO_DATA_RWA | 0x20 | DATA_RWA | データリード/ライトコマンド (非同期) | 必須 | — | ○ | ○ |
| M3CMD_IO_DATA_RWS | 0x21 | DATA_RWS | データリード/ライトコマンド (同期) | 任意 | — | × | ○ |

*1：必須：MECHATROLINK-Ⅲの通信仕様として、対応が必ず必要なコマンドです。

任意：MECHATROLINK-Ⅲの通信仕様として、対応が必ずしも必要ではなく対応するかどうかは製品仕様で決定されます。

*3：通信フェーズの記号は下記ようになります。

- ×：コマンド異常
- ：実行可能
- △：無視
- ：不定な応答データ

データ READ/WRITE_A コマンド (DATA_RWA : 0x20)

機能 I/O データの更新（非同期）を行います。

コマンドパラメータ unsigned char **out**[56]

レスポンスパラメータ unsigned char **in**[56]

out : Output データ

in : Input データ

※ : Input データ、Output データの内容については各スレーブのマニュアルを参照してください。

データ READ/WRITE_S コマンド (DATA_RWS : 0x21)

機能 I/O データの更新（同期）を行います。

コマンドパラメータ unsigned char **out**[56]

レスポンスパラメータ unsigned char **in**[56]

out : Output データ

in : Input データ

※ : Input データ、Output データの内容については各スレーブのマニュアルを参照してください。

2-8 サブコマンド

MECHATROLINK-Ⅲの標準サーボプロファイル用サブコマンド一覧を記します。

表 2-8-1. MECHATROLINK-Ⅲ標準サーボプロファイル用サブコマンド一覧

| コマンドコード | | コマンド | 動作 | 対応 *1 | 通信フェーズ | | |
|--------------------|------|----------|----------------|----------|--------|---|---|
| 定義名 | コード | | | | 1 | 2 | 3 |
| M3CMD_SUB_NOP | 0x00 | NOP | ノーオペレーション | 必須 | — | ○ | ○ |
| M3CMD_SUB_ALM_RD | 0x05 | ALM_RD | アラーム/ワーニング読み出し | 必須 | — | ○ | ○ |
| M3CMD_SUB_ALM_CLR | 0x06 | ALM_CLR | アラーム/ワーニングクリア | 任意 | — | ○ | ○ |
| M3CMD_SUB_MEM_RD | 0x1D | MEM_RD | メモリ読み出し | 任意 | — | ○ | ○ |
| M3CMD_SUB_MEM_WR | 0x1E | MEM_WR | メモリ書き込み | 任意 | — | ○ | ○ |
| M3CMD_SUB_SMON | 0x30 | SMON | サーボステータスマニタ | 必須 | — | ○ | ○ |
| M3CMD_SUB_SVPRM_RD | 0x40 | SVPRM_RD | サーボパラメータ読み出し | 任意 | — | ○ | ○ |
| M3CMD_SUB_SVPRM_WR | 0x41 | SVPRM_WR | サーボパラメータ書き込み | 任意 | — | ○ | ○ |

*1：必須：MECHATROLINK-Ⅲの通信仕様として、対応が必ず必要なコマンドです。

任意：MECHATROLINK-Ⅲの通信仕様として、対応が必ずしも必要ではなく対応するかどうかは製品仕様で決定されます。

*3：通信フェーズの記号は下記ようになります。

- ×：コマンド異常
- ：実行可能
- △：無視
- ：不定な応答データ

※：サブコマンドは標準サーボプロファイルでのみ使用できます。標準 I/O プロファイルでは使用できません。

※：サブコマンドはメインコマンドの組み合わせで実行できるコマンドとできないコマンドがあります。対応状況は各スレーブのマニュアルを参照してください。

標準サーボプロファイル用サブコマンドレスポンスパラメータ

```
typedef union {
    unsigned char          data[12];

    TSUB_COMMAND_NOP      nop;
    TSUB_COMMAND_ALM_RD   alm_rd;
    TSUB_COMMAND_ALM_CLR  alm_clr;
    TSUB_COMMAND_MEM_RD   mem_rd;
    TSUB_COMMAND_MEM_WR   mem_wr;
    TSUB_COMMAND_SMON     smon;
    TSUB_COMMAND_SVPRM_RD svprm_rd;
    TSUB_COMMAND_SVPRM_WR svprm_wr;
} TSUB_COMMAND_DATA;
```

```
typedef union {
    unsigned char          data[12];

    TSUB_RESPONSE_NOP     nop;
    TSUB_RESPONSE_ALM_RD  alm_rd;
    TSUB_RESPONSE_ALM_CLR alm_clr;
    TSUB_RESPONSE_MEM_RD  mem_rd;
    TSUB_RESPONSE_MEM_WR  mem_wr;
    TSUB_RESPONSE_SMON    smon;
    TSUB_RESPONSE_SVPRM_RD svprm_rd;
    TSUB_RESPONSE_SVPRM_WR svprm_wr;
} TSUB_RESPONSE_DATA;
```

data : コマンドレスポンスパラメータバッファ（共用体のベースバッファ）

nop : 無効サブコマンドレスポンスパラメータ

alm_rd : アラーム／ワーニング読み出しサブコマンドレスポンスパラメータ

alm_clr : アラーム／ワーニングクリアサブコマンドレスポンスパラメータ

mem_rd : メモリ読み出しサブコマンドレスポンスパラメータ

mem_wr : メモリ書き込みサブコマンドレスポンスパラメータ

smon : サーボステータスマニタサブコマンドレスポンスパラメータ

svprm_rd : サーボパラメータ読み出しサブコマンドパラメータ

svprm_wr : サーボパラメータ書き込みサブコマンドパラメータ

無効サブコマンド (NOP : 0x00)

| | |
|-------------------|---------------------------|
| 機能 | 無効コマンドです。スレーブは何も処理を行いません。 |
| コマンドパラメータ | なし |
| レスポンスパラメータ | なし |

アラーム・ワーニング読み出しサブコマンド (ALM_RD : 0x05)

機能

アラーム・ワーニング状態の読み出し要求コマンドです。
現在発生しているアラーム状態、ワーニング状態をアラーム、ワーニングコードで ALM_DATA に読み出します。

コマンドパラメータ

unsigned short `mode`;
unsigned short `index`;

レスポンスパラメータ

unsigned short `mode`;
unsigned short `index`;
unsigned char `data`[8];

`mode` : 読み出しモード

0 : 現在のアラーム・ワーニング状態読み出し [必須項目]

1 : アラーム履歴の読み出し [任意項目]

2 : 現在発生中のアラーム・ワーニング詳細読み出し [任意項目]

3 : アラーム履歴の詳細読み出し [任意項目]

`index` : アラームインデックス

0~11 : 発生順位指定 (0 指定で最新の発生アラームが読み出せませす)

※注 : mode が 2 もしくは 3 の時のみ有効です。

`data` : アラーム・ワーニングコード

※ : アラームコードと内容については、各スレーブのマニュアルを参照してください。

アラーム・ワーニングクリアサブコマンド (ALM_CLR : 0x06)

機能

アラーム状態、ワーニング状態を解除します。

コマンドパラメータ

unsigned short `mode`;

レスポンスパラメータ

unsigned short `mode`;

`mode` : アラームクリアモード

0 : 現在のアラーム・ワーニング状態をクリア

[必須項目]

メモリ読み出しサブコマンド (MEM_RD : 0x1D)

機能

メモリの読み出し先頭アドレス、読み出しデータサイズを指定して、仮想メモリ上のデータ読み出しを行います。

コマンドパラメータ

```
unsigned char mode_type;
unsigned short size;
unsigned long address;
```

レスポンスパラメータ

```
unsigned char mode_type;
unsigned short size;
unsigned long address;
unsigned char data[4];
```

`mode_type` : モード/データ型

| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|------|------|------|------|-----------|------|------|------|
| MODE | | | | DATA_TYPE | | | |

MODE : 読み出しモード

- [0 : システム予約]
- [1 : 揮発メモリ (必須項目)]
- [2 : 不揮発メモリ (任意項目)]
- [3~15 : システム予約]

DATA_TYPE : データ型

- [0 : システム予約]
- [1 : BYTE 型]
- [2 : SHORT 型]
- [3 : LONG 型]
- [4~15 : システム予約]

```
size      : 読み出し数
address   : 読み出し先頭アドレス
data     : 読み出しデータ
```

※: 読み出しアドレスおよびデータの内容については、各スレーブのマニュアルを参照してください。

メモリ書き込みサブコマンド (MEM_WR : 0x1E)

機能

メモリの書き込み先頭アドレス、書き込みデータサイズ、書き込みデータを指定して、仮想メモリ上にデータ書き込みを行います。

コマンドパラメータ

```
unsigned char mode_type;
unsigned short size;
unsigned long address;
unsigned char data[4];
```

レスポンスパラメータ

```
unsigned char mode_type;
unsigned short size;
unsigned long address;
unsigned char data[4];
```

`mode_type` : モード/データ型

| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|------|------|------|------|-----------|------|------|------|
| MODE | | | | DATA_TYPE | | | |

MODE : 書き込みモード

- [0 : システム予約]
- [1 : 揮発メモリ (必須項目)]
- [2 : 不揮発メモリ (任意項目)]
- [3~15 : システム予約]

DATA_TYPE : データ型

- [0 : システム予約]
- [1 : BYTE 型]
- [2 : SHORT 型]
- [3 : LONG 型]
- [4~15 : システム予約]

`size` : 書き込み数
`address` : 書き込み先頭アドレス
`data` : 書き込みデータ

※: 書き込みアドレスおよびデータの内容については、各スレーブのマニュアルを参照してください。

サーボステータスマニタサブコマンド (SMON : 0x30)

機能

サーボドライブのアラーム、ステータス、モニタ選択で指定したモニタ情報や入出力信号の状態を読み出す際に使用します。

コマンドパラメータ

なし

レスポンスパラメータ

unsigned long `monitor4`;
unsigned long `monitor5`;
unsigned long `monitor6`;

`monitor4` : モニタデータ 4

`monitor5` : モニタデータ 5

`monitor6` : モニタデータ 6

詳細は「2-9 モニタ情報」を参照してください

サーボパラメータ読み出しサブコマンド (SVPRM_RD : 0x40)

機能

サーボパラメータの番号とデータサイズ、読み出しモードを指定してサーボパラメータの読み出しを行います。

コマンドパラメータ

```
unsigned short no;
unsigned char size;
unsigned char mode;
```

レスポンスパラメータ

```
unsigned short no;
unsigned char size;
unsigned char mode;
unsigned char parameter[8];
```

```
no          : サーボパラメータ番号
size       : サーボパラメータサイズ (byte)
mode       : サーボパラメータ読み出しモード
             [0x00 : 共通パラメータ RAM 領域]          (必須対応)
             [0x01 : 共通パラメータ不揮発メモリ領域] (任意対応)
             [0x10 : 機器パラメータ RAM 領域]         (任意対応)
             [0x11 : 機器パラメータ不揮発メモリ領域] (任意対応)
parameter  : サーボパラメータデータ
```

※ : パラメータ番号とパラメータデータの内容については、各スレーブのマニュアルを参照してください。

サーボパラメータ書き込みサブコマンド (SVPRM_WR : 0x41)

機能

サーボパラメータの番号とデータサイズ、書き込みモードを指定してサーボパラメータの書き込みを行います。

コマンドパラメータ

```
unsigned short no;
unsigned char size;
unsigned char mode;
unsigned char parameter[8];
```

レスポンスパラメータ

```
unsigned short no;
unsigned char size;
unsigned char mode;
unsigned char parameter[8];
```

```
no          : サーボパラメータ番号
size       : サーボパラメータサイズ (byte)
mode      : サーボパラメータ書き込みモード
            [0x00 : 共通パラメータ RAM 領域]          (必須対応)
            [0x01 : 共通パラメータ不揮発メモリ領域] (任意対応)
            [0x10 : 機器パラメータ RAM 領域]         (任意対応)
            [0x11 : 機器パラメータ不揮発メモリ領域] (任意対応)
parameter  : サーボパラメータデータ
```

※ : パラメータ番号とパラメータデータの内容については、各スレーブのマニュアルを参照してください。

2-9 モニタ情報

MECHATROLINK-IIIでは、サーボコマンド制御フィールドのモニタ選択 SEL_MON1~3 と、サブコマンド制御フィールドの SEL_MON4~6 にモニタ選択コードを設定します。

標準サーボプロファイルコマンドのレスポンスデータおよび、サブコマンドの SMON のレスポンスデータで指定されたコードのモニタデータを取得することができます。

表 2-9-1 にモニタデータ一覧を示します。

表 2-9-1. モニタデータ一覧

| 選択コード | モニタ名称 | 内容 | 備考 |
|-------|-------|---------------|---------------------------------|
| 0 | APOS | フィードバック位置 | モータ現在位置 |
| 1 | CPOS | 指令位置 | 加減速フィルタ後の指令位置 |
| 2 | PERR | 位置偏差 | 制御ループの位置偏差 |
| 3 | LPOS1 | ラッチ位置 1 | ラッチ信号によってラッチされたモータ位置 1 |
| 4 | LPOS2 | ラッチ位置 2 | ラッチ信号によってラッチされたモータ位置 2 |
| 5 | FSPD | フィードバック速度 | モータの現在速度 |
| 6 | GSPD | 指令速度 | モータへの指令速度 |
| 7 | TRQ | 指令トルク | モータへの指令トルク |
| 8 | ALARM | 現在発生アラームの詳細情報 | 現在のアラーム・ワーニング |
| 9 | MPOS | 指令位置 (任意) | モニタデータの内容は製品仕様にて規定され ます |
| 10 | - | 予約 | |
| 11 | - | 予約 | |
| 12 | CMN1 | 共通モニタ 1 | 共通パラメータ 0x89 で指定したモニタデータ を選択 |
| 13 | CMN2 | 共通モニタ 2 | 共通パラメータ 0x8A で指定したモニタデータ を選択 |
| 14 | OMN1 | オプションモニタ 1 | パラメータで指定したモニタデータを選択 |
| 15 | OMN2 | オプションモニタ 2 | パラメータで指定したモニタデータを選択 |

※ モニタ情報の詳細は、各スレーブのマニュアルを参照してください。

2-10 アラーム情報

● D_ALM

スレーブ局の機器アラーム状態を示すビットです。

COMM_ALM および CMD_ALM 以外の機器固有のアラームが発生したとき D_ALM=1 となります。COMM_ALM と CMD_ALM とは独立しています。

ALM_CLR コマンドにより、スレーブ局が機器アラーム状態から正常状態へ遷移すると、D_ALM=0 となります。

● D_WAR

スレーブ局の機器ワーニング状態を示すビットです。

COMM_ALM および CMD_ALM 以外の機器固有のワーニングが発生したとき D_WAR=1 となります。COMM_ALM と CMD_ALM とは独立しています。

ALM_CLR コマンドにより、スレーブ局が機器ワーニング状態から正常状態へ遷移すると、D_WAR=0 となります。

● CMD_ALM

コマンド異常状態を通知します。

コマンド異常を示すコードが設定されます。COMM_ALM および D_ALM、D_WAR とは独立しています。

コマンド異常発生後、正常なコマンドを受け付けた場合には、CMD_ALM は自動的にクリアされます。

CMD_ALM が 0 以外のときでも、通信フェーズは変化しません。

CMD_ALM のワーニング/アラームの分類は製品仕様にて規定されます。

CMD_ALM のワーニング・アラームコード一覧を表 2-10-1 に示します。

表 2-10-1. CMD_ALM のワーニング・アラームコード一覧

| コード | 内容 | 備考 | |
|-------|------|---------------|---|
| 0x0 | 正常 | | |
| ワーニング | 0x1 | データ範囲外 | ワーニング状態を通知するが、指令された値もしくは最大値、最小値でクランプされた値で動作します。 |
| | 0x2 | | |
| | 0x3 | | |
| | 0x4 | | |
| | 0x5 | | |
| | 0x6 | | |
| | 0x7 | | |
| | アラーム | 0x8 | |
| 0x9 | | データ範囲外 | |
| 0xA | | コマンド実行条件異常 | |
| 0xB | | サブコマンド組み合わせ異常 | |
| 0xC | | フェーズ異常 | |
| 0xD | | | |
| 0xE | | | |
| 0xF | | | |

● COMM_ALM

通信異常状態を通知します。CMD_ALM および D_ALM、D_WAR とは独立しています。

MECHATROLINK 通信の異常状態を示すコードです。

COMM_ALM は ALM_CLR コマンド実行でクリアされます。

INI ファイルにてマスタプロセスの通信異常自動復帰機能を有効とした場合、通信が復帰すれば自動的に復帰されます。

COMM_ALM のワーニング・アラームコード一覧を表 2-10-2 に示します。

表 2-10-2. COMM_ALM のワーニング・アラームコード一覧

| コード | 内容 | 備考 | |
|-------|------|-----------|--|
| 0x0 | 正常 | | |
| ワーニング | 0x1 | FCS 異常 | フレームチェックシーケンスで異常を検出 |
| | 0x2 | 指令データ未受信 | 自局宛の指令データが受信できなかった |
| | 0x3 | 同期フレーム未受信 | 同期フレームが受信できなかった |
| | 0x4 | | |
| | 0x5 | | |
| | 0x6 | | |
| | 0x7 | | |
| | アラーム | 0x8 | FCS 異常 |
| 0x9 | | 指令データ未受信 | 通信フェーズが同期通信 (3) の場合は、非同期通信 (2) へ移行します。 |
| 0xA | | 同期フレーム未受信 | |
| 0xB | | 同期間隔異常 | 8, 9, A は上記ワーニングが指定された回数連続して起こった場合に設定されます。 |
| 0xC | | WDT 異常 | B, C は 1 回の異常発生で即設定されます。 |
| 0xD | | | |
| 0xE | | | |
| 0xF | | | |

2-11 エラーコード一覧

関数戻り値エラーコード一覧

表 2-11-1. 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 |
|--------------------|--------|------------------------------------|
| ML_ER_OK | 0x0000 | 正常です。 |
| ML_ER_ALREADYOPEN | 0x0001 | すでにオープンしています。 |
| ML_ER_NOMSTPROC | 0x0002 | MECHATROLINK-III マスタプロセスが起動していません。 |
| ML_ER_INVALIDPARAM | 0x0003 | 無効な引数です。 |
| ML_ER_NOTOPEN | 0x0004 | オープンしていません。 |
| ML_ER_ALREADYCOMM | 0x0005 | すでに通信開始されています。 |
| ML_ER_NOTCOMM | 0x0006 | 通信していません。 |
| ML_ER_NOTRESET | 0x0007 | リセットされていません。 |
| ML_ER_NOTSYNC | 0x0008 | 同期通信されていません。 |
| ML_ER_COMMINITERR | 0x0009 | 通信初期化エラーです。 |
| ML_ER_NOSETCMD | 0x000A | コマンドセットされていません。 |
| ML_ER_NORESPONSE | 0x000B | レスポンスなしです。 |
| ML_ER_ERROR | 0x0101 | 内部エラーです。 |
| ML_ER_CREATE | 0x0102 | 各種デバイス生成失敗です。 |
| ML_ER_TIMEOUT | 0x0103 | タイムアウトエラーです。 |

チップエラーコード一覧

表 2-11-2. 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。

76DLH0050C
76DLH0050A

2014年 10月 第3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/>