RCB-4HV　ファームウェアリファレンスマニュアル

はじめに

ファームウェアリファレンスについて

本ファームウェアリファレンスではRCB-4の内部構造、データ構成などについて説明しています。
実際使用できる命令については「RCB-4コマンドリファレンス」を参考にしてください。

第2版　Rev.20131018

近藤科学株式会社

基本仕様

CPU	20MHz
RAM	2kB
ROM	64kB(ファームウェアで使用しているためプログラムできません)
EEPROM	256kB(128x2)
COM通信速度	115200bps、625000bps、1250000bpsのいずれか 1スタートビット、1ストップビット、8ビットデータ 偶数パリティ
ICS通信速度	115200bps、625000bps、1250000bpsのいずれか 1スタートビット、1ストップビット、8ビットデータ 偶数パリティ

ポート

COMポート	外部通信ポートとして x1
SIOポート	ICSデバイス(主にサーボモーター) x8 (2系統 4x2)
ADポート	10ビット x10(うちAD0番は外部電源の電圧測定用で出力は実際の電圧の1/5)
PIOポート	x10(入出力方向の決定が可能、デジタル入力として使用する場合はプルダウン抵抗が必要)

基本機能

- メモリーマップドIOを採用しているので、RAMの読み書きにてハードウェアが駆動できます
- 起動手順をROMに書き込むと、自動的に読み込まれ、ロボットなどを起動できます。

参照

- RAMエリアからRCB-4のデータ（AD値など）を読み込みたい→RAMアドレスマップ
- RCB-4の通信速度などシステム設定値を変えたり、読み出したい→システムレジスタ
- RCB-4の動作内容を確認したい→ROMプログラム
- PIOの設定を行う→PIO
- ADポートから値を読み込み、電圧に変換したい→ADコンバーター
- RCB-4に取りつけられたサーボモーターの値を読み込みたい、データを書き込んでサーボモーターを動かしたい→シリアルサーボ

RAMアドレスマップ

RAMアドレスマップはファームウェア上に仮想的に作成されたメモリーアドレスと機能のマッピングデータです。
基本的に読み書き可能ですが、書き込みに関しては各機能に対して動作が変わります。

W 書き込み
○ 書き込み可能で、書き込むことで動作可能
× 書き込み可能だが、書き込みしてはいけない
△ 特定のデータのみ書き込み可能

アドレス	機能	電源投入時の初期値	W
$0000h	システムレジスタ		○
$0002h	プログラムカウンタ	$000000ｈ	○※１
$0005h	スタックポインタのカウンタ	$00h	×
$0006h	未使用	$00h	○不定
$0007h	ＥＥＰＲＯＭ更新フラグ	$0000000000h	×
$000Ch	AD0基準値レジスタ	$0000h	○
$000Eh	AD1基準値レジスタ	$0000h	○
$0010h	AD2基準値レジスタ	$0000h	○
$0012h	AD3基準値レジスタ	$0000h	○
$0014h	AD4基準値レジスタ	$0000h	○
$0016h	AD5基準値レジスタ	$0000h	○
$0018h	AD6基準値レジスタ	$0000h	○
$001Ah	AD7基準値レジスタ	$0000h	○
$001Ch	AD8基準値レジスタ	$0000h	○
$001Eh	AD9基準値レジスタ	$0000h	○
$0020h	AD10基準値レジスタ	$0000h	○
$0022h	AD0測定値レジスタ	不定	×※２
$0024h	AD1測定値レジスタ	不定	×※２
$0026h	AD2測定値レジスタ	不定	×※２
$0028h	AD3測定値レジスタ	不定	×※２
$002Ah	AD4測定値レジスタ	不定	×※２
$002Ch	AD5測定値レジスタ	不定	×※２
$002Eh	AD6測定値レジスタ	不定	×※２
$0030h	AD7測定値レジスタ	不定	×※２
$0032h	AD8測定値レジスタ	不定	×※２
$0034h	AD9測定値レジスタ	不定	×※２
$0036h	AD10測定値レジスタ	不定	×※２
$0038h	PIO入出力設定レジスタ	$03FFh	○
$003Ah	PIO入出力レジスタ	$0000h	○※３
$003Ch	タイマー0レジスタ	$8000h	○
$003Eh	タイマー1レジスタ	$8000h	○
$0040h	タイマー2レジスタ	$8000h	○
$0042h	タイマー3レジスタ	$8000h	○
$0044h	ICS0割り当てアドレス	$FFFFh	△※４
$0046h	ICS1割り当てアドレス	$FFFFh	△※４
$0048h	ICS2割り当てアドレス	$FFFFh	△※４
$004Ah	ICS3割り当てアドレス	$FFFFh	△※４
$004Ch	ICS4割り当てアドレス	$FFFFh	△※４
$004Eh	ICS5割り当てアドレス	$FFFFh	△※４
$0050h	ICS6割り当てアドレス	$FFFFh	△※４
$0052h	ICS7割り当てアドレス	$FFFFh	△※４
$0054h	ICS8割り当てアドレス	$FFFFh	△※４
$0056h	ICS9割り当てアドレス	$FFFFh	△※４
$0058h	ICS10割り当てアドレス	$FFFFh	△※４
$005Ah	ICS11割り当てアドレス	$FFFFh	△※４
$005Ch	ICS12割り当てアドレス	$FFFFh	△※４
$005Eh	ICS13割り当てアドレス	$FFFFh	△※４
$0060h	ICS14割り当てアドレス	$FFFFh	△※４
$0062h	ICS15割り当てアドレス	$FFFFh	△※４
$0064h	ICS16割り当てアドレス	$FFFFh	△※４
$0066h	ICS17割り当てアドレス	$FFFFh	△※４
$0068h	ICS18割り当てアドレス	$FFFFh	△※４
$006Ah	ICS19割り当てアドレス	$FFFFh	△※４

$006Ch	ICS20割り当てアドレス	$FFFFh	△※４
$006Eh	ICS21割り当てアドレス	$FFFFh	△※４
$0070h	ICS22割り当てアドレス	$FFFFh	△※４
$0072h	ICS23割り当てアドレス	$FFFFh	△※４
$0074h	ICS24割り当てアドレス	$FFFFh	△※４
$0076h	ICS25割り当てアドレス	$FFFFh	△※４
$0078h	ICS26割り当てアドレス	$FFFFh	△※４
$007Ah	ICS27割り当てアドレス	$FFFFh	△※４
$007Ch	ICS28割り当てアドレス	$FFFFh	△※４
$007Eh	ICS29割り当てアドレス	$FFFFh	△※４
$0080h	ICS30割り当てアドレス	$FFFFh	△※４
$0082h	ICS31割り当てアドレス	$FFFFh	△※４
$0084h	ICS32割り当てアドレス	$FFFFh	△※４
$0086h	ICS33割り当てアドレス	$FFFFh	△※４
$0088h	ICS34割り当てアドレス	$FFFFh	△※４
$008Ah	ICS35割り当てアドレス	$FFFFh	×
$008Ch	ジャンプベクタアドレス	$FFFFh	△
$008Eh	未使用	$00h	○不定
$0090h ～ $048Eh	ユーザーエリア（1024byte）	不定	○

※

1 プログラムカウンターは現在実行しているプログラムのアドレスを指すので、書き換えるには下記の手順を守ること

1） システムレジスタのROMからの読み込みフラグを下ろし、EERPOMからのプログラム実行を一時停止する
2） プログラムカウンターを書き換える
3） システムレジスタのROMからの読み込みフラグをセットし、プログラムを再起動する

2 測定値に書き込むと、タイミングによりアナログ変換値が 1 フレーム分有効になるときがある。

3 入出力設定レジスタに依存する

4 ユーザーエリア以降をセットすること

5 HeartToHeart 4で設定しているRAMのユーザーエリアについては「RCB-4プログラミングマニュアル」を参照

システムレジスタ

【機能】

RCB4の各機能のスイッチや動作フラグなどが配置してあるレジスタがシステムレジスタです。
システムレジスタ内の機能は下記の通りとなっています。

・ＩＣＳスイッチ (bit0)
このスイッチを１にすることで、ＩＣＳが有効(サーボモーターが動作可能)になります。

・ＲＯＭプログラムスイッチ (bit1)
このスイッチを１にすることで、ＲＯＭプログラムが動作します。
電源投入時は１になっていますので、電源投入と同時にＲＯＭプログラムの実行が始まります。

・補間動作終了メッセージスイッチ (bit2)
このスイッチを１にすることで、サーボの補間動作終了時にCOMからコマンドを出力します。

・ベクタジャンプスイッチ (bit3)
このスイッチを１にすることで、ベクタジャンプ機能が有効になります。

・出力周期レジスタ (bit4,5)
ＩＣＳ機能の処理と通信を行う周期を設定することが出来ます。
設定は下記の通りです。

b5	b4	周期
0	0	10ms
0	1	15ms
1	0	20ms
1	1	25ms

・ＣＯＭボーレートレジスタ (bit6,7)
ＣＯＭ(ＰＣ)での通信速度を設定することが出来ます。
設定は下記の通りです。

b5	b4	ボーレート
0	0	115.2kbps
0	1	625kbps
1	0	1.25Mbps
1	1	115.2kbps

・ゼロフラグ (bit8)
計算、演算結果が０立った場合に１になります。

・キャリーフラグ (bit9)
計算、演算結果で桁上がり桁下がりが発生したときに１になります。

・プログラムエラーフラグ(bit10)
ＲＯＭプログラム実行時にＲＯＭから読み取ってきたプログラムコードで
チェックサムエラーが発生すると１になります。

・ＩＣＳボーレートレジスタ (bit13,14)
ＩＣＳでの通信速度を設定することが出来ます。
設定は下記の通りです。

b5	b4	ボーレート
0	0	115.2kbps
0	1	625kbps
1	0	1.25Mbps
1	1	115.2kbps

・ＬＥＤレジスタ (bit15)
RCB4の基板上の緑ＬＥＤの制御レジスタです
１で点灯、０で消灯となっています。

【レジスタ】

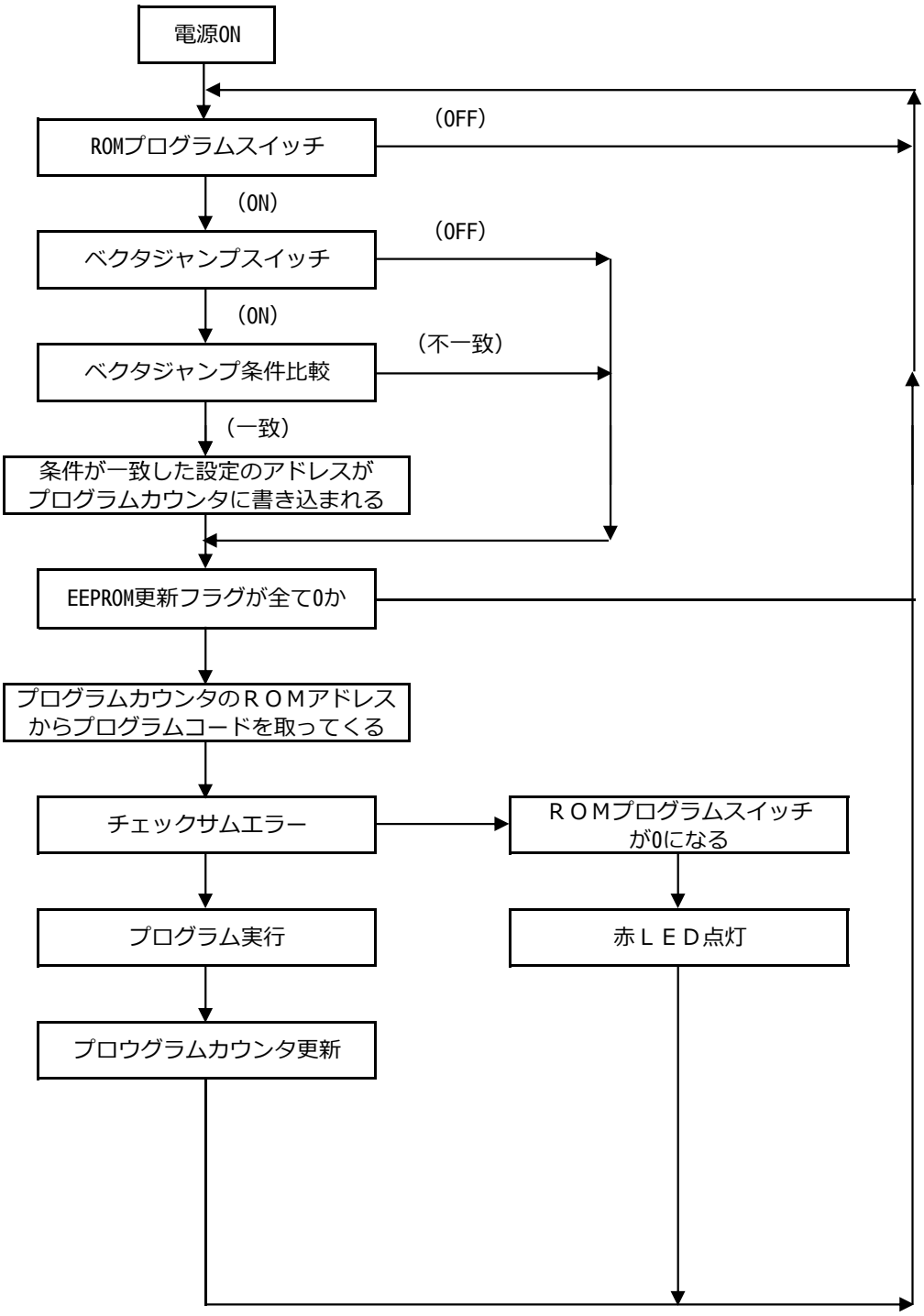
アドレス	機能	電源投入時の初期値
$0000h, 0	ICSスイッチ	0
1	ROMプログラムスイッチ	1
2	補間動作終了メッセージスイッチ	0
3	ベクタジャンプスイッチ	0
4-5	出力周期レジスタ	0 0
6-7	COMボーレートレジスタ	0 0
8	ゼロフラグ	0
9	キャリーフラグ	0
10	プログラムエラーフラグ	0
11	未使用	0
12	未使用	0
13-14	ICSスイッチボーレートレジスタ	0 0
15	LEDレジスタ	0

ROMプログラム

【機能】

電源投入時には、ROMプログラムスイッチが１になりプログラムカウンタが000000hになるので、EEPROMの先頭からプログラム実行が開始されます。
EEPROMからのプログラムコードでチェックサムエラーが発生すると、赤LEDが点灯してROMプログラムスイッチが０になりプログラムが停止します。
プログラムカウンタが実行(読取)ROMアドレスとなっています。システムレジスタのROMプログラムスイッチを１にすることでプログラム実行が開始します。CALLでの呼び出しを行った場合、CALLの次の命令が配置してある先頭アドレスがスタックに保存され、スタックカウンタがインクリメントされます。スタックの上限は16段階となっています。

【処理の流れ】



【主なレジスタ】

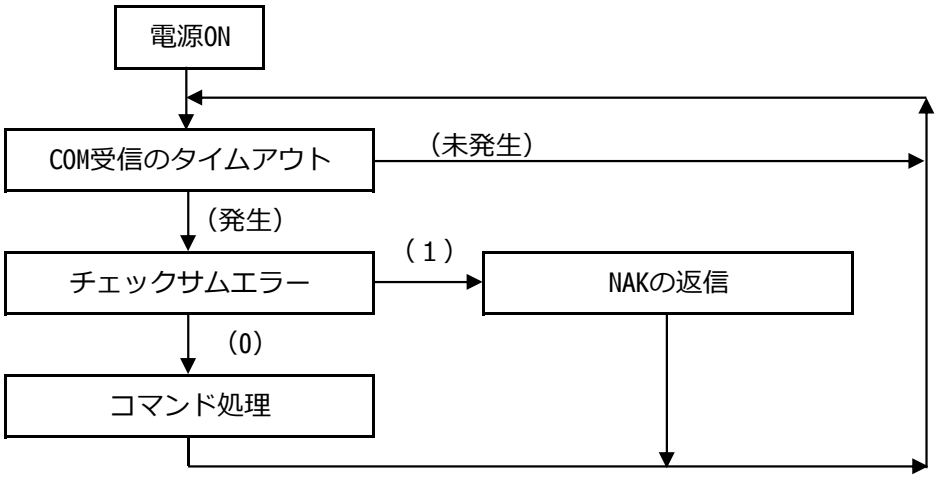
アドレス	機能	電源投入時の初期値
$0000h, 1	ROMプログラムスイッチ	1
$0000h, 3	ベクタジャンプスイッチ	0
$0002h,	プログラムカウンタ	$000000h
$0005h,	スタックポインタのカウンタ	$00h
$0007h,	ＥＥＰＲＯＭ更新フラグ	$0000000000h

COM

【機能】

COMの通信ボーレートはシステムレジスタbit6,7（COMボーレートレジスタ）の設定となります。COMに送るコマンドのタイムアウトは150usとなっていますので、コマンドの1byteと1byteの間隔は150us以内でないと、コマンドとして受信されません。
１回のコマンドで受信できるサイズは最大128byteです。

【処理の流れ】



【レジスタ】

アドレス	機能	電源投入時の初期値
$0000h, 4-5	COMボーレートレジスタ	1

b5	b4	ボーレート
0	0	115.2kbps
0	1	625kbps
1	0	1.25Mbps
1	1	115.2kbps

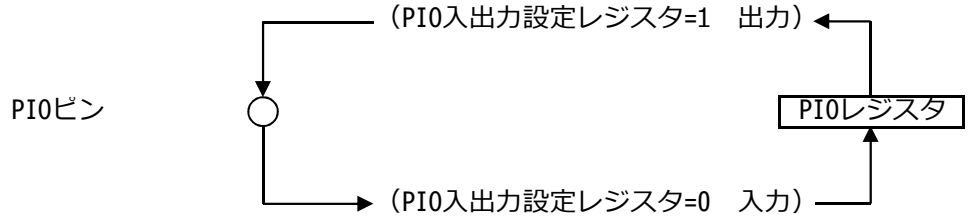
【COMポート設定】

パリティ　偶数

PIO

【機能】

基板上のＰＩＯ(PIO1～PIO10)からTTLレベル(0=GND 1=Vcc)の入出力を行う機能です。
入出力方向の設定はPIO入出力設定レジスタ、出力の設定・入力の状態はPIOレジスタで設定します。



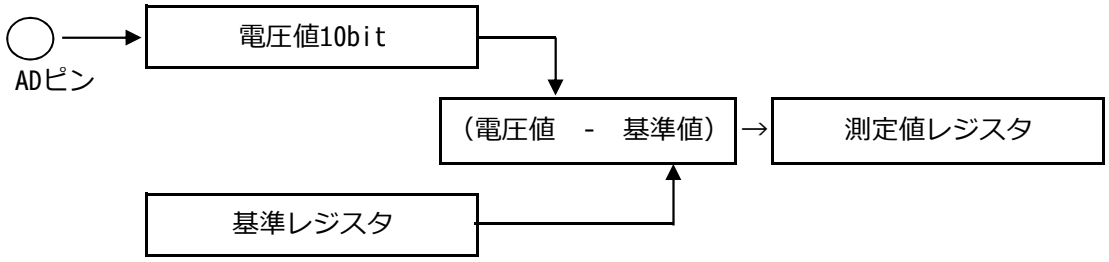
【レジスタ】

アドレス	機能	電源投入時の初期値
$0038h, 0	PIO1入出力設定レジスタ	1
1	PIO2入出力設定レジスタ	1
2	PIO3入出力設定レジスタ	1
3	PIO4入出力設定レジスタ	1
4	PIO5入出力設定レジスタ	1
5	PIO6入出力設定レジスタ	1
6	PIO7入出力設定レジスタ	1
7	PIO8入出力設定レジスタ	1
8	PIO9入出力設定レジスタ	1
9	PIO10入出力設定レジスタ	1
10	未使用	0
11	未使用	0
12	未使用	0
13	未使用	0
14	未使用	0
15	未使用	0
$003Ah, 0	PIO1レジスタ	0
1	PIO2レジスタ	0
2	PIO3レジスタ	0
3	PIO4レジスタ	0
4	PIO5レジスタ	0
5	PIO6レジスタ	0
6	PIO7レジスタ	0
7	PIO8レジスタ	0
8	PIO9レジスタ	0
9	PIO10レジスタ	0
10	未使用	0
11	未使用	0
12	未使用	0
13	未使用	0
14	未使用	0
15	未使用	0

ＡＤコンバーター

【機能】

基板上のＡＤ入力ピン(AD1～AD10、AD0は電源電圧に接続のため入力ピンは無い)に入力している電圧(Vcc[V]～GND[V])を、10bit分解能で計測する機能です。
計測した電圧値は、基準値レジスタの値からの相対値が計測値レジスタに保存されます。



【レジスタ】

アドレス	機能	電源投入時の初期値
$000Ch	AD0基準値レジスタ	$0000ｈ　バッテリー用なので基本的に0
$000Eh	AD1基準値レジスタ	$0000h
$0010h	AD2基準値レジスタ	$0000h
$0012h	AD3基準値レジスタ	$0000h
$0014h	AD4基準値レジスタ	$0000h
$0016h	AD5基準値レジスタ	$0000h
$0018h	AD6基準値レジスタ	$0000h
$001Ah	AD7基準値レジスタ	$0000h
$001Ch	AD8基準値レジスタ	$0000h
$001Eh	AD9基準値レジスタ	$0000h
$0020h	AD10基準値レジスタ	$0000h
$0022h	AD0測定値レジスタ	不定
$0024h	AD1測定値レジスタ	不定
$0026h	AD2測定値レジスタ	不定
$0028h	AD3測定値レジスタ	不定
$002Ah	AD4測定値レジスタ	不定
$002Ch	AD5測定値レジスタ	不定
$002Eh	AD6測定値レジスタ	不定
$0030h	AD7測定値レジスタ	不定
$0032h	AD8測定値レジスタ	不定
$0034h	AD9測定値レジスタ	不定
$0036h	AD10測定値レジスタ	不定

【計算式】

AD1～AD10ポートの電圧

ADポート電圧　＝　（ADポート値 ＋ AD基準値）× 5 ÷ 1024　(V)

AD0ポート(アドレス000Ch：バッテリー電圧専用)は実際の電圧を1/5に分圧したデータとなっています。
AD0ポートにも基準値の設定は可能ですが、基本的には0にしておいてください。

バッテリー電圧　＝　AD0の値 × 25 ÷ 1024　(V)

ダウンタイマー

【機能】

タイマーカウント値レジスタに設定されている値が100ms毎に１づつダウンカウントしていく機能です。設定できる時間は1(100ms)～32767(3276.7s)となっています。
ストップレジスタが０の時にダウンカウントして、０になるとストップレジスタが自動的に１となり、ダウンカウントが停止します。
カウント中にストップレジスタを１にするとダウンカウントを停止させることが出来ます

【レジスタ】

アドレス		機能	電源投入時の初期値
$003Ch	0-14	タイマー０カウント値レジスタ	$0000 h
	15	タイマー０ストップレジスタ	$0000h
$003Eh	0-14	タイマー１カウント値レジスタ	$0000h
	15	タイマー１ストップレジスタ	$0000h
$0040h	0-14	タイマー２カウント値レジスタ	$0000h
	15	タイマー２ストップレジスタ	$0000h
$0042h	0-15	タイマー３カウント値レジスタ	$0000h
	16	タイマー３ストップレジスタ	$0000h

【備考】

設定したいカウント値が255以下でも２バイトで設定すること

比較ジャンプベクタ

【機能】

比較ジャンプベクタとは、EEPROMからのプログラム実行中に指定している比較条件が一致した場合に指定したアドレスにジャンプまたは、アドレスをコールすることが出来る機能です。

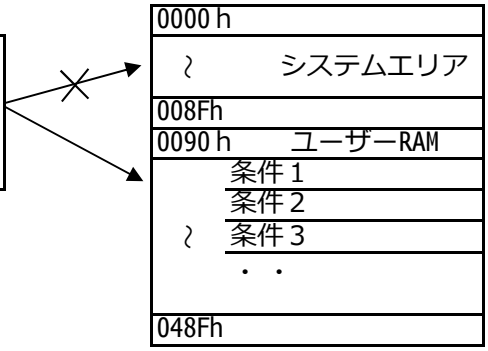
【使用方法】

ジャンプベクタ機能を使用する場合、以下の設定を行う必要があります。

1． ユーザーRAM機能に必要なレジスタを配置する。

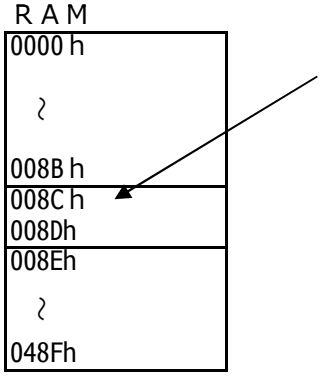
ジャンプベクタに必要な下記の項目を1パッケージとして必要な条件の数だけ
ユーザーRAMの任意の場所に配置します。

配置場所	サイズ	用途
+0バイト目	B	比較用途
+1バイト目	W	比較先の値
+2バイト目	W	比較元の値
+5バイト目	3byte	飛び先番地



2． ユーザーＲＡＭに配置したレジスタの先頭アドレスの登録

ユーザーＲＡＭへの配置先を所定のレジスタに登録します。



アドレス008Chがジャンプベクタの登録先となります。
たとえば、ユーザーエリアの00A0hから比較条件を5個並べた場合（5x8byte=40byte）、アドレス008Chに00A0hを書き込むことでユーザーエリア00A0hからの40byteがジャンプベクタ用のレジスタになります。

3． 動作スイッチの設定

動作スイッチ→SYSTEMの1bit ： 1にすると、プログラム実行中にのみジャンプベクタを使用することが出来ます。

ベクタスイチ→SYSTEMの3bit ： 0：OFF
1：ON（条件が一致してジャンプすると自動的にOFFになる）

【比較条件】

このレジスタは、比較元・先の種別、比較に使用するフラグ、フラグの条件、ユーザーレジスタに並べた条件の末尾であるか？ジャンプかコールか？の設定を入れておくレジスタです。

比較条件の構成

7	6	5	4	3	2	1	0
EOF	JUMP	SIZE	LITE	CE	ZE	C	Z
MSB							LSB

BIT	名称	機能
7	EOF	エンドオブファイル.ユーザーエリアに1パッケージごとに並べた条件の末尾の条件であるか否か？ 1=末尾 0=末尾ではない ※注意：必ず末尾のパッケージには1を入れること。
6	JUMP	条件が一致したときのジャンプ方法 1=JUMP 0＝CALL
5	SIZE	比較元・先のデータのサイズ 1=WORD 0=BYTE
4	LITE	配置場所の+3byte目の設定値がリテラル値であるか否か 1=リテラル 0=それ以外
3	CE	一致条件にCフラグを使用するか否か 1=ON 0=OFF
2	ZE	一致条件にZフラグを使用するか否か 1=ON 0=OFF
1	C	比較計算後に変化するCフラグの一致条件
0	Z	比較系産後に変化するZフラグの一致条件

[比較先・比較元]
このレジスタに設定してあるRAMレジスタ、ICSの割り当て又はリテラル値同士を計算してフラグの変化を求めます。

計算方法

+1byte目（先）	-	+3byte目（元）	⇒	計算結果が0だったらZ=1 そうでなければZ=0 計算結果が0未満の場合C=0 0以上の場合はC=1

*計算するデータサイズは比較条件の5bit目の通りです。

[RAMアドレスの指定方法]

比較先・比較元でRAMアドレスを指定する場合は、2byteのレジスタにそのままRAMアドレスを入れることでRAMアドレス指定となります。

[ICS割り当ての指定方法]

比較先・比較元でICS割り当てを指定する場合は2byteのレジスタを以下の設定にすることで割り当てられます。

0-7bit	ICS割当先のオフセットアドレス
8-14bit	ICS番号
15bit	ICS割り当てを使用のため、1にする。

[リテラルの指定方法]

比較元のみに有効な設定です。
比較条件の4bit目を1にすることで、比較元の2byteの値がリテラル値扱いとなります。

[飛び先番地]
このレジスタは、条件が一致してジャンプ又はコールするときのEEPROMアドレスを入れておくレジスタです。ジャンプ方法は比較条件の6bit目の通りです。

[使用例]
手順1. RAMアドレス008ChにOOAOhを入れる

手順2.　RAMアドレス00A0hから

+0 45	+1 90	+2 00	+3 91	+4 00	+5 21	+6 43	+7 00
+8 BD	+9 02	+10 80	+11 0A	+12 05	+13 00	+14 00	+15 00

の16byteのデータを入れる。

手順3.　プログラム実行中（SYSTEMの1bit=1）である状態で、SYSTEMの3bit目を1にする。

上記の手順の設定を行うと、以下の比較が常時行われる。

比較の順番

1

RAMアドレス0090hの値1byte - RAMアドレス0091hの値1byte

以上の結果が0（Z=1)だったら、ROMアドレス004321hにジャンプする。

2

ICS0の割当先の+2byte目の値1WORD - リテラル値　050Ah

以上の結果が0未満（C=0、Z=0）だったらROMアドレス000000ｈをコールする。

シリアルサーボ

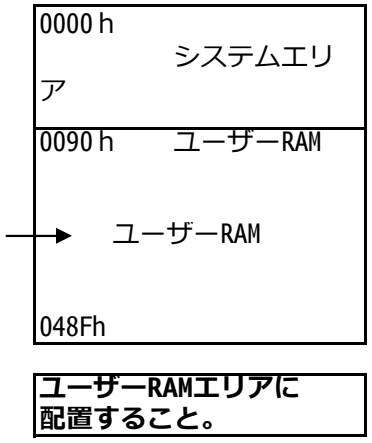
【使用方法】

シリアルサーボを使用する場合、以下の設定を行う必要があります。
1. ユーザーRAMに動作に必要なレジスタを配置する。
2. ユーザーRAMに配置したレジスタの先頭アドレスの登録。
3. 通信速度、通信周期、出力スイッチの設定。

[ユーザーRAMに、動作に必要なレジスタを配置する]

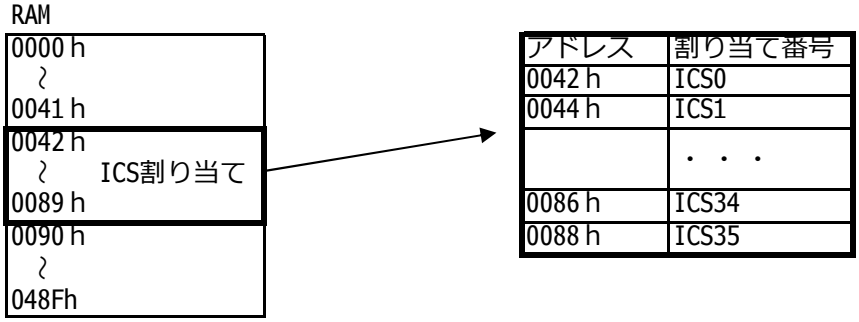
サーボ動作に必要な下記の項目をユーザーRAMの任意の場所に配置します。
HeartToHeart4でプロジェクトを書き込むと、プロジェクトに登録されたサーボモーターの設定を
ROMに保存します。RCB-4起動時にサーボモーターのデータが自動的にユーザーRAMへ配置されます。

配置場所	サイズ	用途
+0byte目	B	レジスタ配置の内容種別
+1byte目	B	シリアルサーボのID
+2byte目	W	基準値、調整用の設定値、トリム
+4byte目	W	実測値、サーボのポジション、現在位置
+6byte目	W	出力値、サーボへ送るポジション、目標位置
+8byte目	B	補間動作時の補間速度
+9byte目	W	補間動作終了時のポジション
+11byte目	B	補間動作中の補間段数
+12byte目	W	補間動作時の動作幅
+14byte目	W	ミキシング１の元となるデータのアドレス
+16byte目	B	ミキシング１の計算方法
+17byte目	W	ミキシング２の元となるデータのアドレス
+19byte目	B	ミキシング２の計算方法



[ユーザーRAMに配置したレジスタの先頭アドレスの登録]

ユーザーRAMへの配置先をICS割り当てに登録します。



例えば、ユーザーエリアの00A0hから20byteのサーボ動作用レジスタを配置した場合、アドレス
0042 h に配置した00A0 h を書き込むと、ICS0番の設定値が00A0hからのレジスタになります。
RCB-4のファームウェアは00A0hのデータを読み書きします。

[通信速度、通信周期、出力スイッチの設定]

サーボとの通信速度、通信周期、出力スイッチは、アドレス0000 h のSYSTEMの設定値
の通りとなります。

通信速度 → SYSTEMの13,14bit
00： 115.2kbps
01： 625kbps
10： 1.25Mbps

通信周期 → SYSTEMの4,5bit
00： 10ms
01： 15ms
10： 20ms
11： 25ms

出力スイッチ→ 0bit
0： OFF（出力せず）

1： ON

［レジスタ配置の種別内容（+0byte目）］

このレジスタは、ユーザーRAMに配置した20byteが、サーボ用のレジスタであるということを
ススステム側が認識するためのれじすたです。ｻｰﾎﾞ用の配置は0を書き込んでください。

［シリアルサーボ（+1byte目）］

このレジスタは、通信を行うシリアルサーボのIDを登録しておくレジスタです。

[基準値（+2byte目） 出力値（+6byte目）]

このレジスタは、サーボへの送信データ（ポジション値）を構成するためのレジスタです。
RCB-4の内部では、以下のような計算が行われ送信データ（ポジション値）が
構成されます。なお、出力値に+32767か-32768を入れるとサーボへの送信データが
その他のレジスタ値とは関係なく0が出力され、サーボはFREEとなります。

ミキシング1の計算値	+	ミキシング2の計算値	+	出力値 +6byte	+	基準値 +2byte	⇒	サーボへの送信データ

出力値の数値について-32768～0～+32767を範囲としているので、
出力値を相対値（-32768～0～+32767）にして、基準値を絶対値（3000～7500～12000）と
したり、出力値を絶対値（3000～7500～12000）にして、基準値を相対値（-32768～0～+32767）
にすることが可能です。

[実測値（+4byte目）]

このレジスタはサーボから戻ってくるポジション値を下記の計算値に通した
後の結果が保存されます。

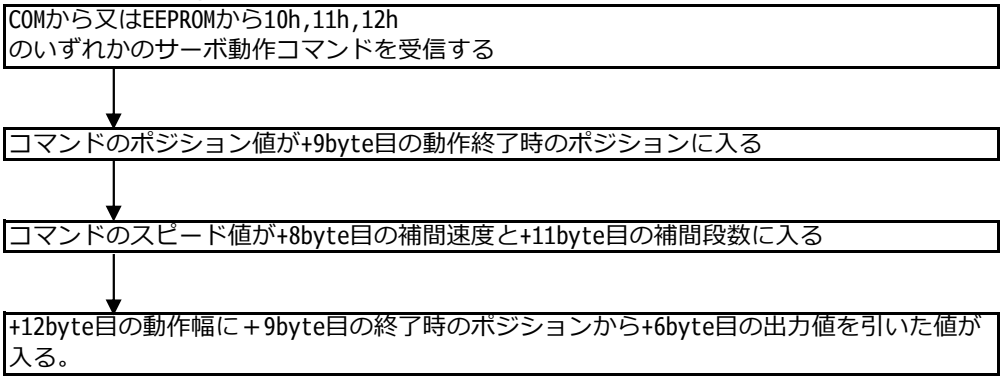
サーボからの	-	基準値 +2byte目	⇒	実測値 +4byte目

【補間動作（+8byte目～+12byte目）】

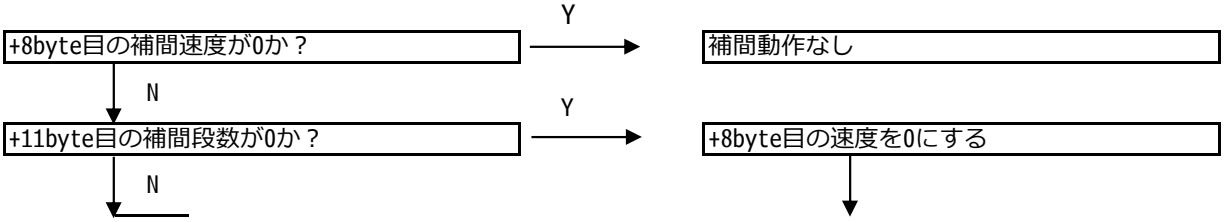
サーボの補間動作を実現するには、+8byte目から+12byte目までの4項目の
の設定値を使用します。

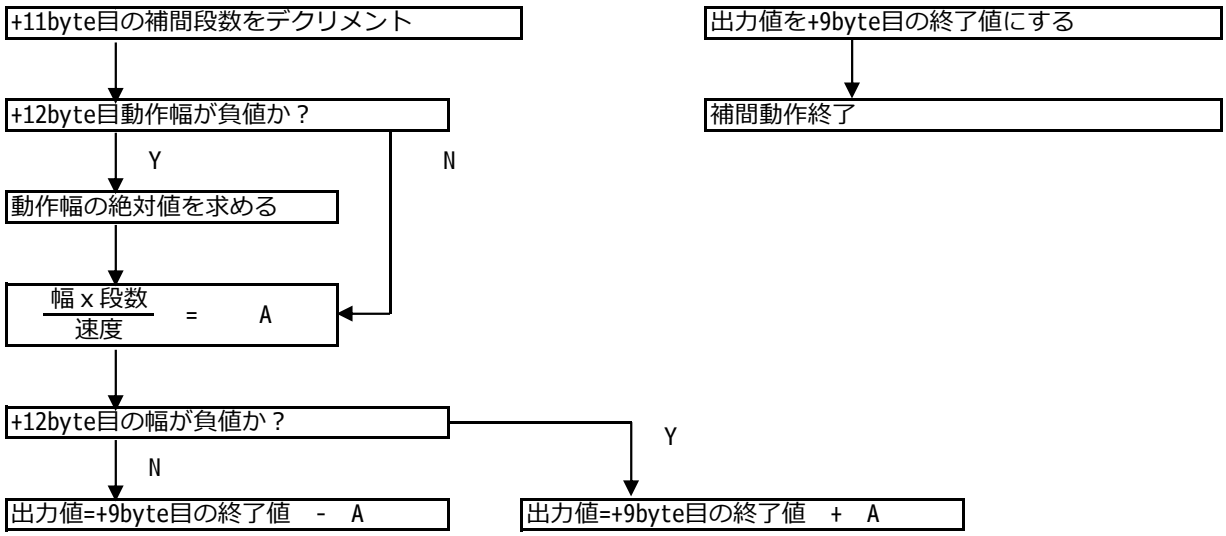
設定から動作までの流れは下記の通りです。

（コマンド処理）



（出力周期毎の処理）

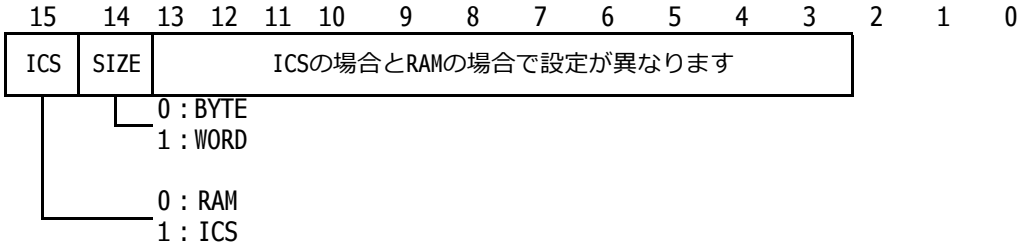




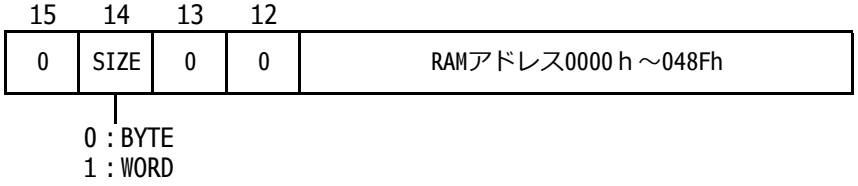
【ミキシング（+14～＋19）】

サーボの出力値にミキシングをかけることが出来ます。
ミキシングに使用する設定値は以下の通り。

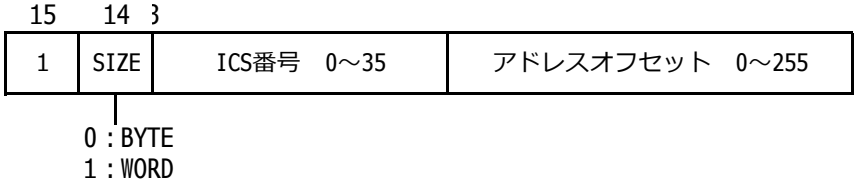
ミキシングの元となるデータのアドレス（+14byte、+17byte目）



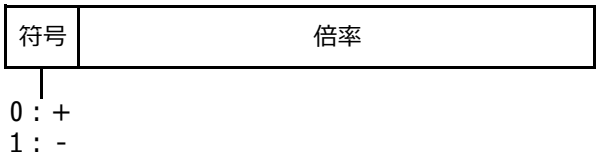
データの種類がRAMの場合



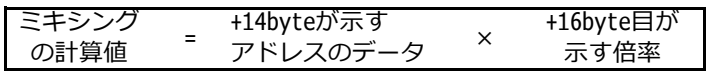
データ種類がICSの場合



ミキシングの計算方法（+16byte目、+19byte目）



ミキシングの計算



動作については、サーボへの信号を出力する毎にミキシングを行っています。
サーボへのポジション出ry口は、下記計算値となります。

サーボへの出力値	=	+6byte目の出力値	+	ミキシング1計算値	+	ミキシング2計算値

【その他】

サーボモーターのアドレス設定や運用は難しいので、HeartToHeart4で登録することをお薦めします。
また、データの読み書き時はアドレスを使わずにICS番号を使うことをお薦めします。詳しくは
RCB-4コマンドリファレンスのMOVコマンドを参照してください。なおRcb4.dllではサーボモーターの
データ読み込み関数があらかじめ準備されています。

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