

RCB-4HV コマンドリファレンスマニュアル

免責事項

本リファレンスおよび内容に関する一切の権利は近藤科学株式会社が有しますが、このリファレンスは参考資料として公開されるものです。このリファレンスを使用したときの障害や損害につきましては、近藤科学株式会社は一切保証いたしませんので、使用者の責任においてご利用ください。

コマンドリファレンスについて

本コマンドリファレンスではRCB-4HVのファームウェアが持つコマンドインターフェイスを説明しています。
通信設定やRCB-4の内部構造については「RCB-4ファームウェアリファレンス」を参考にしてください。

第2版 Rev.20131018

近藤科学株式会社

MOV

【機能】

転送元のデータを転送先にコピーする。
転送先 ← 転送元

【コマンド構成】

1	2	3	4	5	6	7	~	n	n + 1
サイズ	コマンド	種別 classification	転送先 destination			転送元 source			チェックサム

サイズ: コマンド全体のバイト数=n+1
コマンド: MOV命令: 00h

種別: 転送先・転送元の種別									
7	6	5	4	3	2	1	0		
0	0	転送先		0	0	転送元		b1	b0 演算値
								0	0 RAMデータ
								0	1 ICSレジスタ
								1	0 リテラルデータ
								1	1 ROMデータ
								b5	b4 演算先
								0	0 RAM
								0	1 ICSレジスタ
								1	0 COM出力
								1	1 ROM

転送先: 転送先の指定		
RAM	4	5
	RAMアドレス 下位	RAMアドレス 上位
	00h	
RAMアドレス: 0000h~048Fh		
ICSレジスタ	4	5
	オフセット	ICS番号
	00h	00h
オフセット: 00h~FFh		
ICS番号: 0~35		
COM出力	4	5
	00h	00h
	00h	00h
ROM	4	5
	ROMアドレス 0~7bit	ROMアドレス 8~15bit
	ROMアドレス 16~23bit	
ROMアドレス: 000000h~03FFFFh		

転送元: 転送元の指定

	7	8	9
RAMデータ	RAMアドレス 下位	RAMアドレス 上位	転送データ サイズ
	RAMアドレス: 0000h~048Fh		
	データサイズ: 1~128		

	7	8	9
ICSレジスタ	オフセット	ICS番号	転送データ サイズ
	オフセット: 00h~FFh		
	ICS番号: 0~35		
	演算サイズ: 1~128		

	7	...	n
リテラルデータ	リテラルデータ 1byte	...	リテラルデータ n byte
	n: 1~128byte		

	7	8	9	10
ROMデータ	ROMアドレス 0~7bit	ROMアドレス 8~15bit	ROMアドレス 16~23bit	転送データ サイズ
	ROMアドレス: 000000h~03FFFFh			
	データサイズ: 1~128			

チェックサム: 1byte目のサイズからチェックサムの1byte手前までのデータを加算した下位1byte

【COMから実行した場合】

	1	2	3	n	n + 1
・処理が成功した (転送先がCOM)	→	04h	CMD	* 演算結果		SUM
		*結果のサイズは、コマンドで設定した通り				

		1	2	3	4
・処理が成功した (転送先がCOM以外)	→	04h	CMD	ACK (06h)	SUM

		1	2	3	4
・処理が失敗した ・通信エラーが発生した	→	04h	CMD	NAK (15h)	SUM

【ROMから実行した場合】

エラー発生 → ROM_Fが0になりプログラムが停止する

【エラー発生条件】

- ・通信エラーが発生した
- ・ICSの割り当てがないとき(サーボモーターを登録していないにも関わらず、データを読み取ってくるとき)
- ・RAMアドレスの範囲を超えたとき

【Rcb4クラスメソッド名】

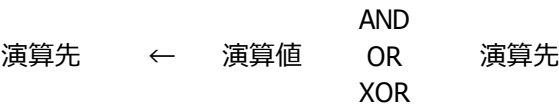
Move

AND/OR/XOR

【機能】

演算値と演算先とで各演算を行う。

【コマンド構成】



結果が0ならばZフラグが1、そうでなければ0。

1	2	3	4	5	6	7	～	n	n + 1
サイズ	コマンド	種別	演 算 先			演算値			チェック サム

サイズ: コマンド全体のバイト数=n+1

コマンド: AND命令: 01h

OR命令: 02h

XOR命令: 03h

種別: 演算先、演算値の種別

7 6 5 4 3 2 1 0

0

0

0

比較フラグ 0:演算結果を演算先に戻す。

1:演算結果を演算先に戻さない。

フラグの反応だけを得る。

b1 b0

0 0 RAMデータ

0 1 ICSデータ

1 0 リテラルデータ

1 1 ROMデータ

b5 b4

0 0 RAM

0 1 ICSレジスタ

1 0 禁止

1 1 ROM

演算先: 演算先の指定

RAM

4 5 6

RAMアドレス
下位

RAMアドレス
上位

00h

RAMアドレス: 0000h～048Fh

ICSレジスタ

4 5 6

オフセット

ICS番号

00h

オフセット: 00h～FFh

ICS番号: 0～35

ROM

4 5 6

ROMアドレス
0～7bit

ROMアドレス
8～15bit

ROMアドレス
16～23bit

ROMアドレス: 000000h～03FFFFh

演算値: 演算値となるデータの場合

	7	8	9
RAMデータ	RAMアドレス 下位	RAMアドレス 上位	演算サイズ
	RAMアドレス: 0000h~048Fh		
	演算サイズ: 1~128byte		

	4	5	6
ICSレジスタ	オフセット	ICS番号	演算サイズ
	オフセット: 00h~FFh		
	ICS番号: 0~35		
	演算サイズ: 1~128byte		

	7	...	n + 1
リテラルデータ	リテラルデータ 1byte	...	リテラルデータ n byte
	n: 1~128byte		

	7	8	9	10
ROMデータ	ROMアドレス 0~7bit	ROMアドレス 8~15bit	ROMアドレス 16~23bit	演算サイズ
	ROMアドレス: 000000h~03FFFFh			
	データサイズ: 1~128			

チェックサム: 1byte目のサイズからチェックサムの1byte手前までのデータを加算した下位1byte

【COMから実行した場合】

	1	2	3	n	n + 1
・処理が成功した	→	04h	CMD	* 演算結果	SUM	
		*結果のサイズは、コマンドで設定した通り				

	1	2	3	4
・処理が失敗した ・通信エラーが発生した	→	04h	CMD	NCK SUM

【ROMから実行した場合】

エラー発生 → ROM_Fが0になりプログラムが停止する

【エラー発生条件】

- ・通信エラーが発生した
- ・ICSの割り当てがないとき(読み取ってくるとき)
- ・RAMアドレスの範囲を超えたとき

【Rcb4クラスメソッド名】

And
Or
Xor

NOT

【機能】

演算先のデータを反転する

演算先 = 演算先

【コマンド構成】

1	2	3	4	5	6	7	8	9	10
サイズ	コマンド	種別	転送先			00h	00h	演算サイズ	チェックサム

サイズ: コマンド全体のバイト数=10
コマンド: NOT命令: 04h
種別: 演算先の種別

76543210

0

0

0

0

0

比較フラグ

0: 演算結果を演算先に戻す。
1: 演算結果を演算先に戻さない。
フラグの反応だけを得る。

b5b4

00
01
10
11

演算先
RAM
ICSLレジスタ
禁止
ROM

転送先: 転送先の指定

RAM

456

RAMアドレス下位

RAMアドレス上位

00h

RAMアドレス: 0000h~048Fh

ICSLレジスタ

456

オフセット

ICS番号

00h

オフセット: 00h~FFh
ICS番号: 0~35

ROM

456

ROMアドレス0~7bit

ROMアドレス8~15bit

ROMアドレス16~23bit

ROMアドレス: 000000h~03FFFFh

チェックサム: 1byte目のサイズからチェックサムの1byte手前までのデータを加算した下位1byte

【COMから実行した場合】

・処理が成功した

→

123.....nn+1

SIZE

CMD

* 演算結果

SUM

*結果のサイズは設定した通り

・処理が失敗した
・通信エラーが発生した

→

1234

04h

CMD

NCK (15h)

SUM

【ROMから実行した場合】

エラー発生 → ROM_Fが0になりプログラムが停止する

【エラー発生条件】

- ・通信エラーが発生した
- ・ICSの割り当てがないとき
- ・RAMアドレスの範囲を超えたとき

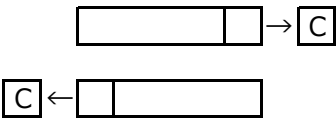
【Rcb4クラスメソッド名】

Not

SHIFT

【機能】

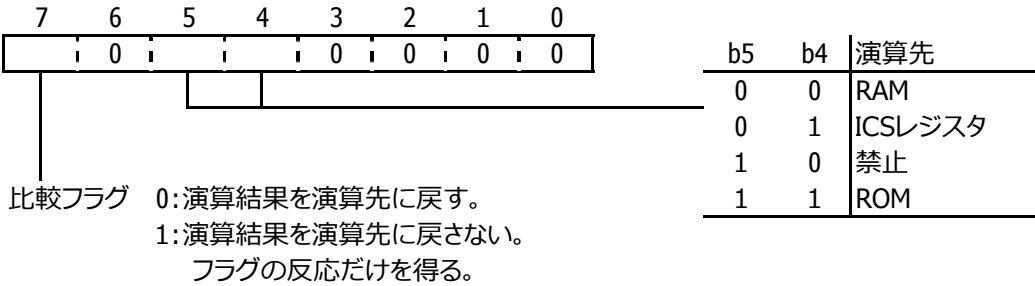
・演算先のデータをシフト演算する。



【コマンド構成】

1	2	3	4	5	6	7	8	9	10
サイズ	コマンド	種別	演 算 先			00h	シフト回数	演算サイズ	チェックサム

サイズ: コマンド全体のバイト数=10
コマンド: SHIFT命令: 05h
種別: 演算先の種別



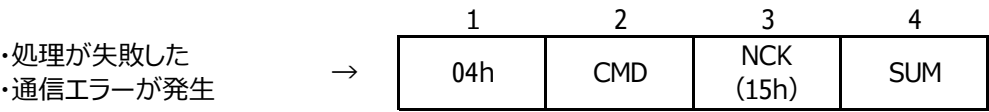
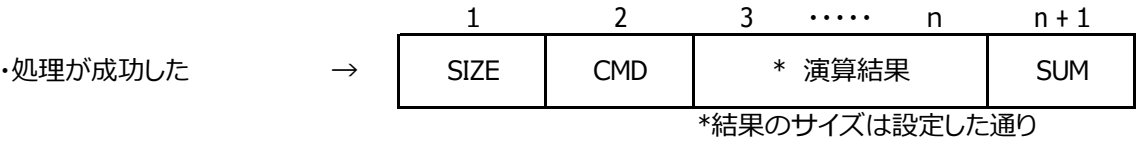
演算先: 演算先の指定

RAM	4	5	6
	RAMアドレス 下位	RAMアドレス 上位	00h
	RAMアドレス: 0000h~048Fh		
ICSレジスタ	4	5	6
	オフセット	ICS番号	00h
	オフセット: 00h~FFh ICS番号: 0~35		
ROM	4	5	6
	ROMアドレス 0~7bit	ROMアドレス 8~15bit	ROMアドレス 16~23bit
	ROMアドレス: 000000h~03FFFFh		

演算サイズ: 演算先のデータバイト数 1~128byte
シフト回数: シフト演算のシフト回数
 0 → シフトしない
 1~127 → 右シフト
 255~128 → 左シフト

チェックサム: 1byte目のサイズから演算サイズまでのデータを加算した、その結果の下位1byte

【COMから実行した場合】



【ROMから実行した場合】

エラー発生

→

ROM_Fが0になりプログラムが停止する

【エラー発生条件】

- ・通信エラーが発生した
- ・ICSの割り当てがないとき
- ・RAMアドレスの範囲を超えたとき

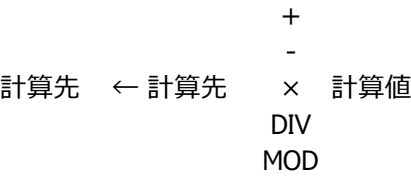
【Rcb4クラスメソッド名】

Shift

四則演算

【機能】

・計算値と計算先とで四則演算を行う



・計算結果

加算:	桁上がりがあればC=1	結果が0ならZ=1
減算:	桁下がりがあればC=0	結果が0ならZ=1
乗算:	桁上がりがあればC=1	結果が0ならZ=1
除算: (余算)	C=0になる	結果が0ならZ=1

【コマンド構成】

1	2	3	4	5	6	7	～	n	n + 1
サイズ	コマンド	種別	計 算 先			計 算 値			チェック サム

サイズ: コマンド全体のバイト数=n+1

コマンド: 加算命令: 06h
減算命令: 07h
乗算命令: 08h
除算命令: 09h
余算命令: 0Ah

種別: 転送先・転送元の種別

76543210

0

転送先

00

転送元

比較フラグ

0:演算結果を演算先に戻す。
1:演算結果を演算先に戻さない。
フラグの反応だけを得る。

b1b0

演算値

00RAMデータ
01ICSデータ
10リテラルデータ
11ROMデータ

b5b4

演算先

00RAM
01ICSレジスタ
10COM出力
11ROM

演算先: 演算先の指定

	4	5	6
RAM	RAMアドレス 下位	RAMアドレス 上位	00h

10/28

RAMアドレス: 0000h~048Fh

ICSレジスタ	4	5	6
	オフセット	ICS番号	00h
	オフセット: 00h~FFh ICS番号: 0~35		

ROM	4	5	6
	ROMアドレス 0~7bit	ROMアドレス 8~15bit	ROMアドレス 16~23bit
	ROMアドレス: 000000h~03FFFFh		

計算値: 計算値の指定

RAMデータ	7	8	9
	RAMアドレス 下位	RAMアドレス 上位	計算サイズ
	RAMアドレス: 0000h~048Fh 計算サイズ: 1,2		

ICSデータ	4	5	6
	オフセット	ICS番号	計算サイズ
	オフセット: 00h~FFh ICS番号: 0~35 計算サイズ: 1,2		

リテラルデータ	7	8
	リテラルデータ 1byte	リテラルデータ n byte
	n: 1,2	

ROMデータ	7	8	9	10
	ROMアドレス 0~7bit	ROMアドレス 8~15bit	ROMアドレス 16~23bit	計算サイズ
	ROMアドレス: 000000h~03FFFFh データサイズ: 1,2			

チェックサム: 1byte目のサイズからチェックサムの1byte手前までのデータを加算した下位1byte

【COMから実行した場合】

1

2

3

.....

n

n + 1

→

04h

CMD

* 演算結果

SUM

*結果のサイズは設定した通り

1

2

3

4

→

04h

CMD

NCK
(15h)

SUM

・処理が成功した

・通信エラーが発生

【ROMから実行した場合】

エラー発生 → ROM_Fが0になりプログラムが停止する
(コールされない状態でRETURN実行)

【エラー発生条件】

- ・通信エラーが発生した
- ・ICSの割り当てがないとき
- ・RAMアドレスの範囲を超えたとき

【Rcb4クラスメソッド名】

Add
Sub
Multiply
Division

JUMP/CALL

【機能】

指定アドレスへのジャンプ(コール)または指定条件でのジャンプ(コール)
コールの場合はコールの次の命令の先頭アドレスを、スタックに入れて指定アドレスに処理が移る

【コマンド構成】

1	2	3	4	5	6	7
サイズ	コマンド	ROMアドレス			条件	チェックサム
		L	M	H		

サイズ: コマンド全体のバイト数=7
コマンド: JUMP命令: 0Bh
 CALL命令: 0Ch

条件: ジャンプするための条件

7	6	5	4	3	2	1	0
0	0	0	0				

Zフラグの一致条件

Cフラグの一致条件

Zフラグ比較のON/OFF
1:ON 0:OFF

Cフラグ比較のON/OFF
1:ON 1:OFF

チェックサム: 1byte目のサイズからチェックサムの1byte手前までのデータを加算した下位1byte

【COMから実行した場合】

・処理が成功した →

1	2	3	4
SIZE	CMD	ACK (06h)	SUM

・処理が失敗した →

1	2	3	4
SIZE	CMD	NCK (15h)	SUM

【ROMから実行した場合】

エラー発生 → ROM_Fが0になりプログラムが停止する
(チェックサムエラー)

【備考】

コール命令が呼び出されたときのプログラムカウンタは最大16階層まで保存されます。
モーションの再生はモーションの先頭アドレスへ、CALL命令で移動します。

【Rcb4クラスメソッド名】

Jump
Call

RETURN

【機能】

- ・コールしたサブルーチンを終了して、コールした場所にジャンプ(戻る)する。
- ・コールされていない状態でRETURNすると、ROM_Fが0になりプログラムが停止します。

【コマンド構成】

1	2	3
サイズ	コマンド	サム

サイズ: コマンド全体のバイト数=3
コマンド: RETURN命令: 0Dh
チェックサム: 1byte目のサイズと2byte目のコマンドを加算した値の下位1byte

【COMから実行した場合】

		1	2	3	4
・処理が成功した	→	SIZE	CMD	ACK (06h)	SUM

		1	2	3	4
・通信エラーが発生	→	SIZE	CMD	NCK (15h)	SUM

(コールされていない状態でRETURN実行)

【ROMから実行した場合】

エラー発生 → ROM_Fが0になりプログラムが停止する
 (コールされていない状態でRETURN実行)

【備考】

モーション再生時はモーションの保存しているROMアドレスへCALL命令で移動し、最後にRETURNで呼び出し元へ戻ります。HeartToHeart 4ではモーションデータの末尾に自動的にRETURN命令を追加しています。

【Rcb4クラスメソッド名】

Return

ICS communication

単独サーボ動作

【機能】

- ・一つのサーボを、スピード・ポジションを指定して動かす
- Moving a single servo

【コマンド構成】

1	2	3	4	5	6	7
サイズ	コマンド	ICS 番号	動作 速度	ポジション L H		チェック サム

サイズ: コマンド全体のバイト数=7

コマンド: 単独動作命令: 0Fh

ICS番号: 動かすサーボの番号

7

6

5

4

3

2

1

0

ICS番号 (0～35)

ICS更新フラグ

0: ICS更新番号で指定した更新フラグを1にする

1: 更新フラグは何も変化しない

動作速度: サーボの動作速度

ポジション: サーボを移動させる場所

0000h～FFFFh

チェックサム: 1byte目のサイズからチェックサムの1byte手前までのデータを加算した下位1byte

【COMから実行した場合】

・処理が成功した →

1

2

3

4

SIZE

CMD

ACK
(06h)

SUM

・チェックサムエラー
・ICSの割り当てがない →

1

2

3

4

SIZE

CMD

NCK
(15h)

SUM

【ROMから実行した場合】

- ・チェックサムエラー → ROM_Fが0になりプログラムが停止する
- ・ICSの割り当てがない

【その他】

更新フラグを1にした場合、ICS番号で指定したサーボの更新フラグ、ROM_UPDが1になり、それ以外はすべて0になる

【Rcb4クラスメソッド名】

RunSingleServo

複数サーボ動作

【機能】

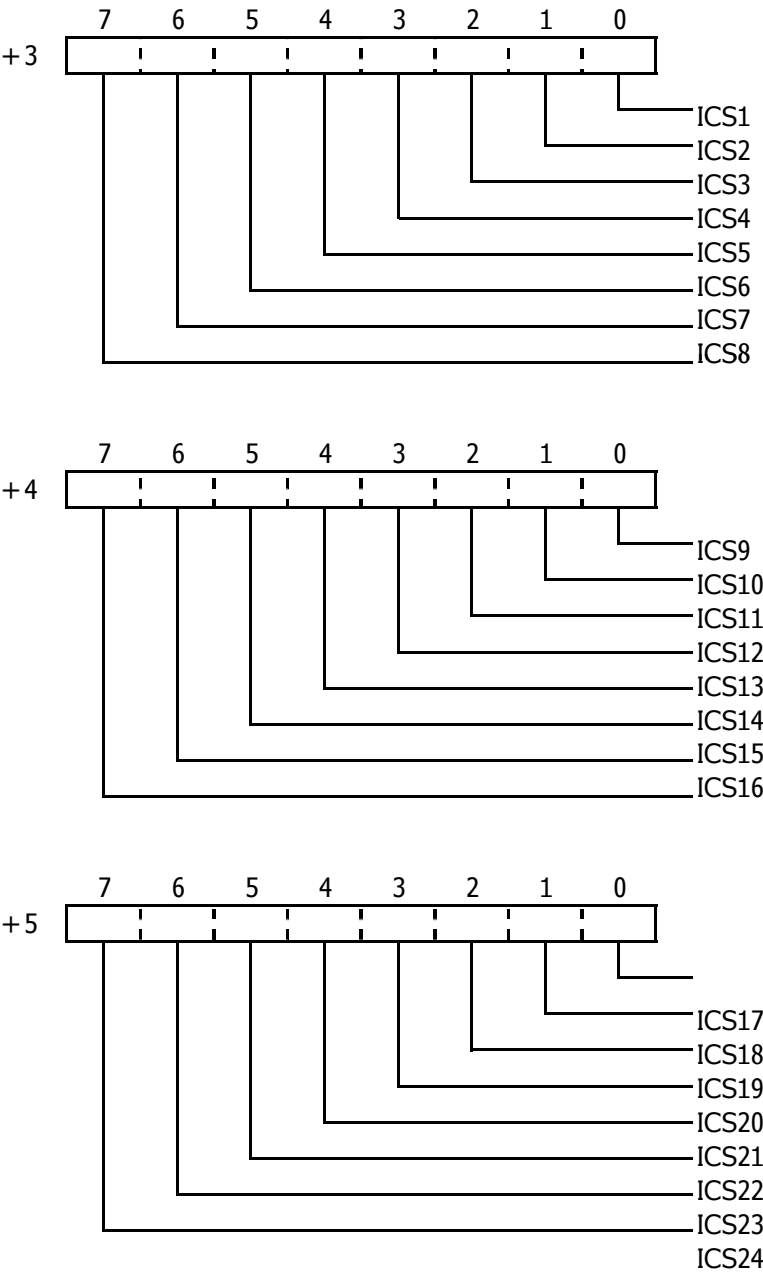
- ・動作スピードは一緒に、複数のサーボを動かす
- Moving multiple servos

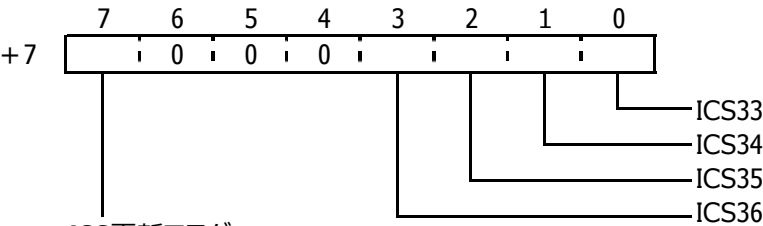
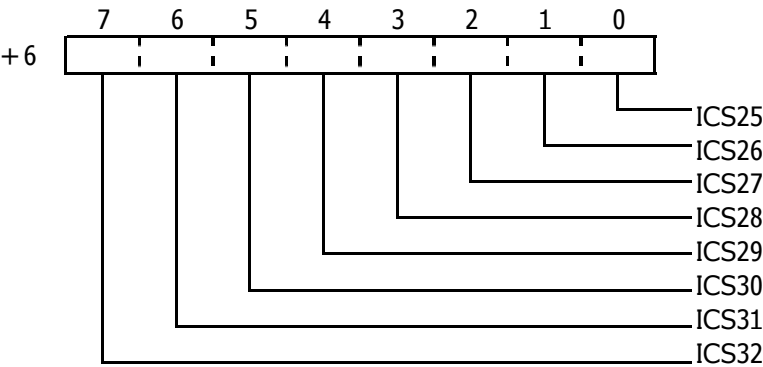
【コマンド構成】

1	2	3 ~ 7	8	9	10	11 ... n	n + 1
サイズ	コマンド	動作させるサーボの指定	動作速度	動作ポジション L : H		...	チェックサム

サイズ: コマンド全体のバイト数=n+1
コマンド: 複数動作命令: 10h

動作させるサーボの指定:
動かすサーボのbitを1にすることで、動かすサーボを決定します。





- ICS更新フラグ
- 0: +3～+7で指定したサーボの更新フラグを1にする
 - 1: 更新フラグは変化させない(使用しない)

動作速度: サーボの動作速度
 1～255

ポジション: サーボを移動させる場所
 0000h～FFFFh
 +3～+7の動作サーボの指定で1にしたサーボのデータが
 順番に並んでいる

チェックサム: 1byte目のサイズからチェックサムの1byte手前までのデータを加算した下位1byte

【COMから実行した場合】

		1	2	3	4
・処理が成功した	→	SIZE	CMD	ACK (06h)	SUM

		1	2	3	4
・チェックサムエラー	→	SIZE	CMD	NCK (15h)	SUM

【ROMから実行した場合】

・チェックサムエラー → ROM_Fが0になりプログラムが停止する

【その他】

- ・ICSの割り当てが無いサーボを指定した場合、更新フラグは0になる。
動作に関しては、割り当てが無いので動作しようとししない。
割り当てのないデータを入れても、特に問題ない。

【Rcb4クラスメソッド名】

RunConstFrameServo

複数サーボの個別動作

【機能】

- ・サーボごとに、動作スピードと動作ポジションを指定して動かす。
- Specify movement speed and movement position of the servo, and move it

【コマンド構成】

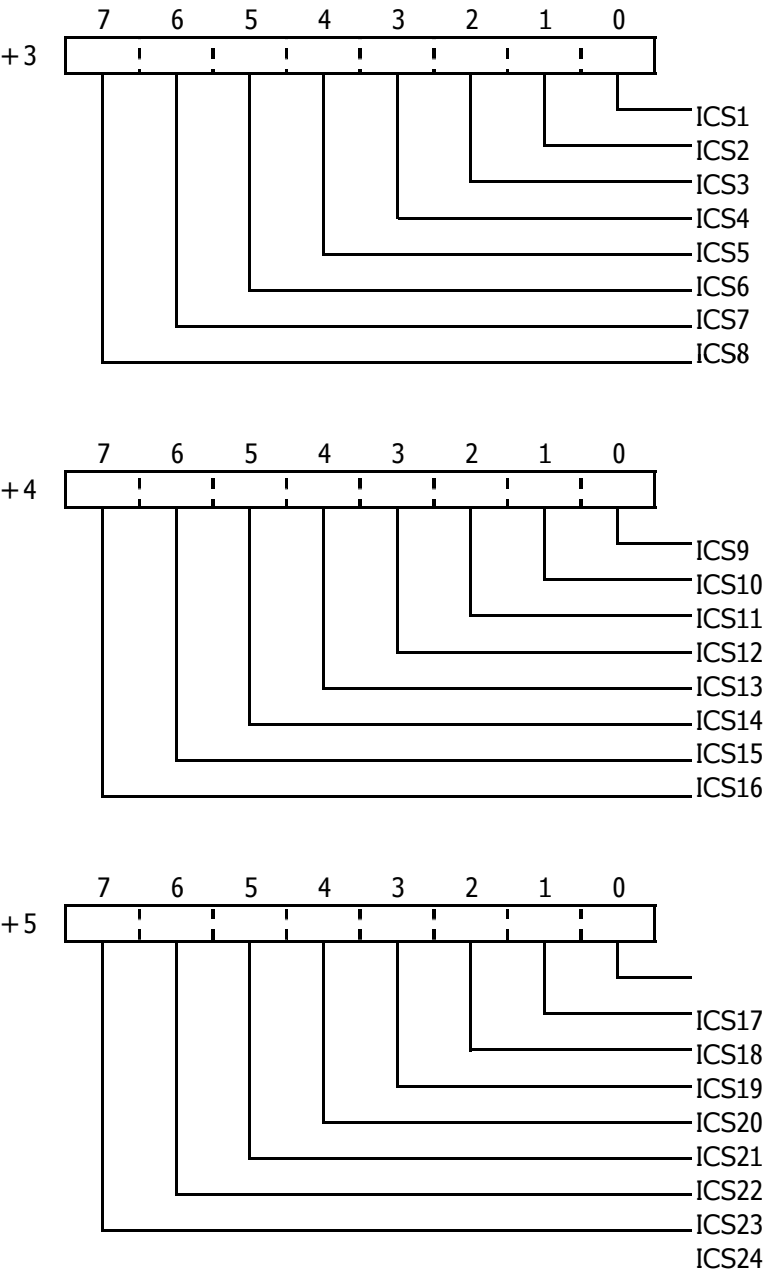
1	2	3 ~ 7	8	9	10	11	...	n	n + 1
サイズ	コマンド	動作させる サーボの指定	動作 速度	動作ポジション L H	速度 L H	速度 L H	速度 L H	速度 L H	チェック サム

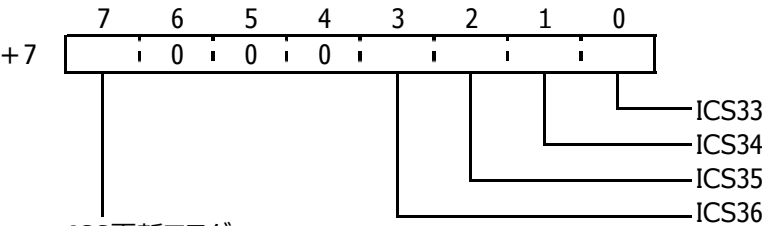
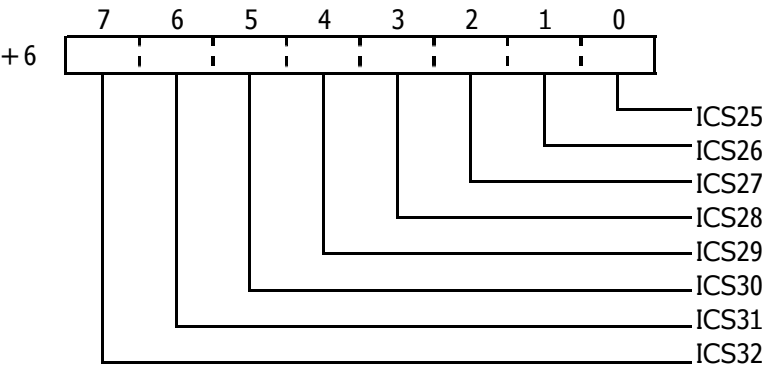
サイズ: コマンド全体のバイト数=n+1

コマンド: 複数動作命令: 11h

動作させるサーボの指定:

動かすサーボのbitを1にすることで、動かすサーボを決定します。





- ICS更新フラグ
- 0: +3～+7で指定したサーボの更新フラグを1にする
 - 1: 更新フラグは変化させない(使用しない)

動作速度: サーボの動作速度
 1～255

ポジション: サーボを移動させる場所
 0000h～FFFFh
 +3～+7の動作サーボの指定で1にしたサーボのデータが
 順番に並んでいる

チェックサム: 1byte目のサイズからチェックサムの1byte手前までのデータを加算した下位1byte

【COMから実行した場合】

		1	2	3	4
・処理が成功した	→	SIZE	CMD	ACK (06h)	SUM

		1	2	3	4
・チェックサムエラー	→	SIZE	CMD	NCK (15h)	SUM

【ROMから実行した場合】

・チェックサムエラー → ROM_Fが0になりプログラムが停止する

【その他】

・ICSの割り当てが無いサーボを指定した場合、更新フラグは0になる。
動作に関しては、割り当てが無いので動作しようとししない。
割り当てのないデータを入れても、特に問題ない。

【Rcb4クラスメソッド名】

RunSeriesServo

サーボのストレッチ・スピードの変更

【機能】

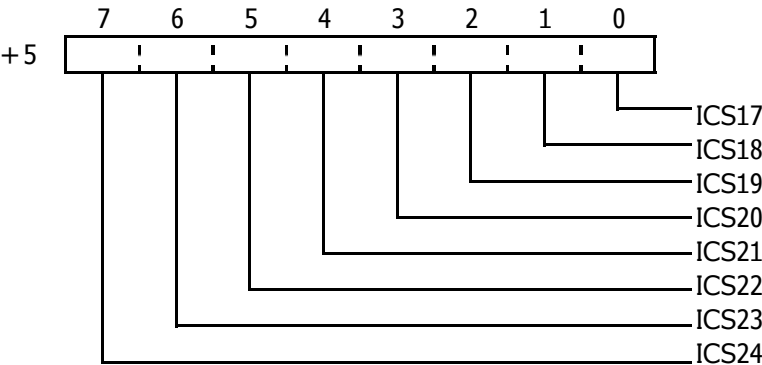
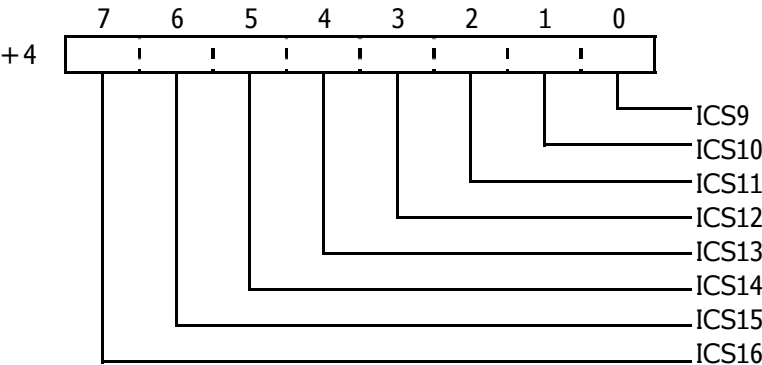
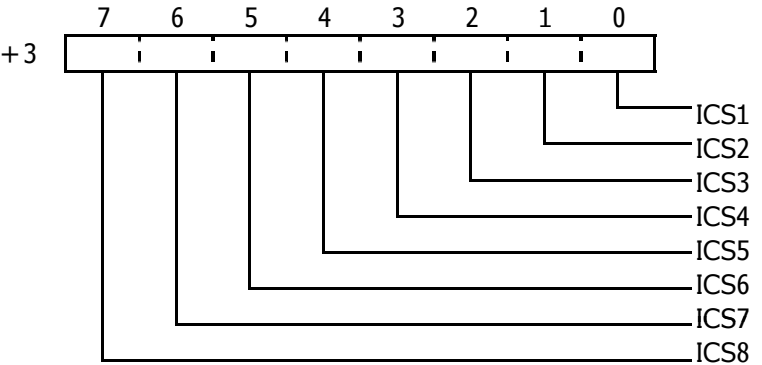
- ・指定したサーボのストレッチやスピードを変更する。
- Change the stretch and speed of the specified servo

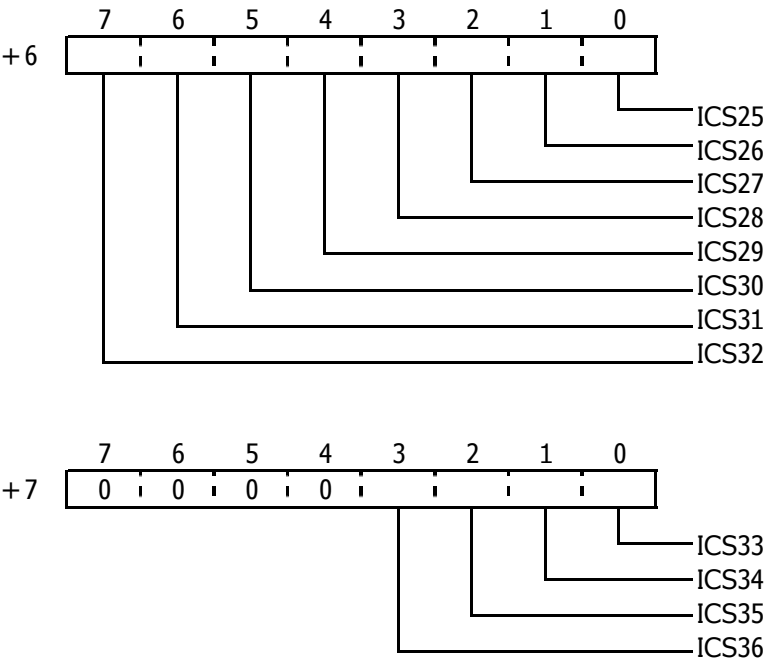
【コマンド構成】

1	2	3	~	7	8	9	~	n	n+1
サイズ	コマンド	ストレッチ・スピードを変更する サーボの設定				サブコマンド	パラメータ		チェック サム

サイズ: コマンド全体のバイト数n+1
コマンド: 複数動作命令: 11h

パラメータを変更するサーボの指定:
 変更するサーボのbitを1にすることで、変更するサーボを決定します。





サブコマンド: ストレッチの変更かスピードの変更かを決定する。

- 01h : ストレッチ
- 02h : スピード

パラメータ: 変更する設定値

- ストレッチ
- スピード : 1~127

+3~+7の変更するサーボの指定で、1にしたサーボの設定値が順番に並んでいる。

チェックサム: 1byte目のサイズからチェックサムの1byte手前までのデータを加算した下位1byte

【COMから実行した場合】

	1	2	3	4
・処理が成功した	SIZE	CMD	ACK (06h)	SUM

	1	2	3	4
・チェックサムエラー	SIZE	CMD	NCK (15h)	SUM

【ROMから実行した場合】

- ・チェックサムエラー → ROM_Fが0になりプログラムが停止する

【その他】

・ICSの割り当てが無いサーボを指定した場合、割り当ての無いサーボへのコマンドだけが出力されません。

【Rcb4クラスメソッド名】

SetSpeed
SetStretch

RCB-4のバージョン情報を確認する

【機能】

RCB-4のバージョン情報を確認する。 Confirm version information of RCB-4

【コマンド構成】

1	2	3
サイズ	コマンド	チェックサム

サイズ: コマンド全体のバイト数=3
コマンド: バージョン: FDh
サム: 1byte目のサイズと2バイト目のコマンドを加算した値の下位1byteをチェックサムとする。

【COMから実行した場合】

1

2

3

~

34

35

04h

CMD

バージョン情報

SUM

→

32byteのバージョン情報文字列が返ってきます。

1

2

3

4

04h

CMD

NCK

SUM

→

コマンドのチェックサムエラー

【ROMから実行した場合】

- ・コマンドのチェックサムエラー → ROM_Fが0になりプログラム実行が停止する。
- ・ROM実行から通信確認コマンドを実行しても、何も起きません。

【Rcb4クラスメソッド名】

GetVersionInfo

変更履歴

2010/8/5	改訂版第1版 HeartToHeart4 Ver.2.0/2.1対応
2013/10/18	改訂版第2版 文章校正のみ行った