標準 SLM4000システム対応

送 受 信 デ ー タ 説 明 書

Ｖｅｒ１．６

2009.6.29

－　目　次　－

１．概要

標準ＳＬＭ－４０００コントローラ（以下ＳＬＭユニット）は、弊社通信ライブラリを利用する事により、データの送受信・各種動作をパソコンから指令して行わせる事ができます。
利用できる通信Ｉ／Ｆは、ＲＳ２３２Ｃ／ＵＳＢとなります。
通信ライブラリの詳細は「標準ＳＬＭ対応 通信ライブラリリファレンスマニュアル(TB00-0803)」を参照下さい。

本説明書は、通信ライブラリの下記の関数にて使用するデータのリファレンスです。

SendData(...)　　：データ送信関数
「3-1.送受信データ詳細」に記述しているデータの内、**～書込** となっているデータを送信する事が出来ます。

ReceiveData(...)：データ受信関数
「3-1.送受信データ詳細」に記述しているデータの内、**～読込** となっているデータを受信する事が出来ます。

SendCommand(...)：動作要求コマンド送信関数
「3-2.動作要求コマンド詳細」に記述しているコマンドを送信する事ができます。

本説明書中に記載されているデータ構造体／データタイプは、通信ライブラリに付属のＣ言語ヘッダーファイル”ＳＬＭＤＡＴＡ．Ｈ”／”ＳＹＮＣＤＥＦ．Ｈ”にて、定義されています。

２．関連資料

「ＳＬＭ－４０００ユーザーズマニュアル」	（ＴＢ００－０８００）
「標準ＳＬＭ対応 セッティングＰＣマニュアル」	（ＴＢ００－０８０２）
「標準ＳＬＭ対応 ＲＯＭＳＷ設定ソフトマニュアル」	（ＴＢ００－０８０１）
「標準ＳＬＭ対応 通信ライブラリリファレンスマニュアル」	（ＴＢ００－０８０３）
「標準ＳＬＭ対応 テクノコード変換ライブラリリファレンスマニュアル」	（ＴＢ００－０８０５）
「標準ＳＬＭ対応 Ｇコード変換ライブラリリファレンスマニュアル」	（ＴＢ００－０８０６）

３．通信データ詳細説明

3-1.データ送受信機能詳細

データ送信機能は、パソコンよりＳＬＭユニットに各種データの設定等を行う機能です。
通信ライブラリのSendData()／ReceiveData()関数を使用して、ＳＬＭユニットへ送信(書込)／受信(読出)を行うことが出来ます。

3-1-1.サーボパラメータ書込／読出 [DAT_PARAMETER(0x00)]

《データタイプ》
　ＤＡＴ＿ＰＡＲＡＭＥＴＥＲ（０ｘ００）

《パラメータ》
　なし

《機能》
　ＳＬＭユニットにサーボパラメータ書込／読出を行うものです。

《書き込み実行条件》
　・セッティングモード

《読み込み実行条件》
　常時実行可能です。

《データフォーマット》

サーボパラメータ １００バイト×９
予　　備 １２０バイト

各軸のサーボパラメータのフォーマットは以下の通りです。

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	Ｄ／Ａ出力ゲイン	4ﾊﾞｲﾄ	0.04～256.0
００００４	ＩＮＰＯＳ量	4ﾊﾞｲﾄ	0～10000[Pluse]
００００８	ＥＲ上限値	4ﾊﾞｲﾄ	100～100000[Pluse]
００００Ｃ	ＥＲサチレーション	4ﾊﾞｲﾄ	100～100000[Pluse]
００１０	ＰＴＰ時定数	4ﾊﾞｲﾄ	20～3000[ms]
００１４	ＰＴＰ速度	4ﾊﾞｲﾄ	1000～4000000[Pluse/Sec]
００１８	ＪＯＧ送り速度	4ﾊﾞｲﾄ	1000～4000000[Pluse/Sec]
００１Ｃ	補間時定数	4ﾊﾞｲﾄ	-1000～1000[ms]
００２０	＋側ソフトリミット	4ﾊﾞｲﾄ	1000～1000000[Pluse]
００２４	－側ソフトリミット	4ﾊﾞｲﾄ	1000～1000000[Pluse]
００２８	原点距離	4ﾊﾞｲﾄ	100～1000000[Pluse]
００２Ｃ	アプローチ速度	4ﾊﾞｲﾄ	1000～4000000[Pluse/Sec]
００３０	原点復帰方向	4ﾊﾞｲﾄ	0:無，1:+2段，2:-2段，3:+1段，4:-1段
００３４	原点復帰順位	4ﾊﾞｲﾄ	0～軸数-1
００３８	原点復帰逃げ量	4ﾊﾞｲﾄ	1000～1000000[Pluse]
００３Ｃ	バックラッシュ補正量	4ﾊﾞｲﾄ	0～30000　[Pluse]
００４０	原点復帰早送り速度	4ﾊﾞｲﾄ	1000～4000000[Pluse/Sec]
００４４	形状補正係数	4ﾊﾞｲﾄ	0～2000
００４８	Ｓ字加減速時定数	4ﾊﾞｲﾄ	0～25×RTC周期［msec］
００４Ｃ	ホームポジション	4ﾊﾞｲﾄ	0～1000000[Pluse]
００５０	ホームポジション位置決め順位	4ﾊﾞｲﾄ	0～軸数-1
００５４	予　　備	１６ﾊﾞｲﾄ	－

・同期軸制御オプションありの場合、同期軸のパラメータ設定については、
「②同期軸パラメータ」を参照して下さい。

・接線制御オプションありの場合、接線制御軸のパラメータ設定については、
「③接線制御軸パラメータ」を参照して下さい。

①位置制御軸パラメータ

（１） Ｄ／Ａ出力ゲイン**（※）**
ＥＲ（偏差パルス）に本設定値を乗算した値が、Ｄ／Ａへの出力値となります。
サーボのＫｐと比例関係にありますので、本設定値を変更することによりサーボの
Ｋｐをソフト的に調整する事ができます。

固定小数点型４バイトデータです。
（上位バイト：整数部 ／ 下位バイト：小数部）
【設定例】 指定値 設定値（ｈｅｘ）
１．０ → ０００００１００ｈ
０．５ → ０００００００８０ｈ
２．７ → ０００００２Ｂ３ｈ

（２） ＩＮＰＯＳ量
インポジションの範囲を、パルス単位の絶対値量で指定します。

（３） ＥＲ上限値**（※）**
偏差の絶対量の上限値を指定します。
ＥＲが本設定値を越えた場合、偏差過大アラームとなります。

（４） ＥＲサチレーション**（※）**
Ｄ／Ａ出力に用いる、偏差量（絶対値）のクランプ値を指定します。
ＥＲが本設定値を越えてかつ、ＥＲ上限まで達していない場合、Ｄ／Ａ出力基数
（通常ＥＲ）に本設定値を使用します。

（５） ＰＴＰ時定数
ＰＴＰ移動時、及びＪＯＧ移動時の直線加減速の時定数を指定します。
移動速度１００Ｋｐｐｓに達するまでの時間で設定します。

（６） ＰＴＰ速度
ＰＴＰ移動時の速度を指定します。

（７） ＪＯＧ速度
ＪＯＧ移動時の速度を指定します。

（８） 補間時定数
補間移動時の指数型加減速の時定数を指定します。
直線加減速オプション有りの時、マイナスの値を入れることで、
直線加減速時定数を指定します。

（９） ソフトリミット
ソフトウェアストロークリミットを、各方向（＋／－）別に指定します。
ポジション値が本設定値を越えた場合、ソフトリミットアラームとなります。

（１０）原点距離
Ｃ相パルス発生位置から機械固有原点までのオフセット量を指定します。

（１１）原点復帰アプローチ速度
原点復帰処理にて、原点ＬＳを通過して、原点へアプローチを行う際の速度を指定します。

（１２）原点復帰方向
原点復帰動作の有無、及び有りの場合の原点復帰方向を指定します。

（ １３ ）原点復帰順位
原点復帰入力にて、全軸原点復帰を行う際の原点復帰順位を指定します。

※ ＳＬＭユニットでは使用しません。 これらのパラメータは将来用です。

（ １４ ）原点復帰逃げ量
全軸原点復帰時の、機械原点位置からの逃げ量を指定します。

（ １５ ）バックラッシュ補正量
モータ回転方向変化時の、バックラッシュの補正量を指定します。

（ １６ ）原点復帰早送り速度
２段原点復帰処理にて、原点ＬＳまでの速度を指定します。

（ １７ ）形状補正係数 〈ｵﾌﾟｼｮﾝ〉

（ １８ ）Ｓ字加減速時定数 〈ｵﾌﾟｼｮﾝ〉
Ｓ字加減速の時定数を設定します。
Ｓ字加減速の詳細については「SLM-4000 ﾕｰｻﾞｰｽﾞﾏﾆｭｱﾙ(TB00-0800)」〈Ⅲ 機能編 4-3.補間加減速〉を参照下さい。

（ １９ ）ホームポジション
ホームポジションを指定します。（機械座標系）

（ ２０ ）ホームポジション位置決め順位
ホームポジション位置決めする順番を０からの連番で設定します。

②同期軸パラメータ ＜オプション＞

（ １ ） Ｄ／Ａゲイン
同期軸のＤ／Ａゲインを指定します。
設定値は、０．０４～２５６．０(8bit.8bitの固定小数点(下位２バイトのみ使用))です。

（ ２ ） ＥＲ上限値
同期偏差上限値を指定します。
設定値は、１～３００００［ﾊﾟﾙｽ］です。

※Ｄ／Ａゲイン以外の、軸制御に必要なパラメータは、Ｘ軸のパラメータを使用します。

③接線制御軸パラメータ ＜オプション＞

（ １ ） ＋方向ソフトリミット
接線制御軸の接線速度上限速度を指定します。
設定値は、１０００～４０００００００［Pluse/Sec］です。

（ ２ ） －方向ソフトリミット
接線制御軸の接線制御基準位置を指定します。
設定値は、０～メカ機構１回転パルス数［ﾊﾟﾙｽ］です。

※上記以外のパラメータについては、「①位置制御軸パラメータ」を参照して下さい。

《構造体定義》

```
/*************************************************************************/
/*  パラメータデータ構造体（１０２０ﾊﾞｲﾄ固定長）                         */
/*************************************************************************/
typedef struct {    /* 各軸独立パラメータ構造体（１００ﾊﾞｲﾄ固定長）      */
    long            Gain;               /* Ｄ／Ａ出力ゲイン              */
    long            InPos;              /* ＩＮＰＯＳ量                  */
    long            ErMax;              /* ＥＲ上限値                    */
    long            ErSat;              /* ＥＲサチレーション            */
    long            Dx;                 /* ＰＴＰ時定数[msec]            */
    long            PtpFeed;            /* ＰＴＰ速度[P/sec]             */
    long            JogFeed;            /* ＪＯＧ送り速度[P/sec]         */
    long            Ka;                 /* 補間時定数[msec]              */
    long            SoftLimP;           /* ソフトリミット＋側            */
    long            SoftLimM;           /* ソフトリミットー側            */
    long            OrgOfs;             /* 原点距離                      */
    long            AprFeed;            /* アプローチ速度[P/sec]         */
    long            OrgDir;             /* 原点復帰方向                  */
    long            OrgPri;             /* 原点復帰順位                  */
    long            OrgPos;             /* 原点復帰逃げ量                */
    long            BackL;              /* バックラッシュ補正量          */
    long            OrgFeed;            /* 原点復帰早送り速度[P/sec]     */
    long            Revise;             /* 形状補正係数                  */
    long            SKa;                /* Ｓ字補間時定数[msec]          */
    long            Homepos;            /* ﾎｰﾑﾎﾟｼﾞｼｮﾝ位置               */
    long            Homepri;            /* ﾎｰﾑﾎﾟｼﾞｼｮﾝ順位               */
    unsigned char   Reserved[16];       /* 未使用                        */
} AXIS_PARAMETER;

typedef struct {
    AXIS_PARAMETER  AxisParam[9];       /* 各軸パラメータ                */
    unsigned char   Reserved[120];      /* 未使用                        */
} PARAMETER_DATA;
```

3-1-2.動作プログラム書込／読出 [DAT_PROGRAM(0x01)]

《データタイプ》
ＤＡＴ＿ＰＲＯＧＲＡＭ（０ｘ０１）

《パラメータ》
プログラム番号（１～１２）

《機能》
動作プログラムを、ＳＬＭユニットのプログラムメモリに書き込み／読み出しを行うものです。
動作プログラムは、テクノ独自フォーマットのバイナリデータで、テキストプログラム処理ライブラリを利用して、テキストデータ⇔バイナリデータへ、変換する事ができます。
動作プログラムの書込時に、格納するプログラム番号を、通信ライブラリ関数のパラメータにて指定します。
指定できるプログラム番号は、プログラム容量により、下表の通りとなります。

	ﾌﾟﾛｸﾞﾗﾑ番号指定範囲	ﾌﾟﾛｸﾞﾗﾑﾊﾞｯﾌｧ容量
888ｽﾃｯﾌﾟ仕様	１～ ３	65535ﾊﾞｲﾄ
444ｽﾃｯﾌﾟ仕様	１～ ６	32767ﾊﾞｲﾄ
222ｽﾃｯﾌﾟ仕様	１～１２	16383ﾊﾞｲﾄ

動作プログラム書込時のプログラム番号指定にて、ＳＬＭユニットが実行中のプログラム番号を指定した場合、動作プログラムの書込は行えません。

プログラムデータは、ＳＬＭユニットに対する動作コマンドの集合体で、作業に必要な動作コマンドをまとめて設定しておく事により、ＳＬＭユニット単体で自動運転を可能にするものです。

《書き込み実行条件》
・指定したプログラム番号が未実行

《読み込み実行条件》
常時実行可能です。

《データフォーマット》

動作プログラムバイナリデータバッファ

（１） 動作プログラムバイナリデータバッファ
必要なバッファ容量は、プログラム容量により変わります。
上表の”ﾌﾟﾛｸﾞﾗﾑﾊﾞｯﾌｧ容量”のサイズのバッファを確保してください。

3-1-3.ＤＮＣデータ書込 [DAT_DNCDATA(0x0e)] 〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＤＡＴ＿ＤＮＣＤＡＴＡ（０ｘ０ｅ）

《パラメータ》
先頭／継続フラグ（０：先頭、１：継続）

《機能》
動作プログラムを、ＳＬＭユニットのプログラムメモリに書き込むものです。
本コマンドの実行には、ＳＬＭユニットにＤＮＣ運転オプションが追加されている必要があります。
ＤＮＣデータの書込は、ＳＬＭユニットの動作モードがＤＮＣ運転モードの時のみ実行可能です。
本データは、テクノ独自フォーマットのバイナリーデータで、テキストプログラム処理ライブラリを利用して、テキストデータより作成する事ができます。

《書き込み実行条件》
・ＤＮＣ運転モード
・データサイズがＤＮＣバッファ空き容量以下

《読み込み実行条件》
読み込み不可

《データフォーマット》
「3-1-2. 動作プログラム書込／読出」と同様です。

3-1-4.マルチタスクプログラム書込／読出 [DAT_TASKPROG(0x17)] 〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＤＡＴ＿ＴＡＳＫＰＲＯＧ（０ｘ１７）

《パラメータ》
タスク番号（０～６）

《機能》
マルチタスク用の動作プログラムを、ＳＬＭユニットのプログラムメモリに書き込み／読み出しを行うものです。
本コマンドの実行には、ＳＬＭユニットにマルチタスクオプションが追加されている必要があります。
本データは、テクノ独自フォーマットのバイナリーデータで、テキストプログラム処理ライブラリを利用して、テキストデータより作成する事ができます。

動作プログラムの書込時に、格納するタスク番号を、通信ライブラリ関数のパラメータにて指定します。
指定できるタスク番号は下表の通りになります。

タスク番号	指定タスク	
０ (MSTTASK)	マスタータスク	※１
１ (SLVTASK)	スレーブタスク	※１
２ (BGTASK)	ﾊﾞｯｸｸﾞﾗﾝﾄﾞﾀｽｸ	※２
３ (RSTTASK)	リセットタスク	※２
４ (ALMTASK)	アラームタスク	※２
５ (EXTTASK)	Ｅｘｉｔタスク	※２
６ (INTTASK)	割り込みタスク	※２

※１　ＳＬＭユニット内で選択されているプログラム番号への書き込み／読み出しです。
（プログラム番号を選択できない以外は、「3-1-2. 動作プログラム書込／読出」と同様です。）

《書き込み実行条件》
・指定したタスクが未実行状態

《読み込み実行条件》
常時実行可能です。

《データフォーマット》
「3-1-2. 動作プログラム書込／読出」と同様です。
（※２のタスクのバッファサイズは8192バイト（200ｽﾃｯﾌﾟ）です。）

3-1-5.ピッチエラー補正用パラメータ書込／読出 [DAT_PITCHERR(0x10)] 〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＤＡＴ＿ＰＩＴＣＨＥＲＲ（０ｘ１０）

《パラメータ》
なし

《機能》
ＳＬＭユニットにピッチエラー補正用データの書込／読出を行うものです。
本コマンドの実行には、ＳＬＭユニットにピッチエラー補正オプションが追加されている必要があります。
ピッチエラー補正データ書込は、ＳＬＭユニットの動作モードがセッティングモードの時のみ実行が可能です。

《書き込み実行条件》
・セッティングモード

《読み込み実行条件》
常時実行可能です。

《データフォーマット》

各軸補正用パラメータ ２０バイト×９
補正データ １０００バイト

各軸補正用パラメータのフォーマットは以下の通りです。

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	補正倍率	４バイト	0～10
０００４	補正間隔	４バイト	1000～1000000
０００８	補正データ先頭番号	４バイト	0～999
００１０	－側補正区間数	４バイト	0～1000
００１４	＋側補正区間数	４バイト	0～1000

補正用データのフォーマットは以下の通りです。

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	補正データ	１バイト	-127～127

《構造体定義》

```
/*************************************************************************/
/*  ピッチエラー補正用パラメータ情報構造体                               */
/*************************************************************************/
typedef struct {                /* 各軸補正用パラメータ                  */
    unsigned long   RevMagnify;     /* 補正倍率                          */
    unsigned long   RevSpace;       /* 補正間隔                          */
    unsigned long   RevTopNo;       /* 補正データ先頭番号                */
    unsigned long   RevMCnt;        /* －側補正区間数                    */
    unsigned long   RevPCnt;        /* ＋側補正区間数                    */
} REV_AX;

typedef struct {                /*  ピッチエラー補正用パラメータ         */
    REV_AX          RevAx[9];       /* 各軸補正用パラメータ              */
    char            RevDt[1000];    /* 補正データ                        */
} PITCH_ERR_REV;
```

3-1-6.工具長補正データ書込／読出 ［DAT_TOOLHOFS(0x12)］ 〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＤＡＴ＿ＴＯＯＬＨＯＦＳ（０ｘ１２）

《パラメータ》
なし

《機能》
ＳＬＭユニットに工具長補正用データの書込／読出を行うものです。
本コマンドの実行には、ＳＬＭユニットに工具長補正オプションが追加されている必要があります。
工具長補正データ書込は、ＳＬＭユニットの動作モードがセッティングモードの時のみ実行が可能です。

《書き込み実行条件》
・セッティングモード

《読み込み実行条件》
常時実行可能です。

《データフォーマット》

各補正用データ ８０バイト

各補正用データのフォーマットは、以下の通りです。

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	補正用データ	４バイト	-99999999～99999999

《構造体定義》

```
/*****************************************************************************/
/*  工具長補正用パラメータ情報構造体                                         */
/*****************************************************************************/
typedef struct {
    long            h[20];              /* 工具長補正データ                 */
} TOOL_H;
```

3-1-7.ポジション・ステータスデータ読出 [DAT_STATUS(0x02)]

《データタイプ》
ＤＡＴ＿ＳＴＡＴＵＳ（０ｘ０２）

《パラメータ》
なし

《機能》
ＳＬＭユニットの動作状態（アラーム状態を含む）や各軸の現在値などのデータです。
有効軸以外の各軸のデータは、無効データです。

《読み込み実行条件》
常時実行可能です。

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	各種状態情報	２バイト	―――
０００２	第１軸ｽﾃｰﾀｽ情報	２バイト	―――
０００４	第２軸ｽﾃｰﾀｽ情報	２バイト	―――
０００６	第３軸ｽﾃｰﾀｽ情報	２バイト	―――
０００８	第４軸ｽﾃｰﾀｽ情報	２バイト	―――
０００Ａ	第５軸ｽﾃｰﾀｽ情報	２バイト	―――
０００Ｃ	ｱﾗｰﾑ情報	２バイト	―――
０００Ｅ	第１軸ｱﾗｰﾑ情報	２バイト	―――
００１０	第２軸ｱﾗｰﾑ情報	２バイト	―――
００１２	第３軸ｱﾗｰﾑ情報	２バイト	―――
００１４	第４軸ｱﾗｰﾑ情報	２バイト	―――
００１６	第５軸ｱﾗｰﾑ情報	２バイト	―――
００１８	第１軸ﾎﾟｼﾞｼｮﾝﾃﾞｰﾀ	20バイト	―――
００２Ｃ	第２軸ﾎﾟｼﾞｼｮﾝﾃﾞｰﾀ	20バイト	―――
００４０	第３軸ﾎﾟｼﾞｼｮﾝﾃﾞｰﾀ	20バイト	―――
００５４	第４軸ﾎﾟｼﾞｼｮﾝﾃﾞｰﾀ	20バイト	―――
００６８	第５軸ﾎﾟｼﾞｼｮﾝﾃﾞｰﾀ	20バイト	―――
００７Ｃ	送りｵｰﾊﾞｰﾗｲﾄﾞ設定	１バイト	0～7
００７Ｄ	選択･実行ﾌﾟﾛｸﾞﾗﾑ番号	１バイト	1～4
００７Ｅ	待機･実行ｽﾃｯﾌﾟ番号	２バイト	1～999

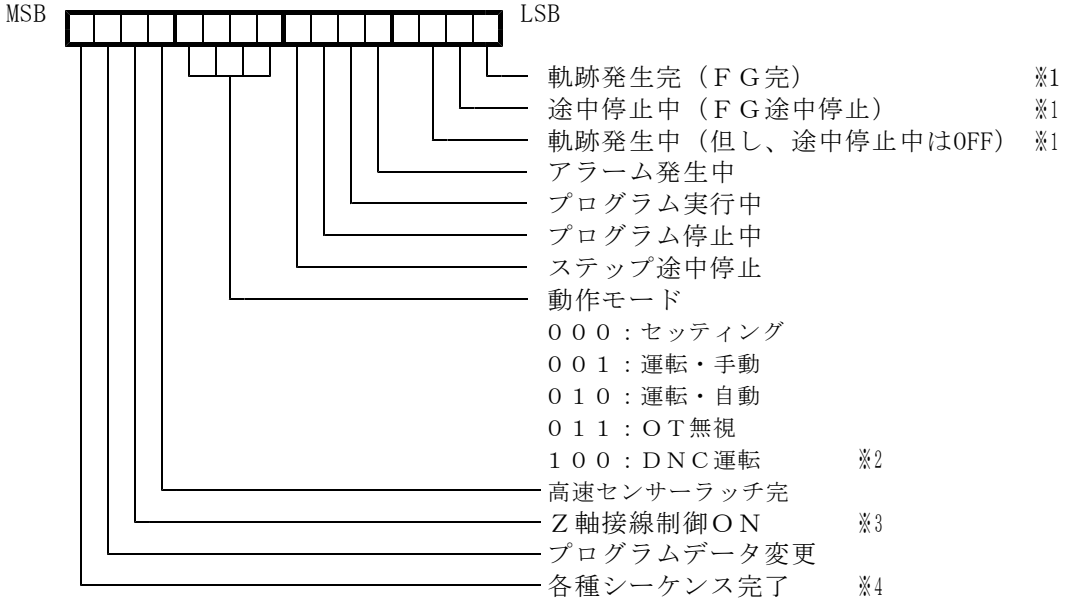
各軸ポジションデータのフォーマットは以下の通りです。

ｵﾌｾｯﾄ	データ名称	データ長	設定値	
００００	指令位置	４バイト	-99999999～99999999[Pulse]	
０００４	機械位置	４バイト	-99999999～99999999[Pulse]	
０００８	偏差量	４バイト	-99999999～99999999[Pulse]	
０００Ｃ	最新ﾌﾞﾛｯｸ払出量	４バイト	-99999999～99999999[Pulse]	※１
００１０	絶対位置	４バイト	-99999999～99999999[Pulse]	

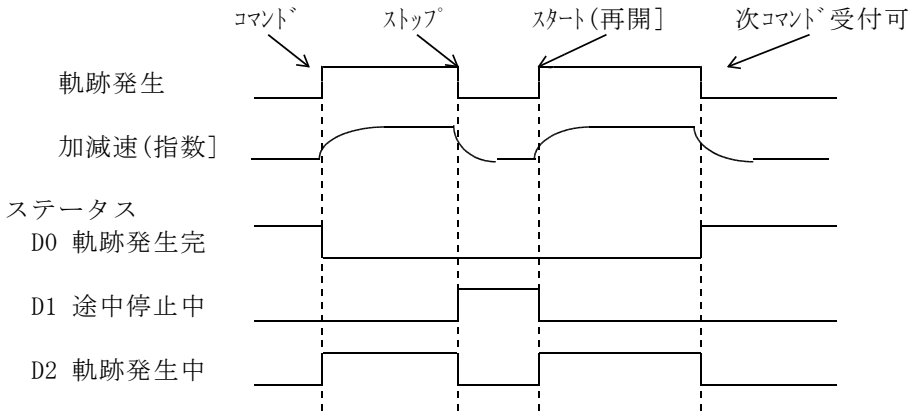
※１　原点復帰完了時には、「原点ドグ」と「Ｃ相位置」との距離（パルス）が返ります。
原点ドグ位置の調整に使用します。
「ＳＬＭ-４０００ ユーザーズマニュアル(TB00-0800]」＜Ⅳ 試運転・調整編 ７－８ 原点復帰＞を参照して下さい。

（ １ ） 各種状態情報

ＳＬＭユニットの現在の状態を示すものです。



※1　ＦＧステータスタイミングチャート



※2 ＤＮＣオプション有りの時

※3 Ｚ軸接線制御ｵﾌﾟｼｮﾝ有りの時

※4 以下の条件が成立しているときにＯＮします。（ＡＮＤ条件）

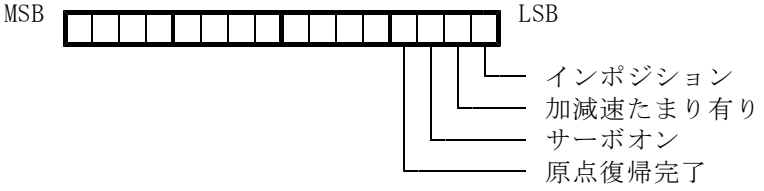
・ＦＧ完

・Ｍコード出力シーケンス未実行

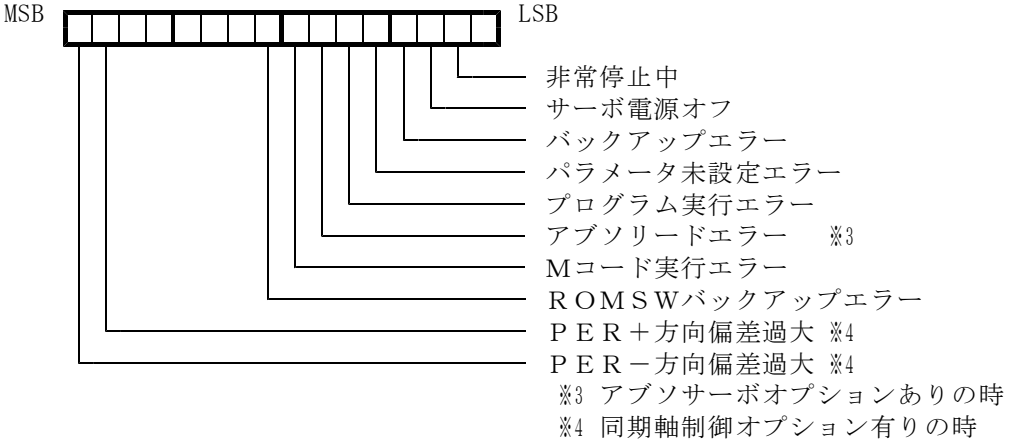
・原点復帰未実行

・ホーム位置決め未実行

（ ２ ） 第１軸～第５軸ステータス情報
各軸の現在の状態を示すものです。

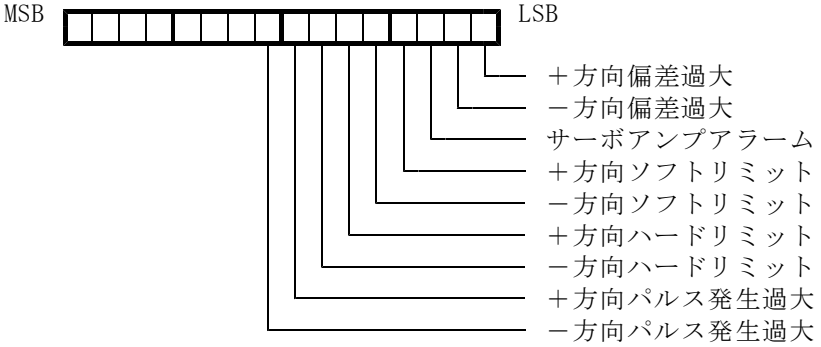


（ ３ ） アラーム情報
ＳＬＭユニットのアラームの要因を示すものです。



※3 アブソサーボオプションありの時
※4 同期軸制御オプション有りの時

（ ４ ） 第１軸～第５軸アラーム情報
各軸のアラームの要因を示すものです。



（ ５ ） 第１軸～第５軸指令位置
論理座標系*における、ＳＬＭユニットから出力した指令位置です。

（ ６ ） 第１軸～第５軸機械位置
論理座標系*における、エンコーダＦＢを積算した位置情報です。

* 論理座標系は、原点設定コマンドにより、原点セットされます。

（ ７ ） 第１軸～第５軸偏差量
指令位置と機械位置の差分です。

（ ８ ） 第１軸～第５軸最新ステップ払出量
最新の移動指令で出力した移動量です。

（ ９ ） 第１軸～第５軸絶対位置
機械原点からの絶対位置です。

（１０） 送りオーバーライド設定
現在の送りオーバーライドの％値です。

（１１） 選択・実行プログラム番号
　　ステータスがプログラム実行中でない場合　→　選択プログラム番号
　　ステータスがプログラム実行中の場合　　→　実行プログラム番号

（１２） 待機・実行ステップ番号
　　ステップ実行中でない場合　　→　待機ステップ番号
　　ステップ実行中の場合　　→　実行ステップ番号

《構造体定義》

```
/************************************************************************/
/*  アンサーステータス情報構造体                                        */
/************************************************************************/
typedef struct {    /* 各軸ポジションデータ構造体                       */
    long            ComReg;             /* 指令位置                     */
    long            PosReg;             /* 機械位置                     */
    long            ErrReg;             /* 偏差量                       */
    long            BlockSeg;           /* 最新ブロック払い出し量       */
    long            AbsReg;             /* 絶対位置                     */
} AXISPOS;

typedef struct {    /* アンサーステータス情報構造体                     */
    unsigned short  Status;             /* 各種状態情報                 */
    unsigned short  AxisSts[5];         /* 各軸ステータス情報           */
    unsigned short  Alarm;              /* アラーム情報                 */
    unsigned short  AxisAlm[5];         /* 各軸アラーム情報             */
    POSITION        AxisPos[5];         /* 各軸位置情報                 */
    unsigned char   Override;           /* 送りオーバーライド設定       */
    unsigned char   ProgramNo;          /* 選択・実行プログラム番号     */
    unsigned short  StepNo;             /* 待機・実行ブロック番号       */
} STATUS;
```

3-1-8.入出力状態データ読出　[DAT_IODATA(0x03)]

《データタイプ》
　ＤＡＴ＿ＩＯＤＡＴＡ（０ｘ０３）

《パラメータ》
　なし

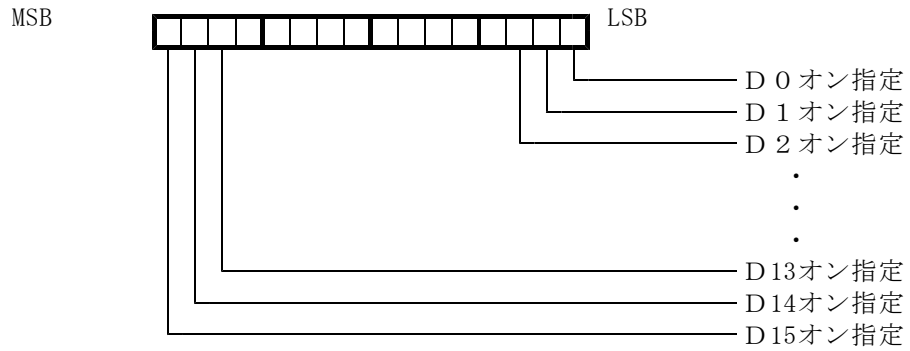
《説明》
　ＳＬＭユニットの汎用入出力の状態です。

《読み込み実行条件》
　常時実行可能です。

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
０００００	入力１ﾊﾟﾀｰﾝ　[#0000]	２バイト	――――
０００００２	入力２ﾊﾟﾀｰﾝ　[#0001]	２バイト	――――
０００００４	入力３ﾊﾟﾀｰﾝ　[#0002]	２バイト	――――
０００００６	出力１ﾊﾟﾀｰﾝ　[#0000]	２バイト	――――
０００００８	出力２ﾊﾟﾀｰﾝ　[#0001]	２バイト	――――

（１）　各入出力のデータは下記の通りになっています。



※各入出力の割付については、「ＳＬＭ－４０００ユーザーズマニュアル[TB00-0800]」
〈Ⅲ 機能編　２．入出力機能〉を参照して下さい。

《構造体定義》

```
/*****************************************************************************/
/*  ＳＬＭ汎用入出力情報構造体                                               */
/*****************************************************************************/
typedef struct {
    unsigned short  Input[3];          /* 汎用入力１～３                  */
    unsigned short  Output[2];         /* 汎用出力１～２                  */
} IODATA;
```

3-1-9.強制入出力状態データ読出　[DAT_FORCEIO(0x13)]

《データタイプ》
ＤＡＴ＿ＦＯＲＣＥＩＯ（０ｘ１３）

《パラメータ》
なし

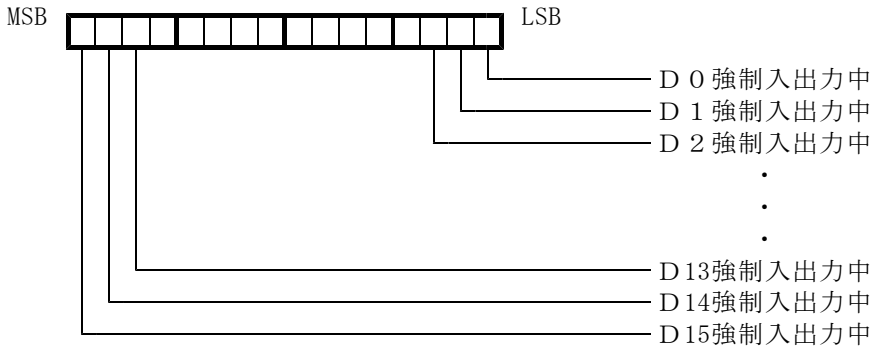
《説明》
通信コマンド等で設定したＳＬＭユニットの強制入出力の状態を読み出します。

《読み込み実行条件》
常時実行可能です。

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
０００００	入力１ﾊﾟﾀｰﾝ　[#0000]	２バイト	――――
００００２	入力２ﾊﾟﾀｰﾝ　[#0001]	２バイト	――――
００００４	入力３ﾊﾟﾀｰﾝ　[#0002]	２バイト	――――
００００６	出力１ﾊﾟﾀｰﾝ　[#0000]	２バイト	――――
００００８	出力２ﾊﾟﾀｰﾝ　[#0001]	２バイト	――――

（１）　各入出力のデータは下記の通りになっています。



※各入出力の割付については、「ＳＬＭ－４０００ユーザーズマニュアル[TB00-0800]」
＜Ⅲ 機能編　2.入出力機能＞を参照して下さい。

《構造体定義》

```
/*************************************************************************/
/*  ＳＬＭ汎用入出力情報構造体                                           */
/*************************************************************************/
typedef struct {
    unsigned short  Input[3];          /* 汎用入力１～３                 */
    unsigned short  Output[2];         /* 汎用出力１～２                 */
} IODATA;
```

3-1-10.動作モードデータ読出 [DAT_MODE(0x04)]

《データタイプ》
ＤＡＴ＿ＭＯＤＥ（０ｘ０４）

《パラメータ》
なし

《説明》
ＳＬＭユニットの動作モードを読み出すものです。

《読み込み実行条件》
常時実行可能です。

《データフォーマット》

ｵﾌｾｯﾄ	デ ー タ 名 称	データ長	設 定 値
０００００	動作モード	２バイト	０ ～ ４

《モードデータ》

設定値	モード
0	セッティングモード
1	手動運転モード
2	自動運転モード
3	ＯＴ無視モード
4	ＤＮＣ運転モード ※

※ＤＮＣオプション有りの時

《構造体定義》

```
/*****************************************************************************/
/*  ＳＬＭ動作モードデータ構造体                                             */
/*****************************************************************************/
typedef struct {
     short          mode;               /* ＳＬＭ動作モード              */
} STMODE;
```

3-1-11.指令位置データ読出 [DAT_COMREG(0x05)]

《データタイプ》
ＤＡＴ＿ＣＯＭＲＥＧ（０ｘ０５）

《パラメータ》
なし

《説明》
ＳＬＭユニットが制御する各軸の指令現在位置を読み出すものです。
有効軸以外の各軸のデータは、無効データです。

《読み込み実行条件》
常時実行可能です。

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	第１軸指令位置	４バイト	-99999999～99999999[Pluse]
００００４	第２軸指令位置	４バイト	-99999999～99999999[Pluse]
００００８	第３軸指令位置	４バイト	-99999999～99999999[Pluse]
００００Ｃ	第４軸指令位置	４バイト	-99999999～99999999[Pluse]
００１０	第５軸指令位置	４バイト	-99999999～99999999[Pluse]

《構造体定義》

```
/*****************************************************************************/
/*  現在位置情報構造体                                                       */
/*****************************************************************************/
typedef struct {
    long            Pos[5];             /* 第１軸～第５軸現在位置       */
} STCOMREG;
```

3-1-12.ステータスフラグデータ読出 [DAT_STATUS_BIT(0x06)]

《データタイプ》
ＤＡＴ＿ＳＴＡＴＵＳ＿ＢＩＴ（０ｘ０６）

《パラメータ》
なし

《説明》
ＳＬＭユニットの現在の動作状態を読み出すものです。
有効軸以外の各軸のデータは、無効データです。

《読み込み実行条件》
常時実行可能です。

《データフォーマット》

ｵﾌｾｯﾄ	デ ー タ 名 称	データ長	設 定 値
００００	各種状態情報	２バイト	―――――
０００２	第１軸ｽﾃｰﾀｽ情報	２バイト	―――――
０００４	第２軸ｽﾃｰﾀｽ情報	２バイト	―――――
０００６	第３軸ｽﾃｰﾀｽ情報	２バイト	―――――
０００８	第４軸ｽﾃｰﾀｽ情報	２バイト	―――――
０００Ａ	第５軸ｽﾃｰﾀｽ情報	２バイト	―――――

（１） 各種状態情報
詳しくは、**「3-1-7.ポジション・ステータスデータ読出」**の**各種状態情報**を参照下さい

（２） 第１～第５軸ステータス情報
詳しくは、**「3-1-7.ポジション・ステータスデータ読出」**の**第１軸～第５軸ステータス情報**を参照下さい

《構造体定義》

```
/*****************************************************************************/
/*  ビット情報ステータス構造体                                               */
/*****************************************************************************/
typedef struct {
    unsigned short  Status;             /* 各種状態情報                  */
    unsigned short  AxisSts[5];         /* 各軸ステータス情報            */
} STSBIT;
```

3-1-13.アラームフラグデータ読出 [DAT_ALARM_BIT(0x07)]

《データタイプ》
ＤＡＴ＿ＡＬＡＲＭ＿ＢＩＴ（０ｘ０７）

《パラメータ》
なし

《説明》
ＳＬＭユニットのアラーム発生情報です。
有効軸以外の各軸のデータは、無効データです。

《読み込み実行条件》
常時実行可能です。

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
０００Ｏ	ｱﾗｰﾑ情報	２バイト	―――
０００２	第１軸ｱﾗｰﾑ情報	２バイト	―――
０００４	第２軸ｱﾗｰﾑ情報	２バイト	―――
０００６	第３軸ｱﾗｰﾑ情報	２バイト	―――
０００８	第４軸ｱﾗｰﾑ情報	２バイト	―――
０００Ａ	第５軸ｱﾗｰﾑ情報	２バイト	―――

（１） アラーム情報
詳しくは、**「3-1-7.ポジション・ステータスデータ読出」**の**アラーム情報**を参照下さい

（２） 第１軸～第５軸アラーム情報
詳しくは、**「3-1-7.ポジション・ステータスデータ読出」**の**第１軸～第５軸アラーム情報**を参照下さい

《構造体定義》

```
/*****************************************************************************/
/*  ビット情報アラーム構造体                                                 */
/*****************************************************************************/
typedef struct {
    unsigned short  Alarm;              /* アラーム情報                      */
    unsigned short  AxisAlm[5];         /* 各軸アラーム情報                  */
} ALMBIT;
```

3-1-14.オーバーライド設定データ読出 [DAT_OVERRIDE(0x08)]

《データタイプ》
ＤＡＴ＿ＯＶＥＲＲＩＤＥ（０ｘ０８）

《パラメータ》
なし

《説明》
現在の送りオーバーライド設定です。

《読み込み実行条件》
常時実行可能です。

《データフォーマット》

ｵﾌｾｯﾄ	デ ー タ 名 称	データ長	設 定 値
０００００	送りｵｰﾊﾞｰﾗｲﾄﾞ設定	２バイト	０～７

設定値と送りオーバーライドの関係は下表の通りです。

設定値	送りオーバーライド
0	２５％
1	５０％
2	７５％
3	１００％
4	１２５％
5	１５０％
6	１７５％
7	２００％

《構造体定義》

```
/*****************************************************************************/
/*  送りオーバーライドデータ構造体                                           */
/*****************************************************************************/
typedef struct {
    unsigned short  Override;           /* 送りオーバーライド設定       */
} STOVERRIDE;
```

3-1-15.プログラム実行情報データ読出 [DAT_MSTPRGSTS(0x0c)、DAT_SLVPRGSTS(0x0d)]

《データタイプ》
ＤＡＴ＿ＭＳＴＰＲＧＳＴＳ（０ｘ０ｃ）
ＤＡＴ＿ＳＬＶＰＲＧＳＴＳ（０ｘ０ｄ）

《パラメータ》
なし

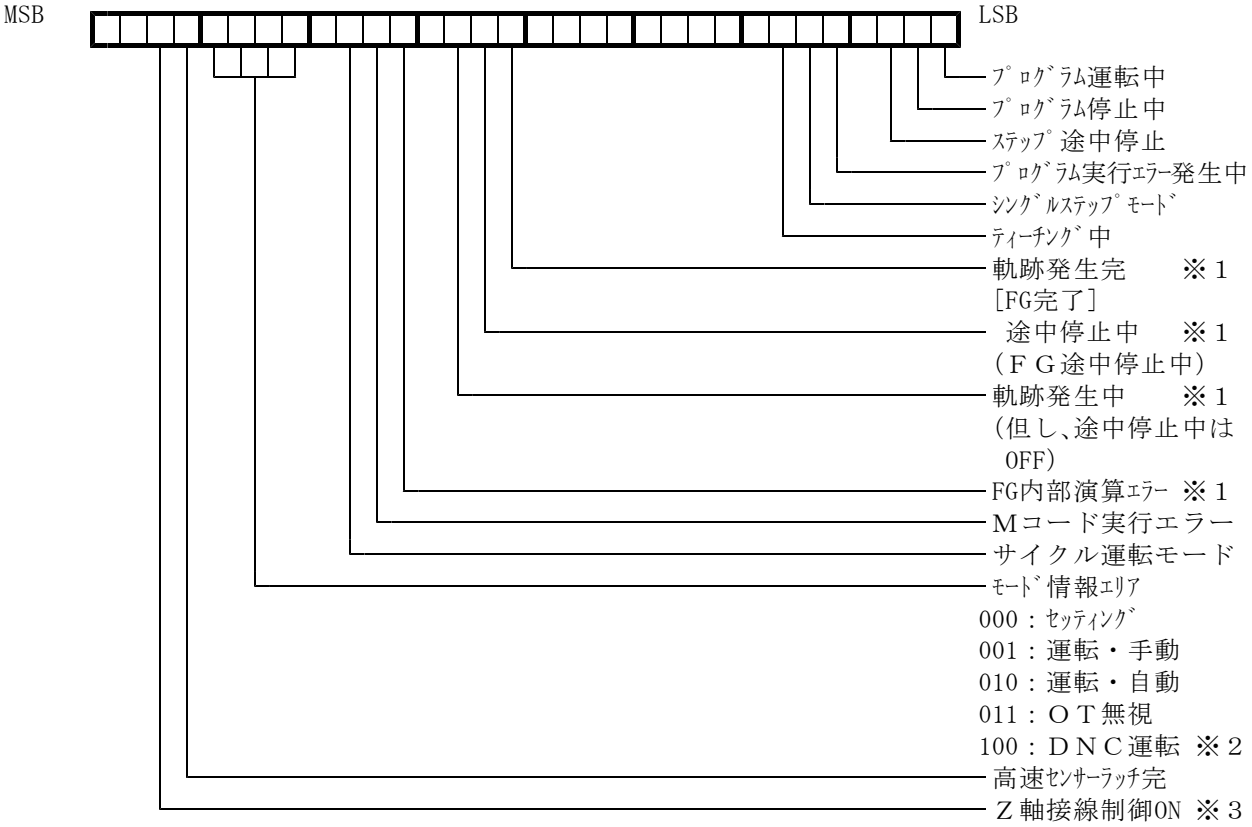
《説明》
現在のプログラム実行状態です。データのフォーマットは以下の通りです。

《読み込み実行条件》
常時実行可能です。

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
０００００	ﾌﾟﾛｸﾞﾗﾑ実行ｽﾃｰﾀｽ	４バイト	－
００００４	選択･実行ﾌﾟﾛｸﾞﾗﾑ番号	１バイト	０～１２
００００５	未使用	１バイト	－
００００６	実行ステップ番号	２バイト	０～９９９

・プログラム実行ステータス



※１ 詳しくは、**「3-1-7.ポジション・ステータスデータ読出」**を参照下さい
※２ ＤＮＣオプション有りの時

※３　Ｚ軸接線制御ｵﾌﾟｼｮﾝ有りの時

《構造体定義》

```
/*************************************************************************/
/*  プログラム情報データ構造体                                           */
/*************************************************************************/
typedef struct {
    unsigned short  PrgExecSts;         /* プログラム実行ステータス     */
    unsigned char   ProgramNo;          /* 選択・実行プログラム番号     */
    unsigned char   Reserved;           /* 未使用                       */
    unsigned short  StepNo;             /* 実行ステップ番号             */
} STPRGSTS;
```

3-1-16.ＤＮＣバッファ情報データ読出 [DAT_DNCBUFI(0x0e)] 〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＤＡＴ＿ＤＮＣＢＵＦＩ（０ｘ０ｅ）

《パラメータ》
なし

《説明》
現在のＤＮＣバッファの状態です。データのフォーマットは以下の通りです。

《読み込み実行条件》
常時実行可能です。

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	バッファ使用容量	４バイト	―――――
０００４	バッファ空き容量	４バイト	―――――

《構造体定義》

```
/*************************************************************************/
/*  ＤＮＣバッファ情報構造体                                             */
/*************************************************************************/
typedef struct {
    long            Size;               /* バッファ使用容量（バイト）   */
    long            Free;               /* バッファ空き容量（バイト）   */
} DNCBUFI;
```

3-1-17.センサーラッチ位置データ読出 [DAT_SENSEPOS(0x11)]〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＤＡＴ＿ＳＥＮＳＥＰＯＳ（０ｘ１１）

《パラメータ》
なし

《説明》
センサーラッチ入力があった時の各軸の位置情報です。データのフォーマットは以下の通りです。
有効軸以外の各軸のデータは、無効データです。

《読み込み実行条件》
常時実行可能です。

《データフォーマット》

各軸論理座標 ４バイト×５
各軸アブソ座標 ４バイト×５

各軸論理座標のフォーマットは以下の通りです。

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	第１軸論理座標	４バイト	-99999999～99999999[Pluse]
０００４	第２軸論理座標	４バイト	-99999999～99999999[Pluse]
０００８	第３軸論理座標	４バイト	-99999999～99999999[Pluse]
０００Ｃ	第４軸論理座標	４バイト	-99999999～99999999[Pluse]
００１０	第５軸論理座標	４バイト	-99999999～99999999[Pluse]

各軸アブソ座標のフォーマットは以下の通りです。

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００１４	第１軸アブソ座標	４バイト	-99999999～99999999[Pluse]
００１８	第２軸アブソ座標	４バイト	-99999999～99999999[Pluse]
００１Ｃ	第３軸アブソ座標	４バイト	-99999999～99999999[Pluse]
００２０	第４軸アブソ座標	４バイト	-99999999～99999999[Pluse]
００２４	第５軸アブソ座標	４バイト	-99999999～99999999[Pluse]

《構造体定義》

```
/*****************************************************************************/
/*  センサーラッチ位置情報構造体                                             */
/*****************************************************************************/
typedef struct {
    long            SenPos[5];          /* ｾﾝｻｰﾗｯﾁﾎﾟｼﾞｼｮﾝ（論理座標系）  */
    long            SenPosA[5];         /* ｾﾝｻｰﾗｯﾁﾎﾟｼﾞｼｮﾝ（ｱﾌﾞｿ座標系）  */
} SENSEPOS;
```

3-1-18.送りオーバーライド%データ読出 [DAT_OVERRIDEP(0x15)]

《データタイプ》
ＤＡＴ＿ＯＶＥＲＲＩＤＥＰ（０ｘ１５）

《パラメータ》
なし

《説明》
ＳＬＭユニットの現在の送りオーバーライドを１％刻みで読み出すものです。

《読み込み実行条件》
常時実行可能です。

《データフォーマット》

ｵﾌｾｯﾄ	デ ー タ 名 称	データ長	設 定 値
０００００	送りｵｰﾊﾞｰﾗｲﾄﾞ設定	２バイト	０～２００

《構造体定義》

```
/*****************************************************************************/
/*  送りオーバーライドデータ構造体                                           */
/*****************************************************************************/
typedef struct {
    unsigned short  Override;           /* 送りオーバーライド設定       */
} STOVERRIDE;
```

3-1-19.プログラム１ステップデータ読出 [DAT_ONEBLOCK(0x30)]

《データタイプ》
ＤＡＴ＿ＯＮＥＢＬＯＣＫ（０ｘ３０）

《パラメータ》
ステップ番号（０～最大ステップ番号）
※ ０を指定すると現在の実行ステップ番号を指定した事になります。

《説明》
ＳＬＭユニットで現在選択されているプログラム番号の任意のステップの動作プログラムのデータを読み出すものです。
読み出したデータには、後述のヘッダデータを付加し、プログラム変換ライブラリにて、テキスト形式のデータへ変換して下さい。

《読み込み実行条件》
・選択されているプログラム番号に動作プログラムダウンロード済み

《データフォーマット》

動作プログラムヘッダデータ ７２バイト
動作プログラム１ステップバイナリデータ ７２バイト

動作プログラムヘッダデータのフォーマットは以下の通りです。

ｵﾌｾｯﾄ	デ ー タ 名 称	データ長	設 定 値
００００	予約	2ﾊﾞｲﾄ	0
０００２	ステップ数	2ﾊﾞｲﾄ	1
０００４	予約	22ﾊﾞｲﾄ	0
００１Ａ	ﾌﾟﾛｸﾞﾗﾑｺｰﾄﾞﾀｲﾌﾟ	2ﾊﾞｲﾄ	[0:Tコード、1:Gコード]
００１Ｃ	予約	44ﾊﾞｲﾄ	0

（ １ ） 動作プログラムヘッダデータ
必ず予約領域を０フィルして下さい。

（ ２ ） 動作プログラム１ステップバイナリデータ
本コマンドで、ＳＬＭユニットより受け取ったデータです。

《構造体定義》

```
/*************************************************************************/
/*  プログラムボリュームラベル構造体（７２ﾊﾞｲﾄ固定長）                   */
/*************************************************************************/
typedef struct {
    unsigned char   Reserved1[2];       /* 予約                          */
    unsigned short  BlockNumber;        /* 有効ブロック長                */
    unsigned char   Reserved2[22];      /* 予約                          */
    unsigned short  ProgramType;        /* ﾌﾟﾛｸﾞﾗﾑｺｰﾄﾞﾀｲﾌﾟ[0:T、1:G]    */
    unsigned char   Reserved[44];       /* 未使用                        */
} BINPRG_LABEL;
/*************************************************************************/
/*  プログラム１ブロックデータ構造体（７２ﾊﾞｲﾄ固定長）                   */
/*************************************************************************/
typedef struct {
    unsigned char   Reserved1[69];      /* 予約                              */
    unsigned char   PrgType;            /* ﾌﾟﾛｸﾞﾗﾑｺｰﾄﾞﾀｲﾌﾟ[0:T、1:G]        */
    unsigned char   Reserved2[2];       /* 未使用                            */
} BINPRG_BLOCK;
```

3-1-20.マルチタスクプログラム実行情報読出 ［DAT_TASKPRGSTS(0x80)］ <ｵﾌﾟｼｮﾝ>

《データタイプ》
ＤＡＴ＿ＴＡＳＫＰＲＧＳＴＳ（０ｘ８０）

《パラメータ》
タスク番号（０～６）
タスク番号についての詳細は「3-1-4.マルチタスクプログラム書込/読出」を参照下さい。

《説明》
ＳＬＭユニットの各タスクの実行情報を読み出すものです。

《読み込み実行条件》
常時実行可能です。

《データフォーマット》
「**3-1-15. プログラム実行情報読出**」と同様です。

3-1-21.ティーチング情報読出 [DAT_TEACHSTS(0x18)]

《データタイプ》
ＤＡＴ＿ＴＥＡＣＨＳＴＳ（０ｘ１８）

《パラメータ》
なし

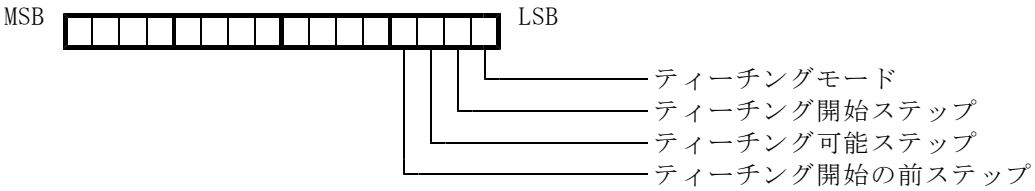
《説明》
ＳＬＭユニットのティーチングの状態を読み出すものです。

《読み込み実行条件》
常時実行可能です。

《データフォーマット》

ｵﾌｾｯﾄ	デ ー タ 名 称	データ長	設 定 値
０００００	ﾃｨｰﾁﾝｸﾞｽﾃｰﾀｽ	２バイト	－
００００２	実行ｽﾃｯﾌﾟ番号	２バイト	－
００００４	ﾃｨｰﾁﾝｸﾞｽﾃｯﾌﾟ番号	２バイト	－
００００６	未使用	２バイト	－

（１） ティーチングステータス



（２） 実行ステップ番号
現在のステップ番号です。

（３） ティーチングステップ番号
ティーチングを開始したステップ番号です。

《構造体定義》

```
/*****************************************************************************/
/*  ティーチング情報用パラメータ情報構造体                                   */
/*****************************************************************************/
typedef struct {
    unsigned short  Status;          /* ティーチングステータス       */
    unsigned short  StepNo;          /* 実行ステップ番号             */
    unsigned short  TchStepNo;       /* ティーチングステップ番号     */
    unsigned short  Reserved;        /* 未使用                       */
} TEACHSTS;
```

3-1-22.ＲＯＭバージョン情報読出 ［DAT_VERSION(0x27)］

《データタイプ》
ＤＡＴ＿ＶＥＲＳＩＯＮ（０ｘ２７）

《パラメータ》
なし

《説明》
ＳＬＭユニットのシステムのバージョン情報を読み出すものです。

《読み込み実行条件》
常時実行可能です。

《データフォーマット》

ｵﾌｾｯﾄ	デ ー タ 名 称	データ長	設 定 値
００００	バージョン文字列	16バイト	－
００１０	PROM SUM [EVEN]	２バイト	－
００１２	PROM SUM [ODD]	２バイト	－
００１４	FLASH SUM	２バイト	－
００１６	FLASH使用フラグ	２バイト	0:未使用、1:使用
００１８	機種ＩＤ	４バイト	－
００１Ｃ	シリアルＩＤ	４バイト	－
００２０	予約	32バイト	－

《構造体定義》

```
/*****************************************************************************/
/*  ＲＯＭバージョン情報構造体                                               */
/*****************************************************************************/
typedef struct {
    char            Version[16];        /* バージョン文字列             */
    unsigned short  EvenSum;            /* SUM:even [rom]               */
    unsigned short  OddSum;             /* SUM:odd  [rom]               */
    unsigned short  FlashSum;           /* SUM:ＳＨ内部ＦＬＡＳＨ       */
    short           FlashFlg;           /* ＳＨ内部ＦＬＡＳＨ使用フラグ */
    unsigned long   KindID;             /* 機種ＩＤ                     */
    unsigned long   SerialID;           /* シリアルＩＤ                 */
    char            Reserved[32];       /* 予約                         */
} ROMVERSION;
```

3-1-23.マクロ変数（一般レジスタ）読出 ［DAT_VARIABLE(0x19)］〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＤＡＴ＿ＶＡＲＩＡＢＬＥ（０ｘ１９）

《パラメータ》
なし

《説明》
ＳＬＭユニットの一般レジスタの値を読み出すものです。
一般レジスタの詳細は、「ＳＬＭ－４０００ ユーザーズマニュアル[TB00-0800]」＜Ⅲ 機能編 ６－４．マクロ機能＞を参照下さい。

《関連項目》
3-1-31.マクロ変数読出
3-2-55.マクロ変数書込コマンド

《読み込み実行条件》
常時実行可能です。

《データフォーマット》

マクロ変数データ ４００バイト

ｵﾌｾｯﾄ	デ ー タ 名 称	データ長	設 定 値
００００	マクロ変数	４ﾊﾞｲﾄ	

《構造体定義》
```
/*************************************************************************/
/*  マクロ変数データ                                                     */
/*************************************************************************/
    typedef long VARIABLE[100];
```

3-1-24.ＴＰＣロギングデータ読出 ［DAT_TPCDATA(0x25)］

《データタイプ》
ＤＡＴ＿ＴＰＣＤＡＴＡ（０ｘ２５）

《パラメータ》
なし

《説明》
ＳＬＭユニットでロギングしたＴＰＣデータを読み出すものです。
本機能はロギング処理を終了してから実行してください。
データの読み出しを行うと、ロギングバッファを解放します。
読み出したデータは、バイナリ形式のままで、拡張子”ＡＳ１”のファイルに書き出して下さい。
このファイルは弊社「ＴＰＣソフトウェア」によって解析することが可能です。

《読み込み実行条件》
・ロギング停止中
・ロギングデータあり

《関連項目》
3-1-25.ＴＰＣロギング情報読出
3-2-41.ＴＰＣデータ選択コマンド
3-2-51.ＴＰＣロギング開始コマンド

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
０００００	第１軸機械位置	４バイト	-99999999～99999999[Pluse]
００００４	汎用入力１状態	２バイト	－
００００６	汎用出力１状態	２バイト	－
００００８	第２軸機械位置	４バイト	-99999999～99999999[Pluse]
００００Ｃ	汎用入力２状態	２バイト	－
００００Ｅ	汎用出力２状態	２バイト	－

《構造体定義》
```
/*************************************************************************/
/*  ＴＰＣデータ構造体                                                   */
/*************************************************************************/
typedef struct {    // for slm
    long           pr1;              /* 第１軸機械位置                 */
    unsigned short hi1;              /* 汎用入力１状態                 */
    unsigned short ho1;              /* 汎用出力１状態                 */
    long           pr2;              /* 第２軸機械位置                 */
    unsigned short hi2;              /* 汎用入力２状態                 */
    unsigned short ho2;              /* 汎用出力２状態                 */
} TPC;
```

3-1-25.ＴＰＣロギング情報読出 ［DAT_TPCINFO(0x24)］

《データタイプ》
ＤＡＴ＿ＴＰＣＩＮＦＯ（０ｘ２４）

《パラメータ》
なし

《説明》
ＳＬＭユニットの現在のＴＰＣデータのロギング情報を読み出すものです。

《読み込み実行条件》
常時実行可能です。

《関連項目》
3-1-24.ＴＰＣロギングデータ読出
3-2-41.ＴＰＣデータ選択コマンド
3-2-51.ＴＰＣロギング開始コマンド

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
０００００	ＴＰＣﾃﾞｰﾀﾛｷﾞﾝｸﾞﾌﾗｸﾞ	２バイト	０：未実行、１：実行中
０００２	ＴＰＣﾃﾞｰﾀﾛｷﾞﾝｸﾞﾎﾟｲﾝﾄ数	２バイト	０～４０９５

《構造体定義》

```
typedef struct {
    short           Log;            /* ＴＰＣﾃﾞｰﾀﾛｷﾞﾝｸﾞﾌﾗｸﾞ        */
    short           Num;            /* ＴＰＣﾃﾞｰﾀﾛｷﾞﾝｸﾞﾎﾟｲﾝﾄ数     */
} TPCINFO;
```

3-1-26.軸ネグレクト設定データ読出 [DAT_AXNEGLECT(0x28)]

《データタイプ》
ＤＡＴ＿ＡＸＮＥＧＬＥＣＴ（０ｘ２８）

《パラメータ》
なし

《説明》
各軸の軸指令の有効/無効状態を取得します。

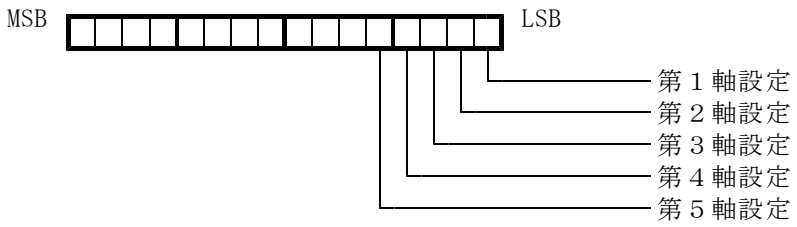
《読み込み実行条件》
常時実行可能です。

《関連項目》
3-2-46.軸ネグレクト設定コマンド

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
０００００	ネグレクト軸フラグ	４バイト	－

（１）ネグレクト軸フラグ
各軸のネグレクト設定を示します。
ネグレクト設定の軸に対応するビットに１が設定されます。



《構造体定義》
構造体定義はありません。
４バイトデータ（long）として定義して下さい。

3-1-27.軸インタロック設定データ読出 [DAT_AXINTLOCK(0x29)]

《データタイプ》
ＤＡＴ＿ＡＸＩＮＴＬＯＣＫ（０ｘ２９）

《パラメータ》
なし

《説明》
各軸のインタロックの有効/無効状態を取得します。

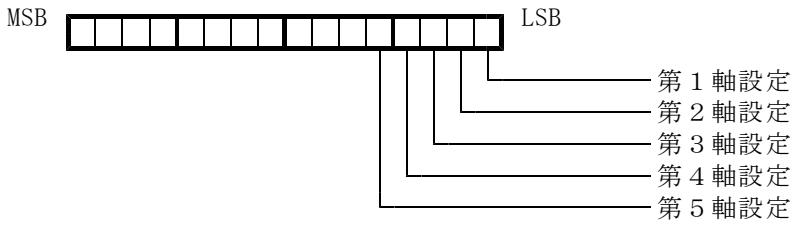
《読み込み実行条件》
常時実行可能です。

《関連項目》
3-2-45.軸インタロック設定コマンド

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	インタロック軸フラグ	４バイト	－

（１）インタロック軸フラグ
各軸のインタロック設定を示します。
インタロック設定の軸に対応するビットに１が設定されます。



《構造体定義》
構造体定義はありません。
４バイトデータ（long）として使用下さい。

3-1-28.各軸サーボON/OFF設定データ読出 [DAT_AXSVONEN(0x31)]

《データタイプ》
ＤＡＴ＿ＡＸＳＶＯＮＥＮ（０ｘ３１）

《パラメータ》
なし

《説明》
各軸のサーボＯＮ／ＯＦＦ設定状態を取得します。

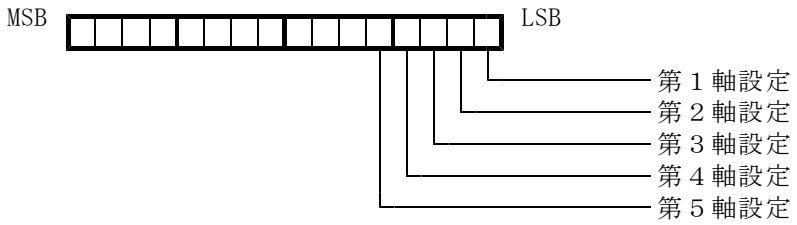
《読み込み実行条件》
常時実行可能です。

《関連項目》
3-2-47.各軸サーボON/OFFコマンド

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
０００００	サーボＯＮ軸フラグ	４バイト	－

（１）サーボＯＮ軸フラグ
各軸のサーボＯＮ設定を示します。
サーボＯＮ設定の軸に対応するビットに１が設定されます。



《構造体定義》
構造体定義はありません。
４バイトデータ（long）として使用下さい。

3-1-29.手動ﾊﾟﾙｻｰ設定情報読出 ［DAT_HANDLESTS(0x32)］〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＤＡＴ＿ＨＡＮＤＬＥＳＴＳ（０ｘ３２）

《パラメータ》
なし

《説明》
手動パルサーの設定状態を取得します。

《読み込み実行条件》
常時実行可能です。

《関連項目》
3-2-48.手動パルサーモードON/OFF設定コマンド
3-2-49.手動パルサー倍率設定コマンド
3-2-50.手動パルサー有効軸設定コマンド

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	手動パルサー有効ﾌﾗｸﾞ	２バイト	０：無効、１：有効
０００２	手動パルサー倍率	２バイト	1,10,100,1000
００１４	手動パルサー第１軸	２バイト	軸番号（０～軸数）
００１６	手動パルサー第２軸	２バイト	軸番号（０～軸数）

（１）手動パルサー有効フラグ
手動パルサーの有効/無効状態を示します。

（２）手動パルサー倍率
手動パルサーの倍率（手動パルサー１パルスあたりの軸移動量）を示します。

（３）手動パルサー第１軸
手動パルサー第１軸にて、移動を行わせる軸を示します。

（４）手動パルサー第２軸
手動パルサー第２軸にて、移動を行わせる軸を示します。

《構造体定義》

```
/*****************************************************************************/
/*  手パ操作情報構造体                                                       */
/*****************************************************************************/
typedef struct {
    short   handle_mode;            /* 手パ有効フラグ                 */
    short   kp;                     /* 手パ設定倍率                   */
    short   ax1;                    /* 手パ第１軸                     */
    short   ax2;                    /* 手パ第２軸                     */
} HANDLESTS;
```

3-1-30.機械操作パネルＳＷ情報読出 ［DAT_MPDATA(0x26)］〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＤＡＴ＿ＭＰＤＡＴＡ（０ｘ２６）

《パラメータ》
なし

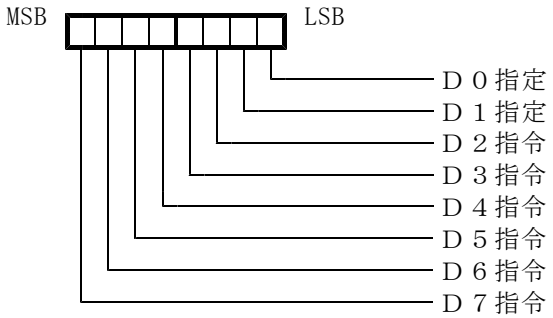
《説明》
機械操作パネルの入力チャンネルの状態を読み出します。
機械操作パネルの詳細については、「ＳＬＭ－４０００ユーザーズマニュアル[TB00-0800]」
＜Ⅲ 機能編 ７．機械操作パネル＞を参照下さい。

《読み込み実行条件》
常時実行可能です。

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	機械パネル入力ＣＨ１	１バイト	－
０００１	機械パネル入力ＣＨ２	１バイト	－
０００２	機械パネル入力ＣＨ３	１バイト	－
０００３	機械パネル入力ＣＨ４	１バイト	－
０００４	機械パネル入力ＣＨ５	１バイト	－
０００５	機械パネル入力ＣＨ６	１バイト	－
０００６	機械パネル入力ＣＨ７	１バイト	－
０００７	機械パネル入力ＣＨ８	１バイト	－

（１）各入出力のデータは下記の通りになっています。



《構造体定義》

```
/*****************************************************************************/
/*  機械パネルＳＷ情報構造体                                                 */
/*****************************************************************************/
typedef struct {
    unsigned char   PnlSw[8];           /* 機械パネル入力１～８          */
} MPDATA;
```

3-1-31.マクロ変数読出 [DAT_MCRREG(0x33)] 〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＤＡＴ＿ＭＣＲＲＥＧ（０ｘ３３）

《パラメータ》
マクロ変数番号

《説明》
パラメータで指定された番号のマクロ変数に設定されている値を読み出します。
このコマンドでは、読出可能であれば全てのマクロ変数を指定する事が出来ます。
マクロ変数詳細については、「ＳＬＭ－４０００ユーザーズマニュアル［TB00-0800］」
＜Ⅲ 機能編 ６－４．マクロ機能＞を参照下さい。。

《読み込み実行条件》
常時実行可能です。

《関連項目》
3-1-23.マクロ変数（一般レジスタ）読出
3-2-55.マクロ変数書込コマンド

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	マクロ変数値	４バイト	-2147483647 ～ 2147483647

《構造体定義》

```
/*****************************************************************************/
/*  マクロ変数データ構造体                                                   */
/*****************************************************************************/
typedef struct {
    long            Val;                /* マクロ変数値                     */
} MCRREG;
```

3-1-32.フィードバックカウンタ積算値読出 [DAT_FBCOUNT(0x50)]

《データタイプ》
ＤＡＴ＿ＦＢＣＯＵＮＴ（０ｘ５０）

《パラメータ》
なし

《説明》
ＦＢカウンタ１と２の積算値を読み出します。
本データは、軸の割り当てや設定に関わらず、ＦＢ入力に入力されたパルスをそのまま積算します。
（Ａ相進みで＋カウント、電源投入時は０）
ＦＢ１／２を軸に割り当てていても、本データには、割り当てた軸の任意分周やＦＢ入力極性は反映されません。
動作モードや軸動作に関わらず、ＦＢ入力に入力されたパルスをそのまま積算したデータです。

《読み込み実行条件》
常時実行可能です。

《関連項目》
3-2-56.フィードバックカウンタ積算値セットアップコマンド

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	ＦＢ１カウンタ積算値	４バイト	-2147483647 ～ 2147483647
０００４	ＦＢ２カウンタ積算値	４バイト	-2147483647 ～ 2147483647

○ Ｃ言語構造体

```
/*************************************************************************/
/*  フィードバックカウンタデータ構造体（８ﾊﾞｲﾄ固定長）                   */
/*************************************************************************/
typedef struct {
    long            cntr[2];            /* ＦＢ１・ＦＢ２積算値         */
} FBCOUNT;
```

3-1-33.工具径補正データ書込／読出 [DAT_TOOLDIA(0x1a)] 〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＤＡＴ＿ＴＯＯＬＤＩＡ（０ｘ１ａ）

《パラメータ》
なし

《機能》
ＳＬＭユニットに工具径補正用データの書込／読出を行うものです。
本コマンドの実行には、ＳＬＭユニットに工具径補正オプションが追加されている必要があります。
工具径補正データ書込は、ＳＬＭユニットの動作モードがセッティングモードの時のみ実行が可能です。

《書き込み実行条件》
・セッティングモード

《読み込み実行条件》
常時実行可能です。

《データフォーマット》

各補正用データ ８０バイト

各補正用データのフォーマットは、以下の通りです。

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	補正用データ	４バイト	-99999999～99999999

《構造体定義》

```
/*****************************************************************************/
/*  ツール径補正用パラメータ情報構造体                                       */
/*****************************************************************************/
typedef struct {
    long            d[20];              /* ツール径補正データ                */
} TOOL_D;
```

3-1-34.工具長補正設定情報読出　[DAT_TOOLHSTS($39)]　〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
　ＤＡＴ＿ＴＯＯＬＨＳＴＳ（＄３９）

《パラメータ》
　なし

《説明》
　現在の工具長補正情報を読み出します。

《読み込み実行条件》
　常時実行可能です。

《データフォーマット》

オフセット	データ名称	データ長	データ範囲［単位］
０００００	補正有効フラグ	WORD	0:長補正未実行、1:長補正中
００００１	選択中補正No	WORD	0～19

○Ｃ言語構造体

```
/*****************************************************************************/
/*  工具長補正情報構造体                                                     */
/*****************************************************************************/
typedef struct {
    short   toolh_en;                   /* 補正有効フラグ                */
    short   toolh_no;                   /* 選択中補正No                  */
} TOOLHSTS;
```

3-1-35.工具径補正設定情報読出　[DAT_TOOLDSTS($3A)]　〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
　ＤＡＴ＿ＴＯＯＤＳＴＳ（＄３Ａ）

《パラメータ》
　なし

《説明》
　現在の工具径補正情報を読み出します。

《読み込み実行条件》
　常時実行可能です。

《データフォーマット》

オフセット	データ名称	データ長	データ範囲［単位］
０００００	補正有効フラグ	WORD	0:径補正未実行、1:径補正中
００００１	選択中補正No	WORD	0～19

○Ｃ言語構造体

```
/*****************************************************************************/
/*  工具径補正情報構造体                                                     */
/*****************************************************************************/
typedef struct {
short   toold_en;                   /* 補正有効フラグ                 */
short   toold_no;                   /* 選択中補正No                   */
} TOOLDSTS;
```

3-1-36.工具径補正エラー情報読出［DAT_TOOLDERR($3B)］ 〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
　ＤＡＴ＿ＴＯＯＤＥＲＲ（＄３Ｂ）

《パラメータ》
　なし

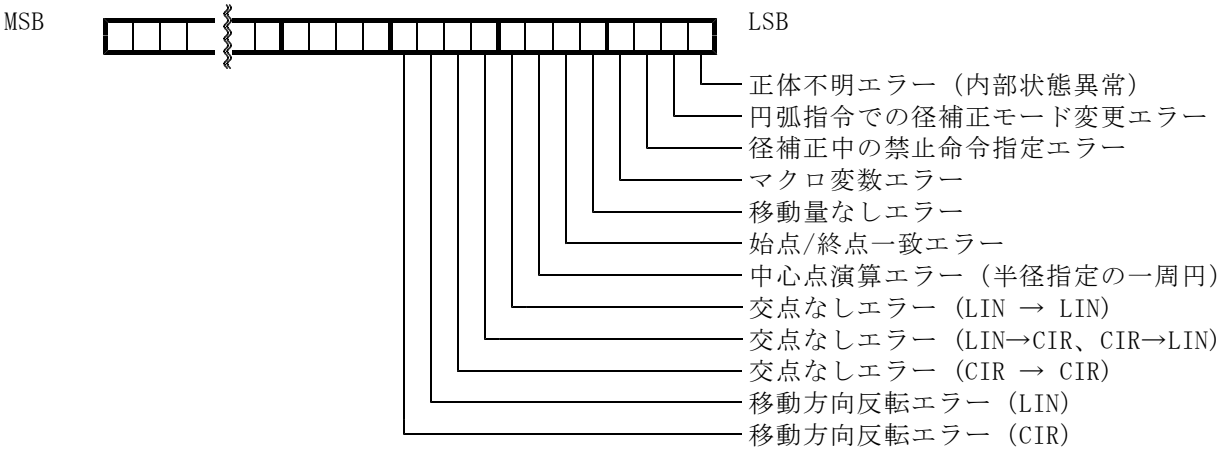
《説明》
　工具径補正のエラー情報を読み出します。

《読み込み実行条件》
　常時実行可能です。

《データフォーマット》

オフセット	データ名称	データ長	データ範囲［単位］
００００	エラー発生ステップ番号	２バイト	1～999
０００２	予約	２バイト	―――
０００４	エラーコード	４バイト	―――
０００８	未使用	８バイト	―――

・エラーコード



※　エラーについての詳細は、「ＳＬＭ４０００　径補正仕様(TB04-2176)」を
　　参照して下さい。

○Ｃ言語構造体

```
/*************************************************************************/
/*  工具径補正エラー情報構造体                                           */
/*************************************************************************/
typedef struct {
    unsigned short  StepNo;          /* エラー発生ステップ番号       */
    unsigned short  Reserved1;       /* 予約                         */
    long            ErrCode;         /* エラーコード                 */
    unsigned char   Reserved2[8];    /* 未使用                       */
} TOOLDERR;
```

3-1-37.補間前加減速ﾊﾟﾗﾒｰﾀ書込/読出 ［DAT_ACOPARAM($37)］ 〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＤＡＴ＿ＡＣＯＰＡＲＡＭ（＄３７）

《説明》
補間前加減速のパラメータを設定します。

《読み込み実行条件》
・軸移動中でない

《データフォーマット》

オフセット	データ名称	データ長	データ範囲［単位］
００００	補間前加減速時定数	４バイト	０～１６３８４［msec］
００００４	補間前加減速切換加速度1	４バイト	０～8000000［pls/sec2］
	：		
００２８	補間前加減速切換加速度7	４バイト	０～8000000［pls/sec2］
００３２	補間前加減速ｵｰﾊﾞﾗｲﾄﾞ1	２バイト	０～１００［%］
	：		
００４４	補間前加減速ｵｰﾊﾞﾗｲﾄﾞ7	２バイト	０～１００［%］
００４６	予約	２バイト	－

○Ｃ言語構造体

```
/*************************************************************************/
/*  補間前加減速用パラメータ情報構造体                                   */
/*************************************************************************/
typedef struct {    /*  （４８バイト）                                   */
    long  aco_acot;                 /* 補間前加減速時定数［msec］    */
    long  aco_accdat[7];            /* 補間前加減速切換加速度［pls/sec］*/
    short aco_ovrdat[7];            /* 補間前加減速ｵｰﾊﾞﾗｲﾄﾞﾃﾞｰﾀ［%］*/
    unsigned short  Reserved;       /* 未使用                        */
} ACO_PRM;
```

《データ詳細》

（１）補間前加減速時定数
現状の速度から仮に速度０まで減速する時間[msec]を設定します。
（速度０から指定の速度まで加速するまでの時間と同じです）

（２）補間前加減速切換加速度、補間前加減速ｵｰﾊﾞﾗｲﾄﾞ
以下の各加減速範囲における最小オーバライドを設定します。

補間前加減速切換加速度1～補間前加減速切換加速度2 => 補間前加減速ｵｰﾊﾞﾗｲﾄﾞ2
補間前加減速切換加速度2～補間前加減速切換加速度3 => 補間前加減速ｵｰﾊﾞﾗｲﾄﾞ3
補間前加減速切換加速度3～補間前加減速切換加速度4 => 補間前加減速ｵｰﾊﾞﾗｲﾄﾞ4
補間前加減速切換加速度4～補間前加減速切換加速度2 => 補間前加減速ｵｰﾊﾞﾗｲﾄﾞ5
補間前加減速切換加速度5～補間前加減速切換加速度2 => 補間前加減速ｵｰﾊﾞﾗｲﾄﾞ6
補間前加減速切換加速度6～ => 補間前加減速ｵｰﾊﾞﾗｲﾄﾞ7

※補間前加減速ｵｰﾊﾞﾗｲﾄﾞ1は使用しません。必ず100を設定して下さい。

3-2.動作要求コマンド詳細

動作要求コマンドは、パソコンよりＳＬＭユニットに各種動作の指示を行う機能です。
通信ライブラリのSendCommand[]関数を使用して、ＳＬＭユニットへ動作要求を行うことが出来ます。

3-2-1.バックアップメモリ初期化コマンド [REQ_GENE(0x14)]

《データタイプ》
ＲＥＱ＿ＧＥＮＥ（０ｘ１４）

《説明》
ＳＬＭユニットのバックアップメモリのデータを初期化するものです。
バックアップされているデータは以下の通りです。

・ サーボパラメータ　　・・・　「**3-1-1.サーボパラメータデータ**」参照
・ 動作プログラム　　・・・　「**3-1-2.プログラムデータ**」参照
・ 疑似アブソポジション　・・・　機械原点を０とする座標上の各軸の機械位置

サーボパラメータを初期化すると、ＳＬＭユニットはパラメータ未設定エラーになります。
パラメータ未設定エラーは、サーボパラメータを再設定する事により解除できます。

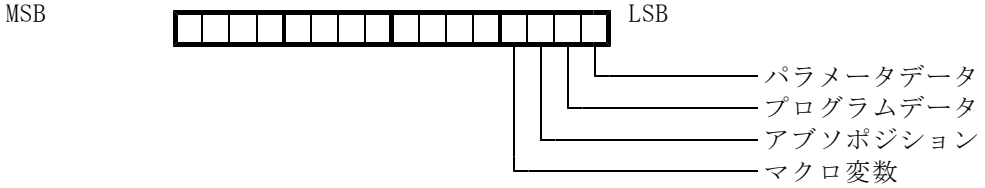
疑似アブソポジションを初期化した場合、原点復帰入力、または全軸原点復帰開始コマンドによる全軸原点復帰処理に障害が発生する可能性があります。
これを回避するためには、各軸個別に手動原点復帰を行う必要があります。
手動原点復帰は、後述の「**3-2-6.各軸原点復帰コマンド**」の**各軸手動原点復帰指定**を利用して実現する事ができます。

《コマンド実行条件》
バックアップデータ初期化は、ＳＬＭの動作モードがセッティングモードの時のみ実行が可能です。

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
０００００	初期化ﾃﾞｰﾀ選択ﾌﾗｸﾞ	２バイト	――――

（１） 初期化データ選択フラグ
初期化を行うデータの種別を指定します。
種別に対応するビットに１を設定するとそのデータが初期化されます。



《構造体定義》

```
/*************************************************************************/
/*  バックアップメモリ初期化コマンド付加データ構造体                     */
/*************************************************************************/
typedef struct {
    unsigned short  InitCmnd;          /* 初期化データ選択フラグ        */
} GENERATION;
```

3-2-2.動作モード変更コマンド [REQ_MODECHG(0x10)]

《データタイプ》
ＲＥＱ＿ＭＯＤＥＣＨＧ（０ｘ１０）

《説明》
ＳＬＭユニットの動作モードを設定するものです。
ＳＬＭユニットはスタンドアロン状態では動作モードの変更はできません。
本機能を使用して、パソコンからのみ動作モードの変更が可能です。
ＳＬＭユニットの電源投入後のデフォルト動作モードは、自動運転モードです。

《コマンド実行条件》
動作モード設定は、プログラム実行中または軸移動中は実行できません。

《データフォーマット》

ｵﾌｾｯﾄ	デ ー タ 名 称	データ長	設 定 値
０００００	動作モード	２バイト	０～４

（１） 動作モード
設定する動作モードを指定します。

設定値	モード
0	セッティングモード
1	手動運転モード
2	自動運転モード
3	ＯＴ無視モード
4	ＤＮＣ運転モード ※

※ＤＮＣ運転オプション有りの時

《構造体定義》

```
/*****************************************************************************/
/*  ＳＬＭ動作モードデータ変更コマンド付加データ構造体                       */
/*****************************************************************************/
typedef struct {
    short           mode;               /* ＳＬＭ動作モード                 */
} MODECHG;
```

3-2-3.軸移動停止コマンド [REQ_STOP(0x13)]

《データタイプ》
ＲＥＱ＿ＳＴＯＰ（０ｘ１３）

《説明》
プログラム実行や移動コマンド発行等によって移動中の軸を停止させるものです。
軸移動停止は、ＳＬＭユニットの動作モードや動作状態に関わらず実行可能です。

《コマンド実行条件》
・サーボＯＮ状態

《データフォーマット》
コマンド付加データは有りません。

3-2-4.軸移動再開コマンド [REQ_RESTART(0x1b)]

《データタイプ》
ＲＥＱ＿ＲＥＳＴＡＲＴ（０ｘ１ｂ）

《説明》
軸移動停止コマンドにて、停止中の軸の移動を再開させるものです。
軸移動再開は、以下のような移動を軸移動停止コマンドにて停止させた場合に実行可能です。

《コマンド実行条件》
・インクレＰＴＰ位置決めコマンドによる軸移動が停止中
・アブソＰＴＰ位置決めコマンドによる軸移動が停止中
・インクレ補間位置決めコマンドによる軸移動が停止中
・アブソ補間位置決めコマンドによる軸移動が停止中
※詳細条件が少し複雑です。このコマンドをお使いになる場合は、恐縮ですがテクノに問い合わせお願い致します。

《データフォーマット》
コマンド付加データは有りません。

3-2-5.ＪＯＧ移動ｺﾏﾝﾄﾞ [REQ_JOGSTART(0x11)]

《データタイプ》
ＲＥＱ＿ＪＯＧＳＴＡＲＴ（０ｘ１１）

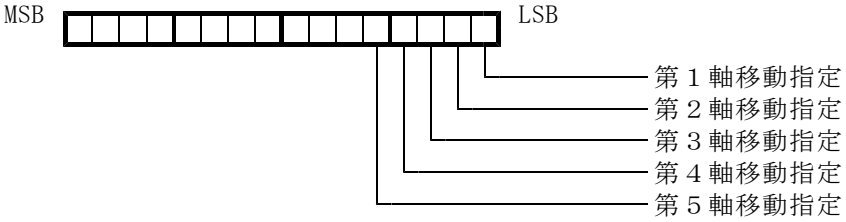
《説明》
任意の軸にＪＯＧ移動を開始させるものです。
有効軸以外の軸指定は、無効です。
ＪＯＧ移動は、以下の条件が満たされているときに実行可能です。

《コマンド実行条件》
・サーボＯＮ状態
・動作モードが手動運転モードか自動運転モードかＯＴ無視モード
・動作プログラム未実行 又は ティーチング中
・移動中の軸がない
・手動パルサー選択無効

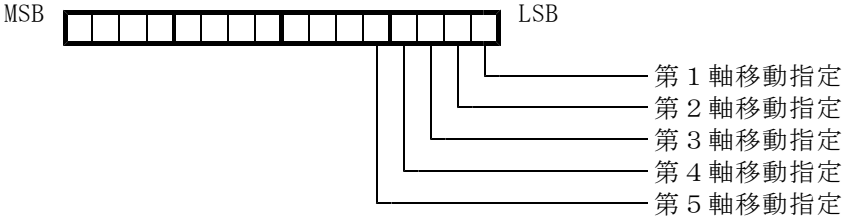
《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
０００００	移動軸選択ﾌﾗｸﾞ	２バイト	――――
０００００２	軸移動方向ﾌﾗｸﾞ	２バイト	――――

（１） 移動軸選択フラグ
ＪＯＧ移動コマンドにて、移動を行わせる軸を指定します。
各軸に対応するビットに１を設定するとその軸が移動を行います。



（２） 軸移動方向フラグ
ＪＯＧ移動コマンドにて、各軸の移動方向を指定します。
各軸に対応するビットに、１を設定するとその軸がー方向へ、０を設定すると＋方向へ移動を行います。



《構造体定義》

```
/*****************************************************************/
/*  ＪＯＧ移動開始コマンド付加データ構造体                       */
/*****************************************************************/
typedef struct {
    unsigned short  AxisFlag;          /* 移動軸選択フラグ        */
    unsigned short  JogVect;           /* 軸移動方向フラグ        */
} JOGSTART;
```

3-2-6.各軸原点復帰コマンド [REQ_ZRNSTART(0x12),REQ_AXAUTOZRN(0x33)]

《データタイプ》
ＲＥＱ＿ＺＲＮＳＴＡＲＴ（０ｘ１２）
ＲＥＱ＿ＡＸＡＵＴＯＺＲＮ（０ｘ３３）

《説明》
任意の軸の原点復帰を開始させるものです。
REQ_ZRNSTART ：各軸手動原点復帰指定（逃げ動作を行いません）
REQ_AXAUTOZRN：各軸自動原点復帰指定（逃げ動作を行います）

有効軸以外の軸指定は、無効です。
原点復帰は、以下の条件が満たされているときに実行可能です。

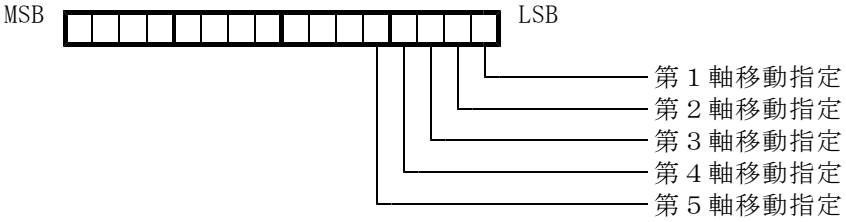
《コマンド実行条件》
・サーボＯＮ状態
・動作モードが手動運転モードか自動運転モード
・動作プログラム未実行
・移動中の軸がない
・手動パルサー選択無効

《関連項目》
3-2-20.全軸原点復帰開始コマンド

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
０００２	原点復帰軸選択ﾌﾗｸﾞ	２バイト	―――――

（１） 原点復帰軸選択フラグ
原点復帰コマンドにて、原点復帰を行わせる軸を指定します。
各軸に対応するビットに１を設定するとその軸が原点復帰を行います。



《構造体定義》
```
/*************************************************************************/
/*  原点復帰移動開始コマンド付加データ構造体                             */
/*************************************************************************/
typedef struct {
    unsigned short  AxisFlag;          /* 移動軸選択フラグ              */
} ZRNSTART;
```

3-2-7.インクレＰＴＰ位置決めコマンド ［REQ_PTPSTART(0x15)］

《データタイプ》
ＲＥＱ＿ＰＴＰＳＴＡＲＴ（０ｘ１５）

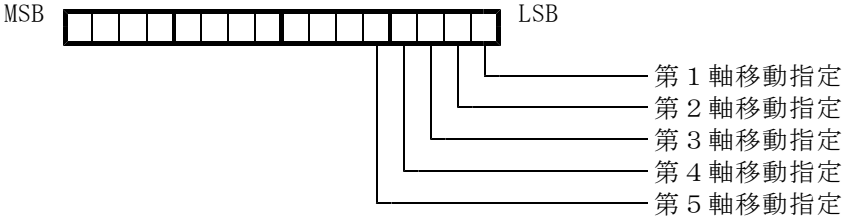
《説明》
インクレモードのＰＴＰ位置決めコマンドを行わせるものです。
有効軸以外の軸指定は、無効です。
［コマンド実行条件］
・ ＳＬＭユニットの動作モードが、手動運転モードか自動運転モード
・ 移動中の軸がない
・ プログラム未実行

《コマンド実行条件》
・サーボＯＮ状態
・動作モードが手動運転モードか自動運転モード
・動作プログラム未実行 又は ティーチング中
・移動中の軸がない
・手動パルサー選択無効

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
０００００	移動軸選択ﾌﾗｸﾞ	４バイト	―――――
００００４	第１軸ｲﾝｸﾘﾒﾝﾄ移動量	４バイト	-99999999～99999999[Pluse]
００００８	第２軸ｲﾝｸﾘﾒﾝﾄ移動量	４バイト	-99999999～99999999[Pluse]
００００Ｃ	第３軸ｲﾝｸﾘﾒﾝﾄ移動量	４バイト	-99999999～99999999[Pluse]
００００１０	第４軸ｲﾝｸﾘﾒﾝﾄ移動量	４バイト	-99999999～99999999[Pluse]
００００１４	第５軸ｲﾝｸﾘﾒﾝﾄ移動量	４バイト	-99999999～99999999[Pluse]

（ １ ） 移動軸選択フラグ
移動コマンドにて、移動を行わせる軸を指定します。
各軸に対応するビットに１を設定するとその軸が移動を行います。



（ ２ ） 第１軸～第５軸インクリメント移動量
各軸の現在位置からの移動量を指定します。 設定単位は［Ｐｕｌｓｅ］です。

《構造体定義》

```
/*****************************************************************************/
/*  ＰＴＰ移動開始コマンドパラメータ構造体                                   */
/*****************************************************************************/
typedef struct {
    unsigned long   AxisFlag;           /* 移動軸選択フラグ                 */
    long            IncAxis[5];         /* 第1軸～第5軸ｲﾝｸﾘﾒﾝﾄ移動量       */
} PTPSTART;
```

3-2-8.アブソＰＴＰ位置決めコマンド ［REQ_PTPASTART(0x16),REQ_PTPBSTART(0x34)］

《データタイプ》
ＲＥＱ＿ＰＴＰＡＳＴＡＲＴ（０ｘ１６）
ＲＥＱ＿ＰＴＰＢＳＴＡＲＴ（０ｘ３４）

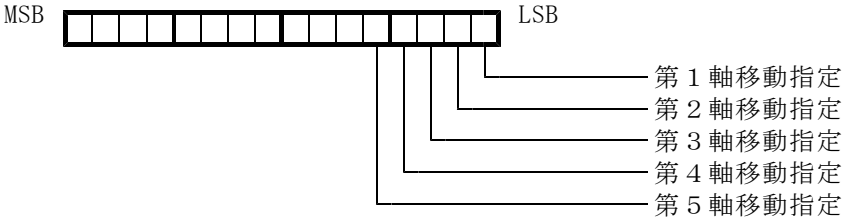
《説明》
アブソモードのＰＴＰ位置決めコマンドを行わせるものです。
REQ_PTPASTART：論理座標系アブソ指定
REQ_PTPBSTART：機械座標系アブソ指定
有効軸以外の軸指定は、無効です。

《コマンド実行条件》
・サーボＯＮ状態
・動作モードが手動運転モードか自動運転モード
・動作プログラム未実行 又は ティーチング中
・移動中の軸がない
・手動パルサー選択無効

《データフォーマット》

ｵﾌｾｯﾄ	デ ー タ 名 称	データ長	設 定 値
０００Ｏ	移動軸選択ﾌﾗｸﾞ	４バイト	―――――
０００４	第１軸目標位置	４バイト	-99999999～99999999[Pluse]
０００８	第２軸目標位置	４バイト	-99999999～99999999[Pluse]
０００Ｃ	第３軸目標位置	４バイト	-99999999～99999999[Pluse]
００１０	第４軸目標位置	４バイト	-99999999～99999999[Pluse]
００１４	第５軸目標位置	４バイト	-99999999～99999999[Pluse]

（ １ ） 移動軸選択フラグ
移動コマンドにて、移動を行わせる軸を指定します。
各軸に対応するビットに１を設定するとその軸が移動を行います。



（ ２ ） 第１軸～第５軸目標位置
各軸の位置決め目標位置を指定します。 設定単位は［Ｐｕｌｓｅ］です。

《構造体定義》

```
/*****************************************************************************/
/*  ＰＴＰ移動開始コマンドパラメータ構造体（ＡＢＳＯ）                      */
/*****************************************************************************/
typedef struct {
    unsigned long   AxisFlag;          /* 移動軸選択フラグ              */
    long            PosAxis[5];        /* 第1軸～第5軸位置決めﾎﾟｼﾞｼｮﾝ  */
} PTPASTART;
```

3-2-9.インクレ直線補間位置決めコマンド [REQ_LINSTART(0x17)]

《データタイプ》
ＲＥＱ＿ＬＩＮＳＴＡＲＴ（０ｘ１７）

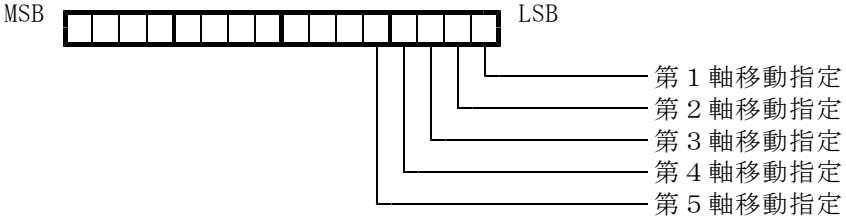
《説明》
インクレモードの補間位置決めコマンドを行わせるものです。
有効軸以外の軸指定は無効です。

《コマンド実行条件》
・サーボＯＮ状態
・動作モードが手動運転モードか自動運転モード
・動作プログラム未実行 又は ティーチング中
・移動中の軸がない
・手動パルサー選択無効

《データフォーマット》

ｵﾌｾｯﾄ	デ ー タ 名 称	データ長	設 定 値
００００	移動軸選択ﾌﾗｸﾞ	４バイト	―――――
００００４	第１軸ｲﾝｸﾘﾒﾝﾄ移動量	４バイト	-99999999～99999999[Pluse]
００００８	第２軸ｲﾝｸﾘﾒﾝﾄ移動量	４バイト	-99999999～99999999[Pluse]
００００Ｃ	第３軸ｲﾝｸﾘﾒﾝﾄ移動量	４バイト	-99999999～99999999[Pluse]
００１０	第４軸ｲﾝｸﾘﾒﾝﾄ移動量	４バイト	-99999999～99999999[Pluse]
００１４	第５軸ｲﾝｸﾘﾒﾝﾄ移動量	４バイト	-99999999～99999999[Pluse]
００１８	補間送り速度	４バイト	1～4095875[PPS]

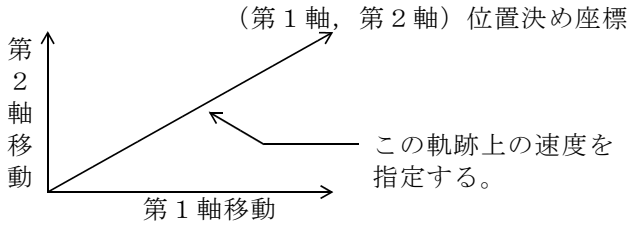
（１） 移動軸選択フラグ
移動コマンドにて、移動を行わせる軸を指定します。
各軸に対応するビットに１を設定するとその軸が移動を行います。



（２） 第１軸～第５軸インクリメント移動量
各軸の現在位置からの移動量を指定します。 設定単位は［Ｐｕｌｓｅ］です。

（３） 補間送り速度
補間送り速度を指定します。 設定単位は［ＰＰＳ］です。

【例：２軸移動の場合】



《構造体定義》

```
/*****************************************************************************/
/*  補間移動開始コマンドパラメータ構造体                                     */
/*****************************************************************************/
typedef struct {
    unsigned long   AxisFlag;          /* 移動軸選択フラグ                 */
    long            IncAxis[5];        /* 第1軸～第5軸ｲﾝｸﾘﾒﾝﾄ移動量       */
    long            Feed;              /* 補間送り速度                     */
} LINSTART;
```

3-2-10.アブソ直線補間位置決めコマンド [REQ_LINASTART(0x18),REQ_LINBSTART(0x35)]

《データタイプ》
ＲＥＱ＿ＬＩＮＡＳＴＡＲＴ（０ｘ１８）
ＲＥＱ＿ＬＩＮＢＳＴＡＲＴ（０ｘ３５）

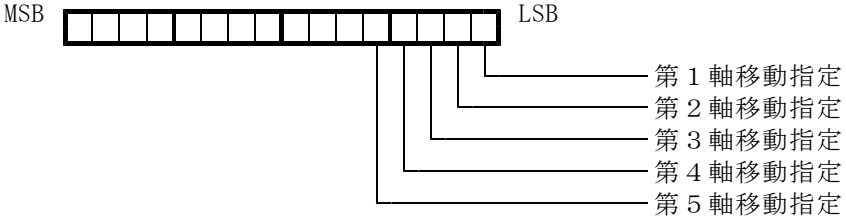
《説明》
アブソモードの補間位置決めコマンドを行わせるものです。
REQ_LINASTART：論理座標系アブソ指定
REQ_LINBSTART：機械座標系アブソ指定
有効軸以外の軸指定は、無効です。

《コマンド実行条件》
・サーボＯＮ状態
・動作モードが手動運転モードか自動運転モード
・動作プログラム未実行 又は ティーチング中
・移動中の軸がない
・手動パルサー選択無効

《データフォーマット》

ｵﾌｾｯﾄ	デ ー タ 名 称	データ長	設 定 値
００００	移動軸選択ﾌﾗｸﾞ	４バイト	―――――
０００４	第１軸目標位置	４バイト	-99999999～99999999[Pluse]
０００８	第２軸目標位置	４バイト	-99999999～99999999[Pluse]
０００Ｃ	第３軸目標位置	４バイト	-99999999～99999999[Pluse]
００１０	第４軸目標位置	４バイト	-99999999～99999999[Pluse]
００１４	第５軸目標位置	４バイト	-99999999～99999999[Pluse]
００１８	補間送り速度	４バイト	1～ 4095875[PPS]

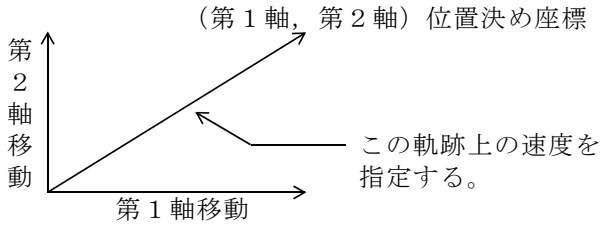
（１） 移動軸選択フラグ
移動コマンドにて、移動を行わせる軸を指定します。
各軸に対応するビットに１を設定するとその軸が移動を行います。



（２） 第１軸～第５軸目標位置
各軸の位置決め目標位置を指定します。　設定単位は［Ｐｕｌｓｅ］です。

（３） 補間送り速度
補間送り速度を指定します。　設定単位は［ＰＰＳ］です。

【例：２軸移動の場合】



《構造体定義》

```
/*********************************************************************/
/*  補間移動開始コマンドパラメータ構造体（ＡＢＳＯ）                 */
/*********************************************************************/
typedef struct {
    unsigned long   AxisFlag;          /* 移動軸選択フラグ             */
    long            PosAxis[5];        /* 第1軸～第5軸位置決めﾎﾟｼﾞｼｮﾝ  */
    long            Feed;              /* 補間送り速度                 */
} LINASTART;
```

3-2-11.リセットコマンド [REQ_RESET(0x1a)]

《データタイプ》
ＲＥＱ＿ＲＥＳＥＴ（０ｘ１ａ）

《説明》
ＳＬＭユニットのリセットを行わせるものです。
ＳＬＭユニットは、リセット処理にて以下の動作を行います。

［リセット動作］
- プログラム実行中の場合は、プログラム実行を終了させる。
- 移動中の軸がある場合は、移動を終了させる。
- アラーム発生中の場合は、アラーム解除／サーボＯＮを行う。
  （但し、アラーム要因が解除されていない場合、再びアラームになります。）

《コマンド実行条件》
ＳＬＭの動作モードや動作状態に関わらず常時実行可能です。

《データフォーマット》
コマンド付加データは有りません。

3-2-12.原点設定コマンド [REQ_ORGSET(0x19)]

《データタイプ》
ＲＥＱ＿ＯＲＧＳＥＴ（０ｘ１９）

《説明》
現在の軸指令位置を指令座標の原点として設定するものです。
原点設定は、全ての軸が停止しているときのみ実行可能です。

《コマンド実行条件》
- サーボＯＮ状態
- 動作モードが手動運転モードか自動運転モード
- 動作プログラム未実行 又は ティーチング中
- 移動中の軸がない
- 手動パルサー選択無効

《データフォーマット》
コマンド付加データは有りません。

3-2-13.汎用出力直接制御コマンド ［REQ_OUTPUT(0x1d)］

《データタイプ》
ＲＥＱ＿ＯＵＴＰＵＴ（０ｘ１ｄ）

《説明》
ＳＬＭユニットの汎用出力を直接ＯＮ／ＯＦＦするものです。
汎用出力直接制御コマンドは、ＳＬＭユニットの動作モードや動作状態に関わらず実行可能です。
（出力ポート／ビットにより一部制限があります。）

変更出来る出力は汎用出力（Ｒｏ００～Ｒｏ３０）です。
但し、動作プログラムにて出力命令を実行すると変更されます。
（後に指定された方が有効になります。）

《コマンド実行条件》
常時実行可能です。

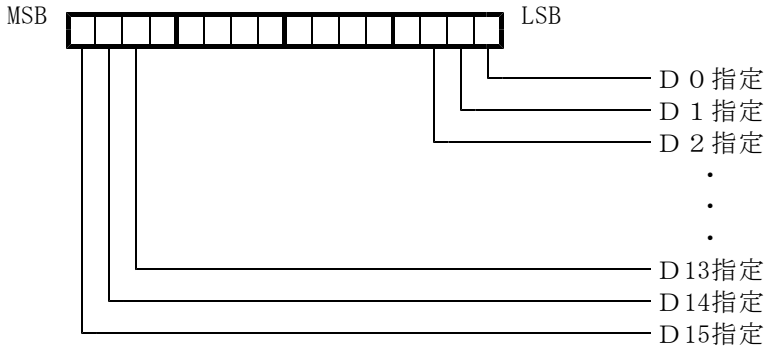
《関連項目》
3-2-23.汎用入力強制制御コマンド
3-2-24.汎用出力強制制御コマンド
3-2-25.汎用入出力ビット強制制御コマンド、汎用出力ビット直接制御コマンド

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
０００００	出力１ﾊﾟﾀｰﾝ ［#0000］	２バイト	――――
０００００２	出力２ﾊﾟﾀｰﾝ ［#0001］	２バイト	――――

※各出力の割付については、「ＳＬＭ－４０００ユーザーズマニュアル[TB00-0800]」
＜Ⅲ 機能編 2.入出力機能＞を参照して下さい。

（１） 出力パターン
各ビットに１を設定した場合、そのビットに対応する出力がＯＮになります。
０を設定した場合、ＯＦＦになります。



《構造体定義》

```
/*************************************************************************/
/*  ＳＬＭ汎用出力直接制御コマンドパラメータ構造体                       */
/*************************************************************************/
typedef struct {
    unsigned short  OutputPat[2];       /* 汎用出力１～２               */
} OUTPUTPAT;
```

3-2-14.サーボ電源ＯＮコマンド [REQ_SERVOON(0x1e)]

《データタイプ》
ＲＥＱ＿ＳＥＲＶＯＯＮ（０ｘ１ｅ）

《説明》
ＳＬＭユニットに対して、サーボ電源ＯＮを行わせるものです。

《コマンド実行条件》
常時実行可能です。

《データフォーマット》
コマンド付加データは有りません。

3-2-15.サーボ電源ＯＦＦコマンド [REQ_SERVOOFF(0x1f)]

《データタイプ》
ＲＥＱ＿ＳＥＲＶＯＯＦＦ（０ｘ１ｆ）

《説明》
ＳＬＭユニットに対して、サーボ電源ＯＦＦを行わせるものです。

《コマンド実行条件》
常時実行可能です。

《データフォーマット》
コマンド付加データは有りません。

3-2-16.プログラム実行開始コマンド [REQ_PROGSTRT(0x20)]

《データタイプ》
ＲＥＱ＿ＰＲＯＧＳＴＲＴ（０ｘ２０）

《説明》
ＳＬＭユニットに対して、プログラム実行を行わせるものです。
プログラム実行開始は、以下の条件が満たされているときに実行可能です。

《コマンド実行条件》
・サーボＯＮ状態
・アラーム未発生
・動作モードが自動運転モードかＤＮＣ運転モード
（自動運転モードなら、現在のプログラム番号に動作プログラムダウンロード済み）
・動作プログラム未実行（開始時） 又は 停止中（再開時）
・移動中の軸がない
・手動パルサー選択無効
・原点復帰無効の軸を除く、全ての軸が原点復帰済み

《データフォーマット》
コマンド付加データは有りません。

3-2-17.プログラム実行停止コマンド [REQ_PROGSTOP(0x21)]

《データタイプ》
ＲＥＱ＿ＰＲＯＧＳＴＯＰ（０ｘ２１）

《説明》
ＳＬＭユニットに対して、プログラム実行の一時停止を行わせるものです。
プログラム実行停止コマンドは、プログラム実行が行われている時のみ有効です。
本コマンドにて一時停止したプログラム実行は、プログラム実行開始コマンドにて再開できます。

《コマンド実行条件》
常時実行可能です。ただし、プログラム実行中／軸移動中でないときはなにもしません。

《データフォーマット》
コマンド付加データは有りません。

3-2-18.実行プログラム選択コマンド [REQ_PROGSLCT(0x22)]

《データタイプ》
ＲＥＱ＿ＰＲＯＧＳＬＣＴ（０ｘ２２）

《説明》
ＳＬＭユニットに対して、実行するプログラムの指定を行うものです。
プログラム選択コマンドは、動作モード・状態に関わらず実行可能です。

《コマンド実行条件》
常時実行可能です。
ただし、プログラム実行中 又は 機械パネルにてプログラム番号選択中は、ステータスデータの現在プログラム選択番号に反映されません。
プログラム未実行 かつ 機械パネルのプログラム番号選択無効 の時に設定した番号が反映されます。

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
０００００	選択プログラム番号	２バイト	―――――

（１） 選択プログラム番号
実行するプログラムを番号で指定します。
指定できるプログラム番号はプログラム容量により、下表の通りとなります。

	プログラム番号指定範囲 [選択されるプログラム番号]
888ｽﾃｯﾌﾟ仕様	０ ～ ２（１～ ３）
444ｽﾃｯﾌﾟ仕様	０ ～ ５（１～ ６）
222ｽﾃｯﾌﾟ仕様	０ ～１１（１～１２）

《構造体定義》
```
/*****************************************************************************/
/*  プログラム選択コマンドパラメータ構造体                                   */
/*****************************************************************************/
typedef struct {
    unsigned short  ProgSel;            /* 選択プログラム番号    */
} PROGSEL;
```

3-2-19.オーバーライド設定変更コマンド [REQ_OVRDCHG(0x23)]

《データタイプ》
ＲＥＱ＿ＯＶＲＤＣＨＧ（０ｘ２３）

《説明》
ＳＬＭユニットに対して、送りオーバーライド設定の変更を行わせるものです。
送りオーバーライドコマンドは、動作モード・状態に関わらず実行可能です。

《コマンド実行条件》
・機械パネルのオーバーライド選択無効

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	ｵｰﾊﾞｰﾗｲﾄﾞ設定値	２バイト	０～７

（１） オーバーライド設定値
オーバーライドを番号で指定します。
番号とオーバーライド設定の関係は下表の通りです。

番号	オーバーライド
0	２５％
1	５０％
2	７５％
3	１００％
4	１２５％
5	１５０％
6	１７５％
7	２００％

《構造体定義》
```
/*****************************************************************************/
/*  軸速度オーバーライド変更コマンドパラメータ構造体                         */
/*****************************************************************************/
typedef struct {
    unsigned short   Override;          /* オーバーライド設定値 */
} OVERCHG;
```

3-2-20.全軸原点復帰開始コマンド ［REQ_ALLZRN(0x24)］

《データタイプ》
ＲＥＱ＿ＡＬＬＺＲＮ（０ｘ２４）

《説明》
ＳＬＭユニットに対して、全軸の原点復帰を行わせるものです。
本コマンドにて、ＳＬＭユニットが実行する全軸原点復帰処理は、原点復帰入力による全軸原点復帰処理と等価なものです。
原点復帰は、以下の条件が満たされているときに実行可能です。

《コマンド実行条件》
・サーボＯＮ状態
・アラーム未発生
・動作モードが手動運転モード／自動運転モード／ＤＮＣ運転モード
・動作プログラム未実行
・移動中の軸がない
・手動パルサー選択無効

《関連項目》
3-2-6.原点復帰コマンド
3-2-20.全軸原点復帰開始コマンド

《データフォーマット》
コマンド付加データは有りません。

3-2-21.高速センサーラッチインクレ補間位置決めコマンド［REQ_SLINSTART(0x28)］〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＲＥＱ＿ＳＬＩＮＳＴＡＲＴ（０ｘ２８）

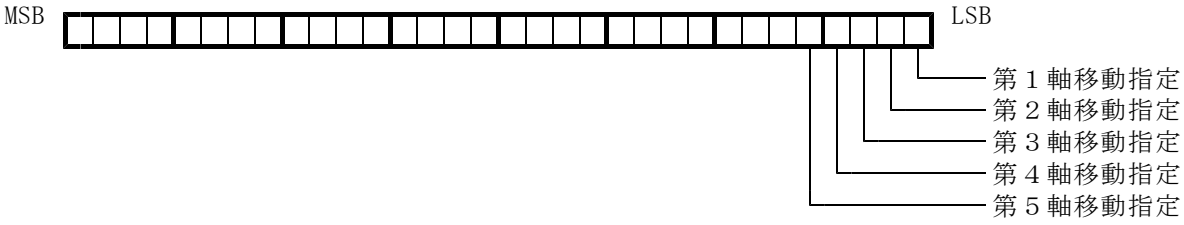
《説明》
ＳＬＭユニットに対して、高速センサーラッチインクレ補間位置決め動作を行わせるものです。
有効軸以外の軸指定は、無効です。

《コマンド実行条件》
・サーボＯＮ状態
・動作モードが手動運転モード／自動運転モード／ＤＮＣ運転モード
・動作プログラム未実行 又は ティーチング中
・移動中の軸がない
・手動パルサー選択無効

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
０００００	移動軸選択ﾌﾗｸﾞ	４バイト	――――
００００４	第１軸ｲﾝｸﾘﾒﾝﾄ移動量	４バイト	-99999999～99999999[Pluse]
００００８	第２軸ｲﾝｸﾘﾒﾝﾄ移動量	４バイト	-99999999～99999999[Pluse]
００００Ｃ	第３軸ｲﾝｸﾘﾒﾝﾄ移動量	４バイト	-99999999～99999999[Pluse]
００１０	第４軸ｲﾝｸﾘﾒﾝﾄ移動量	４バイト	-99999999～99999999[Pluse]
００１４	第５軸ｲﾝｸﾘﾒﾝﾄ移動量	４バイト	-99999999～99999999[Pluse]
００１８	補間送り速度	４バイト	1～4095875[PPS]
００１Ｃ	スキップ抑制フラグ	２バイト	０：スキップ、１：継続

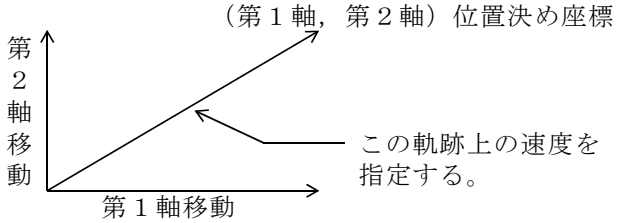
（１） 移動軸選択フラグ
移動コマンドにて、移動を行わせる軸を指定します。
各軸に対応するビットに１を設定するとその軸が移動を行います。



（２） 第１軸～第５軸インクリメント移動量
各軸の現在位置からの移動量を指定します。　設定単位は［Ｐｕｌｓｅ］です。

（３） 補間送り速度
補間送り速度を指定します。　設定単位は［ＰＰＳ］です。

【例：２軸移動の場合】



（４） スキップ抑制フラグ
軸移動中にセンサーラッチ入力が入った時に、移動をキャンセルするかどうかを
指定します。

《構造体定義》

```
/*****************************************************************************/
/*  高速センサーラッチ補間移動開始コマンドパラメータ構造体                   */
/*****************************************************************************/
typedef struct {
    unsigned long   AxisFlag;           /* 移動軸選択フラグ                  */
    long            IncAxis[5];         /* 第1軸～第5軸ｲﾝｸﾘﾒﾝﾄ移動量        */
    long            Feed;               /* 補間送り速度                      */
    short           NoSkipF;            /* スキップ抑制フラグ                */
} SLINSTART;
```

3-2-22.高速センサーラッチアブソ補間位置決めコマンド［REQ_SLINASTART(0x29)］〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＲＥＱ＿ＳＬＩＮＡＳＴＡＲＴ（０ｘ２９）

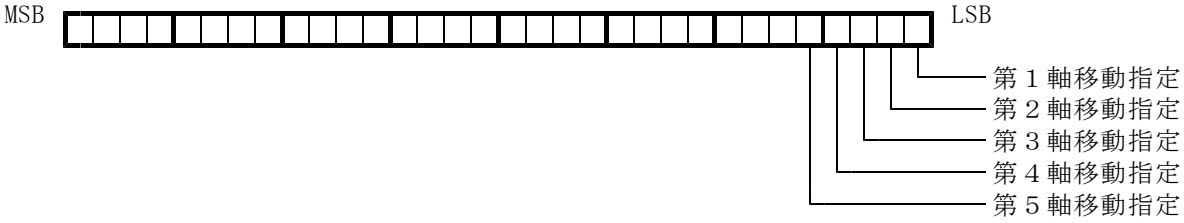
《説明》
ＳＬＭユニットに対して、高速センサーラッチアブソ補間位置決め動作を行わせるものです。
有効軸以外の軸指定は、無効です。

《コマンド実行条件》
・サーボＯＮ状態
・動作モードが手動運転モード／自動運転モード／ＤＮＣ運転モード
・動作プログラム未実行 又は ティーチング中
・移動中の軸がない
・手動パルサー選択無効

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	移動軸選択ﾌﾗｸﾞ	４バイト	―――――
０００４	第１軸目標位置	４バイト	-99999999～99999999[Pluse]
０００８	第２軸目標位置	４バイト	-99999999～99999999[Pluse]
０００Ｃ	第３軸目標位置	４バイト	-99999999～99999999[Pluse]
００１０	第４軸目標位置	４バイト	-99999999～99999999[Pluse]
００１４	第５軸目標位置	４バイト	-99999999～99999999[Pluse]
００１８	補間送り速度	４バイト	1～4095875[PPS]
００１Ｃ	スキップ抑制フラグ	２バイト	０：スキップ、１：継続

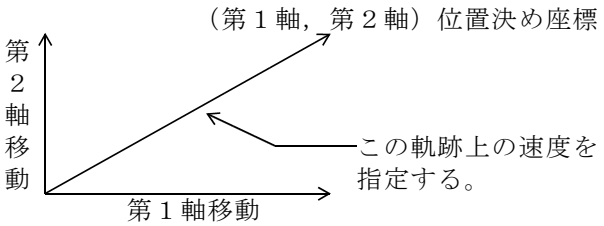
（１） 移動軸選択フラグ
移動コマンドにて、移動を行わせる軸を指定します。
各軸に対応するビットに１を設定するとその軸が移動を行います。



（２） 第１軸～第５軸目標位置
各軸の位置決め目標位置を指定します。　設定単位は［Ｐｕｌｓｅ］です。

（３） 補間送り速度
補間送り速度を指定します。　設定単位は［ＰＰＳ］です。

【例：２軸移動の場合】



（４） スキップ抑制フラグ
軸移動中にセンサーラッチ入力が入った時に、移動をキャンセルするかどうかを指定します。

《構造体定義》

```
/*************************************************************************/
/*  高速センサーラッチ補間移動開始コマンドパラメータ構造体（ＡＢＳＯ）  */
/*************************************************************************/
typedef struct {
    unsigned long   AxisFlag;          /* 移動軸選択フラグ              */
    long            PosAxis[5];        /* 第1軸～第5軸位置決めﾎﾟｼﾞｼｮﾝ  */
    long            Feed;              /* 補間送り速度                  */
    short           NoSkipF;           /* スキップ抑制フラグ            */
} SLINASTART;
```

3-2-23.汎用入力強制制御コマンド ［REQ_COMPINPUT(0x2a)］

《データタイプ》
ＲＥＱ＿ＣＯＭＰＩＮＰＵＴ（０ｘ２ａ）

《説明》
ＳＬＭユニットに対して、任意の入力の強制的なＯＮ／ＯＦＦを一括して指定するものです。

《コマンド実行条件》
常時実行可能です。

《関連項目》
3-2-13.汎用出力直接制御コマンド
3-2-24.汎用出力強制制御コマンド
3-2-25.汎用入出力ビット強制制御コマンド、汎用出力ビット直接制御コマンド

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	入力１ｺﾏﾝﾄﾞ［#0000］	16バイト	―――
００１０	入力２ｺﾏﾝﾄﾞ［#0001］	16バイト	―――
００２０	入力３ｺﾏﾝﾄﾞ［#0002］	16バイト	―――

各入力コマンドのフォーマットは以下の通りです。

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	D00ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００１	D01ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００２	D02ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００３	D03ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００４	D04ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００５	D05ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００６	D06ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００７	D07ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００８	D08ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００９	D09ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００Ａ	D10ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００Ｂ	D11ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００Ｃ	D12ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００Ｄ	D13ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００Ｅ	D14ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００Ｆ	D15ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３

（１） ビット変更コマンド
０：無変更
１：強制ＯＮ指定
２：強制ＯＦＦ指定
３：強制制御解除

※各入出力の割付については、「ＳＬＭ－４０００ユーザーズマニュアル[TB00-0800]」
＜**Ⅲ 機能編** 2.入出力機能＞を参照して下さい。

《構造体定義》

```
/*****************************************************************************/
/*  汎用入力／出力強制制御コマンドパラメータサブ構造体                       */
/*****************************************************************************/
typedef struct {
    unsigned char   IoD00Cmd;           /* 汎用入出力D00ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD01Cmd;           /* 汎用入出力D01ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD02Cmd;           /* 汎用入出力D02ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD03Cmd;           /* 汎用入出力D03ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD04Cmd;           /* 汎用入出力D04ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD05Cmd;           /* 汎用入出力D05ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD06Cmd;           /* 汎用入出力D06ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD07Cmd;           /* 汎用入出力D07ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD08Cmd;           /* 汎用入出力D08ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD09Cmd;           /* 汎用入出力D09ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD10Cmd;           /* 汎用入出力D10ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD11Cmd;           /* 汎用入出力D11ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD12Cmd;           /* 汎用入出力D12ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD13Cmd;           /* 汎用入出力D13ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD14Cmd;           /* 汎用入出力D14ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD15Cmd;           /* 汎用入出力D15ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
} IO_CMD;

/*****************************************************************************/
/*  汎用入力強制制御コマンドパラメータ構造体                                 */
/*****************************************************************************/
typedef struct {
    IO_CMD          InCmd[3];
} COMPINPUT;
```

3-2-24.汎用出力強制制御コマンド ［REQ_COMPOUTPUT(0x2b)］

《データタイプ》
ＲＥＱ＿ＣＯＭＰＯＵＴＰＵＴ（０ｘ２ｂ）

《説明》
ＳＬＭユニットに対して、任意の出力の強制的なＯＮ／ＯＦＦを一括して指定するものです。

《コマンド実行条件》
常時実行可能です。

《関連項目》
3-2-13.汎用出力直接制御コマンド
3-2-23.汎用入力強制制御コマンド
3-2-25.汎用入出力ビット強制制御コマンド、汎用出力ビット直接制御コマンド

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
０００００	出力１ｺﾏﾝﾄﾞ［#0000］	16バイト	―――
０００１０	出力２ｺﾏﾝﾄﾞ［#0001］	16バイト	―――

各入力コマンドのフォーマットは以下の通りです。

ｵﾌｾｯﾄ	データ名称	データ長	設定値
０００００	D00ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００１	D01ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００２	D02ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００３	D03ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００４	D04ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００５	D05ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００６	D06ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００７	D07ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００８	D08ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００９	D09ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００Ａ	D10ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００Ｂ	D11ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００Ｃ	D12ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００Ｄ	D13ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００Ｅ	D14ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３
００００Ｆ	D15ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ	１バイト	０～３

（１） ビット変更コマンド
０：無変更
１：強制ＯＮ指定
２：強制ＯＦＦ指定
３：強制制御解除

※各入出力の割付については、「ＳＬＭ－４０００ユーザーズマニュアル[TB00-0800]」
＜**Ⅲ 機能編** 2.入出力機能＞を参照して下さい。

《構造体定義》

```
/*****************************************************************************/
/*  汎用入力／出力強制制御コマンドパラメータサブ構造体                       */
/*****************************************************************************/
typedef struct {
    unsigned char   IoD00Cmd;           /* 汎用入出力D00ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD01Cmd;           /* 汎用入出力D01ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD02Cmd;           /* 汎用入出力D02ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD03Cmd;           /* 汎用入出力D03ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD04Cmd;           /* 汎用入出力D04ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD05Cmd;           /* 汎用入出力D05ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD06Cmd;           /* 汎用入出力D06ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD07Cmd;           /* 汎用入出力D07ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD08Cmd;           /* 汎用入出力D08ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD09Cmd;           /* 汎用入出力D09ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD10Cmd;           /* 汎用入出力D10ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD11Cmd;           /* 汎用入出力D11ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD12Cmd;           /* 汎用入出力D12ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD13Cmd;           /* 汎用入出力D13ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD14Cmd;           /* 汎用入出力D14ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
    unsigned char   IoD15Cmd;           /* 汎用入出力D15ﾋﾞｯﾄ変更ｺﾏﾝﾄﾞ   */
} IO_CMD;

/*****************************************************************************/
/*  汎用出力強制制御コマンドパラメータ構造体                                 */
/*****************************************************************************/
typedef struct {
    IO_CMD          OutCmd[2];
} COMPOUTPUT;
```

3-2-25.汎用入出力ビット強制制御コマンド、汎用出力ビット直接制御コマンド [REQ_COMPIOBIT(0x2c),REQ_OUTPUTBIT(0x44)]

《データタイプ》
ＲＥＱ＿ＣＯＭＰＩＯＢＩＴ（０ｘ２ｃ）：汎用入出力ビット強制制御コマンド
ＲＥＱ＿ＯＵＴＰＵＴＢＩＴ（０ｘ４４）：汎用出力ビット直接制御コマンド

《説明》
汎用入出力ビット強制制御コマンド
ＳＬＭユニットに対して任意の入出力の強制的なＯＮ／ＯＦＦを指定するものです。
汎用出力ビット直接制御コマンド
ＳＬＭユニットに対して任意の出力のＯＮ／ＯＦＦを指定するものです。

《コマンド実行条件》
常時実行可能です。

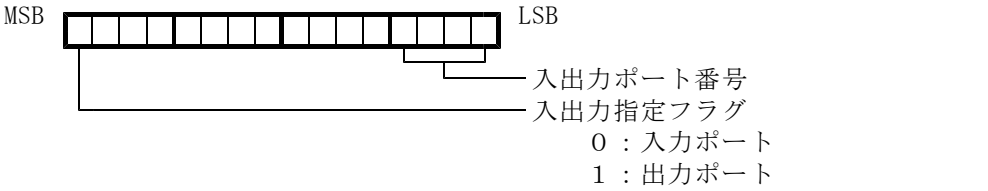
《関連項目》
3-2-13.汎用出力直接制御コマンド
3-2-23.汎用入力強制制御コマンド
3-2-24.汎用出力強制制御コマンド

《データフォーマット》

ｵﾌｾｯﾄ	デ ー タ 名 称	データ長	設 定 値
０００００	入出力ポート番号	２バイト	入力：0～2、出力：0～1
００００２	制御ビット番号	２バイト	０～１５
００００４	制御フラグ	２バイト	―――――

※各出力の割付については、「ＳＬＭ－４０００ユーザーズマニュアル[TB00-0800]」〈Ⅲ 機能編　２．入出力機能〉を参照して下さい。

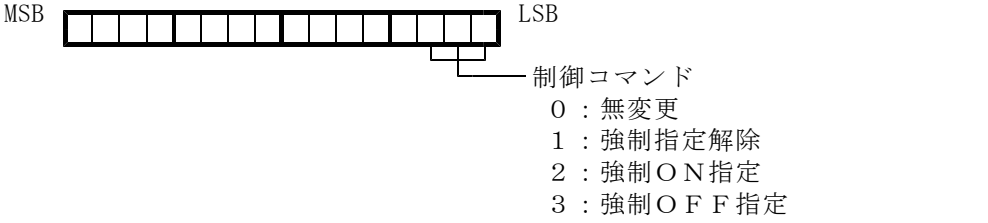
（１） 入出力ポート番号



（汎用出力ビット直接制御コマンドでは”入出力指定フラグ”は無視します）

（２） 制御ビット番号
制御対象のビット（Ｄ００～Ｄ１５）を指定します。

（３） 制御フラグ



（汎用出力ビット直接制御コマンドでは”強制指定解除”は無視します）

《構造体定義》

```
/*****************************************************************************/
/*  汎用入出力強制制御コマンドパラメータ構造体                                 */
/*****************************************************************************/
typedef struct {
    unsigned short      Pno;                /* 入出力ポート番号              */
    unsigned short      Bno;                /* 制御ビット番号                */
    unsigned short      Flg;                /* 制御フラグ                    */
} COMPIOBIT;
```

3-2-26.送りオーバーライド%変更コマンド [REQ_OVRDCHGP(0x2d)]

《データタイプ》

ＲＥＱ＿ＯＶＲＤＣＨＧＰ（０ｘ２ｄ）

《説明》

送りオーバーライドを０％～２００％の範囲で１％刻みで設定できます。

《コマンド実行条件》

・機械パネルのオーバーライド選択が無効

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	ｵｰﾊﾞｰﾗｲﾄﾞ設定値	２バイト	０～２００

《構造体定義》

```
/*****************************************************************************/
/*  軸速度オーバーライド変更コマンドパラメータ構造体                           */
/*****************************************************************************/
typedef struct {
    unsigned short  Override;           /* オーバーライド設定値 */
} OVERCHG;
```

3-2-27.主軸回転ＯＮ／ＯＦＦコマンド［REQ_SPCMND(0x50)］〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＲＥＱ＿ＳＰＣＭＮＤ（０ｘ５０）

《説明》
ＳＬＭユニットに対して、主軸回転のＯＮ／ＯＦＦを行わせるものです。
主軸回転ＯＮを司令する場合、回転方向は正転のみ指定できます。
本コマンドの実行には、ＳＬＭユニットに主軸オプションが追加されている必要があります。
主軸回転は、以下の条件が満たされているときに実行可能です。

《コマンド実行条件》
・サーボＯＮ状態
・アラーム未発生
・動作モードが手動運転モード／自動運転モード／ＤＮＣ運転モード
・移動中の軸がない
・Ｍコード出力中でない

《データフォーマット》

ｵﾌｾｯﾄ	デ ー タ 名 称	データ長	設 定 値
０００００	主軸指令フラグ	２バイト	0:停止、1:正転、2:逆転

（ １ ） 主軸指令フラグ
主軸の動作を番号で指定します。
番号と主軸動作の関係は下表の通りです。

番号	主軸動作
0	主軸停止
1	主軸正転
2	主軸逆転

《構造体定義》
```
/*****************************************************************************/
/*  主軸回転ＯＮ／ＯＦＦコマンドパラメータ構造体                             */
/*****************************************************************************/
typedef struct {
    short           SpOut;              /* 主軸指令フラグ       */
} SPCMND;
```

3-2-28.Z軸接線制御ON/OFFコマンド [REQ_TLINE(0x54)] 〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＲＥＱ＿ＴＬＩＮＥ（０ｘ５４）

《説明》
ＳＬＭユニットに対して、無限回転軸であるＺ軸を第１軸と第２軸の合成移動方向に、
自動的に回転動作させるものです。

《コマンド実行条件》
・アラーム未発生
・動作プログラム未実行 又は 停止中
・移動中の軸がない

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	Ｚ軸接線制御機能ｽｲｯﾁ	２バイト	０：ＯＦＦ、１：ＯＮ

《構造体定義》
```
/*****************************************************************************/
/*  Ｚ軸接線制御ON/OFFコマンドパラメータ構造体                                 */
/*****************************************************************************/
typedef struct {
    unsigned short  tlinesw;            /* Ｚ軸接線制御機能ｽｲｯﾁ */
} TLINESW;
```

3-2-29.シングルステップモード設定コマンド [REQ_SINGLE(0x60)]

《データタイプ》
ＲＥＱ＿ＳＩＮＧＬＥ（０ｘ６０）

《説明》
ＳＬＭユニットに対して、シングルステップモードへの移行と解除をトグルに行わせるものです。
本コマンドは、ＳＬＭユニットが自動運転モードの時のみ、実行することが出来ます。

シングルステップモードに移行すると、「**3-1-15.プログラム実行情報データ**」のプログラム実行
ステータスにP_SINGLEビットが立ちます。

《コマンド実行条件》
機械パネルのモード選択無効

《データフォーマット》
コマンド付加データは有りません。

3-2-30.ティーチング設定コマンド [REQ_TEACH(0x61)]

《データタイプ》
ＲＥＱ＿ＴＥＡＣＨ（０ｘ６１）

《説明》
ＳＬＭユニットに対して、ティーチングモードへの移行と解除をトグルに行わせるものです。

ティーチングモードに移行すると、「**3-1-15.プログラム実行情報データ**」のプログラム実行ステータスにP_TEACHビットが立ちます。

※ティーチングモードの解除時に移動軸が存在した時は、ティーチングモードを解除出来ません。

《コマンド実行条件》（ティーチング有効）
・機械パネルのティーチングモード選択無効
・ＰＬＭＣの動作モードが自動運転モード
・シングルステップモードでのプログラム実行中
・ステップ間停止中（ステップ途中停止中は不可）

《コマンド実行条件》（ティーチング解除）
・機械パネルのティーチングモード選択無効
・ティーチングで軸を移動した後、元の位置へ戻していない。（挿入・置換をした場合は不要）

《データフォーマット》
コマンド付加データは有りません。

3-2-31.プログラムステップ挿入コマンド [REQ_PRGINS(0x62)]

《データタイプ》
ＲＥＱ＿ＰＲＧＩＮＳ（０ｘ６２）

《説明》
ＳＬＭユニットに対して、現在ステップへの移動命令の挿入を行わせるものです。
ティーチングモードへ移行してからの各軸の移動量をＰＴＰ[ﾃｸﾉｺｰﾄﾞ]／Ｇ００[Gｺｰﾄﾞ]命令で挿入します。
ステップ挿入の結果、現在ステップ以降のステップは、１つづつ後ろへずれます。
本コマンドが、正常に終了すると、ティーチングモードを自動的に解除します。

《コマンド実行条件》
・ＳＬＭのプログラム実行状態が、ティーチングモード
・ステップ挿入の結果が最大ステップ数を越えない
・ステップ挿入可能ステップ

《データフォーマット》
コマンド付加データは有りません。

3-2-32.プログラムステップ置換コマンド [REQ_PRGALT(0x63)]

《データタイプ》
ＲＥＱ＿ＰＲＧＡＬＴ（０ｘ６３）

《説明》
ＳＬＭユニットに対して、現在ステップへの移動命令の置換を行わせるものです。
本コマンドが、正常に終了すると、ティーチングモードを自動的に解除します。

《コマンド実行条件》
・ＳＬＭのプログラム実行状態が、ティーチングモード
・ステップ置換可能ステップ

《データフォーマット》
コマンド付加データは有りません。

3-2-33.プログラムステップ削除コマンド [REQ_PRGDEL]

《データタイプ》
ＲＥＱ＿ＰＲＧＤＥＬ

《説明》
ＳＬＭユニットに対して、現在ステップの削除を行わせるものです。

《コマンド実行条件》
・ＳＬＭの動作モードが自動運転モード
・シングルステップモードでステップ間停止中（ステップ途中停止中は不可）
・ＳＬＭのプログラム実行状態が、ティーチングモードでない

《データフォーマット》
コマンド付加データは有りません。

3-2-34.プログラムステップ逆行コマンド [REQ_PRGREV(0x65)]

《データタイプ》
ＲＥＱ＿ＰＲＧＲＥＶ（０ｘ６５）

《説明》
ＳＬＭユニットに対して、以下の動作を行わせるものです。

《コマンド実行条件》
１．ティーチング中
ティーチング開始位置への復帰

２．シングルステップモードでステップ間停止中（ステップ途中停止中は不可）＜ｵﾌﾟｼｮﾝ＞
前回ステップを逆行します。
（インクレ移動命令のみ逆行可能です。）

《データフォーマット》
コマンド付加データは有りません。

3-2-35.プログラムステップ変更コマンド [REQ_STEPCHG(0x66)]

《データタイプ》
ＲＥＱ＿ＳＴＥＰＣＨＧ（０ｘ６６）

《説明》
ＳＬＭユニットに対して、ティーチング対象のステップの変更を行わせるものです。

《コマンド実行条件》
・ティーチングモード
・ティーチングの移動をしていない

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	ステップナンバー	２バイト	１～９９９

《構造体定義》

```
/*****************************************************************************/
/*  ティーチングステップ変更コマンドパラメータ構造体                         */
/*****************************************************************************/
typedef struct {
    unsigned short  StepNo;
} STEPCHG;
```

3-2-36.プログラムステップスキップコマンド [REQ_STEPSKIP(0x67)]

《データタイプ》
ＲＥＱ＿ＳＴＥＰＳＫＩＰ（０ｘ６７）

《コマンド実行条件》
・ステップ間停止中
・ティーチングモード無効

《説明》
ＳＬＭユニットに対して、実行中ステップのスキップを行わせるものです。

《データフォーマット》
コマンド付加データは有りません。

3-2-37.マルチタスクプログラム開始コマンド［REQ_TASKSTART(0x80)］〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＲＥＱ＿ＴＡＳＫＳＴＡＲＴ（０ｘ８０）

《説明》
ＳＬＭユニットに対して、指定したタスクの起動を行わせるものです。
マルチタスクの詳細については「SLM-4000ユーザーズマニュアル[TB00-0800]」＜Ⅲ 機能編 6-5.マルチタスク＞を参照下さい。

《コマンド実行条件》
○マスタータスク
・サーボＯＮ状態
・アラーム未発生
・動作モードが自動運転モードかＤＮＣ運転モード
（自動運転モードなら、現在のプログラム番号に動作プログラムダウンロード済み）
・動作プログラム未実行（開始時） 又は 停止中（再開時）
・移動中の軸がない
・手動パルサー選択無効
・原点復帰無効の軸を除く、全ての軸が原点復帰済み

○スレーブタスク
・プログラム停止中（再開）
プログラム新規開始は不可（マスタータスクからのみ起動できます。）

○バックグランドタスク
・バックアップエラー未発生
・動作プログラムダウンロード済み

○リセットタスク／EXITタスク／割り込みタスク
・アラーム未発生
・動作モードが自動運転モードかＤＮＣ運転モード
・移動中の軸がない
・原点復帰無効の軸を除く、全ての軸が原点復帰済み

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	タスク番号	４バイト	－

タスク番号
動作を指定するタスクに対応する番号を指定します。
・マスタータスク　：０
・スレーブタスク　：１
・バックグランドタスク　：２
・リセットタスク　：３
・アラームタスク　：４
・ＥＸＩＴタスク　：５
・割り込みタスク　：６

《列挙型定義》
```
typedef enum {
    MSTTASK = 0,                    /* マスタータスク          */
    SLVTASK,                        /* スレーブタスク          */
    BGTASK,                         /* バックグランドタスク    */
    RSTTASK,                        /* リセットタスク          */
    ALMTASK,                        /* アラームタスク          */
    EXTTASK,                        /* Ｅｘｉｔタスク          */
    INTTASK,                        /* 割り込みタスク          */
} PRGTASK;
```

3-2-38.マルチタスクプログラム停止コマンド［REQ_TASKSTOP(0x81)］〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＲＥＱ＿ＴＡＳＫＳＴＯＰ（０ｘ８１）

《説明》
ＳＬＭユニットに対して、指定したタスクの停止を行わせるものです。

《コマンド実行条件》
常時実行可能です。

《データフォーマット》
「**3-2-37.マルチタスクプログラム開始コマンド**」と同様です。

3-2-39.マルチタスクプログラムリセットコマンド［REQ_TASKRESET(0x82)］〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＲＥＱ＿ＴＡＳＫＲＥＳＥＴ（０ｘ８２）

《説明》
ＳＬＭユニットに対して、指定したタスクのリセットを行わせるものです。

《コマンド実行条件》
常時実行可能です。

《データフォーマット》
「**3-2-37.マルチタスクプログラム開始コマンド**」と同様です。

◆1.2

3-2-40.回転軸回転動作コマンド ［REQ_SPINAX(0x52)］

《データタイプ》
ＲＥＱ＿ＳＰＩＮＡＸ（０ｘ５２）

《説明》
ＳＬＭユニットに対して、任意の無限回転軸を定速回転運動の開始及び停止させるものです。
回転動作の実行中に再度回転指令を行う事により、停止することなく回転速度の変更が行えます。
無限回転軸の設定は「標準SLM対応 ROMSW 設定ｿﾌﾄﾏﾆｭｱﾙ[TB00-0801]」＜4-4.軸設定ﾊﾟﾗﾒﾀ＞を参照下さい。
以下の場合、コマンドの実行に失敗します。
・[無限]回転軸以外の軸に対する回転動作要求
・動作中の軸に対する回転動作要求
・セッティングモード、ＯＴ無視モードの時

尚、回転動作中の軸に対して、他の移動命令（位置決め等）を行うことは出来ません。

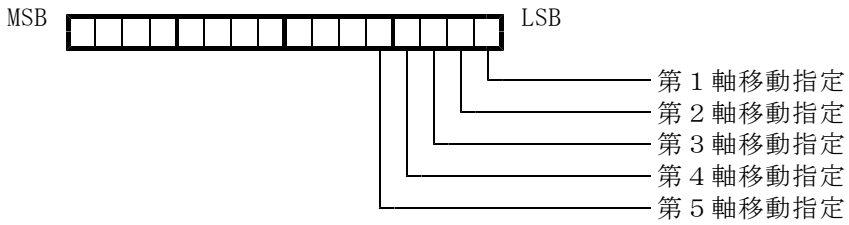
《コマンド実行条件》
・サーボＯＮ状態
・アラーム未発生
・動作モードが手動運転モード／自動運転モード／ＤＮＣ運転モード
・指定した軸が移動中の軸でない
・指定した軸が無限回転軸

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値	
００００	オーバライドフラグ	２バイト	０：無効、１：有効	
０００２	移動軸選択フラグ	２バイト	－	
０００４	第１軸回転数	４バイト	0～[MAXpps＊600/1回転ﾊﾟﾙｽ数]	※１
０００８	第２軸回転数	４バイト	0～[MAXpps＊600/1回転ﾊﾟﾙｽ数]	
０００Ｃ	第３軸回転数	４バイト	0～[MAXpps＊600/1回転ﾊﾟﾙｽ数]	
００１０	第４軸回転数	４バイト	0～[MAXpps＊600/1回転ﾊﾟﾙｽ数]	
００１４	第５軸回転数	４バイト	0～[MAXpps＊600/1回転ﾊﾟﾙｽ数]	

（１）オーバーライド有効フラグ
オーバーライドの変更に伴い、回転速度を変更するかどうかを指定します。

（２）移動軸選択フラグ
移動コマンドにて、移動を行わせる軸を指定します。
各軸に対応するビットに１を設定するとその軸が移動を行います。



（３）各軸回転数
０．１ｒｐｍ単位で指定します。
尚、回転数に０を指定した場合、回転動作を終了します。

※１　"MAXpps"はROMSW設定の「パルスジェネレータクロック」設定により、変わります。
詳細は「標準SLM対応 ROMSW 設定ｿﾌﾄﾏﾆｭｱﾙ[TB00-0801]」＜4-1.基本ﾊﾟﾗﾒｰﾀ＞を参照下さい。

《構造体定義》

```
/************************************************************************/
/*  回転軸回転動作コマンドパラメータ構造体                              */
/************************************************************************/
typedef struct {
    short           OverFlag;          /* オーバーライドフラグ          */
    unsigned short  AxisFlag;          /* 移動軸選択フラグ              */
    long            RevAx[5];          /* 第１軸～第５軸回転数          */
} SPINAX;
```

3-2-41.ＴＰＣデータ選択コマンド ［REQ_TPCSEL(0x31)］

《データタイプ》
ＲＥＱ＿ＴＰＣＳＥＬ（０ｘ３１）

《説明》
ＴＰＣデータのロギングを実行する軸と入出力の設定をします。

《コマンド実行条件》
・ＴＰＣロギング未実行

《関連項目》
3-1-25.ＴＰＣロギング情報読出
3-1-24.ＴＰＣロギングデータ読出
3-2-51.ＴＰＣロギング開始コマンド

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	第１ポジション番号	２バイト	０～１１
０００２	第１入力信号番号	２バイト	０～８
０００４	第１出力信号番号	２バイト	０～８
０００６	第２ポジション番号	２バイト	０～１１
０００８	第２入力信号番号	２バイト	０～８
０００Ａ	第２出力信号番号	２バイト	０～８

（１）各設定内容は下図の通りです。
（２）コマンド発行前のデフォルト設定は、表 □、 ■部が
それぞれ第１データ、第２データとなっています。
（３）ポジションデータとしては通常、機械位置を使用します。

ポジション番号 [ax1,ax2]		入力信号番号 [hi1,hi2]		出力信号番号 [ho1,ho2]	
設定値	指定内容	設定値	指定内容	設定値	指定内容
0	第１軸指令位置	0	＃０００	0	＃０００
1	第２軸指令位置	1	＃００１	1	＃００１
2	第３軸指令位置	2	＃００２	－	－
3	第４軸指令位置	－	－	－	－
4	第５軸指令位置	－	－	－	－
5	第１軸機械位置	－	－	－	－
6	第２軸機械位置	－	－	－	－
7	第３軸機械位置	－	－	－	－
8	第４軸機械位置	－	－	－	－
9	第５軸機械位置	－	－	－	－

《構造体定義》

```
/*************************************************************************/
/*  ＴＰＣデータ選択コマンドパラメータ構造体                               */
/*************************************************************************/
typedef struct {
    short          ax1;              /* 第１ポジション番号      */
    short          hi1;              /* 第１入力信号番号        */
    short          ho1;              /* 第１出力信号番号        */
    short          ax2;              /* 第２ポジション番号      */
    short          hi2;              /* 第２入力信号番号        */
    short          ho2;              /* 第２出力信号番号        */
} TPCSEL;
```

3-2-42.論理原点シフトコマンド [REQ_ZRNSHIFT(0x30)]

《データタイプ》
ＲＥＱ＿ＺＲＮＳＨＩＦＴ（０ｘ３０）

《説明》
ティーチングで移動した量だけ、論理原点をシフトします。

《コマンド実行条件》
・ティーチング中

《データフォーマット》
コマンド付加データは有りません。

3-2-43.ホームポジション位置決めコマンド [REQ_HOME(0x83)]

《データタイプ》
ＲＥＱ＿ＨＯＭＥ（０ｘ８３）

《説明》
設定されたホーム位置へ移動します。ホーム位置決め移動時には、原点復帰同様ＪＯＧ指令、インチング指令、手動パルサーの指令原点復帰は行えません。
また、移動中にソフトリミット／ハードリミットエラーが発生した場合は即停止します。

《コマンド実行条件》
・サーボＯＮ状態
・アラーム未発生
・動作モードが手動運転モード／自動運転モード／ＤＮＣ運転モード
・動作プログラム未実行 又は ティーチング中
・移動中の軸がない
・手動パルサー選択無効

《データフォーマット》
コマンド付加データは有りません。

3-2-44.座標系設定コマンド [REQ_COORDSET(0x37)]

《データタイプ》
ＲＥＱ＿ＣＯＯＲＤＳＥＴ（０ｘ３７）

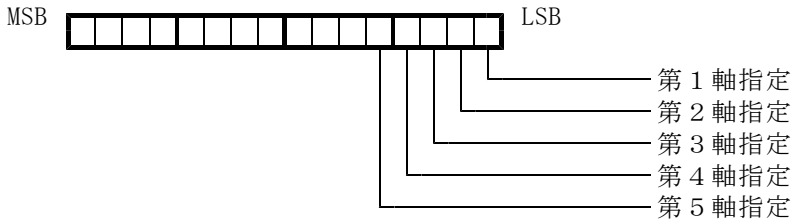
《説明》
各軸の現在位置を指定の論理座標値に設定します。

《コマンド実行条件》
・サーボＯＮ状態
・動作モードが手動運転モード／自動運転モード／ＤＮＣ運転モード
・動作プログラム未実行 又は ティーチング中
・移動中の軸がない
・手動パルサー選択無効

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	移動軸選択フラグ	４バイト	－
０００４	第１軸論理座標値	４バイト	-99999999～99999999
０００８	第２軸論理座標値	４バイト	-99999999～99999999
０００Ｃ	第３軸論理座標値	４バイト	-99999999～99999999
００１０	第４軸論理座標値	４バイト	-99999999～99999999
００１４	第５軸論理座標値	４バイト	-99999999～99999999

（１）設定軸選択フラグ
論理座標値を設定する軸を指定します。
設定を行う軸に対応するビットに１を設定します。



《構造体定義》

```
/*****************************************************************************/
/*  座標系設定コマンドパラメータ構造体                                       */
/*****************************************************************************/
typedef struct {
    unsigned long   AxisFlag;           /* 移動軸選択フラグ              */
    long            PosAxis[5];         /* 第1軸～第5軸位置決めﾎﾟｼﾞｼｮﾝ  */
} COORDSET;
```

3-2-45.軸インタロック設定コマンド [REQ_AXISINTLK(0x38)]

《データタイプ》
ＲＥＱ＿ＡＸＩＳＩＮＴＬＫ（０ｘ３８）

《説明》
各軸のインタロックの有効無効を設定します。
インタロック有効軸に対して指令を行うと、エラーになります。

《コマンド実行条件》
・移動中の軸がない
・ＡＣＣたまりなし

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	第１軸ｲﾝﾀﾛｯｸ指定	２バイト	0:無変更,1:無効,2:有効
０００２	第２軸ｲﾝﾀﾛｯｸ指定	２バイト	0:無変更,1:無効,2:有効
０００４	第３軸ｲﾝﾀﾛｯｸ指定	２バイト	0:無変更,1:無効,2:有効
０００６	第４軸ｲﾝﾀﾛｯｸ指定	２バイト	0:無変更,1:無効,2:有効
０００８	第５軸ｲﾝﾀﾛｯｸ指定	２バイト	0:無変更,1:無効,2:有効

《構造体定義》
```
/*****************************************************************************/
/*  軸インタロック設定コマンドパラメータ構造体                                 */
/*****************************************************************************/
typedef struct {
    unsigned short  IntlkSw[5];         /* 軸インタロック指定スイッチ   */
} AXINTLK;
```

3-2-46.軸ネグレクト設定コマンド [REQ_AXISNEG(0x39)]

《データタイプ》
ＲＥＱ＿ＡＸＩＳＮＥＧ（０ｘ３９）

《説明》
各軸の軸指令の有効無効を設定します。
軸指令を無効にした軸に対する指令は無視します。

《関連項目》
3-1-26.軸ネグレクト設定データ読出

《コマンド実行条件》
・移動中の軸がない
・ＡＣＣたまりなし

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	第１軸ﾈｸﾞﾚｸﾄ指定	２バイト	0:無変更,1:無効,2:有効
０００２	第２軸ﾈｸﾞﾚｸﾄ指定	２バイト	0:無変更,1:無効,2:有効
０００４	第３軸ﾈｸﾞﾚｸﾄ指定	２バイト	0:無変更,1:無効,2:有効
０００６	第４軸ﾈｸﾞﾚｸﾄ指定	２バイト	0:無変更,1:無効,2:有効
０００８	第５軸ﾈｸﾞﾚｸﾄ指定	２バイト	0:無変更,1:無効,2:有効

《構造体定義》

```
/*****************************************************************************/
/*  軸ネグレクト設定コマンドパラメータ構造体                                 */
/*****************************************************************************/
typedef struct {
    unsigned short  NeglectSw[5];       /* 軸ネグレクト指定スイッチ     */
} AXNGLCT;
```

3-2-47.各軸サーボON/OFFコマンド [REQ_SVONOFF(0x3a)]

《データタイプ》
ＲＥＱ＿ＳＶＯＮＯＦＦ（０ｘ３ａ）

《説明》
各軸のサーボＯＮ／ＯＦＦを設定します。
本コマンドはサーボＯＮ状態の時に強制的にサーボＯＦＦするためのコマンドです。
エラーが発生しているときなど、サーボＯＮできない状態の時にはサーボＯＮ出来ません。

《コマンド実行条件》
・移動中の軸がない
・ＡＣＣたまりなし

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	第１軸ﾓｰﾄﾞ設定	２バイト	0:無変更,1:ｻｰﾎﾞON,2:ｻｰﾎﾞOFF
０００２	第２軸ﾓｰﾄﾞ設定	２バイト	0:無変更,1:ｻｰﾎﾞON,2:ｻｰﾎﾞOFF
０００４	第３軸ﾓｰﾄﾞ設定	２バイト	0:無変更,1:ｻｰﾎﾞON,2:ｻｰﾎﾞOFF
０００６	第４軸ﾓｰﾄﾞ設定	２バイト	0:無変更,1:ｻｰﾎﾞON,2:ｻｰﾎﾞOFF
０００８	第５軸ﾓｰﾄﾞ設定	２バイト	0:無変更,1:ｻｰﾎﾞON,2:ｻｰﾎﾞOFF

《構造体定義》

```
/*************************************************************************/
/*  各軸サーボＯＮ／ＯＦＦコマンドパラメータ構造体                       */
/*************************************************************************/
typedef struct {
    unsigned short  SvOnSw[5];          /* ｻｰﾎﾞON/OFF指定スイッチ       */
} SVONOFFCHG;
```

◆1.2

3-2-48.手動パルサーモードON/OFF設定コマンド［REQ_HANDLE(0x5c)］ 〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＲＥＱ＿ＨＡＮＤＬＥ（０ｘ５ｃ）

《説明》
手動パルサーの有効/無効を指定します。

《コマンド実行条件》
・機械パネルの手パ軸選択無効
・サーボＯＮ状態
・動作モードが手動運転モード／自動運転モード／ＤＮＣ運転モード／ＯＴ無視モード
・動作プログラム未実行 又は ティーチング中
・移動中の軸がない

《関連項目》
3-1-29.手動パルサー設定情報読出
3-2-49.手動パルサー倍率設定コマンド
3-2-50.手動パルサー有効軸設定コマンド

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
０００２	手動パルサー有効/無効設定	２バイト	０：無効、１：有効、２：ｼﾞｮｲｽﾃｨｯｸ

《構造体定義》

```
/*********************************************************************/
/*  手パモード構造体                                                 */
/*********************************************************************/
typedef struct {
    short           mode;                       /* 手パモード       */
} HANDLEMODE;
```

3-2-49.手動パルサー倍率設定コマンド [REQ_HANDLEKP(0x5d)] 〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＲＥＱ＿ＨＡＮＤＬＥＫＰ（０ｘ５ｄ）

《説明》
手動パルサーの倍率（手動パルサー１パルス毎に動く量）を指定します。
1,10,100,1000の内から任意の倍率を指定します。

《コマンド実行条件》
・機械パネルの手パ軸選択無効

《関連項目》
3-1-29.手動パルサー設定情報読出
3-2-48.手動パルサーモードON/OFF設定コマンド
3-2-50.手動パルサー有効軸設定コマンド

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	手動パルサー設定倍率	４バイト	1,10,100,1000

《構造体定義》

```
/*************************************************************************/
/*  手パ倍率パラメータ構造体                                             */
/*************************************************************************/
typedef struct {
    long            kp;                         /* 手パ倍率             */
} HANDLEKP;
```

3-2-50.手動パルサー有効軸設定コマンド
[REQ_HANDLEAXIS1(0x5e),REQ_HANDLEAXIS2(0x5f)] 〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＲＥＱ＿ＨＡＮＤＬＥＡＸＩＳ１（０ｘ５ｅ）
ＲＥＱ＿ＨＡＮＤＬＥＡＸＩＳ２（０ｘ５ｆ）

《説明》
手動パルサーにて、移動を行わせる第１軸、第２軸を指定します。

《コマンド実行条件》
・機械パネルの手パ軸選択無効

《関連項目》
3-1-29.手動パルサー設定情報読出
3-2-48.手動パルサーモードON/OFF設定コマンド
3-2-49.手動パルサー倍率設定コマンド

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
０００００	手動パルサー有効軸	４バイト	０～軸数－1

《構造体定義》
```
/************************************************************************/
/*  手パ軸パラメータ構造体                                              */
/************************************************************************/
typedef struct {
    long            axis;                   /* 手パ有効軸       */
} HANDLEAXIS;
```

3-2-51.ＴＰＣロギング開始コマンド [REQ_TPCLOG(0x32)]

《データタイプ》
ＲＥＱ＿ＴＰＣＬＯＧ（０ｘ３２）

《説明》
ＴＰＣロギングの開始／停止／再開を指定します。

《コマンド実行条件》
・ＴＰＣロギング未実行 又は 停止中

《関連項目》
3-1-24.ＴＰＣロギングデータ読出
3-1-25.ＴＰＣロギング情報読出
3-2-41.ＴＰＣデータ選択コマンド

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	ＴＰＣロギングフラグ	２バイト	０：停止、１：開始、２：再開

《構造体定義》
構造体定義はありません。
２バイト（short）のデータとして定義して下さい。

3-2-52.Mコード出力コマンド [REQ_MCDOUT(0x3d)]

《データタイプ》
ＲＥＱ＿ＭＣＤＯＵＴ（０ｘ３ｄ）

《説明》
指定したMコードを出力するように指令します。
本コマンドでは、Ｍ０３（主軸正転）のような内部Mコード処理は行いません。

《コマンド実行条件》
・サーボＯＮ状態
・動作モードが手動運転モード／自動運転モード／ＤＮＣ運転モード
・動作プログラム未実行 又は ティーチング中
・移動中の軸がない
・手動パルサー選択無効

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	出力Mコード番号	２バイト	０～２５５

《構造体定義》
```
/*****************************************************************************/
/*  Mコード出力コマンドパラメータ構造体                                      */
/*****************************************************************************/
typedef struct {
    unsigned short  mcd;                /* 出力Mコード                   */
} OUTMCD;
```

3-2-53.サイクル運転モード設定コマンド [REQ_CYCLE(0x36)]

《データタイプ》
ＲＥＱ＿ＣＹＣＬＥ（０ｘ３６）

《説明》
サイクル運転モードの有効/無効を選択します。
サイクル運転については「SLM-4000 ユーザーズマニュアル[TB00-0800]」＜Ⅲ 機能編 ４－１２．サイクル運転＞を参照下さい。

《コマンド実行条件》
常時実行可能です。

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	ｻｲｸﾙ運転有効ﾌﾗｸﾞ	２バイト	０：無効、１：有効

《構造体定義》
```
/*****************************************************************************/
/*  サイクル運転モード変更コマンドパラメータ構造体                           */
/*****************************************************************************/
typedef struct {
    short           cycle;              /* サイクル運転有効フラグ        */
} CYCLECHG;
```

3-2-54.主軸回転数設定コマンド［REQ_SPREVSET(0x51)］〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＲＥＱ＿ＳＰＲＥＶＳＥＴ（０ｘ５１）

《説明》
主軸の回転数を指定します。
主軸回転中に本コマンドにて主軸回転を変更した場合は、直ちに指定された回転数に変わります。

《コマンド実行条件》
常時実行可能です。

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	主軸回転数	４バイト	0～最高主軸回転数［３］

（１） 主軸回転数
主軸回転数を指定します。

《構造体定義》
```
/*****************************************************************************/
/*  主軸回転数変更コマンドパラメータ構造体                                   */
/*****************************************************************************/
typedef struct {
    unsigned long   SpRevo;             /* 主軸回転数           */
} SPREVSET;
```

3-2-55.マクロ変数書込コマンド ［REQ_MCRREG(0x84)］〈ｵﾌﾟｼｮﾝ〉

《データタイプ》
ＲＥＱ＿ＭＣＲＲＥＧ（０ｘ８４）

《関連項目》
3-1-23.マクロ変数（一般レジスタ）読出
3-1-31.マクロ変数読出

《説明》
任意の番号のマクロ変数にデータを設定します。
書き込み可能な全てのマクロ変数に対して書き込む事が出来ます。
（ただし、指定したマクロ変数の範囲外のデータを書き込む事はできません。）
マクロ変数詳細については、「SLM-4000 ユーザーズマニュアル[TB00-0800]」＜Ⅲ 機能編 ６－４．マクロ機能＞を参照下さい。

《コマンド実行条件》
常時実行可能です。

《データフォーマット》

ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	マクロ変数番号	４バイト	1000 ～ 9999［－］
０００４	マクロ変数値	４バイト	-2147483647～2147483647［－］

《構造体定義》
```
/*****************************************************************************/
/*  マクロ変数書込コマンドパラメータ構造体                                   */
/*****************************************************************************/
typedef struct {
    long            RegNo;                  /* マクロ変数番号       */
    long            Val;                    /* マクロ変数値         */
} MCRVALSET;
```

《データ詳細》
（１） マクロ変数番号
データを書き込む対象のマクロ変数の番号を指定します。

（２） マクロ変数値
指定したマクロ変数に書き込む値を指定します。

3-2-56.フィードバックカウンタ積算値セットアップコマンド [REQ_FBSETUP(0x78)]

《データタイプ》
　ＲＥＱ＿ＦＢＳＥＴＵＰ（０ｘ７８）

《関連項目》
　3-1-32.フィードバックカウンタ積算値読出

《説明》
　ＦＢカウンタ積算値を変更します。
　ＦＢを軸に割り当てていた場合でも、本コマンドでＦＢカウンタ積算値を変更する事ができます。
　（ＦＢを割り当てている軸の座標には影響しません。）

《コマンド実行条件》
　常時実行可能です。

《データフォーマット》

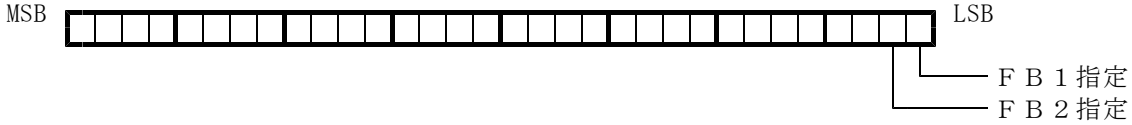
ｵﾌｾｯﾄ	データ名称	データ長	設定値
００００	ＦＢ１カウンタ設定値	４バイト	-2147483647～2147483647[pls]
０００４	ＦＢ２カウンタ設定値	４バイトD	-2147483647～2147483647[pls]
０００８	設定選択フラグ	２バイト	0x01 ～ 0x03 [－]

○Ｃ言語構造体

```
/*************************************************************************/
/*  フィードバックカウンタクリアコマンドパラメータ構造体                 */
/*************************************************************************/
typedef struct {
    long            cntr[2];                    /* FB1･FB2設定値        */
    short           set;                        /* FB1･FB2設定フラグ    */
} FBSETUP;
```

《データ詳細》
（１）　ＦＢ１／２カウンタ設定値
　　設定する値を指定します。

（２）　設定選択フラグ
　　設定を行うカウンタを指定します。
　　各カウンタに対応するビットに１を設定するとそのカウンタの値を変更します。



４．改版版履歴

4-1.　[Ver1.1→Ver1.2]　2004.08.12

項番	新ﾍﾟｰｼﾞ	内容
3-2-40	0804-65	無限回転軸説明を追加
3-2-48	0804-70	設定値にｼﾞｮｲｽﾃｨｯｸを追加

4-2.　[Ver1.2→Ver1.3]　2006.07.22

項番	新ﾍﾟｰｼﾞ	内容
3-1	―	全てのデータ送受信機能に《パラメータ》《書き込み実行条件》《読み込み実行条件》を追加（実行条件は必要に応じて）
3-2	―	全ての動作要求コマンドに《コマンド実行条件を追加》
3-1-32	0804-41	フィードバックカウンタ積算値読出を追加
3-2-56	0804-87	フィードバックカウンタ積算値セットアップコマンドを追加

4-3.　[Ver1.3→Ver1.4]　2008.06.26

項番	新ﾍﾟｰｼﾞ	内容
3-1-33	0804-43	工具径補正データ書込／読出コマンドを追加
3-1-34	0804-44	工具長補正設定情報読出コマンドを追加
3-1-35	0804-45	工具径補正設定情報読出コマンドを追加

4-4.　[Ver1.4→Ver1.5]　2009.05.22

項番	新ﾍﾟｰｼﾞ	内容
3-1-36	0804-46	工具径補正エラー情報読出コマンドを追加

4-5.　[Ver1.4→Ver1.5]　2009.05.22

項番	新ﾍﾟｰｼﾞ	内容
3-1-37	0804-47	補間前加減速パラメータ書込/読出を追加