

B3M Series SoftwareManual  
Command Reference

Kondo Kagaku Co., Ltd.

Ver. 1.2.0.2

注意事項

このマニュアルはB3Mシリーズと通信を行うコマンドのリファレンスマニュアルです。

内容についての著作権など法的な権利は、近藤科学株式会社にあります。  
このコマンドをご使用になった結果については責任は負いません。

誤字脱字などについては弊社窓口までお申し出ください。ただし、内容についてのご質問  
及びプログラミングについてのご質問については、お答え出来ない場合がございますので  
あらかじめご了承ください。

なお本マニュアルの内容、各種名称については予告なく変更される場合があります。

## 目次

### 1.B3Mサーボについて

- 製品説明
- 製品特徴と概要
- 製品構成
- サーボの種類とスペック
- LEDの機能
- コネクタとピン配置
- システム構成
- インターフェース
- 電源
- マルチドロップ接続について

### 2.通信仕様について

- シリアル通信設定
- 通信モード
- データ構造
- データフォーマットの名称説明

### 3.コマンド詳細

- コマンド一覧
- データの統合と分割について
- LOADコマンド
- SAVEコマンド
- READコマンド
- WRITEコマンド
- RESETコマンド
- SET POSITIONコマンド

### 4.オプション/ステータス詳細

- オプション
- ステータス

### 5.サーボの動作について

- サーボの状態について
- 制御モードについて
- 位置制御モードとは
- 速度制御モードとは
- 電流（トルク）制御モードとは
- フォワードモードとは

## 6.B3Mサーボを動かしてみる

B3Mサーボを動かすには

## 7.メモリーマップについて

単位記号について  
属性表記について  
型表記について  
メモリー領域について

## 8.メモリーマップ/各機能詳細

SYSTEM領域  
サーボパラメータ領域  
制御パラメータ領域  
STATUSパラメータ領域  
バージョン情報領域  
システム初期値領域

## 9.B3MシリーズPC用ライブラリーについて

使用環境  
メモリーマップ

## 10.列挙型変数一覧(bitwiseオプション)

## 11.変更履歴

## 1.B3Mサーボについて

### ■ 製品説明

B3Mはブラシレスモータや非接触磁気式エンコーダを使用し、高信頼性、長寿命を目指したサーボです。ケースのトップとミドルにアルミを採用、さらにアルミギヤに硬質アルマイト処理を施した強度の高いギヤを採用することで、高い耐久性を実現しました。

制御プログラムを一新し、高機能で使いやすいシステムを実装しました。PID制御の設定や温度制限や動作電圧制限などの各種リミッターの搭載しました。それらをシンプルな6種類のコマンドだけで制御可能です。制御は、位置制御、速度制御から選択できます。位置制御では、片側320°の広範囲にわたって角度を指定することができます。また、制御方式も4-1-4多項式補完法など5種類の軌道生成タイプから選択できます。

### ■ 製品特徴と概要

#### 【高精度12bit磁気式エンコーダーを採用】

非接触磁気式エンコーダーは接触部分がないので長寿命。  
さらに12bitの解像度で最小分解能は0.088°（0～4095）

#### 【信頼のRS-485規格】

通信はRS-485規格に準拠しています。差動信号のためノイズに強く、3Mbpsの高速通信が可能。デジチェーン接続で最大255個接続可能です（理論値）。

#### 【6種類の省コマンドでシンプルな構成】

僅か6種類のコマンドのみでサーボの制御が可能です。  
メモリーマップドIOを採用しているので、メモリーのデータを読み書きするだけで実際にサーボモーターが動作します。

#### 【マルチキャストコマンド】

サーボごとにID番号を設定できます。また、ID=255はブロードキャストIDとして使用。デジチェーン接続された全てのサーボモーターを同時にコマンド実行させることが可能です。

#### 【位置ゲイン、微分ゲイン、積分ゲイン】

位置制御・速度制御時はPID制御パラメーターを調整可能です。また、ゲインプリセット機能を使えば3種類のゲインをあらかじめ登録しておくことができますので、プリセット番号ですばやくゲインを変更することができます。

#### 【システムエラー検知機能・ソフトウェアリミット】

入力電圧やMCU温度、モーター温度、モーターロックなど各種システムエラーをメモリー上に保存しています。また、安全のためのリミット機能を搭載しています。モーターやMCUの温度に対して制限をかけることで、故障の原因を減らすことが可能です。

## ■ 製品構成

サーボ本体には、アルミクランプホーンが標準装備されています。また、接続ケーブルもサーボの台数分付属します。

## 【サーボ本体】



## 【付属品】

## &lt;単品&gt;

XH接続ケーブル（400mm） ×1本

アルミクランプホーン ×1個

## &lt;5個セット&gt;

XH接続ケーブル（400mm） ×5本

アルミクランプホーン ×5個



XH接続ケーブル（400mm）



アルミクランプホーン

## ■ 用意するもの

サーボの制御に必要なパーツをご紹介します。

## ・ RS485USB/シリアル変換アダプター

No.02133 ¥6,000

サーボと通信するための中継用アダプターです。

## ・ HV電源スイッチハーネス

No.01213 ¥800

上記のアダプターと電源を接続するためのスイッチ付きケーブルです。

## ・ 電源

B3Mサーボに合った電源として9～12Vのバッテリー、またはAC電源などを準備してください。

・ SB-1040-A・・・サーボ1個当たり**最大**2.8Aの電流容量が必要です。

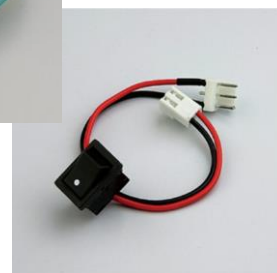
・ SC-1040-A・・・サーボ1個当たり**最大**3.6Aの電流容量が必要です。

・ SC-1170-A・・・サーボ1個当たり**最大**5.4Aの電流容量が必要です。

## ・ USBケーブル

PCからサーボにコマンドを送る場合は、こちらの準備もお願いします。

コネクタは、USB-A ⇔ mini-B です。



## ■オプション製品



**XHコネクター用ハブ typeA**

No.02136      ¥2,500

複数のサーボをつなぐ際に使用します。



**ベアリング付アルミフリーホーン  
(B3M/KRS-6000シリーズ用)**

No.02137      ¥1,200

ボトムケース側につけるホーンです。



**サーボ用4本ケーブルセット**

No.02135      ¥1,000

ケーブルセット、コネクターセットを組み合わせることで自由な長さのケーブルを作成することができます。



**XHケーブル用コネクターセット**

No.02134      ¥1,500



**サーボアーム**

No.02150      ¥3,000



**サーボブラケットS**

No.02151      ¥3,000



**サーボブラケットB**

No.02152      ¥3,000



組み合わせ例

サーボアームとブラケットを組み合わせることで、複数のパターンの組み込み方法に対応できます。

※これらのフレームはB3M-1000シリーズ用です。

## ■ サーボの種類とスペック

B3Mシリーズは、大きく二種類のサーボに分かれます。ブラシレスモータを採用したB3M-SBと、コアレスモータのB3M-SCです。ブラシレスモータは、損耗部品のブラシが無いいため長くご使用いただけます。また、コアレスモータは、ハイスpekででありながらSBシリーズに比べ低価格に設定されており、採用しやすい価格を実現しています。

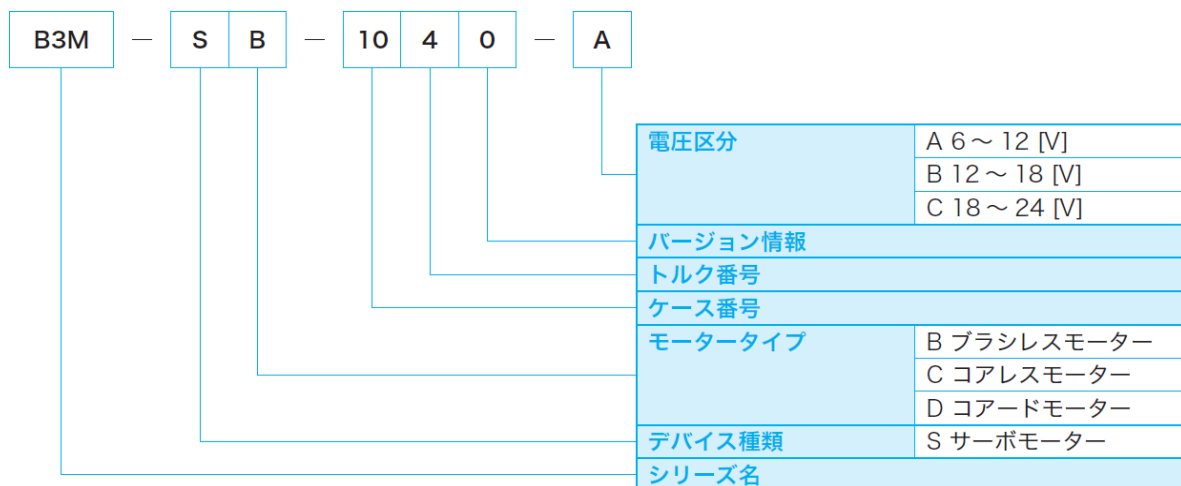
項目名	SB-1040-A	SC-1040-A	SC-1170-A
動作環境温度[℃]	0～40		
動作電圧[V]	6～12		
待機電流※1(フリーモード時) [mA]	80	68	68
ストール電流[A]	2.8	3.6	5.4
最大トルク※1 [Nm]	4.1	4.6	7.6
[kgf・cm]	42	47	78
最大連続トルク※1[Nm]	2.5	2.5	4.0
無負荷回転数※1 [s/60°]	0.22	0.18	0.21
[rpm]	45	54	46
減速比	362.88:1	381.2:1	362.88:1
動作環境温度[℃]	0～40		
モーター種類	ブラシレスモータ      コアレスモータ		
重量[g]	86	82	105
最大動作角度※2 位置制御	±320° (±327.67°)		
その他制御	無限回転※3		
制御分解能	12 [bit]/1round (0～4095)		
動作方式	位置制御、速度制御、トルク制御		
通信方式	RS485通信準拠、半二重通信		
主なパラメータ	ID番号、各種制限項目、PID制御各種ゲイン		
固有ID数	0～ 254 (255はブロードキャストIDとして予約)		

※1 12Vで使用した場合のスペック値です。

※2 原点位置からの動作角度(原点位置は任意に設定できます)。

※3 角度を取得する場合は原点位置±180degになります。

## B3Mシリーズの見方



■ LEDの機能

サーボ本体に実装されているLEDの点滅パターンでサーボの状態が確認できます。



LED

状態	点滅パターン
正常	点灯
システムエラー	100ms周期の点滅
モータエラー	250ms周期の点滅
UARTエラー	500ms周期の点滅
コマンドエラー	1秒周期の点滅
電源OFF	消灯

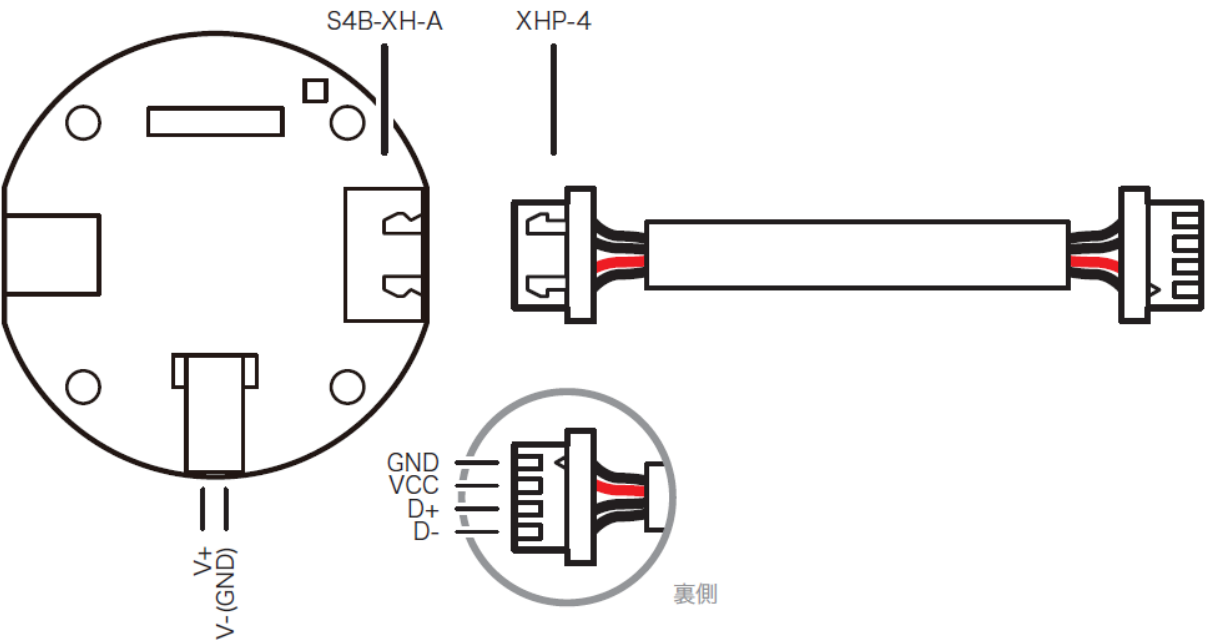
※エラーの解除はOPTIONのCLEARビットに1をセットしてください。

※Normalモード時はLED点滅の優先度が低いいため、規則正しく点滅しません。

FreeモードまたはHoldモードでエラーステータスを確認してください。

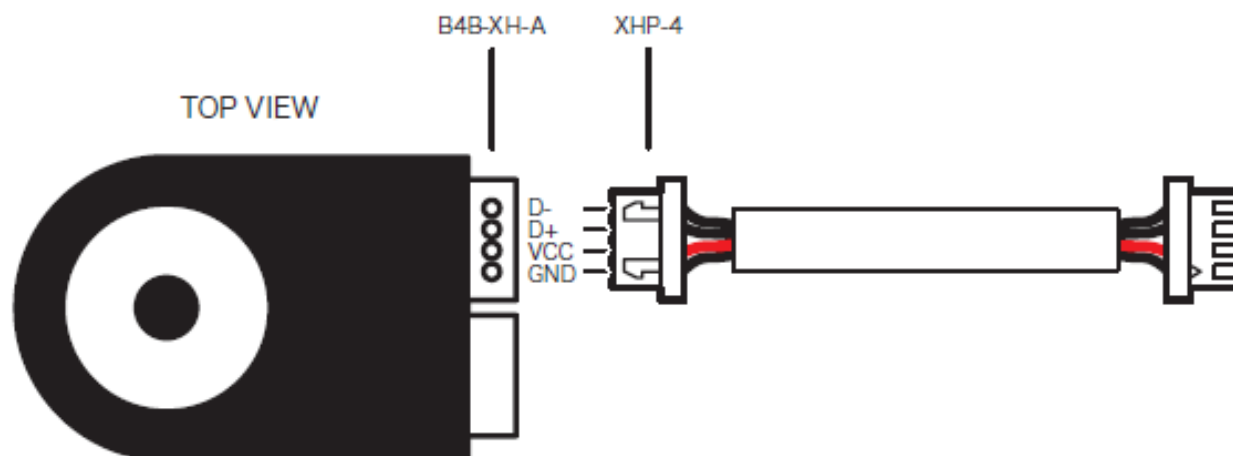
■ コネクタとピン配置

■ S4B-XH-A側接続図





## ■ サーボ側接続図



## ■ ケーブル品名・コード

品名: XH接続ケーブル (400mm)

コード : 02144

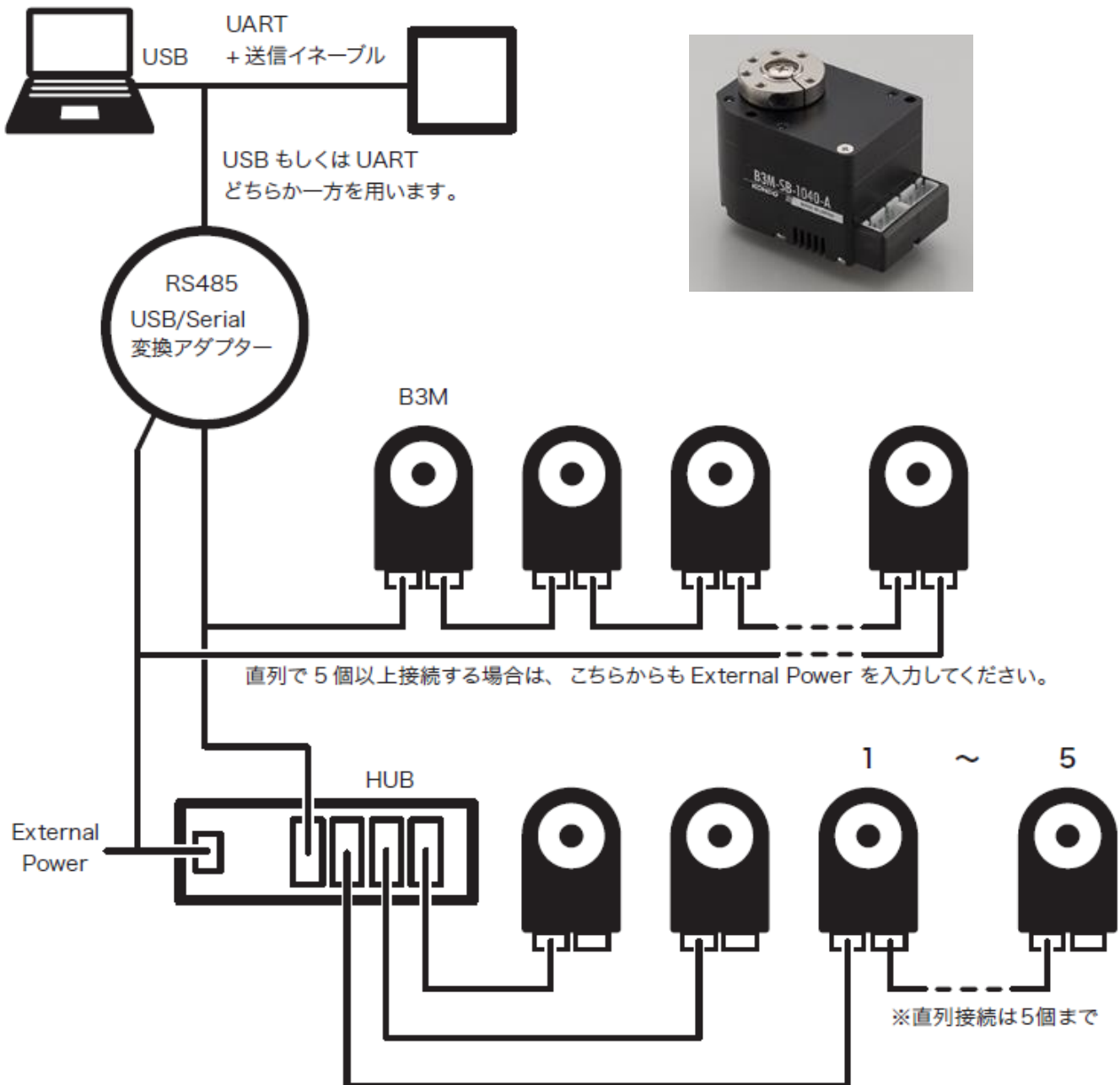
## ■ コネクタセット品名・コード

品名: XHケーブル用コネクターセット (30セット入り)

コード : 02134

## ■システム構成

B3Mは以下のシステム構成で制御することが可能です。コマンドを送るホストは、PCのUSB接続またはマイコンボードのUARTを使用します。RS485USB/シリアル変換アダプターにて中継し、サーボをマルチドロップ接続にて複数個つなぐことが可能です。電源は、USBアダプターから供給することができますし、ハブを挟んでの供給も可能です。



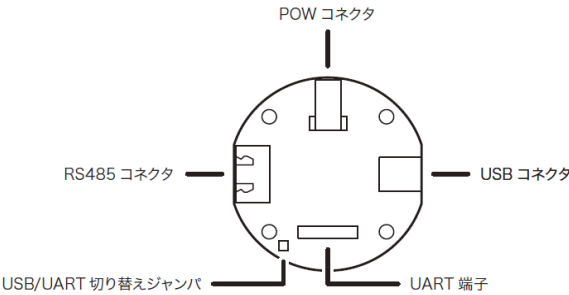
※多数接続すると、末端のサーボは電力不足になります。

1 列に対し 5 個直列接続までとし、ハブを用い分散させ接続してください。

■ インターフェース

【RS485USB/シリアル変換アダプター】

このアダプターにて、B3MサーボのRS-485規格に通信を合わせることができます。  
PCから直接動作する場合とマイコンのUART端子から制御する場合のどちらにも対応できます。  
サーボへの電源供給用の端子も装備しています。



【機械特性】

項目名	内 容
使用温度	0～40 [℃]

【電気・通信特性】

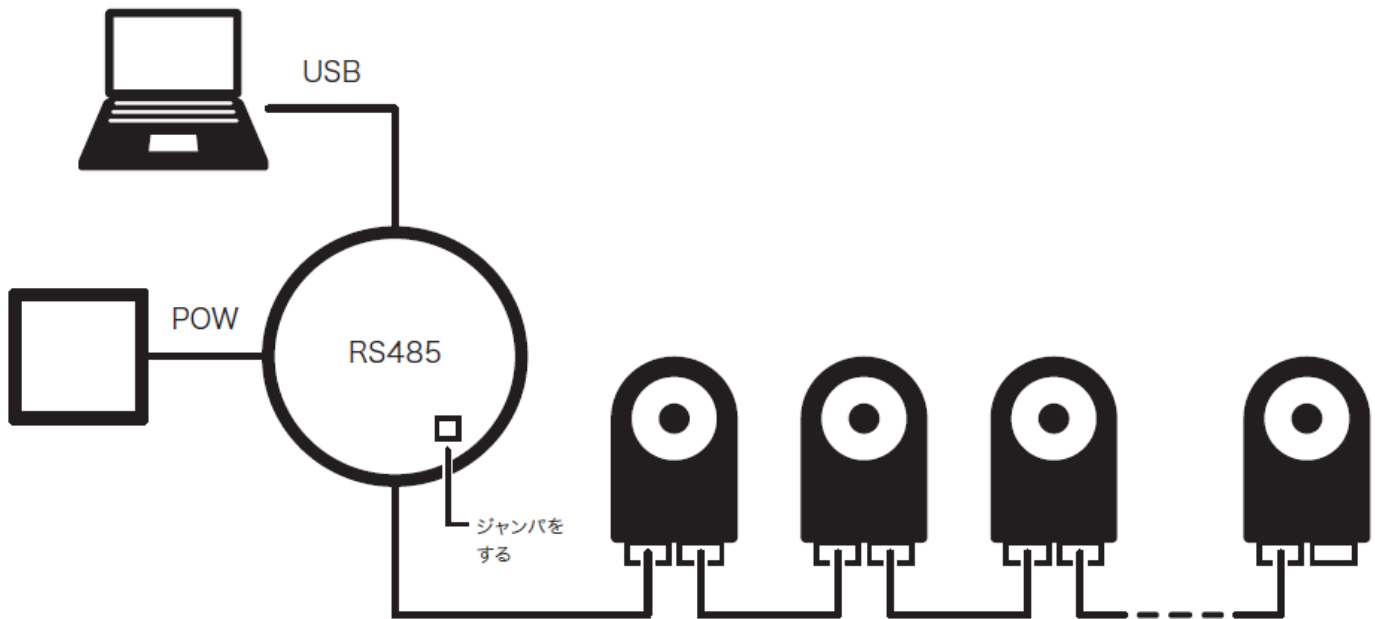
項目名	内 容
接続機器印加電圧	接続機器による
USB	USB2.0 および USB1.1
RS485	
転送モード	半二重非同期通信
最大通信速度	3 [Mbps] (USB使用時) 16 [Mbps] (UART使用時)
バス上ノード数	最大256まで
RS485ライン (A,B)	-8～13 [V]
UART接続ピン	
Vcc電圧	+3.0 ～+6.0 [V]
R出力電圧	H: Vcc -0.6 [V] L: 0.4 [V]
ENおよびD入力電圧	0～+6.0 [V]
ロジックレベル	H: 2.0 [V]以上 L: 0.8 [V]以下

## 【コネクタ仕様】

項目名	内 容
USB コネクタ	USBミニB
POWコネクタ	日本圧着端子製造株式会社 B2PS-VH
	1 V+ (機器用電源)
	2 V- (GND)
RS485コネクタ	日本圧着端子製造株式会社 S4B-XH-A
	1 GND
	2 電源出力
	3 A (D+ RS485 Line)
	4 B (D- RS485 Line)
UART接続ピン	(2.54 [mm]ピッチで配置)
	EN 送信、受信切り替え
	R データ受信ピン
	D データ送信ピン
	GND GND
	Vcc ロジックIC用電源 (3.0 ~6.0 [V])

## 【使用方法】

## ■ USB通信の場合



1. JP1を半田でジャンパします（出荷時は接続されています）。
2. PCに接続し、KOドライバをインストールします。ドライバのインストール後は作成された仮想COMポートと通信できます（ドライバのインストール方法は、KO Driverのインストールマニュアルを参照してください）。

【KONDO website】カスタマーセンター→サポート情報→ソフトウェア→「KO Driver2015」

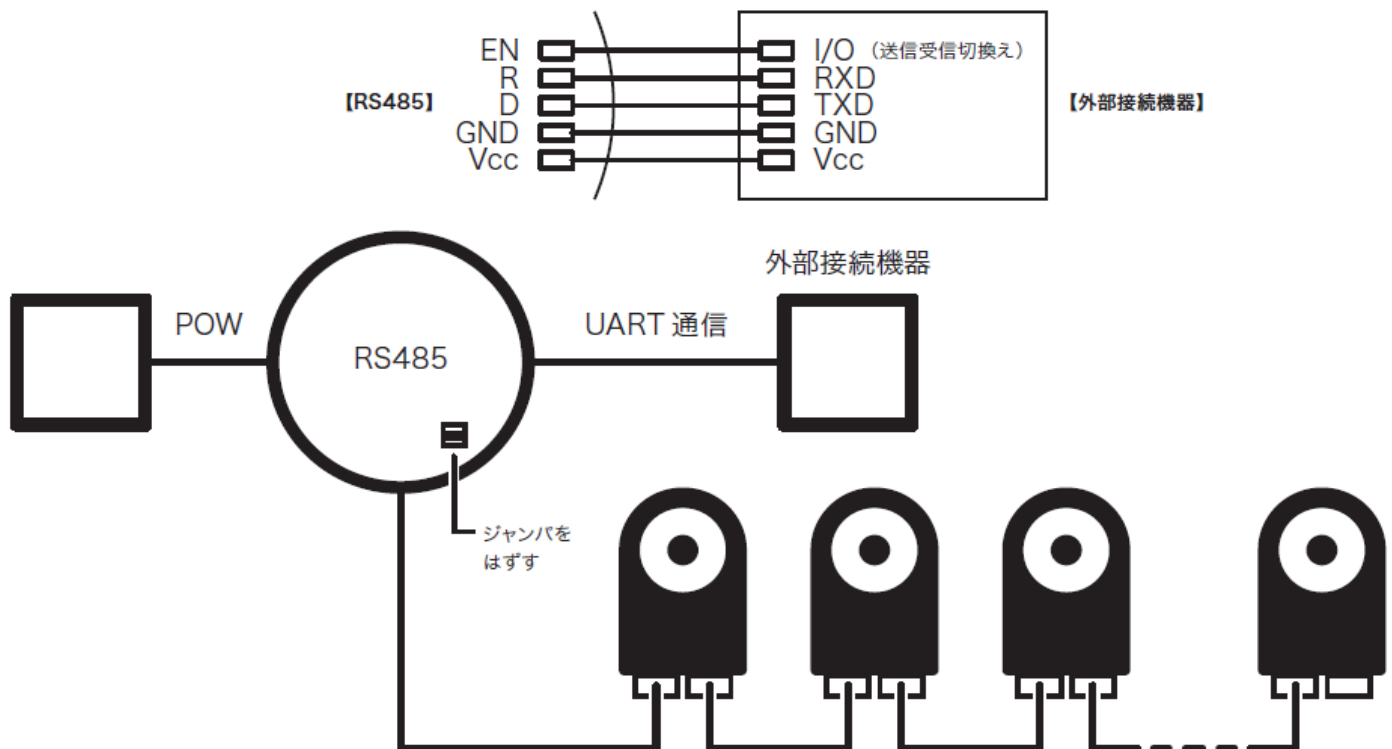
<http://kondo-robot.com/fag/ko-driver-2015>

3. B3Mマネージャーソフトをインストールします。（マネージャーソフトの使用方法は、B3M Managerマニュアルを参照してください）。

【KONDO website】カスタマーセンター→サポート情報→B3M開発資料→「B3M Managerソフトウェア」

<http://kondo-robot.com/fag/b3m-manager-software-2>

## ■ UART通信の場合



1. JP1の半田ジャンパをはずします。
2. UART端子に外部接続機器を接続します。
3. 外部接続機器から485対応機器にデータを送信する時は、EN端子をHレベル、それ以外はLレベルにしてください。

## ■ 外部電源使用時

RS485対応機器に電力を供給する場合に使用します。RS485のVcc/GNDラインを使って外部より電力を供給する場合は不要です。

## ■ 電源

電源は12Vの直流電源またはバッテリーをご用意ください。B3Mに電源を接続する場合は、極性を間違えないように接続してください。必要な電流値（アンペア）は使用するモーターの種類や数などによって変わります。詳しくは使用するモーターの仕様書などをご覧ください。



## ■ リチウムフェライトバッテリーラインアップ

**ROBOパワーセル F3-850タイプ (Li-Fe)**

NO.02171 ¥3,000（税別）

**ROBOパワーセル F3-1450タイプ (Li-Fe)**

NO.02172 ¥3,500（税別）

**ROBOパワーセル F3-2100タイプ (Li-Fe)**

NO.02174 ¥4,500（税別）

## サーボの最大消費電流

下記の最大消費電流を目安に電源を選択してください。

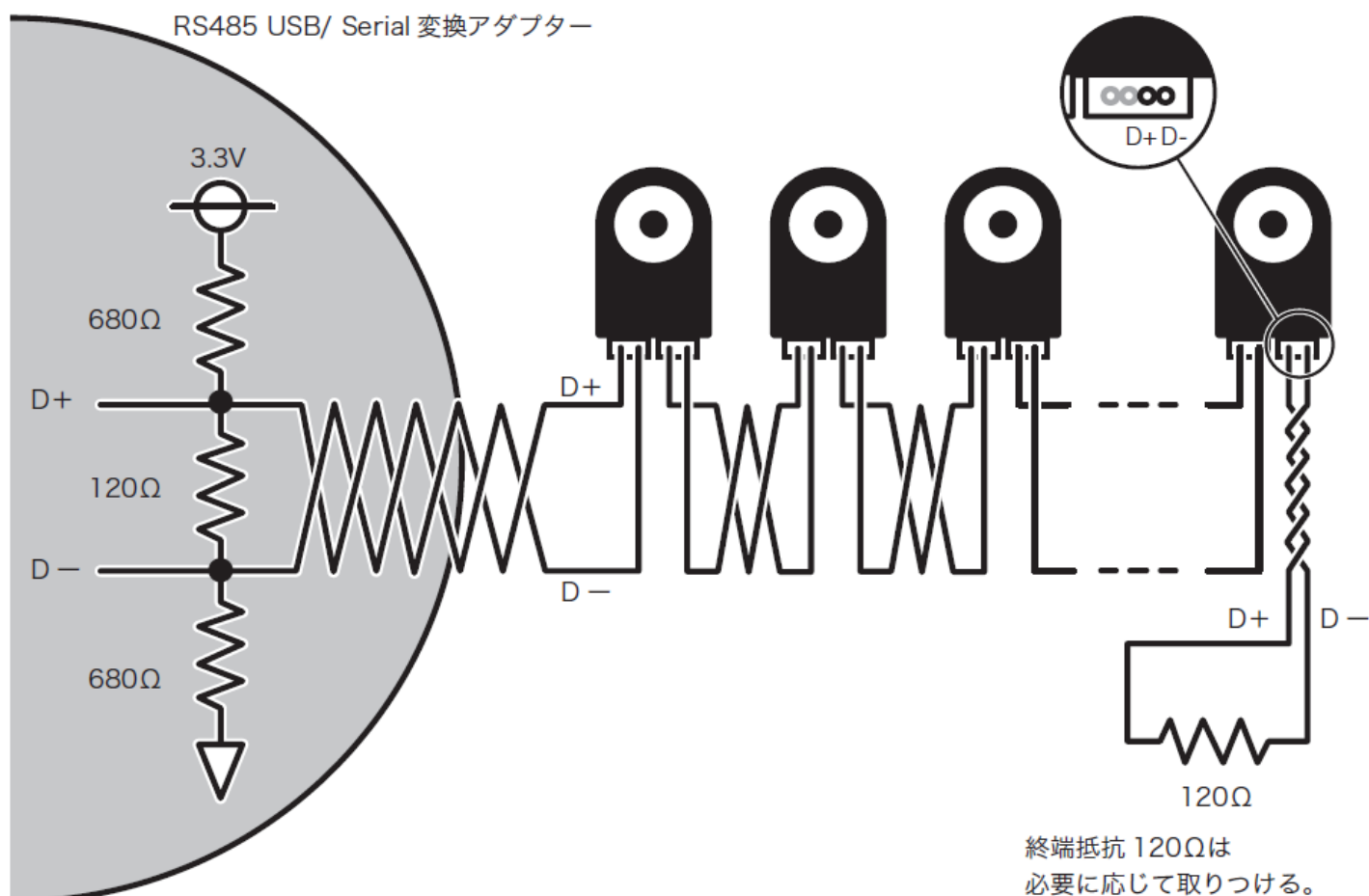
- ・ SB-1040-A・・・サーボ1個当たり**最大**2.8Aの電流容量が必要です。
- ・ SC-1040-A・・・サーボ1個当たり**最大**3.6Aの電流容量が必要です。
- ・ SC-1170-A・・・サーボ1個当たり**最大**5.4Aの電流容量が必要です。

## 【ご注意ください】

B3Mシリーズの電源回路は、ソフトスタート機能が装備されているため、電源の立ち上がりが遅い場合はサーボが正常に起動しない場合があります。安定化電源を電源とする場合は、サーボと電源の間にスイッチを入れ、安定化電源をONにしてからスイッチをONにすると、サーボの起動が安定します。

## ■ マルチドロップ接続について

サーボ同士をつなげたマルチドロップ接続が可能です。接続には、XH接続ケーブルがお勧めです。  
サーボが複数台になる場合は、ハブを使用することもできます。



### ■ 備考

- ・ D+ と D- はノイズ軽減のためよじってください。
- ・ 終端抵抗を取りつける場合は D+ と D- 端子を 120Ω でショートさせてください。
- ・ 40cm ケーブル使用時で 10 個以上のサーボモーターを直列に取りつける場合、または伝送距離（ケーブル）が長い場合は 120Ω の抵抗で終端してください。10 個以下でも終端抵抗を付けた方が信号の劣化が少なくなります。



## 2.通信仕様について

### ■シリアル通信設定

B3Mシリーズは、ホストより送信されるコマンドにより、サーボの動作やサーボのステータスを取得する事ができます。

名称	機 能
通信形式	RS-485準拠
転送モード	半二重非同期式通信（調歩同期式通信）
通信速度	115200～3000000bps（115200, 1000000, 1500000, 2000000, 3000000bpsのいずれかを推奨）ガードタイム200 [us]以上
スタートビット	1bit
データビット	8bit
パリティビット	なし
ストップビット	1bit
備考	<p>※通信速度2Mbps以上で通信を行った時、通信エラーになる場合があります。その場合、ストップビットを2にすることで改善されることがあります。</p> <p>※コマンドは2バイト分以上の間隔が空くと一連のコマンドとして認識されません。</p> <p>※連続して通信を行う場合は通信間隔を2バイト分のデータ+220us以上に空けて下さい。これは送受信データ量および通信速度により変わります。</p>

### ■通信モード

B3Mは通常の1対1の通信（シングルモード）に加え、一度に多数のデバイスと通信するマルチモードを搭載しています。モード切替は、1台目のIDと各データに続けて2台目以降のIDとデータを連続して送信することで自動的にマルチモードとして送信されます。

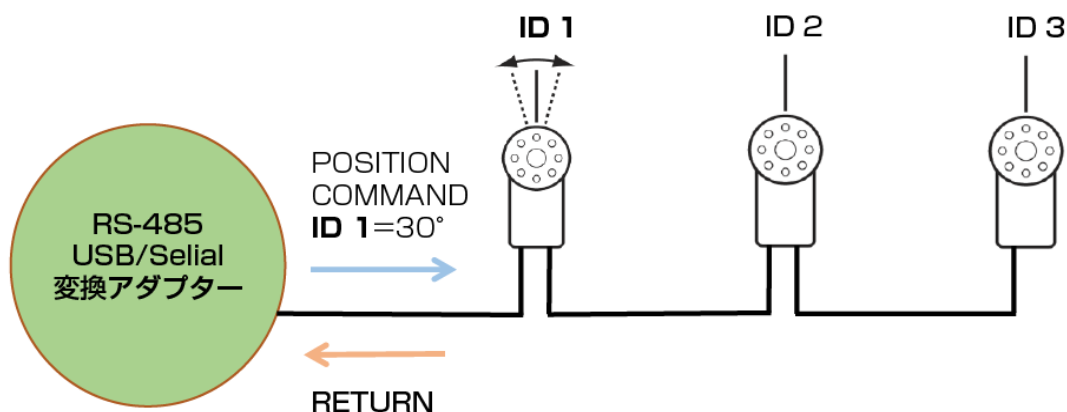
名称	内 容
シングル	1個のIDまたはブロードキャストIDを指定してコマンドを実行します。
マルチ	2個以上のIDを指定してコマンドを実行します。 ブロードキャストIDは使用できません(無視されます)。

### ■ブロードキャストIDについて

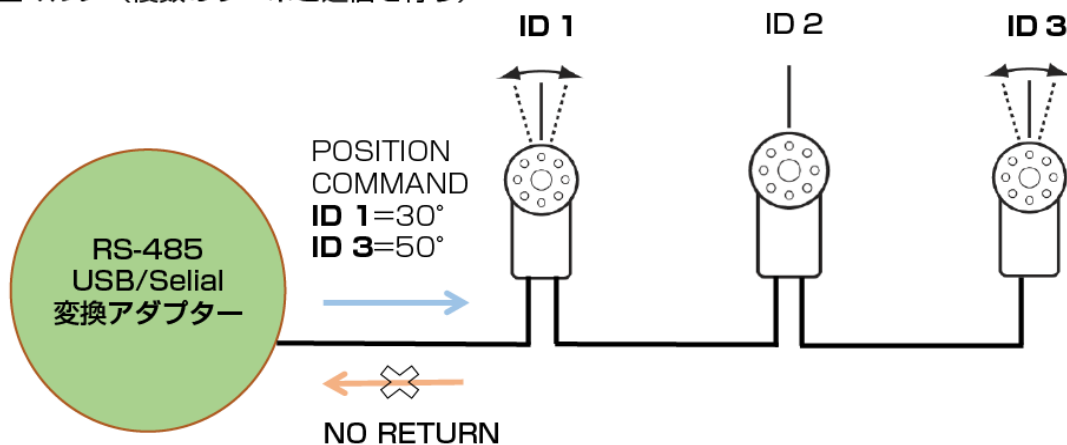
- ・ID 255を使用すると全てのデバイスに一斉に命令を送ることができます。
- ・ブロードキャストIDは特殊なID番号で255に固定。

※マルチコマンドおよびブロードキャストIDを指定したときは返信データはありません。

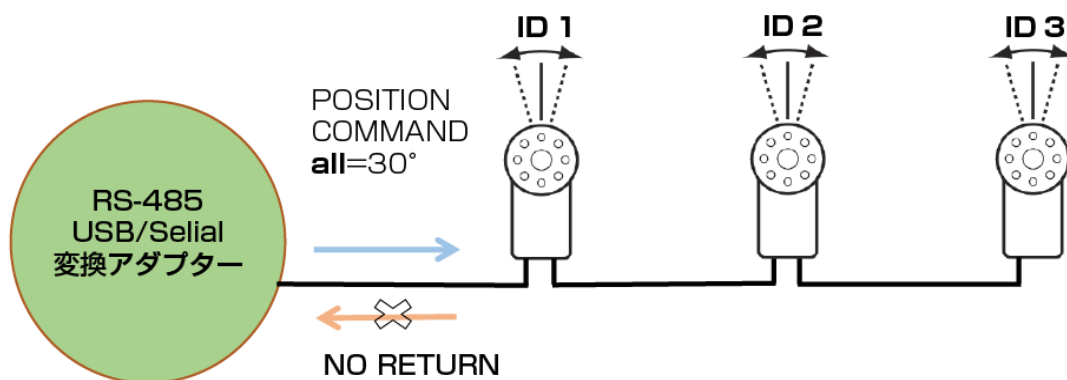
■シングル（1個のサーボと通信を行う）



■マルチ（複数のサーボと通信を行う）



■ブロードキャスト（全てのサーボと通信を行う）



## ■ データ構造

データ構造は、ホストより送信する【送信フォーマット】と、サーボより送信する【返信フォーマット】の2種類あります。

### ■ シングルモード

【送信フォーマット】

内容	SIZE	COMMAND	OPTION	ID	S-DATA	SUM
バイト	1	1	1	1	0～N	1

【返信フォーマット】

内容	SIZE	COMMAND	STATUS	ID	R-DATA	SUM
バイト	1	1	1	1	0～N	1

### ■ マルチモード

【送信フォーマット】

内容	SIZE	COMMAND	OPTION	ID1	S-DATA	SUM
バイト	1	1	1	1	0～N	1



マルチモードでは、S-DATAに連結して複数のサーボIDに対してコマンドを送ることができます。

【返信フォーマット】

・返信データはありません。

マルチモードでのフォーマットについては、「3.コマンド詳細」をご覧ください。

## ■データフォーマットの名称説明

各フォーマットのそれぞれの項目を紹介します。「3.コマンド詳細」と合わせてご覧ください。

### 【SIZE】データ長

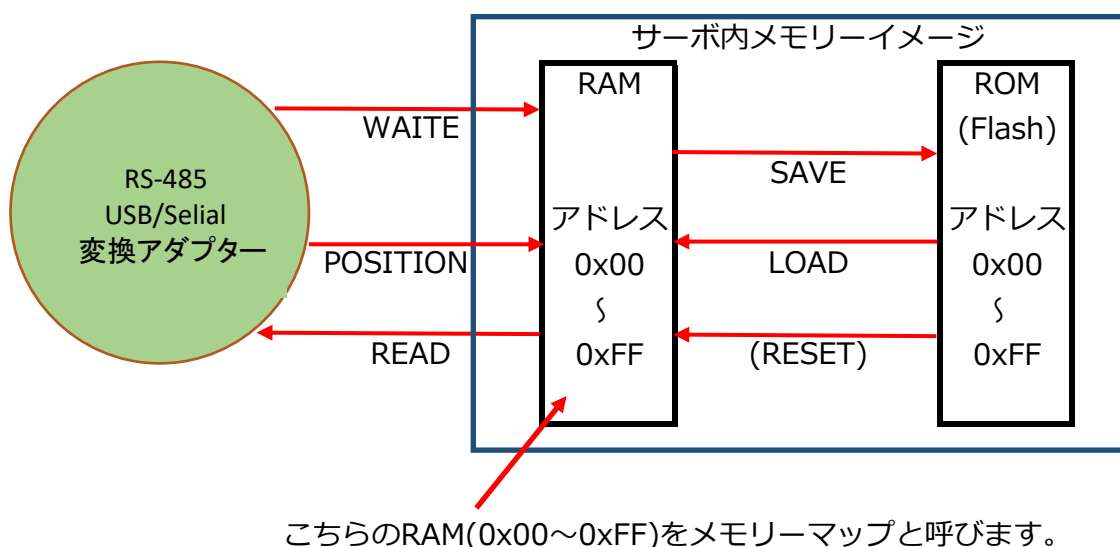
- ・フォーマット(SIZE～SUM)の長さを指定します。最大256バイトです。

### 【COMMAND】コマンド

- ・B3Mは、シンプルな6種類のコマンドによりサーボ及びデータ制御ができます。  
「3.コマンド詳細」にて、各コマンドの使用例を紹介しています。

名称	CODE	機 能
LOAD	0x01	ROMの内容をIOマップ（RAM）に読み出します
SAVE	0x02	IOマップ（RAM）の内容をROMに保存します
READ	0x03	IOマップ(RAM)のアドレス指定でデバイスからデータ(RAM)を読み込みます
WRITE	0x04	IOマップ(RAM)のアドレス指定でデバイスにデータ(RAM)を書き込みます
RESET	0x05	デバイスを再起動します
POSITION	0x06	サーボモーターのポジションを変更します

※READコマンドで取得したデータは通信中にデータの値が変化している場合があります。



### 【OPTION】オプション / 【STATUS】ステータス

- ・送信フォーマットのOPTIONで指定したエラー項目を、返信フォーマットのSTATUSにて状態を返信します。また、OPTIONでステータスエラーのクリアの指示を行います。  
詳細は、「4.オプション/ステータス詳細」をご覧ください。

## 【ID】 ID番号

- ・サーボなどデバイスのID番号です。

## 【S-DATA】 データ(送信フォーマット)

各コマンドの送信データをS-DATAとして送信します。

RESETでは時間（TIME）をS-DATAで送信しますが、これは何秒後にリセットをかけるかを指定できます。

POSITIONは、角度と時間を送ることで動作角と目標に到達するための時間を指定することができます。ただし、軌道生成タイプ（0x29）がNomal以外に設定されているときのみ時間の設定が有効になります。Noemalの場合は、目標移動時間の設定値に関係なく最速で動きます。

マルチモードのときは、LOADとSAVEに複数のサーボIDを入れることで同時に動作指示をだすことができます。WRITEでは複数のデータとIDを交互に送ることで、同時に複数台に書き込みが可能です。RESETでは指定した複数のサーボを同時にリセットをかけることができます。

POSITIONも同様です。READは、マルチモードに対応していません。

S-DATAの詳細な使用方法は「3.コマンド詳細」を参照してください。

名称	内容	シングルモード	マルチモード
LOAD	S-DATAは使用しません	データ無し	1番目のIDに続けて2番目以降のサーボID番号を指定します(ID2・・・IDn)
SAVE	S-DATAは使用しません	データ無し	1番目のIDに続けて2番目以降のサーボID番号を指定します(ID2・・・IDn)
READ	読み出す情報のアドレスとバイト数	アドレス(ADDRES) +バイト数(LENGTH)	
WRITE	書き込む情報とアドレス	書き込みデータ(DATA1) +アドレス (ADDRES)	1番目の書き込みデータ(DATA1) +2番目以降の書き込みID(IDn) +2番目以降の書き込みデータ(DATAN) +書き込みの先頭アドレス(ADDRES) +指定するデバイスの数(COUNT)
RESET	TIME（何秒後にリセットするか）	時間(TIME)	2番目以降のサーボID番号(ID2・・・IDn)+ 時間(TIME)
POSITION	角度と到達までの時間を指定	角度(POS)+時間(TIME)	1番目のPOS(POS1) +2番目以降のID(IDn) +2番目以降の角度(POSn) +時間(TIME)

### 【R-DATA】データ(返信フォーマット)

READでは、各パラメータのデータが、POSITIONではデータ送信時の角度が返信されてきます。

LOAD、SAVE、WRITEでは返信されるデータにR-DATAは含まれません。RESETは、サーボからの返信がありません。また、マルチモードではすべてのコマンドにおいて返信はありません。

名称	シングルモード	マルチモード
LOAD	データ無し	
SAVE	データ無し	
READ	指定したアドレスのデータ	
WRITE	データ無し	
RESET		
POSITION	データ送信時の角度(POS)	

### 【ADDRESS】アドレス

・WRITE（書き込み）、READ（読み出し）コマンドで対象となるデータのメモリーマップ上のアドレスを指定します。4BYTEデータなど複数に渡って連続してアドレスがある場合は、先頭アドレスを指定します。詳細は、「7.メモリーマップについて」をご参照ください。

### 【LENGTH】長さ

・READコマンドの読み取るR-DATAのバイト数を指定します。

### 【COUNT】カウント

・WRITEコマンドの指定する対象デバイスのID数（台数）を指定します。

### 【POS】ポジション(Position)データ

・SET POSITIONコマンドで角度を指定します。

### 【TIME】ポジションデータ

・RESETでは、リセットがかかるまでの時間を指定できます。  
 ・SET POSITIONコマンドで軸の到達時間を指定します。ただし、軌道生成タイプ（0x29）がNormal以外に設定されているときのみ時間の設定が有効になります。「Normal」の場合は、目標移動時間の設定値に関係なく最速で動きます。

### 【SUM】チェックサム

・SUMを除くコマンドのデータ総和の下位1バイトです。

## 3.コマンド詳細

### ■コマンド一覧

シンプルな6種類のコマンドによりサーボ及びデータの制御ができます。

シングルモードでは、送信後にデバイスからの返事を受け取ることができます。この中に、デバイスの状態を知ることができる「STATUS」が含まれていますので、デバイスが正しく動作できているかなどの状態を確認することができます。READ以外は、マルチモードにて複数のサーボに同時に命令を送ることができます。ただし、マルチモードではデバイスからの返事を受け取ることはできません。

名称	モード	コマンド	機 能
LOAD	送信 (S/M)	0x01	ROMの内容をメモリーマップ（RAM）に読み出します
	返信	0x81	STATUSとIDが返ってきます。 ※マルチモード時は返事はありません
SAVE	送信 (S/M)	0x02	メモリーマップ（RAM）の内容をROMに保存します
	返信	0x82	STATUSとIDが返ってきます。 ※マルチモード時は返事はありません
READ	送信 (Sのみ)	0x03	メモリーマップ(RAM)のアドレス指定でデバイスからデータ(RAM)を読み込みます
	返信	0x83	指定したアドレスの読み取ったデータが返ってきます。
WRITE	送信 (S/M)	0x04	メモリーマップ(RAM)のアドレス指定でデバイスにデータ(RAM)を書き込みます
	返信	0x84	STATUSとIDが返ってきます。 ※マルチモード時は返事はありません
RESET	送信 (S/M)	0x05	デバイスを再起動します
	返信	—	返事はありません
POSITION	送信 (S/M)	0x06	サーボモーターのポジションを変更します
	返信	0x86	STATUSとID、データ送信時の角度が返ってきます。 ※マルチモード時は返事はありません

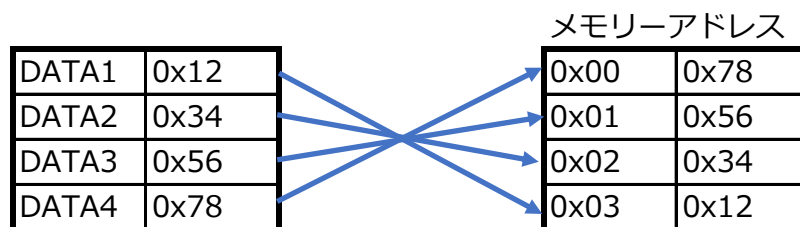
※モード欄；S=シングルモード M=マルチモード

”S/M”は両モードに対応していることを意味します。

## ■データの統合と分割について

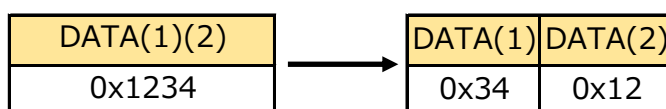
READコマンド、WRITEコマンドでメモリーマップ(RAM)の内容を読み込み、書き込みする時に、2バイト又は4バイトの項目は、リトルエンディアン方式でデータが格納されています。

- ・通信フォーマットのデータ項目がlong(4byte)型の場合

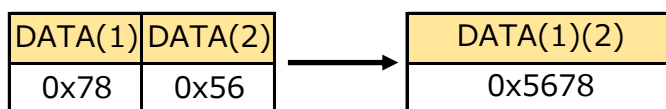


- ・2バイトのデータの場合

- ・書き込みデータが0x1234の時

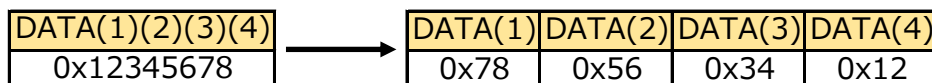


- ・読み込みデータが0x5678の時

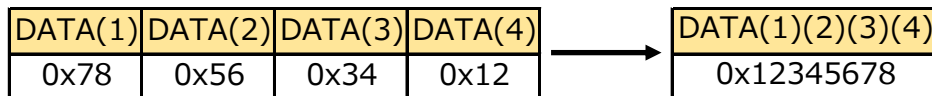


- ・4バイトのデータの場合

- ・書き込みデータが0x12345678の時



- ・読み込みデータが0x12345678の時





## ■ LOADコマンド

ROMの内容をメモリーマップ（RAM）に読み出します

通信モード	シングル	マルチ
-------	------	-----

### ■ シングルモードの送信フォーマット

桁	1	2	3	4	5
項目	SIZE	COMMAND	OPTION	ID	SUM

データ詳細				
桁	項目	設定範囲		内 容
		最小	最大	
1	SIZE	0x05		コマンド一連のデータバイト数
2	COMMAND	0x01		
3	OPTION			「4.オプション/ステータス詳細」参照
4	ID	0x00	0xFF	LOAD対象デバイスのID番号
5	SUM			SIZE～IDまでの総和の下位1バイト

### ■ マルチモードの送信フォーマット

1番目のIDに続けて2番目以降のサーボID番号を指定することで、一度で複数のデバイス宛てに命令を送ることができます。

桁	1	2	3	4		n+2	n+3	n+4
項目	SIZE	COMMAND	OPTION	ID1	...	ID(n-1)	ID(n)	SUM

(n:指定するデバイスの数)

データ詳細				
桁	項目	設定範囲		内 容
		最小	最大	
1	SIZE	0x05	0xFF	コマンド一連のデータバイト数
2	COMMAND	0x01		
3	OPTION			「4.オプション/ステータス詳細」参照
4	ID1	0x00	0xFE	LOAD対象デバイスのID番号
...				
n+3	ID(n)	0x00	0xFE	LOAD対象デバイスのID番号
n+4	SUM			SIZE～ID(n)までの総和の下位1バイト

## ■ 返信データ（シングルモードのみ）

※マルチモードやブロードキャストIDを指定したときときは返事なし

桁	1	2	3	4	5
項目	SIZE	COMMAND	STATUS	ID	SUM

データ詳細				
桁	項目	設定範囲		内 容
		最小	最大	
1	SIZE	0x05		コマンド一連のデータバイト数
2	COMMAND	0x81		
3	STATUS	OPTIONにより変化		「4.オプション/ステータス詳細」参照
4	ID	送信IDと同じ		返信したデバイスのID
5	SUM			SIZE～IDまでの総和の下位1バイト

## ■ 使用例

ID1のデバイスに対して、LOADコマンドによりROMの内容をメモリーマップ（RAM）に呼び出します。

【送信データ】

桁	項目	データ
1	SIZE	0x05
2	COMMAND	0x01
3	OPTION	0x00
4	ID	0x01
5	SUM	0x07

【返信データ】

桁	項目	データ
1	SIZE	0x05
2	COMMAND	0x81
3	STATUS	0xXX
4	ID	0x01
5	SUM	0xXX

※0xXXは不定

## ■SAVEコマンド

メモリーマップ（RAM）の内容をROMに保存します。

通信モード	シングル	マルチ
-------	------	-----

### ■シングルモードの送信フォーマット

桁	1	2	3	4	5
項目	SIZE	COMMAND	OPTION	ID	SUM

データ詳細				
桁	項目	設定範囲		内 容
		最小	最大	
1	SIZE	0x05		コマンド一連のデータバイト数
2	COMMAND	0x02		
3	OPTION			「4.オプション/ステータス詳細」参照
4	ID	0x00	0xFF	SAVE対象デバイスのID番号
5	SUM			SIZE～IDまでの総和の下位1バイト

### ■マルチモードの送信フォーマット

1番目のIDに続けて2番目以降のサーボID番号を指定することで、一度で複数のデバイス宛てに命令を送ることができます。

桁	1	2	3	4		n+2	n+3	n+4
項目	SIZE	COMMAND	OPTION	ID1	...	ID(n-1)	ID(n)	SUM

(n:指定するデバイスの数)

データ詳細				
桁	項目	設定範囲		内 容
		最小	最大	
1	SIZE	0x05	0xFF	コマンド一連のデータバイト数
2	COMMAND	0x02		
3	OPTION			「4.オプション/ステータス詳細」参照
4	ID1	0x00	0xFE	SAVE対象デバイスのID番号
...				
n+3	ID(n)	0x00	0xFE	SAVE対象デバイスのID番号
n+4	SUM			SIZE～ID(n)までの総和の下位1バイト

## ■ 返信データ（シングルモードのみ）

※マルチモードやブロードキャストIDを指定したときときは返事なし

桁	1	2	3	4	5
項目	SIZE	COMMAND	STATUS	ID	SUM

データ詳細				
桁	項目	設定範囲		内 容
		最小	最大	
1	SIZE	0x05		コマンド一連のデータバイト数
2	COMMAND	0x82		
3	STATUS	OPTIONにより変化		「4.オプション/ステータス詳細」参照
4	ID	送信IDと同じ		返信したデバイスのID
5	SUM			SIZE～IDまでの総和の下位1バイト

## ■ 使用例

ID1のデバイスに対して、SAVEコマンドによりメモリーマップ（RAM）の内容をROMに保存します。

## 【送信データ】

桁	項目	データ
1	SIZE	0x05
2	COMMAND	0x02
3	OPTION	0x00
4	ID	0x01
5	SUM	0x08

## 【返信データ】

桁	項目	データ
1	SIZE	0x05
2	COMMAND	0x82
3	STATUS	0xXX
4	ID	0x01
5	SUM	0xXX

※0xXXは不定

## ■ READコマンド

メモリーマップ(RAM)のアドレスを指定してデバイスからRAMデータを読み込みます。

通信モード	シングル
-------	------

### ■ シングルモードの送信フォーマット

桁	1	2	3	4	5	6	7
項目	SIZE	COMMAND	OPTION	ID	ADDRESS	LENGTH	SUM

データ詳細				
桁	項目	設定範囲		内 容
		最小	最大	
1	SIZE	0x07		コマンドの総データバイト数
2	COMMAND	0x03		コマンド
3	OPTION			「4.オプション/ステータス詳細」参照
4	ID	0x00	0xFE	ID番号(0xFF(ブロードキャスト)はエラー)
5	ADDRESS	0x00	0xFF	データ読み出す先頭のアドレス(メモリーマップ参照)
6	LENGTH	0x01	0xFA	読み出すデータの長さ(0x00はエラー、最大250byteしか読み出せません)
7	SUM			SIZE～LENGTHまでの総和の下位1バイト

### ■ 返信フォーマット

桁	1	2	3	4	5	...	n+4	n+5
項目	SIZE	COMMAND	STATUS	ID	DATA(1)	...	DATA(n)	SUM

(n:返信したデータの数)

データ詳細				
桁	項目	設定範囲		内 容
		最小	最大	
1	SIZE	DATA(n)に対してn+5		コマンドの総データバイト数
2	COMMAND	0x83		コマンド
3	STATUS	OPTIONにより変化		「4.オプション/ステータス詳細」参照
4	ID	送信IDと同じ		返信したデバイスのID
5	DATA	0x01	0xFF	実際のデータ(データ数は送信フォーマットのLENGTHによる)
...				
n+5	SUM			SIZE～DATA(n)までの総和の下位1バイト

## ■ 使用例

ID0のデバイスに対して、READコマンドによりアドレス0xA2番地から4バイト分のデータを読み込む。

【送信データ】

桁	項目	データ
1	SIZE	0x07
2	COMMAND	0x03
3	OPTION	0x00
4	ID	0x00
5	ADDRESS	0xA2
6	LENGTH	0x04
7	SUM	0xB0

【返信データ】

桁	項目	データ
1	SIZE	0x09
2	COMMAND	0x83
3	STATUS	0xFF
4	ID	0x00
5	DATA1	0xFF
6	DATA2	0xFF
7	DATA3	0xFF
8	DATA4	0xFF
9	SUM	0xFF

※0xFFは不定

## ■ WRITEコマンド

メモリーマップ(RAM)のアドレス指定でデバイスのRAM上に書き込みます。

通信モード	シングル	マルチ
-------	------	-----

### ■ シングルモードの送信フォーマット

桁	1	2	3	4	5		m+4
項目	SIZE	COMMAND	OPTION	ID	DATA(1)	...	DATA(m)

桁	(m+1)+4	(m+1)+5	(m+1)+6
項目	ADDRESS	COUNT	SUM

m:1個のデバイスに対して書き込むByte数

データ詳細				
桁	項目	設定範囲		内 容
		最小	最大	
1	SIZE	0x08	0xFF	コマンドの総データバイト数
2	COMMAND	0x04		コマンド
3	OPTION			「4.オプション/ステータス詳細」参照
4	ID	0x00	0xFF	ID番号(※)
5	DATA(1)	0x00	0xFF	書き込むデータ(下位バイトが先になる)
...				(short型・long型はリトルエンディアン型)
(m+1)+4	ADDRESS	0x00	0xFF	データを書き込む先頭のアドレス(メモリーマップ参照)
(m+1)+5	COUNT	0x01	0x01	指定するデバイスの数(0x00はエラー)
(m+1)+6	SUM			SIZE～COUNTまでの総和の下位1バイト

## ■ マルチモードの送信フォーマット

1番目のID、DATAに続けて2番目以降のIDとDATAを続けて指定することで、一度で複数のデバイス宛てに命令を送ることができます。

桁	1	2	3
項目	SIZE	COMMAND	OPTION

桁	4	5		m+4	(m+1)+3	(m+1)+4		(m+1)+3
項目	ID1	DATA1(1)	...	DATA1(m)	ID2	DATA2(1)	...	DATA2(m)

ID1
ID2

桁	$n*(m+1)+3$	$n*(m+1)+4$		$n*(m+1)+3$	$n*(m+1)+4$	$n*(m+1)+5$	$n*(m+1)+6$
項目	ID(n)	DATA(n)(1)	...	DATA(n)(m)	ADDRESS	COUNT	SUM

ID(n)

m: 1個のデバイスに対して書き込むByte数  
 n: 書き込むデバイスの数(COUNTと同じ数)

データ詳細				
桁	項目	設定範囲		内 容
		最小	最大	
1	SIZE	0x08	0xFF	コマンドの総データバイト数
2	COMMAND	0x04		コマンド
3	OPTION			「4.オプション/ステータス詳細」参照
4	ID1	0x00	0xFF	ID番号(※)
5	DATA1(1)	0x00	0xFF	書き込むデータ(下位バイトが先になる)
...				(short型・long型はリトルエンディアン型)
$n*(m+1)+4$	ADDRESS	0x00	0xFF	データを書き込む先頭アドレス(メモリーマップ参照)
$n*(m+1)+5$	COUNT	0x01	0xFF	指定するデバイスの数(0x00はエラー)
$n*(m+1)+6$	SUM			SIZE～COUNTまでの総和の下位1バイト



### ■ 返信データ（シングルモードのみ）

※マルチモードやブロードキャストIDを指定したときときは返事なし

桁	1	2	3	4	5
項目	SIZE	COMMAND	STATUS	ID	SUM

データ詳細				
桁	項目	設定範囲		内 容
		最小	最大	
1	SIZE	0x05		コマンドの総データバイト数
2	COMMAND	0x84		コマンド
3	STATUS	OPTIONにより変化		「4.オプション/ステータス詳細」参照
4	ID	送信IDと同じ		返信したデバイスのID
5	SUM			SIZE～IDまでの総和の下位1バイト

※ 下記の条件を満たす場合はインデックス番号の早いものが処理され、それ以外は無効となる。

1. 同一IDが存在する
2. 通常のIDとブロードキャストIDを両方指定している

※ ID(1)～DATA(n)(m)までの長さがCOUNTで割り切れない場合はコマンドが無効となる。

## ■ 使用例

IDが1、3、5のデバイスに対して、WRITEコマンドにより先頭アドレス0x05番地から4バイト分のデータをマルチモードで書き込む。

ID=1 最小位置制限(0x05.0x06番地)=-32000(0x8300)

ID=1 最大位置制限(0x07.0x08番地)=32000(0x7D00)

ID=3 最小位置制限(0x05.0x06番地)=-31000(0x86E8)

ID=3 最大位置制限(0x07.0x08番地)=32000(0x7918)

ID=5 最小位置制限(0x05.0x06番地)=-30000(0x8AD0)

ID=5 最大位置制限(0x07.0x08番地)=30000(0x7530)

## 【送信フォーマット】

桁	項目	データ	ID番号	アドレス
1	SIZE	0x07		
2	COMMAND	0x04		
3	OPTION	0x00		
ID1	ID(1)	0x01	1	
	DATA1(1)	0x00		0x05
	DATA1(2)	0x83		0x06
	DATA1(3)	0x00		0x07
ID3	DATA1(4)	0x7D		0x08
	ID(2)	0x03	3	
	DATA2(1)	0xE8		0x05
	DATA2(2)	0x86		0x06
ID5	DATA2(3)	0x18		0x07
	DATA2(4)	0x79		0x08
	ID(3)	0x05	5	
	DATA3(1)	0xD0		0x05
	DATA3(2)	0x8A		0x06
	DATA3(3)	0x30		0x07
	DATA3(4)	0x75		0x08
19	ADDRESS	0x05		
20	COUNT	0x03		
21	SUM	0x1A		

※上記コマンドは、マルチモードのため返信データはありません。

## ■ RESETコマンド

デバイスを再起動します。

リセット後にROMの内容をメモリーマップ（RAM）に読み出します。

※RAMの内容は上書きされますので、データを保存したい場合はSAVEコマンドでROMにデータを保存してから実行してください。

通信モード	シングル	マルチ
-------	------	-----

### ■ シングルモードの送信フォーマット

桁	1	2	3	4	n+4	n+5
項目	SIZE	COMMAND	OPTION	ID	TIME	SUM

データ詳細					
桁	項目	設定範囲		内 容	
		最小	最大		
1	SIZE	0x06	0xFF	コマンドの総データバイト数	
2	COMMAND	0x05		コマンド	
3	OPTION			「4.オプション/ステータス詳細」参照	
4	ID	0x00	0xFF	ID番号(0xFFですべてリセットをする)	
5	TIME	0x00	0xFF	指定時間は0~25.5sec（100ms単位）	
6	SUM			SIZE~TIMEまでの総和の下位1バイト	

※ Resetコマンドに返信はありません

## ■ マルチモードの送信フォーマット

桁	1	2	3
項目	SIZE	COMMAND	OPTION

桁	4	5		n+2	n+3	n+4	n+5
項目	ID1	ID2	...	ID(n-1)	ID(n)	TIME	SUM

n:指定するデバイスの数

データ詳細				
桁	項目	設定範囲		内 容
		最小	最大	
1	SIZE	0x06	0xFF	コマンドの総データバイト数
2	COMMAND	0x05		コマンド
3	OPTION			「4.オプション/ステータス詳細」参照
4	ID	0x00	0xFF	ID番号(0xFFですべてリセットをする)
...				
n+4	TIME	0x00	0xFF	指定時間は0~25.5sec (100ms単位)
n+5	SUM			SIZE~TIMEまでの総和の下位1バイト

※ Resetコマンドに返信はありません

## ■ 使用例

ID0のデバイスをRESETコマンドにより300ms後に再起動する

## 【送信フォーマット】

桁	項目	データ
1	SIZE	0x06
2	COMMAND	0x05
3	OPTION	0x00
4	ID	0x00
5	TIME	0x03
6	SUM	0x0E

## ■ SET POSITIONコマンド

サーボモータに動作角と目標に到達するまでの時間を指定しポジションを変更します。

目標移動時間は、軌道生成タイプ（0x29）を「Even」「ThirdPoly」「ForthPoly」「FifthPoly」に設定したときのみ有効です。

「Noemal」の場合は、目標移動時間の設定値に関係なく最速で動きます。

通信モード	シングル	マルチ
-------	------	-----

### ■ シングルモードの送信フォーマット

桁	1	2	3	4	5	6
項目	SIZE	COMMAND	OPTION	ID	POS_L	POS_H

桁	7	8	9
項目	TIME_L	TIME_H	SUM

TIME

POS

データ詳細				
桁	項目	設定範囲		内 容
		最小	最大	
1	SIZE	0x09	0xFF	コマンドの総データバイト数
2	COMMAND	0x06		コマンド
3	OPTION			「4.オプション/ステータス詳細」参照
4	ID	0x00	0xFF	
5	POS_L	0x00	0xFF	ポジションデータ下位バイト
6	POS_H	0x00	0xFF	ポジションデータ上位バイト
7	TIME_L	0x00	0xFF	目標移動時間は0~65535ms、軌道生成タイプ(0x29)が0(Noemal)以外の時有効
8	TIME_H	0x00	0xFF	
9	SUM			SIZE~TIME_Hまでの総和の下位1バイト

ポジションデータの詳細は、下記「ポジションデータについて」をご参照ください。

#### 【軌道生成タイプ（アドレス：0x29）】

軌道生成タイプは、アドレス0x29のデータをWRITEコマンドで書き換えて変更します。

詳しくは、「8.メモリーマップ詳細について」の「起動生成タイプ」の項をご参照ください。

0:Normal・・・最速で回転します。

1:Even・・・目標時間を均等分割して始点から終点までを補完します（軌道生成）。

3:ThirdPoly・・・3次多項式を使って始点から終点までを補完します。

4:ForthPoly・・・4-1-4多項式補完法で始点から終点までを補完します。

5:FifthPoly・・・5次多項式を使って始点から終点までを補完します。

※動作モード（0x28）がNormalで位置制御モード時のみ有効です。

## ■ マルチモードの送信フォーマット

桁	1	2	3
項目	SIZE	COMMAND	OPTION

4	5	6	7	8	9	
ID1	POS(1)_L	POS(1)_H	ID2	POS(2)_L	POS(2)_H	...

ID1

ID2

3n+4	3n+5	3n+6	3n+7	3n+8	3n+9
ID(n)	POS(n)_L	POS(n)_H	TIME_L	TIME_H	SUM

ID(n)

TIME

n:指定するデバイスの数

データ詳細				
桁	項目	設定範囲		内 容
		最小	最大	
1	SIZE	0x09	0xFF	コマンドの総データバイト数
2	COMMAND	0x06		コマンド
3	OPTION			「4.オプション/ステータス詳細」参照
4	ID1	0x00	0xFF	ID番号(同一ID指定は後のデータを優先する)
5	POS(1)_L	0x00	0xFF	ポジションデータ下位バイト
6	POS(1)_H	0x00	0xFF	ポジションデータ上位バイト
...				
3n+7	TIME_L	0x00	0xFF	目標時間は0~65535ms、軌道生成タイプ (0x29)が0(Normal)以外の時有効
3n+8	TIME_H	0x00	0xFF	
3n+9	SUM			SIZE~TIME_Hまでの総和の下位1バイト

ポジションデータの詳細は、下記「ポジションデータについて」をご参照ください。

## ■ ポジションデータについて

B3Mは、動作角-320°～320°の範囲で動作します。  
 パラメータでは、10進数で-32000～32000の範囲ですが、  
 16進数では左右で指定する数値が異なります。詳細は以下をご参照ください。

		左回り 最大値	センター	右回り 最大値
角度		-320°	0°	320°
パラメータ	10進数	-32000	0	32000
	16進数	<b>0x8300</b>	<b>0x00</b>	<b>0x7d00</b>

上記一覧の最大値は、出荷時の数値です。

最大動作角は、アドレス0x05（最小）、0x07（最大）にて変更できます。

設定した最大動作角以上のパラメータを指定してもそれ以上の角度には動作しません。

右回りは10進数でパラメータ"1"のとき16進数では"0x0001"になりますが、  
 左回りでは10進数"-1"のとき"0xFFFF"になり、最大値"-32000"の時は"0x8300"になりますので  
 数値が減る方向になります。

左右ともに、設定できる範囲は10進数で1～32000の31999、16進数で7CFFです。

センターは0です。

		左回り	センター	右回り
パラメータ	10進数	-1	0	1
	16進数	0xFFFF	0x00	0x0001



## ■ 返信データ

桁	1	2	3	4	5	6	7
項目	SIZE	COMMAND	STATUS	ID	POS_L	POS_H	SUM



※マルチモードやブロードキャストIDを  
指定したときときは返事なし

POS

データ詳細				
桁	項目	設定範囲		内 容
		最小	最大	
1	SIZE	0x07		コマンドの総データバイト数
2	COMMAND	0x86		コマンド
3	STATUS	OPTIONにより変化		「4.オプション/ステータス詳細」参照
4	ID	送信IDと同じ		返信したデバイスのID
5	POS_L	0x00	0xFF	現在位置の下位バイト
6	POS_H	0x00	0xFF	現在位置の上位バイト
7	SUM	SIZE～POS_Hまでの総和の下位1バイト		

## ■ 使用例

ID2のデバイスを180度の位置に3000ms（3秒）で移動させる。

## 【送信フォーマット】

桁	項目	データ
1	SIZE	0x09
2	COMMAND	0x06
3	OPTION	0x00
4	ID1	0x02
5	POS(1)_L	0x50
6	POS(1)_H	0x46
7	TIME_L	0xB8
8	TIME_H	0x0B
9	SUM	0x6A

## 【返信フォーマット】

桁	項目	データ
1	SIZE	0x07
2	COMMAND	0x86
3	STATUS	0xFF
4	ID	0x02
5	POS_L	0xFF
6	POS_H	0xFF
7	SUM	0xFF

※0xFFは不定



## 4.オプション/ステータス詳細

「オプション」「ステータス」は、サーボの状態を取得することができる機能です。

「オプション」は読み出すデータを指定でき、それに対してサーボが「ステータス」を返信します。独立したコマンドではなく、各命令の送信、返信フォーマット内の一部として使用します。

※エラーステータスは、自動でクリアになりません。必ずオプションのCLEAR（7ビット）にて手動でクリアしてください。

### ■オプション

・オプションは、「エラーステータス」「システムステータス」「モーターステータス」「通信ステータス」「コマンドステータス」の5種類から読み取るステータスを指定することができます。

ビット	名称	機能	備考
0	STATUS SELECT	0~2bitを使って受け取る STATUSの種類を決定する	000: ERROR STATUSを返す
1			001: SYSTEM STATUSを返す
2			010: MOTOR STATUSを返す
			011: UART STATUSを返す
3	—		100: COMMAND STATUSを返す
4	—		
5	—		
6	—		
7	CLEAR	STATUSクリア	全てのSTATUSを0でクリア、返信コマンドにはクリア前のステータスが保存される

## ■ ステータス

- ・ オプションで指定されたステータスが返信されます。

## 【ERROR STATUS (OPTIONのSTATUS SELECTを"000"に指定したとき)】

- ・ 各エラーステータスの代表ビットです。

ビット	内 容	
	0	1
0	エラー無し	SYSTEM STATUSのいずれかのエラービットが1になっている時
1	エラー無し	MOTOR STATUSのいずれかのエラービットが1になっている時
2	エラー無し	UART STATUSのいずれかのエラービットが1になっている時
3	エラー無し	COMMAND STATUSのいずれかのエラービットが1になっている時
4	—	—
5	—	—
6	—	—
7	—	—

## 【SYSTEM STATUS (OPTIONのSTATUS SELECTを"001"に指定したとき)】

- ・ システム関連のエラービットです。

ビット	内 容	
	0	1
0	エラー無し	Watchdog Timerが起動したときに1になります
1	エラー無し	MCUのROMに保存されているデータに何らかの不都合があった場合に1になります。
2	エラー無し	メモリーに何らかの不具合があり、RAM割り当てに失敗したときに1になります。起動時に1度だけチェックします。
3	エラー無し	入力電圧が上限値を超えたか、下限値を下回った場合に1になります
4	エラー無し	MCU温度が上限値を超えた場合に1になります
5	エラー無し	AD変換に失敗したときに1になります
6	エラー無し	I2C通信に失敗したときに1になります (未使用)
7	エラー無し	SPI通信に失敗したときに1になります

※ エラーステータスはサーボモーターの種類により、無効なものがあります。

B3M-SB-1040A

SYSTEM:Watchdog Timer

B3M-SC-1040A,B3M-SC-1170A

SYSTEM:Watchdog Timer Motor:HallIC

## 【MOTOR STATUS (OPTIONのSTATUS SELECTを"010"に指定したとき)】

・モーター関係のエラービットです。

ビット	内 容	
	0	1
0	エラー無し	モーター温度が上限値を超えた場合に1になります
1	エラー無し	モーターロックが検知された場合に1になります
2	エラー無し	モーターに流れる電流が上限値を超えた場合に1になります
3	エラー無し	ブラシレスモーターのホールICに不具合があった場合に1になります
4	エラー無し	—
5	エラー無し	—
6	エラー無し	—
7	エラー無し	—

## 【UART STATUS (OPTIONのSTATUS SELECTを"011"に指定したとき)】

・通信関係のエラービットです。

ビット	内 容	
	0	1
0	エラー無し	フレミングエラー発生時に1になります
1	エラー無し	パリティエラー発生時に1になります
2	エラー無し	ブレークエラー発生時に1になります
3	エラー無し	オーバーランエラー発生時に1になります
4	エラー無し	—
5	エラー無し	—
6	エラー無し	—
7	エラー無し	—

## 【COMMAND STATUS (OPTIONのSTATUS SELECTを"100"に指定したとき)】

・コマンド関係のエラービットです。

ビット	内 容	
	0	1
0	エラー無し	コマンドのチェックサムが間違っている場合に1になります
1	エラー無し	コマンドのデバイス数が多すぎるあるいは少なすぎる場合に1になります
2	エラー無し	取得するデータ長さがアドレスを越えるほど長い場合に1になります
3	エラー無し	アドレスが指定範囲外だった場合に1になります
4	エラー無し	コマンド自身が間違っている場合に1になります
5	エラー無し	—
6	エラー無し	—
7	エラー無し	—

## 【CLR (OPTIONのCLEARビット)】

ビット	内 容	
	0	1
7	クリア無し	全てのSTATUSを0でクリア、返信コマンドにはクリア前のステータスが保存される

※ 各種エラーステータスは自動でクリアされません。

## 5.サーボの動作について

### ■サーボの動作モードについて

■サーボの動作モードは以下の3種類があります。

- ・Free・・・モーターに力を加えていない状態にします。
- ・Hold・・・モーターの電磁ブレーキをかける状態にします。
- ・Normal・・・各制御モードの目標値により制御を開始します。

■電源を入れるとサーボはFreeモードで起動します。

■各動作モードへの切り替えはアドレス0x28の動作モードで変更する事ができます。

### ■制御モードについて

サーボの制御方法は、位置制御モード・速度制御モード・電流（トルク）制御モード・フィードフォワードモードの4種類が有り、使用する用途により選択できます。

※全てのモードでPID制御を行っています。

制御モード (0x28)	目標値	軌道生成タイプ (0x29)	目標時間 (0x36)	制御パラメータ (0x5c~0x8c)
位置制御モード	目標位置 (0x2A・0x2B)	0:Normal・・・最速で回転します。		比例ゲイン 微分ゲイン 積分ゲイン 静止摩擦係数 動摩擦係数
		1:Even・・・目標時間を均等分割して始点から終点までを補完します（軌道生成）。	目標 移動時間	
		3:ThirdPoly・・・3次多項式を使って始点から終点までを補完します。		
		4:ForthPoly・・・4-1-4多項式補完法で始点から終点までを補完します。		
		5:FifthPoly・・・5次多項式を使って始点から終点までを補完します。		
速度制御モード	目標速度 (0x30・0x31)			
電流(トルク)制御モード	目標トルク (0x3C・0x3D)			
フィードフォワードモード	目標位置 (PWM) (0x2A・0x2B)			

## ■動作モード/制御モードの変更について

動作モード、制御モードの切り替えは、アドレス0x28を書き換えることによって設定します。

アドレス : 0x28	モード
BIT0	動作モード
BIT1	
BIT2	制御モード
BIT3	

### 【動作モード】

00:Normal:サーボモーターがアクティブになるモードです。

10:Free:サーボモーターが脱力するモードです。モーターは制御されていません。

11:Hold:サーボモーターが現在位置を保持するモードです。モーターは制御されていません。

### 【制御モード】

00:Position:位置制御モード

01:Speed:速度制御モード

10:Current:電流制御モード

11:Feed Forward:フィードフォワードモード

フィードフォワード時は目標位置にPWMのカウント値を入力すると、モーターが回転します。

※制御モードは、動作モードをNormalにしない限り動作しません。

※**制御モードを切り替えるときは、一度動作モードをFreeにしてください。**

**また、PIDゲインプリセット (0x5C) を変更が必要です。**

詳細は「8.メモリーマップ詳細」の0x28をご一読ください。

## ■動作ゲインのプリセットについて

出荷時に動作ゲインのプリセットとして下記のデータが用意されています。

プリセット0 : 位置制御

プリセット1 : 速度制御

プリセット2 : 電流 (トルク) 制御

WRITEコマンドによりアドレス0x5Cのゲインプリセット番号を変えると即座にゲインが変更され、サーボモーターの動作に反映されます。位置制御・速度制御・電流 (トルク) 制御ではゲインが違うので、制御モードを変更する場合は必ずゲインまたはゲインプリセットを変更してください。

### ■ 位置制御モードとは

- ・ 指定した位置（角度）に移動します。
- ・ 目標位置と現在位置を元に制御を行います。
- ・ 原点(中央値オフセットで設定した値)から $\pm 320$ 度の範囲で制御できます。  
 ※Free, Holdにすると現在位置 $\pm 320$ 度に制限され、それ以上の角度の場合は、角度は保障されません。  
 Normalモードに戻す前に一度現在値を取得してから角度を指定してください。  
 ※リセットがかかった場合、 $\pm 180$ 度の範囲に無い場合は、強制的に $\pm 180$ 度の位置に変換されます。
- ・ 目標位置を指定するにはSET POSITIONコマンドで指定する方法と、目標位置(0x2A・0x2B)にデータを書き込む方法の二通りがあります。
- ・ 軌道生成タイプは(0x29)で指定できます。
- ・ 軌道生成タイプの目標時間(0x36)を変更する事により、サーボの動きを変化させる事ができます。
- ・ ゲインの初期状態は、ゲイン0に保存されています(プリセット番号アドレス(0x5C)で指定)。

### ■ 速度制御モードとは

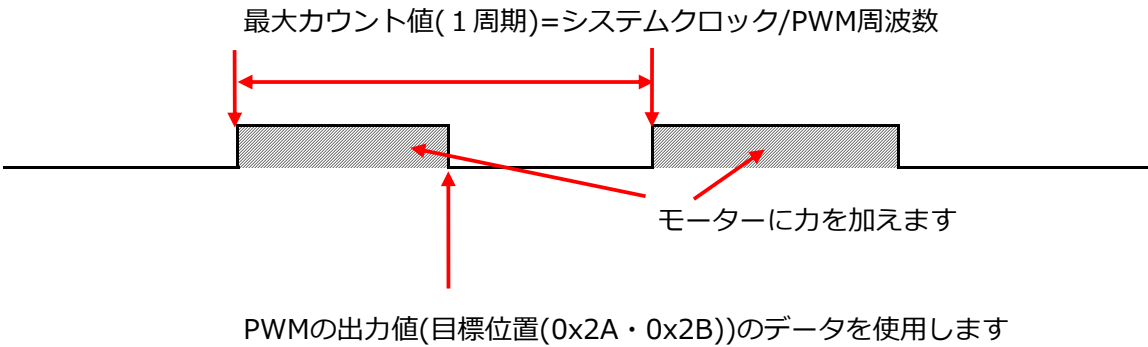
- ・ 指定した一定速度で、回転を行います。
- ・ 目標速度と現在速度を元に制御を行います。
- ・ 目標速度を変更するには、(0x30・0x31)の値を変更します。
- ・ 現在位置(0x2C・0x2D)を取得出来ますが、 $\pm 180$ 度の範囲内になります。
- ・ 累積の回転角を取得する場合、エンコーダーのトータルカウント数(0x52～0x55)から逆算する事ができます。
- ・ ゲインの初期状態は、ゲイン1に保存されています(プリセット番号アドレス(0x5C)で指定)。

### ■ 電流(トルク)制御モードとは

- ・ 目標トルクから電流値を計算し、電流値をフィードバックする制御方法です。
- ・ 指定した一定の電流になる様に制御します。
- ・ 電流(仕事量)とトルクには一定の関係があるため、一定のトルクで力を加える事ができます。
- ・ 一定の力を出力しながら無限回転します。
- ・ 目標トルクを変更するには、(0x3C・0x3D)の値を変更します。
- ・ モータの温度等により、正確なトルクを出力できない可能性があります。
- ・ 累積の回転角を取得する場合、エンコーダーのトータルカウント数(0x52～0x55)から逆算する事ができます。
- ・ ゲインの初期状態は、ゲイン2に保存されています(プリセット番号アドレス(0x5C)で指定)。

■ フィードフォワードモードとは

- ・ モーターの出力を直接指定します。
- ・ 累積の回転角を取得する場合、エンコーダーのトータルカウント数(0x52～0x55)から逆算する事ができます。



- ・ 周波数が8kHzの時のカウント最大値は6250なので、フィードフォワードモードの場合、目標位置(PWMカウ

周波数 (Hz)	カウント最大
1000	50000
1250	40000
1562	32010.24328
1563	31989.76328
1600	31250
2000	25000
2500	20000
4000	12500
5000	10000
8000	6250
10000	5000
12500	4000
20000	2500
25000	2000
40000	1250
50000	1000

設定できるデットライン

(目標位置の最大値となります)

## 6. B 3 Mサーボモータを動かしてみる

### ■B3Mサーボモータを動かすには

B 3 Mサーボモータを動かすには、下記の手順が必要です。

- ① サーボの状態をFreeモードにします。
- ② 制御モードを変更します。
- ③ 制御モードに応じてゲインを変更します。
- ④ サーボの状態をNomalモードにします。
- ⑤ 目標値を設定します。

ここでは、位置制御モードでのサーボモータの動かし方を説明します。

※今回は、ID=0のB3Mのサーボモータで説明します。

### ■パケットの名称について

【SIZE】 . . . . . データ長  
 【CMD】 は【COMMAND】 の略式 . . . コマンド  
 【OP】 は【OPTION】 の略式 . . . . . オプション  
 【STTS】 は【STATUS】 の略式 . . . ステータス  
 【ID】 . . . . . ID番号  
 【DATA】 は【S-DATA】 の略式 . . . データ  
 【ADR】 は【ADDRESS】 の略式 . . . アドレス  
 【CNT】 は【COUNT】 の略式 . . . カウント（サーボのID数（台数））  
 【SUM】 . . . . . チェックサム

### ①サーボの状態をFreeモードにします

まず初めにサーボモータを急に動作しないように脱力させます。モータは制御されていません。

#### 【動作モードをFreeモードにしてコマンド送信】

WRITEコマンドで、0x28番地に0x02の値を書き込みます。

Tx	1	2	3	4	5	6	7	8
	SIZE	CMD	OP	ID	DATA	ADR	CNT	SUM
	0x08	0x04	0x00	0x00	0x02	0x28	0x01	0x37

アドレス : 0x28				制御モード		動作モード	
BET7	BET6	BET5	BET4	BET3	BET2	BET1	BET0
0	0	0	0	0	0	1	0

【動作モード】

00:Normal

**10:Free**

11:Hold



(返信データ)

	1	2	3	4	5
<b>Rx</b>	SIZE 0x05	CMD 0x84	STTS 0x00	ID 0x00	SUM 0x89

※上記はサーボの動作にエラーがない場合。エラーを検出した場合はSTATUSが変化します。

※以下のWRITEコマンドも、送信したDATA数に関係なく5BYTEのデータがサーボから返信されます。

## ②制御モードを変更します

次に、制御モードを位置制御モードにします。

### 【制御モードを位置制御モードにしてコマンド送信】

WRITEコマンドで、0x28番地に0x02の値を書き込みます。

※動作モードの0x02と制御モードの0x00を論理和(OR)した値になります。

※下記のコマンドの内容は以前のFreeに設定するコマンドと同じですが、制御手順として掲載します。

	1	2	3	4	5	6	7	8
<b>Tx</b>	SIZE 0x08	CMD 0x04	OP 0x00	ID 0x00	DATA 0x02	ADR 0x28	CNT 0x01	SUM 0x37

アドレス : 0x28				制御モード		動作モード	
BET7	BET6	BET5	BET4	BET3	BET2	BET1	BET0
0	0	0	0	0	0	1	0

※動作モードはFreeの状態です。

【制御モード】

**00:Position:位置制御モード**

01:Speed:速度制御モード

10:Current:電流制御モード

11:Feed Forward:フィードフォワードモード

## ■軌道生成の設定を行います

位置制御モードに変更してから軌道生成を設定します。

(軌道生成タイプをNormalのまま使用の場合はスキップします)

### 【軌道生成タイプをEvenにしてコマンド送信】

WRITEコマンドで、0x29番地に0x01の値を書き込みます。

Tx	1	2	3	4	5	6	7	8
	SIZE	CMD	OP	ID	DATA	ADR	CNT	SUM
	0x08	0x04	0x00	0x00	0x01	0x29	0x01	0x37

【軌道生成タイプ（アドレス：0x29）】

0:Normal・・・最速で回転します。

**1:Even**・・・目標時間を均等分割して始点から終点までを補完します（軌道生成）。

3:ThirdPoly・・・3次多項式を使って始点から終点までを補完します。

4:ForthPoly・・・4-1-4多項式補完法で始点から終点までを補完します。

5:FifthPoly・・・5次多項式を使って始点から終点までを補完します。

## ③制御モードに応じてゲインを変更します

ゲインを設定します。出荷時にはプリセットNo.0に位置制御用パラメータがセットされています。

### 【ゲインをプリセットNo.0にしてコマンド送信】

WRITEコマンドで、0x5C番地に0x00の値を書き込みます。

Tx	1	2	3	4	5	6	7	8
	SIZE	CMD	OP	ID	DATA	ADR	CNT	SUM
	0x08	0x04	0x00	0x00	0x00	0x5C	0x01	0x69

【PIDゲインのプリセット番号（アドレス：0x5C）】

**0：位置制御**

1：速度制御

2：電流（トルク）制御

## ④サーボの状態をNormalモードにします

サーボの状態をNormalモードにすると、制御が始まりサーボが動きだします。

## 【動作モード：Normalの設定コマンド送信】

WRITEコマンドで、0x28番地に0x00の値を書き込みます。

Tx	1	2	3	4	5	6	7	8
	SIZE	CMD	OP	ID	DATA	ADR	CNT	SUM
	0x08	0x04	0x00	0x00	0x00	0x28	0x01	0x35

アドレス：0x28				制御モード		動作モード	
BET7	BET6	BET5	BET4	BET3	BET2	BET1	BET0
0	0	0	0	0	0	0	0

## 【制御モード】

00:位置制御モード

01:速度制御モード

10:電流制御モード

11:フィードフォワードモード

## 【動作モード】

00:Normal

10:Free

11:Hold

## ⑤目標値を設定します

目標値を変更する事によりサーボの位置を変更できます。

## 【目標位置の設定コマンド送信】

目標位置を設定する方法は、WRITEコマンドで指定の0x2A/0x2B番地に目標位置を書き込む方法とSET POSITIONコマンドで目標位置、目標移動時間を指定する方法の2種類あります。

## ■WRITEコマンドで目標位置を指定する場合

例) 目標位置を180度に設定・・・18000(0x4650)

WRITEコマンドで、0x2A番地に0x50の値を、0x2B番地に0x46の値を書き込みます。

アドレスは先頭アドレスの0x2Aを指定します。

Tx	1	2	3	4	5	6	7	8	9
	SIZE	CMD	OP	ID	DATA1	DATA2	ADR	CNT	SUM
	0x09	0x04	0x00	0x00	0x50	0x46	0x2A	0x01	0xCE

リトルエンディアン(2バイト)  
(下位バイトからデータをおくります)

## ■SET POSITIONコマンドで目標位置、目標移動時間を指定する場合

例) 目標位置を180度に設定・・・18000(0x4650)

目標移動時間を3000ms (3秒) に設定・・・3000(0x0BB8)

	1	2	3	4	5	6	7	8	9
<b>Tx</b>	SIZE	CMD	OP	ID	DATA1	DATA2	DATA3	DATA4	SUM
	0x09	0x06	0x00	0x00	0x50	0x46	0xB8	0x0B	0x68

目標位置(POS)(2バイト)  
リトルエンディアン

目標移動時間(TIME)(2バイト)  
リトルエンディアン

(返信データ (SET POSITION) )

	1	2	3	4	5	6	7
<b>Rx</b>	SIZE	CMD	STTS	ID	DATA1	DATA2	SUM
	0x07	0x86	0x00	0x00	0xXX	0xXX	0xXX

※上記はサーボの動作にエラーがない場合。

コマンド送信時の位置データが返信データに含まれます。

※目標移動時間は、軌道生成タイプ (0x29) を「Even」「ThirdPoly」「ForthPoly」「FifthPoly」に設定したときのみ有効です。  
「Noemal」の場合は、目標移動時間の設定値に関係なく最速で動きます。

## 7.メモリーマップについて

B3Mには、データを処理する為のメモリーがあります。

このメモリーのことを『メモリーマップ』と呼びます。

メモリーには、電源を切るとデータが消えてしまう『RAM領域』と、電源を切ってもデータを保持できる『ROM領域』があります。

### ■単位表記について

・メモリー領域の単位についての説明です。

単位	内 容
bps	データ転送レートの単位
deg	角度の単位（度）
℃	温度の単位
mA	電流の単位
mV	電圧の単位
deg/sec	1秒間で移動する角度
ns	1,000,000,000分の1秒( $10^{-9}$ 秒)
ms	1,000分の1秒
sec	秒の単位
Hz	周波数の単位

### ■属性表記について

・メモリー領域の属性についての説明です。

属性	内 容
RW	(Read Write) 読み込み・書き込みが出来ます。 書き換えるとシステムの挙動が変化するパラメーターです。
RO	(Read) 読み込みのみ出来ます。 システム側でデータを上書きするパラメーターです。
#	備考に#マークがついているものは書き込み、再起動後に有効になります。 有効にするには再起動前に必ずROMにSAVEすること。
x	100xなどと記載されている場合は実際の100倍の数値を使用すること。 例えばサーボの角度で90度を指定する場合は9000とすること。

## ■ 型表記について

・メモリー領域の型についての説明です。

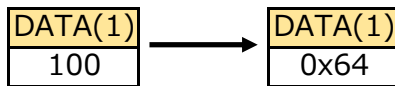
型	内 容
char	符号付きchar型変数、1byte
byte	符号無しchar型変数、1byte
short	符号付きshort int型変数、2byte (リトルエンディアン)※
ushort	符号無しshort int型変数、2byte (リトルエンディアン)※
long	符号付きlong int型変数、4byte (リトルエンディアン)※
ulong	符号無しlong int型変数、4byte (リトルエンディアン)※

※リトルエンディアンとは、2バイト以上のデータ量を持つ数値データを1バイトごとに分割するが、これを最下位のバイトから順番に記憶する方式です。

- データの符号には、符号付き(char・short・long)と符号無し(byte・ushort・ulong)があります。

例)1バイトのデータの場合

- データが100の時



- データが-100の時

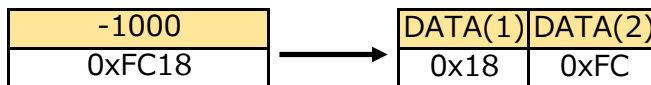


例)2バイトのデータの場合

- データが1000の時

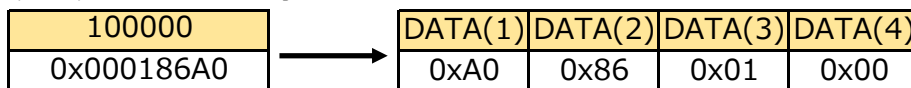


- データが-1000の時

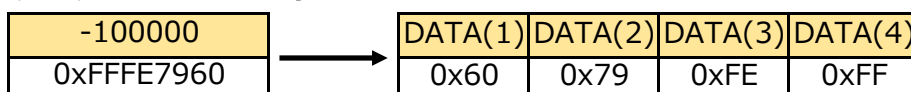


例)4バイトのデータの場合

- データが100000の時



- データが-100000の時



## ■メモリー領域について

・メモリーはSYSTEM領域、サーボパラメータ領域、制御パラメータ領域、STATUS領域、バージョン情報領域、システム初期値領域に分かれています。

## 【システム情報が格納されています】

SYSTEM領域						
アドレス	バイト数	内 容	単位	型	属性	注記
0x00	1	ID番号	—	byte	RW	*1
0x01	4	ボーレート、通信速度	bps	ulong	RW #	*2
0x05	2	最小位置制限	100x deg	short	RW	*3
0x07	2	最大位置制限	100x deg	short	RW	*3
0x09	2	中央値オフセット	100x deg	short	RW	*3
0x0B	2	MCU温度リミット	100x °C	short	RW	
0x0D	1	MCU温度制限時のPWM制限率	%	byte	RW	
0x0E	2	モーター温度リミット	100x°C	short	RW	
0x10	1	モーター温度制限時のPWM制限率	%	byte	RW	
0x11	2	電流制限値	mA	ushort	RW	
0x13	1	電流制限時のPWM制限率	%	byte	RW	
0x14	1	モーターロックと認識されるまでの時間	100ms	byte	RW	
0x15	1	ロックと認識される最大出力からの割合	%	byte	RW	
0x16	1	ロック検知時の脱力度合い	%	byte	RW	*4
0x17	2	有効入力電圧最小値	mv	ushort	RW	*5
0x19	2	有効入力電圧最大値	mv	ushort	RW	*5
0x1B	1	PWM制限	%	byte	RW	
0x1C	2	不感帯の幅	100x deg	ushort	RW	
0x1E	2	Reserved	—	short	—	
0x20	2	Reserved	—	short	—	
0x22	1	モーター回転特性CW方向の割合	—	byte	RW #	
0x23	1	モーター回転特性CCW方向の割合	—	byte	RW #	

## 【サーボを制御する情報が格納されています】

サーボパラメータ領域						
アドレス	バイト数	内 容	単位	型	属性	注記
0x27	1	サーボオプション	—	byte	RW #	*6
	ビット	内 容	0		1	
	0	—				
	1					
	2					
	3					
	4					
	5					
	6	クローン(Ver.1.0.2.0より有効)	返信データ有り		返信データ無し	
	7	リバーサ(Ver.1.0.2.0より有効)	OFF		ON	
0x28	1	Normal/FREE/Holdモード切り替え	—	byte	RW	*6
	ビット	内 容	0		1	
	0	動作モード	00:ノーマル			
	1		10:フリー			
	2	制御モード(※)	11:ホールド			
	3		00:位置制御 01:速度制御			
	4	—	10:トルク(電流)制御			
	5	—	11:フィードフォワード			
	6	Servo(Ver.1.0.1.0まで有効)	00:ノーマル			
	7		01:クローン			
0x29	1	軌道生成タイプ	—	byte	RW	*6
	ビット	内 容	0		1	
	0	Trajectory	0: 直線移動 (加速・減速区間無し)			
	1		1: 均等分割式 (時間で等間隔移動とする)			
	2		3: 3次多項式補間 (加速区間を3次多項式で生成、最大速度になってから停止します)			
	3	—	4: 4-1-4多項式補間 (加速・等速・減速区間を4次多項式と1次式の組み合わせで生成、等速度区間は固定)			
	4	—	5: 5次多項式補間 (加速・減速区間を5次多項式で生成、速度0で停止します)			
	5	—				
	6	—				
	7	—				

※ フィードフォワード制御では目標位置(アドレス0x2A)にPWMのデューティ比をPWMカウント数(符号で向きが変わる)で入力します。ただし、最大カウント数=システムクロック÷PWM周期で計算され、この値を越えて設定しても、上限値に自動修正されます。



サーボパラメータ領域						
0x2A	2	目標位置	100x deg	short	RW	
0x2C	2	現在位置	100x deg	short	RO	
0x2E	2	1サンプリング回前の位置	100x deg	short	RO	
0x30	2	目標速度	100x deg/sec	short	RW	*7
0x32	2	現在速度	100x deg/sec	short	RO	
0x34	2	1サンプリング回前の速度	100x deg/sec	short	RO	
0x36	2	目標時間	ms	ushort	RW	
0x38	2	コマンド実行中時間	ms	ushort	RO	*8
0x3A	2	起動時からの経過時間	sec	ushort	RO	*8
0x3C	2	目標トルク	mN.m	short	RW	*7
0x3E	4	システムのクロック	20x ns	ulong	RO	
0x42	2	サンプリングタイム	20x ns	ushort	RO	
0x44	2	現在のMCU温度	100x℃	short	RO	
0x46	2	現在のモーター温度	100x℃	short	RO	
0x48	2	現在の(負荷)電流値	mA	short	RO	
0x4A	2	現在の入力電圧値	mV	ushort	RO	
0x4C	2	現在のPWMデューティ比		ushort	RW	
0x4E	2	PWM周期	Hz	ushort	RW #	
0x50	2	エンコーダーの現在値	—	ushort	RO	
0x52	4	エンコーダーのトータルカウント数	—	long	RW	
0x56	1	ホールICの状態	—	byte	RO	

## 【ゲイン情報が格納されています】

制御パラメータ領域						
アドレス	バイト数	内 容	単位	型	属性	注記
0x5C	2	PIDゲインのパターンを変更する	—	ushort	RW	
0x5C	1	PIDゲインのプリセット番号		byte		
0x5D	1	Reserved	—	byte		
0x5E	4	比例ゲイン0	100x	ulong	RW	
0x62	4	微分ゲイン0	100x	ulong	RW	
0x66	4	積分ゲイン0	100x	ulong	RW	
0x6A	2	静止摩擦係数0	—	ushort	RW	
0x6C	2	動摩擦係数0	—	ushort	RW	
0x6E	4	比例ゲイン1	100x	ulong	RW	
0x72	4	微分ゲイン1	100x	ulong	RW	
0x76	4	積分ゲイン1	100x	ulong	RW	
0x7A	2	静止摩擦係数1	—	ushort	RW	
0x7C	2	動摩擦係数1	—	ushort	RW	
0x7E	4	比例ゲイン2	100x	ulong	RW	
0x82	4	微分ゲイン2	100x	ulong	RW	
0x86	4	積分ゲイン2	100x	ulong	RW	
0x8A	2	静止摩擦係数2	—	ushort	RW	
0x8C	2	動摩擦係数2	—	ushort	RW	

## 【ステータス情報が格納されています】

STATUS領域						
アドレス	バイト数	内 容	単位	型	属性	注記
0x9D	1	エラーステータス	—	byte		
0x9E	4	エラーなどのステータス	—	ulong		
0x9E	1	システムエラー	—	byte		
0x9F	1	モーターステータス異常	—	byte		
0xA0	1	UART受信エラー	—	byte		
0xA1	1	コマンドエラーなど	—	byte		

## 【バージョン情報が格納されています】

バージョン情報領域						
アドレス	バイト数	内 容	単位	型	属性	注記
0xA2	4	モデル区分	—	ulong	RO	*9
0xA2	1	電圧区分	—	char	RO	
0xA3	1	モデルバージョン	—	byte	RO	*9
0xA4	1	トルク番号	—	byte	RO	*9
0xA5	1	ケース番号	—	byte	RO	*9
0xA6	4	モデル種類	—	ulong	RO	*9
0xA6	1		—		RO	
0xA7	1		—		RO	
0xA8	1	モータータイプ	—	char	RO	*9
0xA9	1	デバイスタイプ	—	char	RO	*9
0xAA	4	ファームウェア情報	—	ulong	RO	
0xAA	1	ビルド番号	—	byte	RO	
0xAB	1	リビジョン番号	—	byte	RO	
0xAC	1	マイナーバージョン	—	byte	RO	
0xAD	1	メジャーバージョン	—	byte	RO	

## 【基準値と実際の位置のずれを補正します】

システム初期値領域						
0xAE	2	絶対00位置からのエンコードのずれ	—	short	RW #	
0xB0	72	相対位置でのエンコードの誤差補正	—	short*36	RW #	
0xB0	2	相対位置01番目の誤差補正	—	short	RW #	
⋮	⋮	⋮	⋮	⋮	⋮	
0xF6	2	相対位置36番目の誤差補正	—	short	RW #	

※システム初期値領域(0xAE～0xF7)は工場出荷時に設定済みですので変更しないでください。

## 注記

- \*1 255(ブロードキャストID)を指定すると接続されているデバイス全てに対する命令となる。
- \*2 115200/1000000/1250000/1500000/2000000/3000000が設定可能。  
上限値または下限値を超えた値を設定すると上限値または下限値に修正される。
- \*3 リバースモード時はノーマルモードと同じ位置に設定され、リバースの影響を受けません。
- \*4 ロック検知タイマー開始まで50ms程度時間がかかります。
- \*5 電流不足で電圧が急に下がった場合も検知されます。
- \*6 1byte単位でしかアクセスできません。書く場合はすべてのbitを設定してください。
- \*7 スペック以上の値は出力できません。
- \*8 最大±0.5%の誤差があります。
- \*9 B3Mシリーズ仕様書を参照のこと。

全般 理論上の最大・最小値が指定値を越えて入力できる場合でも、指定値を超えて入力しないでください。

## 8.メモリーマップ詳細について

ここでは、個々のメモリーアドレスの機能について説明します。

### ■SYSTEM領域

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
ID番号	0x00	byte	－	0	0	255	RW
サーボのID番号。 例) ID番号を12に設定・・・12(0x0c) ※255を指定すると接続されているデバイス全てに対する命令となります。							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
ボーレート、通信速度	0x01	ulong	bps	1500000	115200	3000000	RW #
115200/1000000/1250000/1500000/2000000/3000000が設定可能。 例) ボーレートを1500000bpsに設定・・・1500000(0x0016E360) ※上限値または下限値を超えた値を設定すると上限値または下限値に修正されます。							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
最小位置制限	0x05	short	100×deg	-32000	-32000	32000	RW
最大位置制限	0x07	short	100×deg	32000	-32000	32000	RW

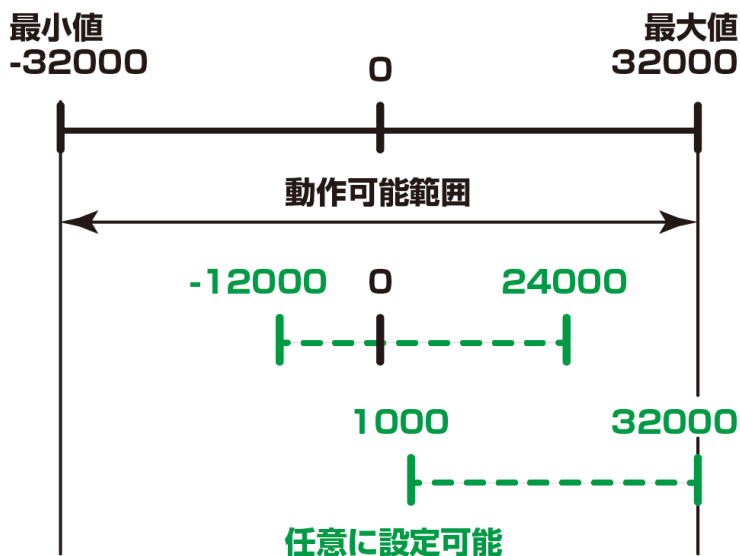
可動範囲の最小値・最大値。

例) 最小値を-30000に設定・・・-30000(0x8AD0)

例) 最大値を 30000に設定・・・ 30000(0x7530)

※最小値・最大値が不正な場合の動作保証はしません。

※リバースモード時はノーマルモードと同じ位置に設定され、リバースの影響を受けません。



名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
中央値オフセット	0x09	short	100×deg	0	-18000	18000	RW
中心位置（取り付け位置オフセット）。 例）オフセットを30度に設定・・・3000(0x0bb8) ※値が不正な場合の動作保証はしません。 ※リバースモード時はノーマルモードと同じ位置に設定され、リバースの影響を受けません。							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
MCU温度リミット	0x0B	short	100×C	8000	-32768	32767	RW
MCUの温度が上がりすぎたときにエラーを発生させる温度閾値。 指定温度を超えると出力トルクに制限がかかります。 各種制限値（温度制限、電流制限など）を越えて出力制限がかかった場合は正常な挙動にはなりません。 例）リミットを50度に設定・・・5000(0x1388)							

名称（説明）	アドレス	型	単位	初期値	最大値	最小値	属性
MCU温度制限時のPWM制限率	0x0D	byte	%	0	0	100	RW
MCU温度リミットエラーが発生したときにモーターパワーを指定したパーセンテージにします。 例）出力トルクを80%制限に設定・・・80(0x50)							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
モーター温度リミット	0x0E	short	100×C	10000	-32768	32767	RW
モーターの温度が上がりすぎたときにエラーを発生させる温度閾値。 指定温度を超えると出力トルクに制限がかかります。 制限値を越えて出力制限がかかった場合は正常な挙動にはなりません。 例）リミットを50度に設定・・・5000(0x1388)							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
モーター温度制限時のPWM制限率	0x10	byte	%	0	0	100	RW
モーター温度リミットエラーが発生したときにモーターパワーを指定したパーセンテージにします。							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
電流制限値	0x11	ushort	mA	※	0	65535	RW
モーターに電流が流れすぎたときにエラーを発生させる閾値。							

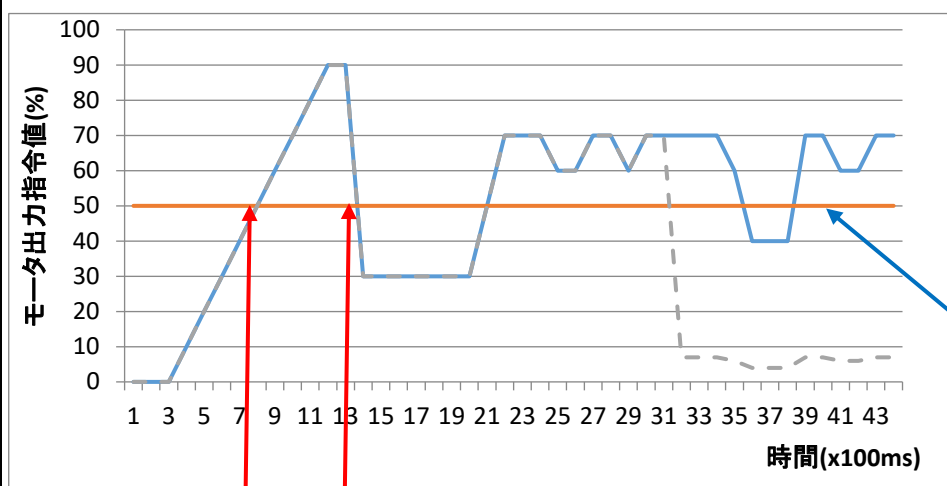
名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
電流制限時のPWM制限率	0x13	byte	%	0	0	100	RW
電流制限値エラーが発生したときにモーターパワーを指定したパーセンテージにします。 例）出力トルクを80%制限に設定・・・80(0x50)							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
モーターロックと認識されるまでの時間	0x14	byte	100ms	0	0	255	RW
モーターロックを検知してからエラーを発生させるまでの時間。							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
ロックと認識される最大出力からの割合	0x15	byte	%	50	0	100	RW
最高出力に対する割合で、この値がモーターロックと認識されるまでの時間だけ続くとロックと検知される。							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
ロック検知時の脱力度合い	0x16	byte	%	100	0	100	RW
モーターロックエラーが発生したときにモーターパワーを指定したパーセンテージにします。 例）出力トルクを90%制限に設定・・・90(0x5A)							

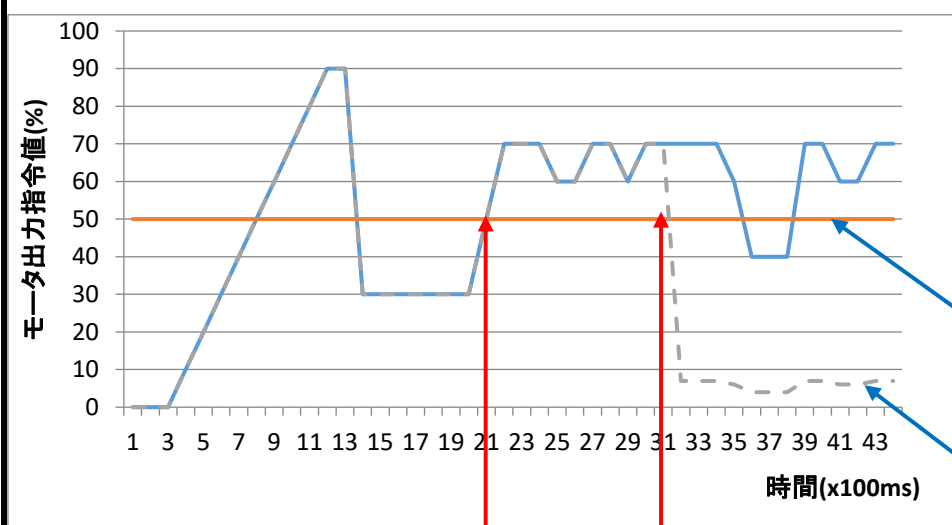
モーターロックの機能が動作しない時と、動作した時のタイミングチャートを図解します。



ロック検知開始

ロック検知

ロック検知開始してからロック終了時間が短い為、ロック機能が動作しない(0x14)



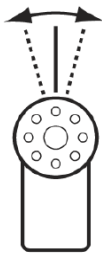
ロック検知開始

ロック検知終了

ロック検知開始してからエラー認識する時間を越えた(0x14)

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
有効入力電圧最小値	0x17	ushort	mv	5500	0	65535	RW
有効入力電圧最大値	0x19	ushort	mv	18000	0	65535	RW
サーボの入力電圧が指定した電圧以下又は電圧以上になったときにモーターを完全脱力します。 例) 最小値を 5Vに設定・・・5000(0x1388) 例) 最大値を13Vに設定・・・13000(0x32C8) ※電流不足で電圧が急に下がった場合も検知されます。							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
PWM制限	0x1B	byte	%	100	0	100	RW
モーターを駆動するPWMデューティに上限をつけます。 例) PWM制限を90%に設定・・・90(0x5A)							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
不感帯の幅	0x1C	ushort	100×deg	15	0	65535	RW
動作不感帯（偏差、静定値誤差）。 例) 幅を5度に設定・・・500(0x01F4) ※値を小さくしすぎると振動的になる恐れがあります。 							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
モーター回転特性CW方向の割合	0x22	byte	%	100	0	100	RW #
モーターの時計回り方向のPWMデューティの上限値。 例) PWM値の割合が90%に設定・・・90(0x5A)							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
モーター回転特性CCW方向の割合	0x23	byte	%	100	0	100	RW #
モーターの反時計回り方向のPWMデューティの上限値。 例) PWM値の割合が80%に設定・・・80(0x50)							



## ■サーボパラメータ領域

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性				
サーボオプション	0x27	byte	－	0			RW #				
<table><tr><td>BIT6</td><td>クローン</td></tr><tr><td>BIT7</td><td>リバース</td></tr></table>								BIT6	クローン	BIT7	リバース
BIT6	クローン										
BIT7	リバース										

動作モード

リバース

- ・サーボを取り付けた時、指令値と逆方向に動かしたいときにこのモードにします。
- ・中央値オフセット(0x09-0x0A)や最大位置制限(0x07-0x08)、最小位置制限(0x05-0x06)はリバースの影響を受けず、ノーマル時の指定範囲内で動作します。
- ・このモードにする場合、7bit目を1にし、ROMに保存、再起動を行います。

クローン

- ・全く同じ動作にしたいサーボがある場合、IDを同じにし、このモードにします。
- ・このモードにすると、返信データが無いため、Maneger等で値を変更しても、servo側は値は変更されていますが、Maneger側でエラーとして認識される場合があります。
- ・このモードにする場合、6bit目を1にし、ROMに保存、再起動を行います。

制御モード（無限回転での制御）

- ・B3Mサーボモーターは位置制御モード時以外は無限回転をします。
- ・B3Mサーボモータは360度読み取れる磁気式エンコーダを搭載しています。
- ・位置制御以外のモードでは現在位置を取得すると原点位置より±180度になります。
- ・積算角度を取得する場合、動作中にアドレス0x50のエンコーダー現在値（ushort）、またはアドレス0x52のエンコーダーのトータルカウント数をもとに算出することができます。
- ・B3Mサーボモーターのエンコーダーは12bitですので、1回転（360度）あたり4096の値を取得できます。トータルカウント数はこの累積値を計算したもので、-2^31～2^31-1までカウントできます。
- ・トータルカウント数に任意の値を入れることも可能で、カウントをリセットする場合は0を入れてください。車輪などに使用した場合、滑りは検知できませんので距離計算をする場合は滑りに注意してください。

※Ver.1.0.2.0より有効。

※1byte単位でしかアクセスできません。書く場合はすべてのbitを設定してください。

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性									
Normal/FREE/Holdモード切り替え	0x28	byte	－	0x02			RW									
<table><tr><td>BIT0</td><td rowspan="2">動作モード</td></tr><tr><td>BIT1</td></tr><tr><td>BIT2</td><td rowspan="2">制御モード</td></tr><tr><td>BIT3</td></tr><tr><td>BIT6</td><td rowspan="2">サーボオプション</td></tr><tr><td>BIT7</td></tr></table>								BIT0	動作モード	BIT1	BIT2	制御モード	BIT3	BIT6	サーボオプション	BIT7
BIT0	動作モード															
BIT1																
BIT2	制御モード															
BIT3																
BIT6	サーボオプション															
BIT7																

動作モード

00:Normal:サーボモーターがアクティブになるモードです。

10:Free:サーボモーターが脱力するモードです。モーターは制御されていません。

11:Hold:サーボモーターが現在位置を保持するモードです。モーターは制御されていません。

制御モード

00:Position:位置制御モード。

01:Speed:速度制御モード。

10:Current:電流制御モード。

11:Feed Forward:フィードフォワードモード。

フィードフォワード時は目標位置にPWMのカウント値を入力すると、モーターが回転します。

※制御モードはRun ModeをNormalにしない限り動作しません。

サーボオプション(※Ver1.0.1.0まで有効 0x27に移動しました)

00:Normal:通常動作オプション

01:Clone:どの命令に対しても返事をしません。同一IDのサーボを同時に操作するときを使用します。

10:Reverse:動作方向を逆向きにします。位置制御モード時のみ有効です。

※速度制御モードにするためには、いったん位置制御モードに切り替え、現在位置を確定してからモードを切り替えて下さい。

※制御モードの切り替え時は必ずFREE、HOLDへ移行して初期値をセットしてからモードの切り替えを行ってください。

※モード切り替え時はゲインプリセット変更またはゲインの調整を行ってください。

※1byte単位でしかアクセスできません。書く場合はすべてのbitを設定してください。

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性				
軌道生成タイプ	0x29	byte	－	0	0	255	RW				
<table><tr><td>BIT0</td><td rowspan="3">軌道生成タイプ</td></tr><tr><td>BIT1</td></tr><tr><td>BIT2</td></tr></table>								BIT0	軌道生成タイプ	BIT1	BIT2
BIT0	軌道生成タイプ										
BIT1											
BIT2											

0:Normal・・・最速で回転します。

1:Even・・・目標時間を均等分割して始点から終点までを補完します（軌道生成）。

3:ThirdPoly・・・3次多項式を使って始点から終点までを補完します。

4:ForthPoly・・・4-1-4多項式補完法で始点から終点までを補完します。

5:FifthPoly・・・5次多項式を使って始点から終点までを補完します。

※動作モードがNormalで位置制御モード時のみ有効です。

Even、ThirdPoly、ForthPoly、FifthPolyモードで目標時間を0にした場合はNormalの動作となります。

軌道生成タイプ別のサーボの軌道をグラフにしました。

目標時間=2000ms 目標位置=90deg

図1：時間に対してのサーボの移動軌道

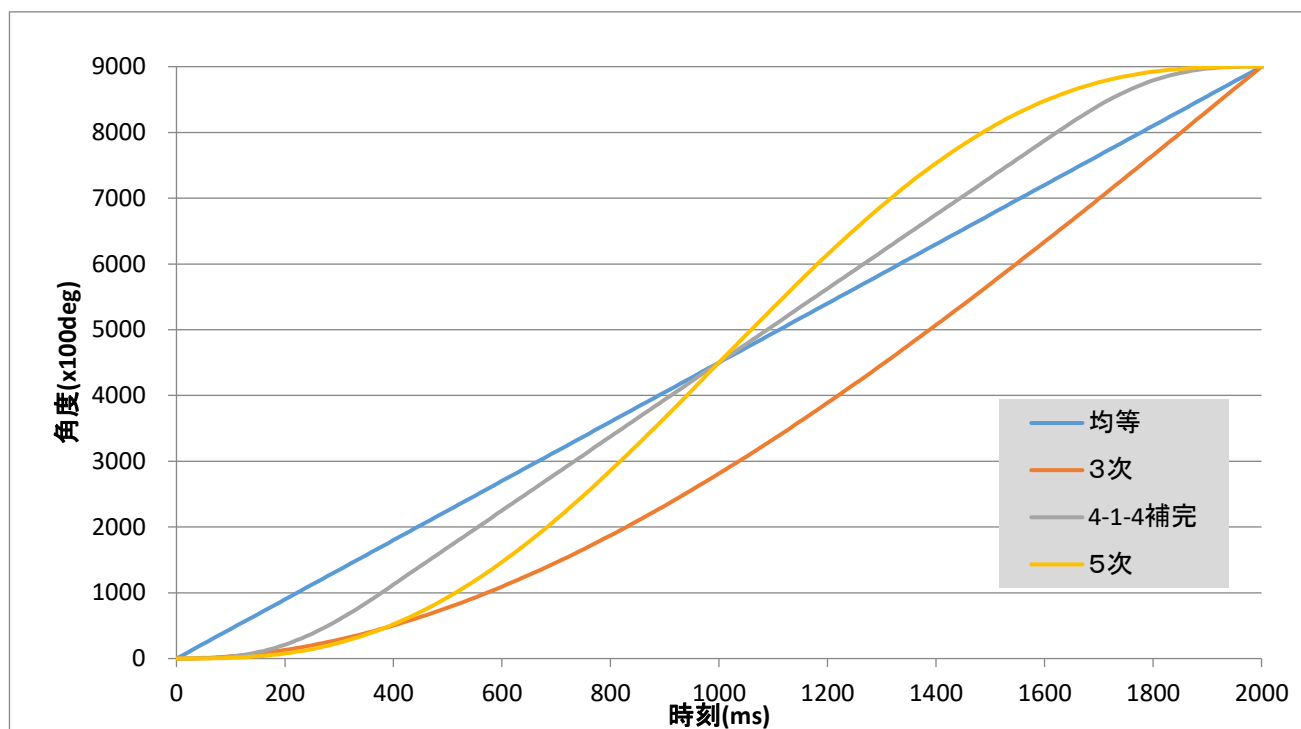
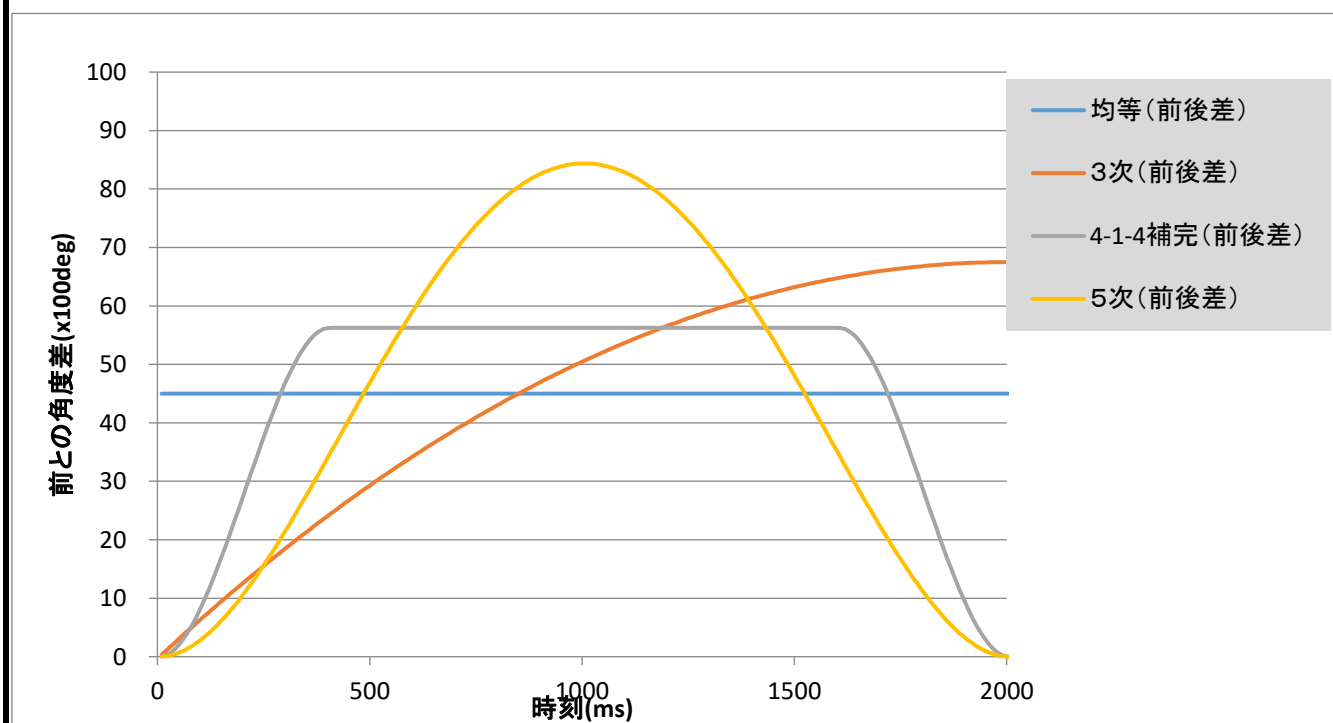


図2：時間に対してのサーボの角度差



名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
目標位置	0x2A	short	100×deg	0	-32000	32000	RW
<p>目標位置を書き込みます。位置制御モードのときは位置制御となります。</p> <p>フィードフォワードモードのときはPWMカウント値となります。</p> <p><u>位置制御</u></p> <ul style="list-style-type: none"> <li>・目標位置と現在位置との差を元に入力値（操作量）が決まる制御方法です。PID制御をしていますので、後述のPゲイン、Dゲイン、Iゲインで挙動が変化します。</li> <li>・原点(中央値オフセットで設定した値)から±320度の範囲で制御できます。 (Free, Holdにすると、現在位置±180度に制限されます)</li> </ul> <p>例) 目標位置を180度に設定・・・18000(0x4650)</p>							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
現在位置	0x2C	short	100×deg	0	-32000	32000	RO
<p>サーボからデータを読み取ったときに現在位置を表示します。</p> <p>可動範囲の中央を0度として、CCW（反時計回転）方向に-320度、CW(時計回転)方向に320度の範囲で、現在いる位置の角度情報を0.01度単位で知ることができます。</p> <p>例) 現在位置が90度の時・・・9000(0x2328)</p>							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
前回のサンプリングの位置	0x2E	short	100×deg	0	-32000	32000	RO
<p>サーボの制御ループで1回前の値を表示します。</p> <p>例) 前回のサンプリングサーボの位置が89度の時・・・8900(0x22C4)</p>							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
目標速度	0x30	short	100×deg/sec	0	-32768	32767	RW
<p>サーボ目標速度を書き込みます。速度制御モードでのみ有効です。</p> <p><u>速度制御</u></p> <ul style="list-style-type: none"> <li>・目標速度と現在速度との差を元に入力値（操作量）が決まる制御方法です。</li> </ul> <p>例) 目標速度を30度/1secに設定・・・3000(0x0BB8)</p> <p>※スペック以上の値は出力できません。</p>							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
現在速度	0x32	short	100×deg/sec	0	-32768	32767	RO
<p>現在速度を0.01deg/sec単位で知ることができます。</p> <p>例) 現在の速度が40度/secの時・・・4000(0x0FA0)</p>							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
前回のサンプリング速度	0x34	short	100×deg/sec	0	-32768	32767	RO
サーボの制御ループで1回前の速度を表示します。 例) 前回のサンプリングサーボの速度が39度/secの時・・・3900(0x0F3C)							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
目標時間	0x36	ushort	ms	0	0	65565	RW
位置制御モードで軌道生成モード時に目標移動時間を書き込みます。 例) 目標時間を3secに設定・・・3000(0x0BB8) ※0を指定すると最高速度で移動します。							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
コマンド実行中時間	0x38	ushort	ms	0	0	65565	RO
位置制御モードで現在時間を表示します。 SET POSITIONコマンド発行時に0になります。 例) コマンド実行時間が2秒の時・・・2000(0x07D0)							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
起動時からの経過時	0x3A	ushort	sec	0	0	65535	RO
制御経過時間を表示します。 例) 経過時間が30秒の時・・・30(0x001e)							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
目標トルク	0x3C	short	mN.m	0	-32768	32767	RW
電流(トルク)制御 ・目標トルクから電流値を計算し、電流値をフィードバックする制御方法です。PI制御を行っています。 例) 目標トルクを1000mN.mに設定・・・1000(0x03E8) ※スペック以上の値は出力できません。							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
システムのクロック	0x3E	ulong	20×ns	0	0	10000000	RO
システム（MCU）クロック値を読み取ります。 例) クロック値が12345678の時・・・12345678(0x00BC614E)							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
サンプリングタイム	0x42	ushort	20×ns	0	0	10000000	RO
現在の制御ループサンプリングタイムを表示します。 例) データ処理時間が30msの時・・・1500(0x05DC)							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
現在のMCU温度	0x44	short	100×℃	0	-5500	12800	RO
MCU温度を表示します。 例) 現在のMCUの温度が40度の時・・・4000(0x0FA0)							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
現在のモーター温度	0x46	short	100×℃	0	-12800	12800	RO
モーター温度を表示します。 例) 現在のモーターの温度が60度の時・・・6000(0x1770)							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
現在の(負荷)電流値	0x48	short	mA	0	-32768	32767	RO
モーター電流値を表示します。 CCW（反時計回転）の時は、-32768～0 CW(時計回転)の時は、0～32767 例) 電流値が300mAの時・・・300(0x012C)							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
現在の入力電圧値	0x4A	ushort	mV	0	0	65535	RO
入力電圧値を表示します。 例) 入力が5.5Vの時・・・5500(0x157C)							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
現在のPWM <b>カウント値</b>	0x4C	ushort	—	0	0	65535	RO
PWMのデューティ値を表示します。 数値はPWM Frequencyに対するカウント数で、いわゆるデューティ比ではありません。 例) カウント数が2000の時・・・2000(0x07D0)							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
PWM周期	0x4E	ushort	Hz	※	0	65535	RW #
PWM Frequencyです。カウンタ最大値としてデューティ比を計算できます。 <u>PWM周期</u> ・最大PWMカウント値＝システムクロック÷PWM周期・・・(1) PWM信号のデューティ比＝指定PWMカウント値÷最大PWMカウント値 最大PWMカウント値は式(1)に依存します。 PWM周期カウント値を大きくすると1周期あたりの最大PWMカウント値が小さくなるので、 デューティ比を細かく指定できなくなります。 例) PWM周期を8KHzに設定・・・8000(0x1F40) ※製品により異なります。 ※書き換えるとPWM周期が変わります。							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
エンコーダーの現在値	0x50	ushort	－	0	0	4095	RO
サーボに取り付けられているエンコーダー値です。 現在位置を求める為のデータ補正前のデータです。 例) 現在値が2048の時・・・2048(0x0800)							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
エンコーダーのトータルカウント数	0x52	long	－	0	-2 <sup>31</sup>	2 <sup>31</sup> -1	RW
エンコーダーの累計カウント数です。 -2147483648~2147483647まで（-524288~524287）回転まで計測できます。 初期化する場合はこのパラメーターを0にしてください。 リバースモード時はノーマルモードと同じ位置に設定され、リバースの影響を受けません。 例) トータルカウント数が123456の時・・・123456(0x0001E240)							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
ホールICの状態	0x56	byte	－	0	0	7	RO
ブラシレスモーターのホールICの入力状態を(ON=1、OFF=0)3ビットで表します。							



## ■ 制御パラメータ領域

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
PIDゲインのプリセット番号	0x5C	byte	—	0	0	2	RW
Reserved	0x5D	byte					
PIDの各ゲインをあらかじめ指定したプリセットから呼び出します。							

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
比例ゲイン0	0x5E	ulong	100x	※	0	2 <sup>32</sup> -1	RW
微分ゲイン0	0x62	ulong	100x	※	0	2 <sup>32</sup> -1	RW
積分ゲイン0	0x66	ulong	100x	※	0	2 <sup>32</sup> -1	RW
静止摩擦係数0	0x6A	ushort	—	※	0	2 <sup>16</sup> -1	RW
動摩擦係数0	0x6C	ushort	—	※	0	2 <sup>16</sup> -1	RW
比例ゲイン1	0x6E	ulong	100x	※	0	2 <sup>32</sup> -1	RW
微分ゲイン1	0x72	ulong	100x	※	0	2 <sup>32</sup> -1	RW
積分ゲイン1	0x76	ulong	100x	※	0	2 <sup>32</sup> -1	RW
静止摩擦係数1	0x7A	ushort	—	※	0	2 <sup>16</sup> -1	RW
動摩擦係数2	0x7C	ushort	—	※	0	2 <sup>16</sup> -1	RW
比例ゲイン2	0x7E	ulong	100x	※	0	2 <sup>32</sup> -1	RW
微分ゲイン2	0x82	ulong	100x	※	0	2 <sup>32</sup> -1	RW
積分ゲイン2	0x86	ulong	100x	※	0	2 <sup>32</sup> -1	RW
静止摩擦係数2	0x8A	ushort	—	※	0	2 <sup>16</sup> -1	RW
動摩擦係数2	0x8C	ushort	—	※	0	2 <sup>16</sup> -1	RW

ゲインについて

B3Mサーボモーターは各ゲイン値を変更すると即座に制御に反映されます。

※製品により異なります。

Pゲイン（Kp0~Kp2）

- ・Pゲインは比例ゲインのことで、目標位置と現在位置の位置誤差（偏差）に対して、比例的に入力値（操作量）を変化させるための倍率（利得）のことです。
- ・Pゲインを減らすと比例的に入力値が減るため、モーターのパワー・スピードともに減ります。大きくすると動作が振動的になります。

Dゲイン（Kd0~Kd2）

- ・Dゲインは微分ゲインのことで、現在速度と目標速度（位置制御時は目標速度は0）の偏差に対して、比例的に操作量を変化させるための倍率（利得）のことです。
- ・Dゲインは位置変化の傾きに寄与するため、応答が振動的になるのを抑制し収束しやすくなります。

## Iゲイン (Ki0~Ki2)

- ・Iゲインは積分ゲインのことで、現在速度と目標速度（位置制御時は目標速度は0）の偏差の累積値（積分値）に対して、比例的に操作量を変化させるための倍率（利得）のことです。
- ・Iゲインを大きくすると位置誤差を小さくするようになり、モーターの保持力が大きくなります。ただし偏差が0の状態または積分値が小さいような、主に応答開始時の反応が遅れるようになります。

## ゲインプリセット

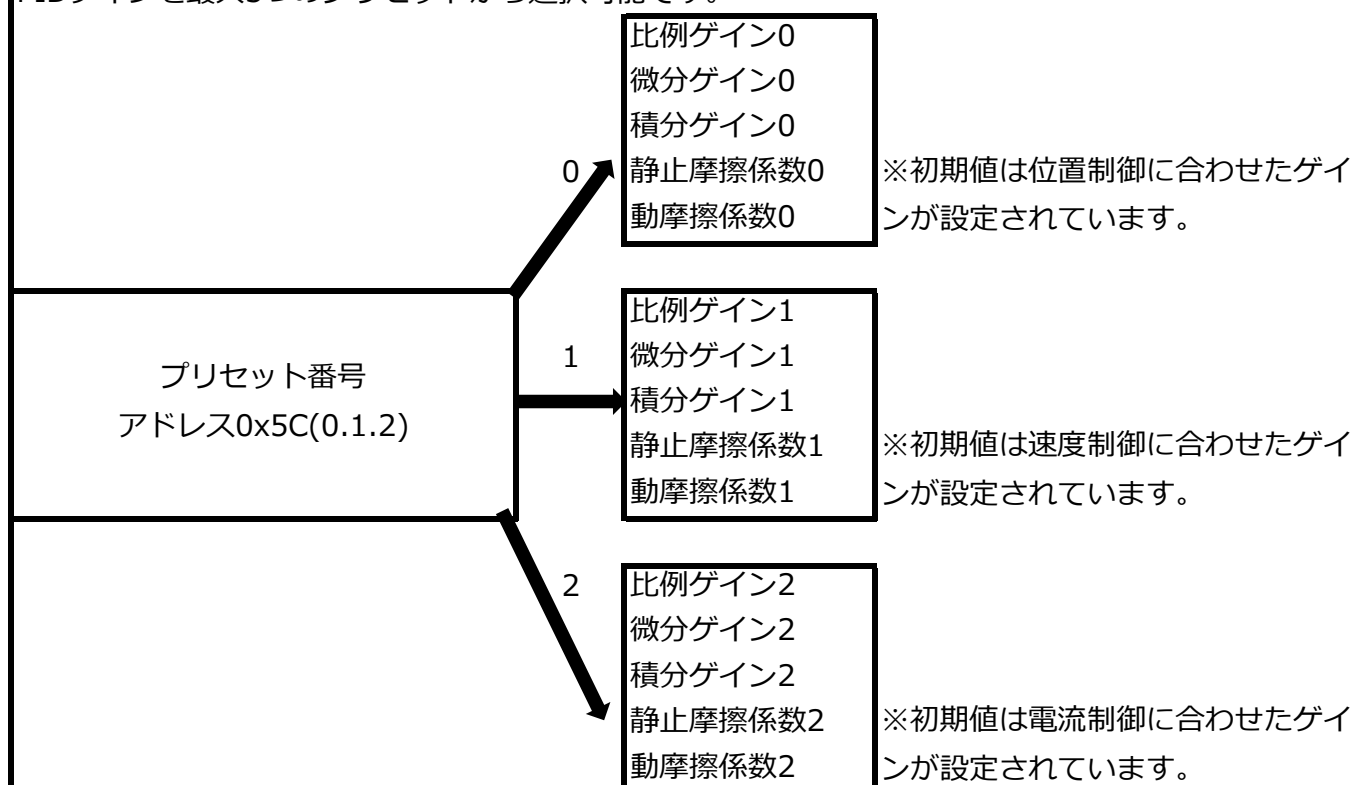
- ・出荷時にはプリセット0、1、2にそれぞれ位置制御、速度制御、トルク制御に合わせたゲインが設定されています。
- ・ゲインプリセット番号を変えると即座にゲインが変更され、サーボモーターの動作に反映されます。位置制御・速度制御・トルク制御ではゲインが違うので、制御モードを変更する場合は必ずゲインまたはゲインプリセットを変更してください。
- ・全てのプリセットに位置制御用のゲインを入れておき、切り替えて使用することもできます。

## 摩擦係数

- ・摩擦係数はモーターおよびギアの摩擦に対する補償で、無次元数です。
- ・静止状態から動き出すときに静止摩擦係数を操作量に加えています。  
また動作中は動摩擦係数を加えています。  
静止摩擦係数と動摩擦係数はモーターの速度により自動的に切り替えています。

初期設定ではゲインプリセット No.0を位置制御、ゲインプリセット No.1を速度制御、ゲインプリセット No.2を電流（トルク）制御に割り当てています。

PIDゲインを最大3つのプリセットから選択可能です。



## ■ STATUSパラメータ領域

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
エラーステータス	0x9D	byte	—	0			
システムエラー	0x9E	byte	—	0			
モーターステータス異常	0x9F	byte	—	0			
UART受信エラー	0xA0	byte	—	0			
コマンドエラーなど	0xA1	byte	—	0			

## ■ バージョン情報領域

名称（説明）	表記	内 容
電圧区分	A	6～12[V]
	B	12～18[V]
	C	18～24[V]
トルク番号	4	40[kgf.cm]
	7	70[kgf.cm]
ケース番号	10	ケースの高さが低いタイプ
	11	ケースの高さが高いタイプ
モータータイプ	B	ブラシレスモーター
	C	コアレスモーター
	D	コアドモーター
デバイスタイプ	S	サーボモーター

## モデル区分

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
電圧区分	0xA2	char	—	※	0x00	0xFF	RO
モデルバージョン	0xA3	byte	—	※	0x00	0xFF	RO
トルク番号	0xA4	byte	—	※	0x00	0xFF	RO
ケース番号	0xA5	byte	—	※	0x00	0xFF	RO

※製品により異なります。

## モデル種類

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
	0xA6	byte	—	※	0x00	0xFF	RO
	0xA7	byte	—	※	0x00	0xFF	RO
モータータイプ	0xA8	char	—	※	0x00	0xFF	RO
デバイスタイプ	0xA9	char	—	※	0x00	0xFF	RO

※製品により異なります。

## ファームウェア情報

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
ビルド番号	0xAA	byte	—	※	0x00	0xFF	RO
リビジョン番号	0xAB	byte	—	※	0x00	0xFF	RO
マイナーバージョン	0xAC	byte	—	※	0x00	0xFF	RO
メジャーバージョン	0xAD	byte	—	※	0x00	0xFF	RO
※製品により異なります。							

## ■ システム初期値領域

## 相対位置でのエンコードの誤差補正

名称（説明）	アドレス	型	単位	初期値	最小値	最大値	属性
絶対00位置からのエンコードのずれ	0xAE	short	—	※	-32768	32767	RW #
相対位置01番目の誤差補正	0xB0	short	—	※	-32768	32767	RW #
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
相対位置36番目の誤差補正	0xF6	short	—	※	-32768	32767	RW #

基準値と実際の位置のずれを補正します。

※工場出荷時に設定済みですので変更しないでください。

## 9.B3MシリーズPC用ライブラリーについて

- ・ B3MサーボモーターのPC用.NETライブラリです。
- ・ B3MLib : B3Mサーボモーター用のコマンドを作成できるB3MLib.dllとExtensions.dllの2つのDLLファイルがあります。

### ■使用環境

- ・ Windows XP (SP2以降) /Windows Vista/Windows 7/8/8.1/10各32ビット・64ビット版。
- ・ B3MLib.DLLとExtensions.DLL (B3Mライブラリセット)。
- ・ .NET Framework 2.0以上。
- ・ Windows XP サービスパック2以降、Windows Vista、Windows 7ではOSにインストールされています。
- ・ Visual Studio 2005以降、または Visual Basic Express/Visual C# Express/Visual C++ Expressの2005以降のバージョンが必要（本解説書ではVisual Basic Express 2010を使用）。

#### 【公開先】

<http://kondo-robot.com/faq/b3m-pclibset>

### ■メモリーマップ

SYSTEM領域				
アドレス	内 容	型	アドレス番号定義名	プロパティ名
0x00	ID番号	byte	SYSTEM_ID	ID
0x01	ボーレート、通信速度	ulong	SYSTEM_BAUDRATE	Baudrate
0x05	最小位置制限	short	SYSTEM_POSITION_MIN	PositionMinLimit
0x07	最大位置制限	short	SYSTEM_POSITION_MAX	PositionMaxLimit
0x09	中央値オフセット	short	SYSTEM_POSITION_CENTER	PositionCenterOffset
0x0B	MCU温度リミット	short	SYSTEM_MCU_TEMP_LIMIT	MCUTempLimit
0x0D	MCU温度制限時のPWM制限率	byte	SYSTEM_MCU_TEMP_LIMIT_PR	MCUTempPowerLimit
0x0E	モーター温度リミット	short	SYSTEM_MOTOR_TEMP_LIMIT	MotorTempLimit
0x10	モーター温度制限時のPWM制限率	byte	SYSTEM_MOTOR_TEMP_LIMIT_PR	MotorTempPowerLimit
0x11	電流制限値	ushort	SYSTEM_CURRENT_LIMIT	CurrentLimit
0x13	電流制限時のPWM制限率	byte	SYSTEM_CURRENT_LIMIT_PR	CurrentPowerLimit
0x14	モーターロックと認識されるまでの時間	byte	SYSTEM_LOCKDETECT_TIME	LockDetectTime
0x15	ロックと認識される最大出力からの割合	byte	SYSTEM_LOCKDETECT_OUTRATE	LockDetectOutputRate
0x16	ロック検知時の脱力度合い	byte	SYSTEM_LOCKDETECT_TIME_PR	LockDetectTimePowerLimit
0x17	有効入力電圧最小値	ushort	SYSTEM_INPUT_VOLTAGE_MIN	InputVoltageMin
0x19	有効入力電圧最大値	ushort	SYSTEM_INPUT_VOLTAGE_MAX	InputVoltageMax
0x1B	PWM制限	byte	SYSTEM_TORQUE_LIMIT	TorqueLimit
0x1C	不感帯の幅	ushort	SYSTEM_DEADBAND_WIDTH	DeadBandWidth
0x22	モーター回転特性CW方向の割合	byte	SYSTEM_MOTOR_CW_RATIO	MotorCWRatio
0x23	モーター回転特性CCW方向の割合	byte	SYSTEM_MOTOR_CCW_RATIO	MotorCCWRatio

サーボパラメーター領域				
アドレス	内 容	型	アドレス番号定義名	プロパティ名
0x27	サーボオプション(※)	byte	SERVO_SERVO_OPTION	ServoOption
0x28	動作モード	ushort	SERVO_SERVO_MODE	ServoMode
0x28	Normal/FREE/Holdモード切り替え	byte	SERVO_TORQUE_ON	TorqueON
0x29	軌道生成タイプ	byte	SERVO_RUN_MODE	RunMode
0x2A	目標位置	short	SERVO_DESIRED_POSITION	DesiredPosition
0x2C	現在位置	short	SERVO_CURRENT_POSITION	CurrentPosition
0x2E	1サンプリング回前の位置	short	SERVO_PREVIOUS_POSITION	PreviousPosition
0x30	目標速度	short	SERVO_DESIRED_VELOCITY	DesiredVelocity
0x32	現在速度	short	SERVO_CURRENT_VELOCITY	CurrentVelocity
0x34	1サンプリング回前の速度	short	SERVO_PREVIOUS_VELOCITY	PreviousVelocity
0x36	目標時間	ushort	SERVO_DESIRED_TIME	DesiredTime
0x38	コマンド実行中時間	ushort	SERVO_RUNNING_TIME	RunningTime
0x3A	起動時からの経過時間	ushort	SERVO_WORKING_TIME	WorkingTime
0x3C	目標トルク	short	SERVO_DESIRED_TORQUE	DesiredTorque
0x3E	システムのクロック	ulong	SERVO_SYSTEM_CLOCK	SystemClock
0x42	サンプリングタイム	ushort	SERVO_SAMPLING_TIME	SamplingTime
0x44	現在のMCU温度	short	SERVO_MCU_TEMP	MCUTemperature
0x46	現在のモーター温度	short	SERVO_MOTOR_TEMP	MotorTemperature
0x48	現在の(負荷)電流値	short	SERVO_CURRENT	Current
0x4A	現在の入力電圧値	ushort	SERVO_INPUT_VOLTAGE	InputVoltage
0x4C	現在のPWMデューティ比	ushort	SERVO_PWM_DUTY	PwmDuty
0x4E	PWM周期	ushort	SERVO_PWM_FREQUENCY	PwmFrequency
0x50	エンコーダーの現在値	ushort	SERVO_ENCODER_VALUE	EncoderValue
0x52	エンコーダーのトータルカウント数	long	SERVO_ENCODER_COUNT	EncoderCount
0x56	ホールICの状態	byte	SERVO_HALLIC_STATE	HallICState

制御パラメーター領域				
アドレス	内 容	型	アドレス番号定義名	プロパティ名
0x5C	PIDゲインのパターンを変更する	ushort	CONTROL_CONTROL_LOW	ControlRow
0x5C	PIDゲインのプリセット番号	byte	CONTROL_GAIN_PRESETNO	GainPresetNo
0x5D	Reserved	byte	CONTROL_TYPE	ControlType
0x5E	比例ゲイン0	ulong	CONTROL_KP0	Kp0
0x62	微分ゲイン0	ulong	CONTROL_KD0	Kd0
0x66	積分ゲイン0	ulong	CONTROL_KI0	Ki0
0x6A	静止摩擦係数0	ushort	CONTROL_STATIC_FRICTION0	StaticFriction0
0x6C	動摩擦係数0	ushort	CONTROL_DYNAMIC_FRICTION0	DynamicFriction0
0x6E	比例ゲイン1	ulong	CONTROL_KP1	Kp1
0x72	微分ゲイン1	ulong	CONTROL_KD1	Kd1
0x76	積分ゲイン1	ulong	CONTROL_KI1	Ki1
0x7A	静止摩擦係数1	ushort	CONTROL_STATIC_FRICTION1	StaticFriction1
0x7C	動摩擦係数1	ushort	CONTROL_DYNAMIC_FRICTION1	DynamicFriction1
0x7E	比例ゲイン2	ulong	CONTROL_KP2	Kp2
0x82	微分ゲイン2	ulong	CONTROL_KD2	Kd2
0x86	積分ゲイン2	ulong	CONTROL_KI2	Ki2
0x8A	静止摩擦係数2	ushort	CONTROL_STATIC_FRICTION2	StaticFriction2
0x8C	動摩擦係数2	ushort	CONTROL_DYNAMIC_FRICTION2	DynamicFriction2

STATUS領域				
アドレス	内 容	型	アドレス番号定義名	プロパティ名
0x9D	エラーステータス	byte	STATUS_BASE_ADDR	StatusError
0x9E	エラーなどのステータス	ulong	STATUS_SYSTEM	Status
0x9E	システムエラー	byte	STATUS_SYSTEM	StatusSystem
0x9F	モーターステータス異常	byte	STATUS_MOTOR	StatusMotor
0xA0	UART受信エラー	byte	STATUS_UART	StatusUart
0xA1	コマンドエラーなど	byte	STATUS_COMMAND	StatusCommand

システム初期値・バージョン情報領域				
アドレス	内 容	型	アドレス番号定義名	プロパティ名
0xA2	モデル区分	ulong	CONFIG_MODEL_NUMBER	ModelNumber
0xA2	電圧区分	char	CONFIG_MODEL_NUMBER_VOLTAGE_CLASS	ModelNumberVoltageClass
0xA3	モデルバージョン	byte	CONFIG_MODEL_NUMBER_VERSION	ModelNumberVersion
0xA4	トルク番号	byte	CONFIG_MODEL_NUMBER_TORQUE	ModelNumberTorque
0xA5	ケース番号	byte	CONFIG_MODEL_NUMBER_CASE	ModelNumberCase
0xA6	モデル種類	ulong	CONFIG_MODEL_TYPE	ModelType
0xA6				
0xA7				
0xA8	モータータイプ	char	CONFIG_MODEL_TYPE_MOTOR	ModelTypeMotor
0xA9	デバイスタイプ	char	CONFIG_MODEL_TYPE_DEVICE	ModelTypeDevice
0xAA	ファームウェア情報	ulong	CONFIG_FW_VERSION	FwVersion
0xAA	ビルド番号	byte	CONFIG_FW_BUID	FwVersionBuild
0xAB	リビジョン番号	byte	CONFIG_FW_REVISION	FwVersionRevision
0xAC	マイナーバージョン	byte	CONFIG_FW_MINOR	FwVersionMinor
0xAD	メジャーバージョン	byte	CONFIG_FW_MAJOR	FwVersionMajor
0xAE	絶対00位置からの誤差	short	CONFIG_ENC_OFFSET_CENTER	EncoderOffsetCenter
0xB0	相対01位置での誤差	short	CONFIG_ENC_OFFSET	EncoderOffset

※ Ver.1.0.2.0より有効

## 10. 列挙型変数一覧(bitwiseオプション)

アドレス(0x27.0x28)	動作モード		
カテゴリー	内容	Enum名称	値
Options	動作モードノーマル	RunNormal	0x00
	フリーモード	RunFree	0x02
	ホールドモード	RunHold	0x03
	位置制御モード	ControlPosition	0x00
	速度制御モード	ControlVelocity	0x04
	電流（トルク）制御モード	ControlTorque	0x08
	フィードフォワード制御モード	ControlFFForward	0x0C
	ノーマル	ServoNormal	0x00
	クローンモード	ServoClone	0x40
	リバースモード	ServoReverse	0x80

アドレス(0x9E)	システムエラー		
カテゴリー	内容	Enum名称	値
SystemErrors	Watchdogタイマーが再起動したか	Watchdog	0x01
	フラッシュメモリーアクセスエラー	FlashAccess	0x02
	フラッシュメモリー初期化エラー	MemoryAllocation	0x04
	入力電圧エラー	InputVoltage	0x08
	MCU温度が範囲外	MCUTemperature	0x10
	AD変換に失敗している	ADConversion	0x20
	I2Cアクセス失敗	I2C	0x40
	SPIアクセスに失敗	SPI	0x80

アドレス(0x9F)	モーターステータス異常		
カテゴリー	内容	Enum名称	値
MotorErrors	モーター温度が範囲外	ExceedMotorTemperature	0x01
	モーターロック検出	LockDetect	0x02
	モーター電流が範囲外	ExceedCurrentLimit	0x04
	Hall ICが誤検知した	HallIC	0x08

アドレス(0xA0)	UART受信エラー		
カテゴリー	内 容	Enum名称	値
UARTErrors	START/STOPビットが不正	Framing	0x01
	パリティチェック失敗	Parity	0x02
	データが終了していない	Break	0x04
	データビット取りこぼし	Overrun	0x08



アドレス(0xA1)	コマンドエラーなど		
カテゴリー	内 容	Enum名称	値
CommandErrors	チェックサムエラー	Checksum	0x01
	データの長さ指定を間違えている	Length	0x02
	指定データの長さが違う	Size	0x04
	アドレスがオーバーフローしている	Address	0x08
	コマンド間違い	WrongCommand	0x10

アドレス(0x29)	軌道生成タイプ		
カテゴリー	内 容	Enum名称	値
Trajectory	直線移動、加減速制御なし	Normal	0x00
	均等分割式（時間で等間隔移動とする）	Even	0x01
	3次多項式補間	ThirdPoly	0x03
	4-1-4次多項式補間	FourthPoly	0x04
	5次多項式補間	FifthPoly	0x05

	コマンド		
カテゴリー	内 容	Enum名称	値
CommandTypes	フラッシュメモリーに保存されているデータをRAMへ展開する	Load	0x01
	RAMのデータをフラッシュメモリーに保存する	Save	0x02
	RAM上のデータを読み取る	Read	0x03
	RAM上へデータを書き込む	Write	0x04
	ポジション指定コマンド	Position	0x05
	リセットコマンド	Reset	0x06

### 注意事項

bitwiseオプションをバイト変数に戻す場合は、(byte)カテゴリー.Enum名称というようにキャストしてください。例えばOptionsのRunFreeをByte変数に取り出す場合は下記のよ  
byte options = (byte)Options.RunFree;

## 11.変更履歴について

Version	日付	変更内容
Ver.1.0.0.0		βリリース
Ver.1.0.1.0	2012/12/15	PositionコマンドとResetコマンドのデータを入れ替え Statusアドレスの一部の記載間違いを修正
Ver.1.1.0.0	2013/07/18	エラーステータスの記載間違いを修正 LockDetectTimeの記載間違いを修正 サーボオプション（クローン・リバース）のアドレスを修正。B3Mバージョン1.0.2.0より適用。再起動後に有効になるように変更 サーボオプションのbitwiseオプションが間違っていたので修正 内部計算ルーチンを修正 その他マニュアル記載間違いを修正
Ver.1.1.3.0	2014/04/07	位置制御以外の現在位置は±180度になるように修正 FreeやHold状態の場合も現在位置は±180度になるように修正 位置制御以外の状態の場合、急に動かないようにするため目標位置に現在の位置を代入するように変更 トルク制御の時、電圧に依存しないように変更 マニュアルに動かし方の部分を追加 下記のバグを修正 <ul style="list-style-type: none"> <li>・コマンド実行中時間(0x38)が反映されていない</li> <li>・ロック検知が正常に反映されていない</li> <li>・エンコーダのカウントが起動直後0にならない</li> <li>・エンコーダのずれ(0xAE~0xF6)が起動中変更しても反映されている</li> <li>・Save,Loadコマンドで、マルチモードのコマンドを送った時、正常に動作しない</li> </ul> その他マニュアル記載間違いを修正
Ver.1.2.0.0	2015/11/27	マニュアルを見やすく修正
Ver.1.2.0.1	2016/09/07	マニュアル記載間違いを修正
Ver.1.2.0.2	2018/08/21	マニュアル記載間違いを修正 解説を追記 それぞれ修正・追記箇所は赤字で表記しています。