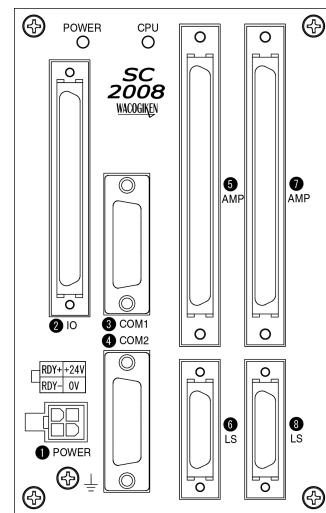
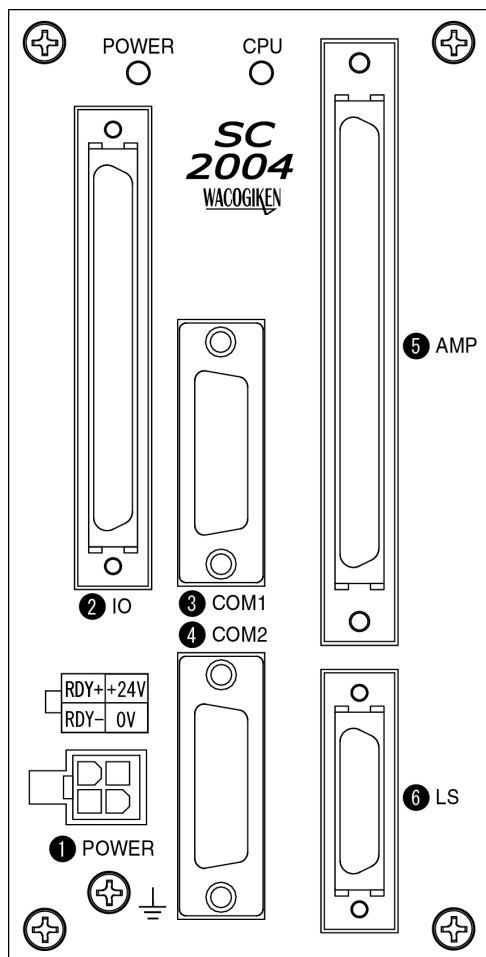


サーボコントローラ

SC2000シリーズ

取扱説明書

V1.0



株式会社 ワコー技研

改定履歴

番号	日付	改定内容
第1版	2001年5月1日	初版
第2版	2001年8月1日	誤記訂正 円弧補間機能の追加 モーション命令追加

目次

目次

1. はじめに	14
1.1. 製品概要	14
1.2. 一般仕様	14
1.3. 保守、点検について	14
1.4. 設置	15
1.5. フラッシュメモリーについて.....	15
1.6. システム構成.....	16
1.7. 外観	17
1.8. 前面パネル	19
1.9. LED表示	21
2. 接続	22
2.1. 各コネクタのピン接続	22
2.1.1. 電源コネクタ.....	22
2.1.2. IOコネクタ.....	23
2.1.3. AMPコネクタ	25
2.1.4. LSコネクタ.....	28
2.1.5. 通信コネクタ.....	29
2.2. 入出力仕様	30
2.2.1. 電源入力	30
2.2.2. 電源出力	30
2.2.3. 入力信号	30
2.2.4. 出力信号	31
2.3. 電源系統図	31
2.4. RDY出力について	32
2.5. サーボアンプについて	32
2.6. エンコーダのフィードバックについて.....	32
2.7. TRIG入力について	32
2.8. 駆動限界リミットスイッチの検出.....	33
2.9. サーボアンプのアラーム入力.....	33
2.10. 電源投入時の信号タイミング	33
3. プログラミング概要	34
3.1. モーション.....	34
3.2. ラダー	34
3.3. 処理の流れ	34
3.4. プログラミングの要点	35
4. モータの運転	36
4.1. モータの運転方法の説明	36

4.1.1.	ジョグ運転	36
4.1.2.	原点復帰	37
4.1.3.	単軸運転	40
4.1.4.	多軸運転	41
4.1.5.	円弧運転	42
4.1.6.	オーダー運転	43
4.1.7.	単位時間運転	45
4.2.	ユーザー単位	45
4.3.	速度設定	45
4.4.	非常停止	46
4.5.	ソフトリミット	46
4.6.	無限送りについて	47
4.7.	速度モデル	47
4.8.	現在値の取り込み機能	47
4.9.	エラーからの復帰について	48
4.10.	オーバーライド	48
5.	パラメータ	49
5.1.	共通パラメータ	49
5.2.	通信パラメータ	50
5.2.1.	SCLinkについて	51
5.2.2.	タッチパネルの接続について	51
5.3.	各軸パラメータ	52
6.	レジスタ	54
6.1.	入出力レジスタ	54
6.2.	軸レジスタ	55
6.3.	システムレジスタ	57
6.4.	駆動レジスタ	59
6.4.1.	位置レジスタ	59
6.4.2.	速度レジスタ	59
6.4.3.	オーダー	60
6.5.	キープレジスタ	60
6.6.	ファイルレジスタ	61
6.7.	リンクレジスタ	61
6.8.	拡張レジスタ	61
6.9.	パラメータレジスタ	61
7.	モーションの説明	62
7.1.	文法	62
7.2.	数値	62
7.3.	レジスタの参照	62
7.4.	モーション専用のレジスタ	62
7.5.	演算子	63
7.6.	数式	63

目次

7.7.	プログラム	63
7.8.	ラベル	63
7.9.	スタック	64
7.10.	命令語	64
7.10.1.	prg	64
7.10.2.	sub	64
7.10.3.	if、elseif、else、endif	65
7.10.4.	goto	65
7.10.5.	gosub	65
7.10.6.	return	66
7.10.7.	wait	66
7.10.8.	stop	66
7.10.9.	movh	66
7.10.10.	movs、movsi	67
7.10.11.	movm、movmi	67
7.10.12.	movc、movci	67
7.10.13.	movo	68
7.10.14.	movt、movti	68
7.10.15.	pset	68
7.10.16.	padj	69
7.10.17.	reset	69
7.10.18.	代入文	69
7.10.19.	コメント文	69
7.11.	関数	69
7.11.1.	sin()、cos()、tan()	70
7.11.2.	asin()、acos()、atan()、atan2()	70
7.11.3.	sqrt()	70
7.11.4.	pow()	70
7.11.5.	ceil()、floor()	70
7.11.6.	float()、int()	71
7.11.7.	abs()	71
7.11.8.	PI	71
7.11.9.	PI2	71
8.	ラダーの説明	72
8.1.	文法	72
8.2.	アキュムレータとスタック	72
8.3.	命令一覧表	73
8.4.	基本命令	74
8.4.1.	LD、LDN	74
8.4.2.	LDA	74
8.4.3.	OUT	74
8.4.4.	AND、ANDN	75
8.4.5.	ANDS	75
8.4.6.	OR、ORN	76

8.4.7.	ORS	76
8.4.8.	LDSA	76
8.4.9.	LDS.....	76
8.4.10.	XOR、XORN	77
8.4.11.	CPL	77
8.4.12.	RAISE	77
8.4.13.	FALL	77
8.4.14.	MC.....	78
8.4.15.	MCR.....	78
8.4.16.	TMR、TMRr、TMRI、TMRIr.....	79
8.4.17.	TMRRST、TMRRSTr.....	79
8.4.18.	EQ、NE、GE、GT、LE、LT、EQI、NEI、GEI、GTI、LEI、LTI.....	80
8.5.	応用命令	81
8.5.1.	SET、SETr.....	81
8.5.2.	RES、RESr.....	81
8.5.3.	MOV、MOVr、MOVI、MOVIr.....	82
8.5.4.	ADD、ADDr、ADDI、ADDIr.....	82
8.5.5.	SUB、SUBr、SUBI、SUBIr.....	83
8.5.6.	MUL、MULr、MULI、MULIr	83
8.5.7.	DIV、DIVr、DIVI、DIVIr	84
8.5.8.	NEG、NEGr.....	84
8.5.9.	SFT、SFTr.....	84
8.5.10.	LDBCD、LDBCDr	85
8.5.11.	OUTBCD、OUTBCDr.....	85
8.5.12.	XTRACT、XTRACTr	86
8.5.13.	MERGE、MERGEr.....	87
8.5.14.	MOVB、MOVBr	88
8.5.15.	MOVbW、MOVbWr.....	89
8.5.16.	MOVBL、MOVBLr.....	90
8.5.17.	SIX、DIX	90
8.5.18.	IXR.....	91
8.5.19.	NOP.....	91
9.	プログラム例.....	92
9.1.	ラダーによるプログラム	92
9.1.1.	ジョグ運転.....	92
9.1.2.	モーションの開始	92
9.2.	モーションによるプログラム.....	92
9.2.1.	原点復帰	92
9.2.2.	インデックスを使った全軸一括位置決め	93
10.	サーボアンプとの接続例.....	94
10.1.	GPP型サーボアンプとの接続	94
10.2.	AFS型サーボアンプとの接続.....	95

目次

安全にお使いいただくために

ご使用前に必ずお読み下さい

取り扱い、注意点について記述しています。

ご使用前に本書を一読され、正しくご使用いただきますようお願いいたします。

なお、本書は、ご使用になるお客様のお手元に届くようご配慮をお願いいたします。

また、必ず保管して下さい。

安全上のご注意

据付、運転、保守・点検の前に必ず本書と取扱説明書および付属書類をすべて熟読し、正しくご使用下さい。機器の知識、安全の情報そして注意事項のすべてについて習熟してからご使用ください。

この取扱説明書では、安全注意事項のランクを「危険」、「注意」として区別してあります。




取扱いを誤った場合に、危険な状況がおりえて、死亡または重傷を受ける可能性が想定される場合。



取扱いを誤った場合に、危険な状況がおりえて、中程度の障害や軽傷を受ける可能性が想定される場合および物的損害だけの発生が想定される場合。



なお、に記載した事項でも、状況によっては重大な結果に結びつく可能性があります。いずれも重要な内容を記載していますので必ず守って下さい。







本書では必要に応じて下記の図記号を用いています。

図記号	図記号の意味
	一般的な禁止の通告
	分解禁止
	強制(必ずすること)
	必ずアース端子を接地して下さい







図記号	図記号の意味
	接触禁止
	発火注意
	感電注意
	破裂注意



【運搬・据付について】

 注意	
	サーボコントローラ・サーボドライバ・サーボモータは、精密機器なので、落下させたり、強い衝撃をあたえないようにしてください。
	製品の過積載は荷崩れの原因となりますのでご注意ください。
	製品の上にとったり、重いものを載せたりしないでください。 けが、製品破壊の恐れがあります。
	サーボモータ運搬時には、モータケーブルやモータ軸を持たないでください。 故障、けがの恐れがあります。
	取付方法は必ずお守りください。機器の放熱は規定に従い行ってください。 火災、故障の恐れがあります。 (詳細は、取扱説明書を参照ください。)
	排気口をふさいだり、異物が入らないようにしてください。 火災の恐れがあります。
	サーボコントローラ、サーボドライバと制御盤内面またはその他の機器との間隔は規定の距離を明けてください。故障の恐れがあります。 (詳細は、取扱説明書を参照ください。)
	外付け回生抵抗器を使用する場合は、異常信号で電源を遮断してください。回生トランジスタの故障などにより、回生抵抗器が異常加熱し火災の恐れがあります。
	腐食性ガス、油沫、塵埃、水蒸気、金属粉等のある場所への据付けはしないでください。故障の恐れがあります。
	質の悪い電源への接続(変動±10%以上、パルスノイズ1KV以上)はしないでください。故障の恐れがあります。
	振動の激しい場所や密閉された場所への据付けはしないでください。 故障の恐れがあります。
	周囲温度:サーボコントローラ、サーボドライバ:0℃~45℃ (製品によって0℃~50℃のものがあります。 取扱説明書参照) サーボモータ:0℃~40℃
	周囲湿度:サーボコントローラ、サーボドライバ、サーボモータ 85%RH 以下(結露のないこと)





 注意	
	サーボモータは確実に機械へ固定してください。固定が不十分だと運転時に外れることがあります。けが、故障の恐れがあります。
	運転中、サーボモータの回転部には絶対触れないよう、軸にはカバーなどを設けてください。けがの恐れがあります。
	サーボモータの軸端へカップリング結合するときに、ハンマーでたたくなどの衝撃をあてないでください。検出器やベアリングの故障の恐れがあります。
	サーボモータ軸へ許容荷重以上の荷重を与えないでください。ベアリングの破損、軸折損の恐れがあります。
	サーボモータの軸や本体の加工は絶対にしないでください。故障の恐れがあります。









【配線について】

 危険	
	配線作業や点検は専門の技術者が行ってください。配線は正しく確実に行ってください。サーボモータの暴走や感電の恐れがあります。
	配線材は規定の容量のものをご使用ください。発熱により火災の恐れがあります。
	サーボドライバのアース端子(E 又はFG 端子)は必ず接地してください。感電の恐れがあります。
	ノイズ耐量の向上、放射ノイズの低減を図るためにも必ず接地してください。接地の方法は、第3種接地(100Ω以下、φ1 . 6mm以上)を推奨します。
	ケーブルは傷つけたり、無理なストレスをかけたり、重いものを載せたり、挟み込んだりしないでください。感電の恐れがあります。




	<p>端子接続を間違えないでください。また、決められた電圧以外は印加しないでください。破裂・破損などの恐れがあります。</p>
	<p>サーボドライバが故障した場合は、サーボドライバの電源側で電源を遮断してください。大電流が流れ続けると火災の恐れがあります。</p>

【操作・運転について】




 危険	
	<p>通電中および運転中は、本体カバーを開けないでください。感電の恐れがあります。</p>
	<p>本体カバーをはずしての運転は行わないでください。高電圧の端子および充電部が露出していますので感電の恐れがあります。</p>
	<p>電源OFF時でも配線作業、定期点検以外では、本体カバーをはずさないでください。ドライバ内部は充電されており、感電の恐れがあります。</p>




 注意	
	<p>運転前に各設定値の確認調整を行ってください。機械によっては予期しない動きとなる場合があります。</p>
	<p>極端な調整変更は動作が不安定になりますので決して行わないでください。又、点検（モニタ）端子をショートさせないよう注意してください。けがの恐れがあります。（詳細は、取扱説明書を参照ください。）</p>
	<p>即時に運転停止し、電源を遮断できるように外部に非常停止回路を設定してください。</p>
	<p>試運転はモータを固定し、機械系と切り放した状態で動作確認してから、機械に取り付けてください。機械の損傷及びけがの恐れがあります。</p>
	<p>アラーム発生時は原因を取り除き、安全を確保してからアラームリセット後、再運転してください。（詳細は、取扱説明書を参照ください。） けがの恐れがあります。</p>
	<p>サーボモータとサーボドライバは指定された組み合わせでご使用ください。 破損の恐れがあります。</p>
	<p>保持ブレーキは、機械の安全を確保するための停止装置ではありません。機械側に安全を確保するための停止装置を設置してください。 けがの恐れがあります。</p>



安全にお使いいただくために

	瞬停復電後、突然再始動する可能性がありますので、十分注意が必要です。 (再始動しても人に対する安全性を確保するよう機械設定を行ってください。) けがの恐れがあります。
	ノイズフィルタなどにより電磁障害の影響を小さくしてください。サーボドライバの近くで使用される電子機器に電磁障害を与える恐れがあります。
	サーボドライバの放熱器や外付け回生抵抗器、サーボモータのフレーム等は高温になることがありますので不用意にふれないでください。 やけどの恐れがあります。


【保守・点検・部品について】


 危険	
	点検は入力電源を遮断(OFF)し、3分以上経過してから行ってください。 感電の恐れがあります。
	通電状態での点検は行わないでください。 感電の恐れがあります。

 禁止	
	モータのセンサー部はメガーテストおよび耐圧試験を行わないでください。 制御回路を破損させます。
	分解、改造、修理は絶対にしないでください。 無断で行った修理により生じた事故については、一切責任を負いません。



 注意	
	電源ラインのコンデンサは、劣化により容量低下をします。故障による二次災害を防止するため、5年程度で交換されることを推奨します。 その際は弊社営業までご連絡ください。

【廃棄について】

 注意	
---	--

	一般産業廃棄物として処理してください。
---	---------------------

【その他】

 注意	
	当製品の品質確保には最大限の努力を払っておりますが、予想以上の外来ノイズ、静電気や部品、端子配線等の万一の異常により設定外の動作をすることがありますので、貴社機械やその周辺の安全性には十分なご配慮をお願いします。 けがの恐れがあります。

【保証について】

無償保証期間とその保証範囲

1. 無償保証期間

適正な使用範囲に於いて、貴社または貴社顧客殿での使用開始から12ヶ月未満、または当社出荷後18ヶ月未満のうちいずれか短い方を採用します。

2. 保証の範囲

下記4項目は有償、これ以外は無償の扱いといたします。これらにのらない場合は双方の協議によることと致します。

- 1) 貴社および貴社顧客殿などに於いて、不適切な保管や取扱い、不注意過失及び貴社側の設計に起因する故障の場合。
- 2) 当社の了解なく、貴社側で当社の製品に改造・分解等の手をかけたことに起因する故障の場合。
- 3) 当社製品の仕様範囲外で使用したことに起因する故障の場合。
- 4) その他に、貴社が当社の責任外と認める故障の場合。

なお原則として、修理の対応は日本国内のみとします。保証期間外並びに海外の修理については、その費用、送料は貴社負担とさせていただきます。

また、当社製品以外への損傷、その他の処置に対する補償は対象外とさせていただきます。

1. はじめに

1.1. 製品概要

SC2000シリーズはサーボモータ用のコントローラで、パルス列入力タイプのサーボアンプに接続して使用します。プログラム機能を備えていますので単独でサーボシステムを制御することができます。モータの運転は、JOG運転、原点復帰、位置決め、連続、時間の駆動が行えさまざまなシステムに対応可能です。

プログラムはラダーとモーションの2つの言語を搭載しています。目的に応じて使い分けることで効率よくプログラム作成が行えます。

SC2000シリーズの標準機種は以下のとおりです。この取扱説明書は以下の機種について書かれています。

型式	SC2004	SC2008
制御軸数	4軸	8軸
入力点数	32点	
出力点数	32点	
シリアルポート	2チャンネル	
付属品	電源コネクタ 1個	
外形(突起含まず)W×H×D	72mm×160mm×105mm	90mm×160mm×105mm
質量	約0.6Kg	約0.8Kg

オプションで最大32軸、IO各128点までの拡張が行えます。詳しくは弊社営業までお問い合わせください。

1.2. 一般仕様

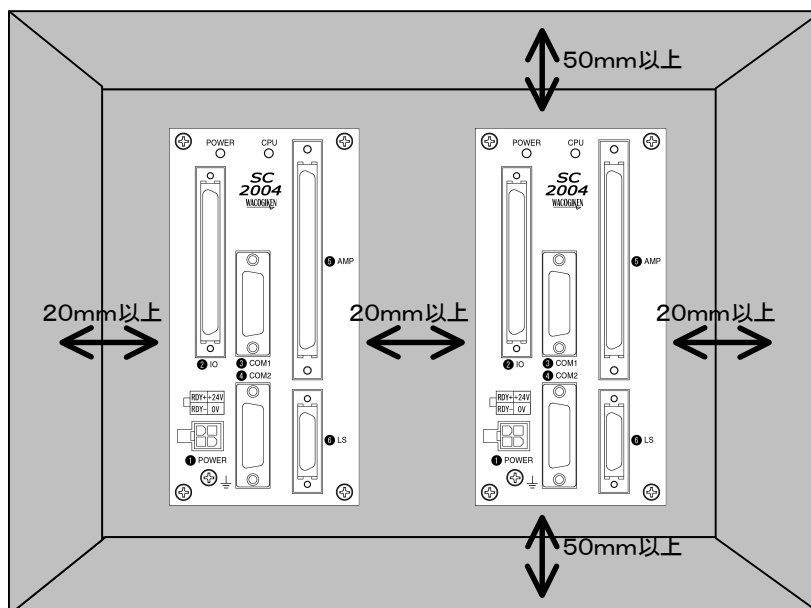
項目	仕様
保存温度	-20~70℃
周囲温度	0~55℃
周囲湿度	35~90RH(結露なきこと)
消費電流	最大1.5A
雰囲気	腐食性ガスのないこと

1.3. 保守、点検について

SC2000シリーズは電池等の消耗品は使用しておりませんので、長期間の使用にも安心してご使用いただけます。

1.4. 設置

本機は垂直面への取り付けを標準としています。上下の通風孔をふさがないように取り付けてください。



1.5. フラッシュメモリーについて

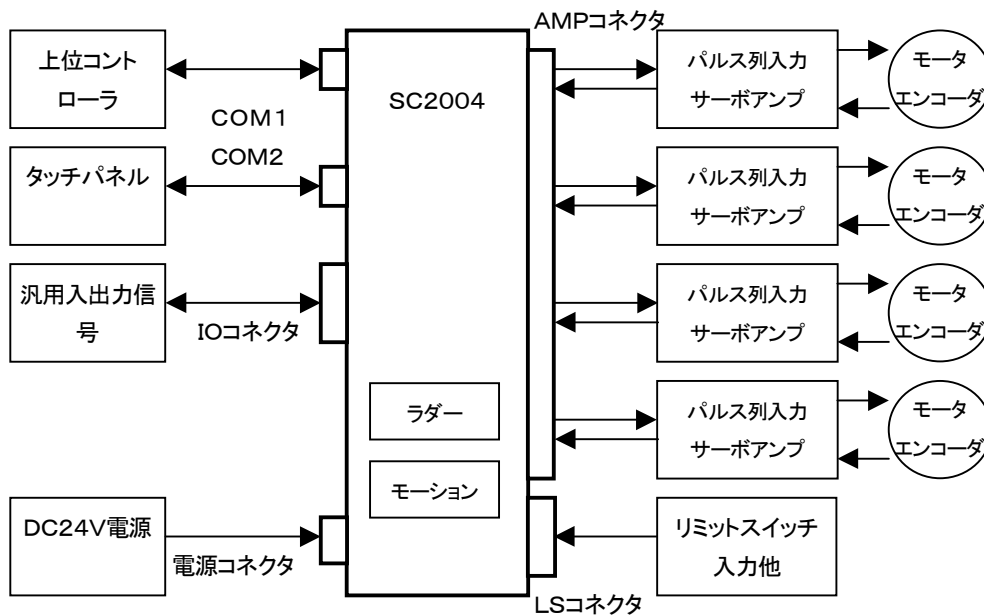
プログラム、位置データと等はフラッシュメモリーによって記憶されています。フラッシュメモリーの書き換え保証回数は10万回ですので、プログラムから書き換えを行う場合には、この回数を超えないよう書き換え頻度を考慮してください。

フラッシュメモリーのデータは電源投入時内部RAMにコピーして使用されます。プログラム、データの変更はRAMのデータを書き換えることで行いますので、RAMのデータをフラッシュメモリーに書き込んでいない状態で電源を切った場合、変更したデータは無くなり以前のデータとなってしまいますので注意してください。

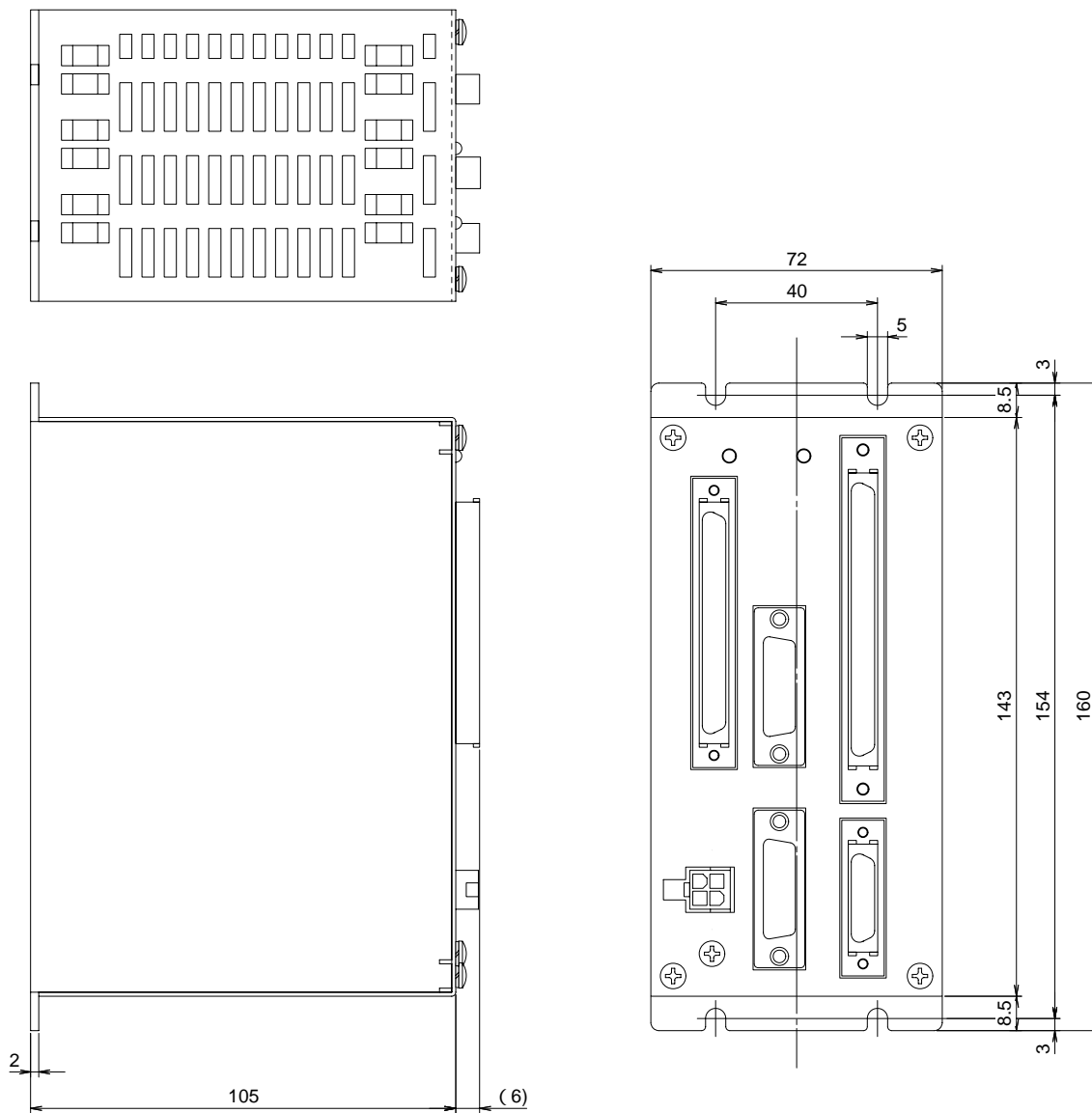
またパラメータをフラッシュメモリーに書き込んだ場合、電源投入時と同様の処理を行いますので、パラメータ以外の変更データは全て失われます。パラメータの保存時には必要なデータを前もってフラッシュメモリーに書き込んでから行ってください。

1.6. システム構成

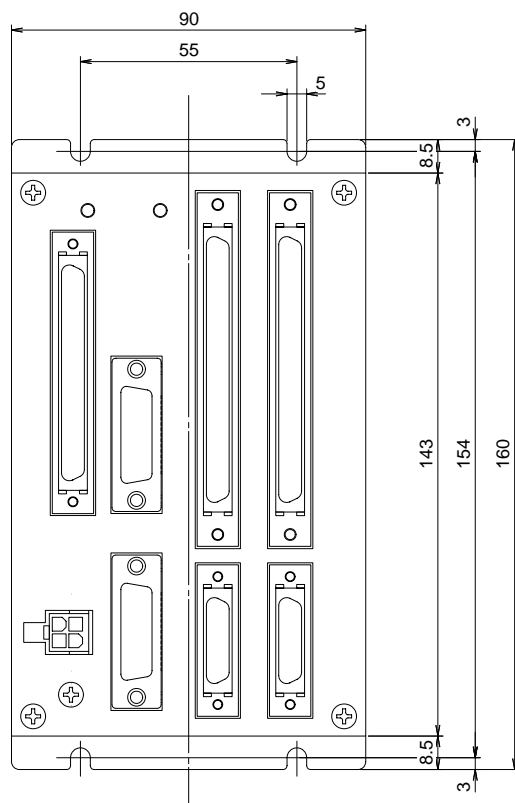
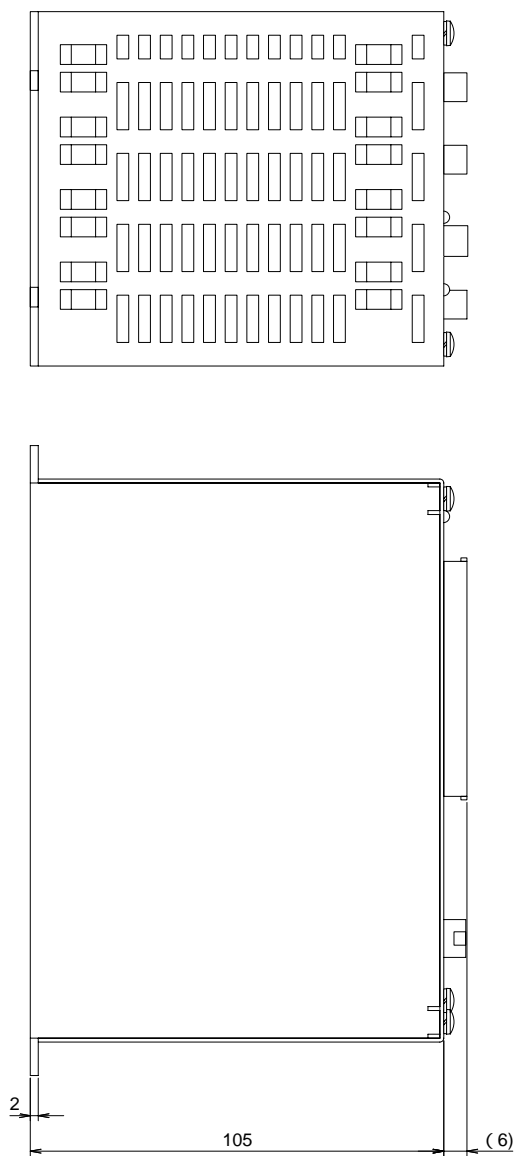
4軸の構成例を以下に示します。



1.7. 外観

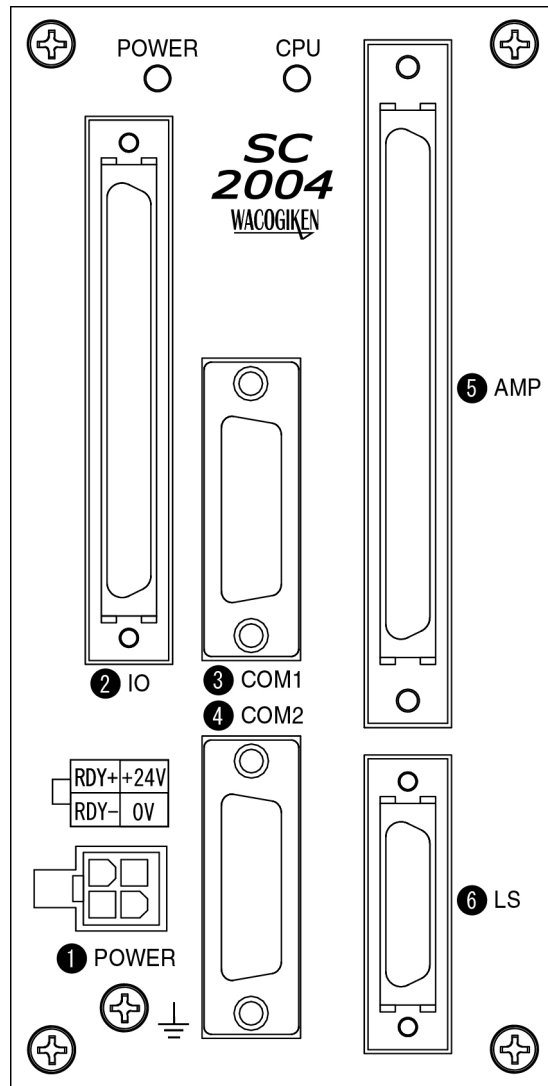


SC2004 外形寸法図

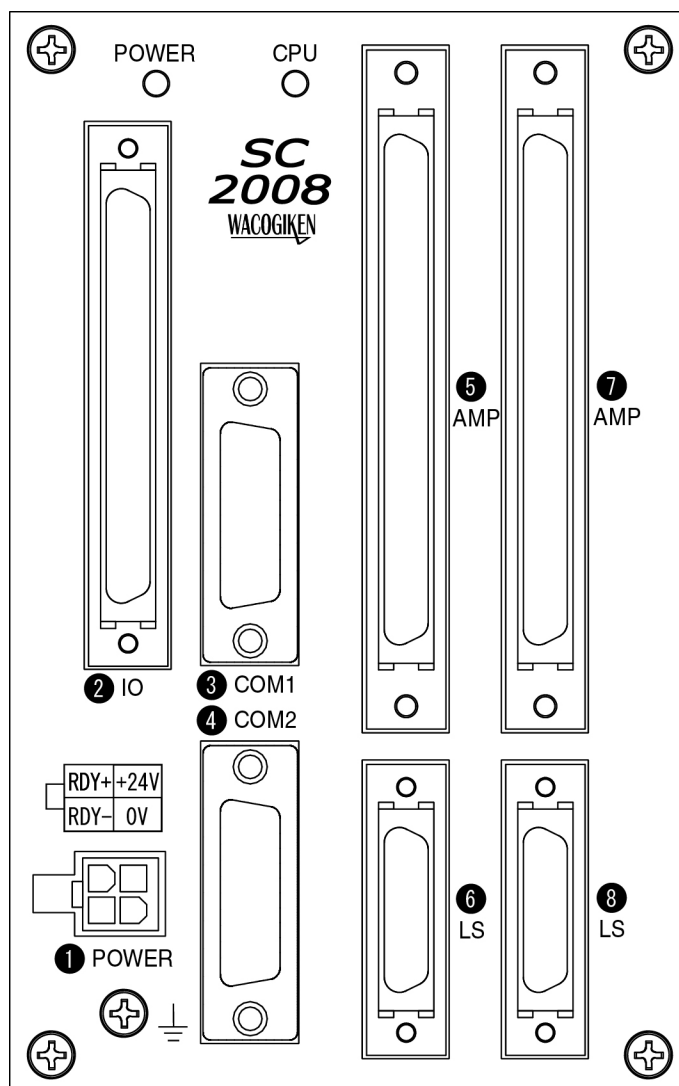


SC2008外形寸法図

1.8. 前面パネル



記号	名称	説明
①	電源コネクタ	電源を接続します。
②	IOコネクタ	入出力信号を接続します。
③	通信コネクタ(1)	通信ポート1。
④	通信コネクタ(2)	通信ポート2。
⑤	アンプコネクタ	0軸～3軸のサーボアンプ制御信号の入出力コネクタです。
⑥	LSコネクタ	0軸～3軸のリミットスイッチの入力コネクタです。
⊕	FG端子	接地線を接続します。
	POWER LED	電源の状態を示すLEDです。
	CPU LED	CPUの状態を示すLEDです。

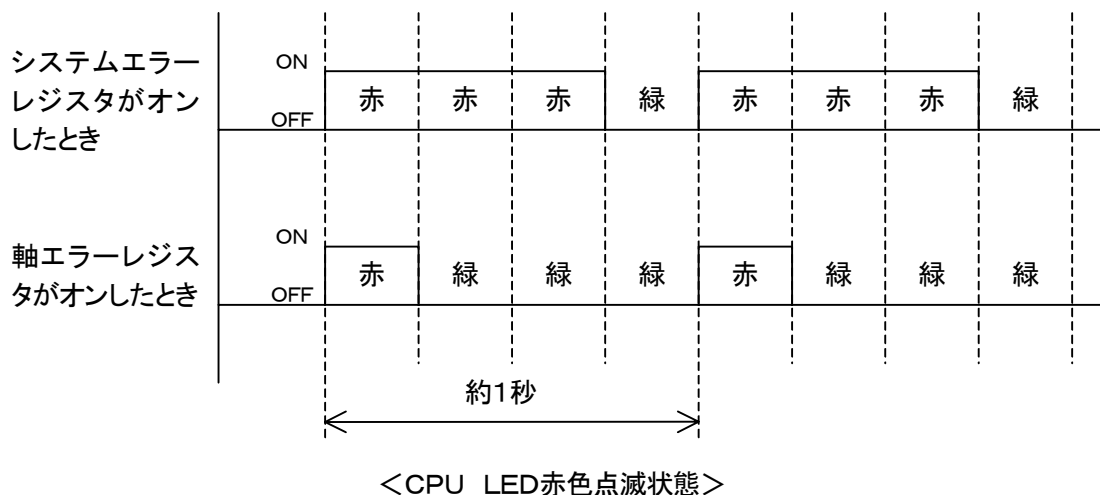


記号	名称	説明
①	電源コネクタ	電源を接続します。
②	IOコネクタ	入出力信号を接続します。
③	通信コネクタ(1)	通信ポート1。
④	通信コネクタ(2)	通信ポート2。
⑤	アンプコネクタ	0軸～3軸のサーボアンプ制御信号の入出力コネクタです。
⑥	LSコネクタ	0軸～3軸のリミットスイッチの入力コネクタです。
⑦	アンプコネクタ	4軸～7軸のサーボアンプ制御信号の入出力コネクタです。
⑧	LSコネクタ	4軸～7軸のリミットスイッチの入力コネクタです。
⊕	FG端子	接地線を接続します。
	POWER LED	電源の状態を示すLEDです。
	CPU LED	CPUの状態を示すLEDです。

1.9. LED表示

システムの状態はパネルの2つのLEDによって表示されます。

色	POWER LED	CPU LED
緑	電源が供給されていて、ハードウェアが正常に動作していることを示します。 電源コネクタのRDY出力はONされています。	CPUが正常に動作していることを示します。 CPUの動作状況によって薄い緑(CPUの稼働時間が少ない)から濃い緑(CPUの稼働時間が長い)に変化します。
赤	電源電圧が20V以下であるか、ハードウェアの異常のため運転できない状態です。電源電圧を確認して、再投入しても復帰しない場合、修理が必要です。 電源コネクタのRDY出力はOFFされています。	赤の点灯時間3に対して緑の点灯時間1で点滅している場合、システムエラーレジスタS7、S10、S11またはS12にエラーが表示されています。 赤の点灯時間1に対して緑の点灯時間3の周期で点滅している場合、軸エラーレジスタA□.7にエラーが表示されています。 どちらもエラーレジスタをクリアすることによって復帰します。 赤が点灯の状態は軸レジスタのエラーとシステムレジスタのエラーが同時に起こった場合です。



2. 接続

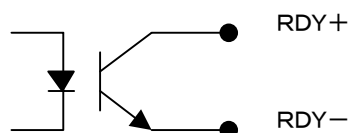
2.1. 各コネクタのピン接続

2.1.1. 電源コネクタ

ピン番号	信号名	説明
1	24VIN	DC24Vの電源入力です。
2	0VIN	
3	RDY+	RDY出力信号です。詳細は【2.4RDY出力について】を参照してください。
4	RDY-	

コネクタ型式 5557-04R ターミナル型式 5556 メーカー モレックス

<RDY出力回路>

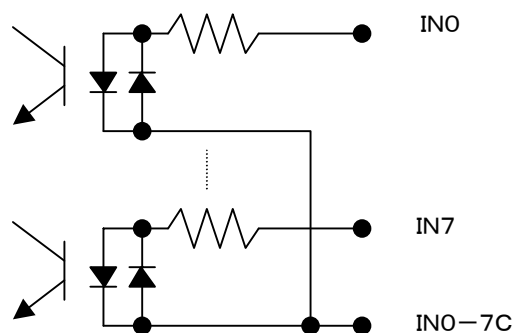


2.1.2. IOコネクタ

ピン番号	信号名	説明	ピン番号	信号名	説明
1	IN0	双方向フォトカプラ 受けの入力信号で す。	41	OUT0	フォトカプラオープン コレクタの出力です。
2	IN1		42	OUT1	
3	IN2		43	OUT2	
4	IN3		44	OUT3	
5	IN4		45	OUT4	
6	IN5		46	OUT5	
7	IN6		47	OUT6	
8	IN7		48	OUT7	
9	IN0-7C	IN0-7のコモン	49	OUT0-7C	OUT0-7のコモン
10	IN8	双方向フォトカプラ 受けの入力信号で す。	50	OUT8	フォトカプラオープン コレクタの出力です。
11	IN9		51	OUT9	
12	IN10		52	OUT10	
13	IN11		53	OUT11	
14	IN12		54	OUT12	
15	IN13		55	OUT13	
16	IN14		56	OUT14	
17	IN15		57	OUT15	
18	IN8-15C	IN8-15のコモン	58	OUT8-15C	OUT8-15のコモン
19	IN16	双方向フォトカプラ 受けの入力信号で す。	59	OUT16	フォトカプラオープン コレクタの出力です。
20	IN17		60	OUT17	
21	IN18		61	OUT18	
22	IN19		62	OUT19	
23	IN20		63	OUT20	
24	IN21		64	OUT21	
25	IN22		65	OUT22	
26	IN23		66	OUT23	
27	IN16-23C	IN16-23のコモン	67	OUT16-23C	OUT16-23のコモン
28	IN24	双方向フォトカプラ 受けの入力信号で す。	68	OUT24	フォトカプラオープン コレクタの出力です。
29	IN25		69	OUT25	
30	IN26		70	OUT26	
31	IN27		71	OUT27	
32	IN28		72	OUT28	
33	IN29		73	OUT29	
34	IN30		74	OUT30	
35	IN31		75	OUT31	
36	IN24-31C	IN24-31のコモン	76	OUT24-31C	OUT24-31のコモン
37	24VOUT	電源出力(24V)	77	0VOUT	電源出力(0V)
38	24VOUT		78	0VOUT	
39	24VOUT		79	0VOUT	
40	24VOUT		80	0VOUT	

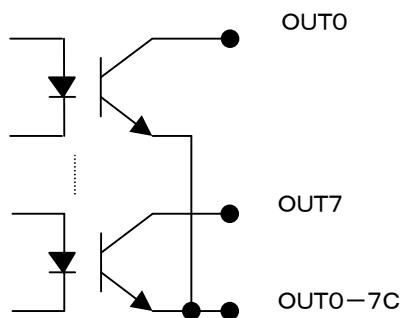
コネクタ型式 8840-080-174BD シェルカバー型式 8840S-080-CVR
メーカー KEL

<INO~31入力回路>



INO-7C、IN8-15C、
IN16-23C、IN24-31C
はそれぞれ独立しています。

<OUT0~31出力回路>



OUT0-7C、OUT8-15C、
OUT16-23C、OUT24-31C
はそれぞれ独立しています。

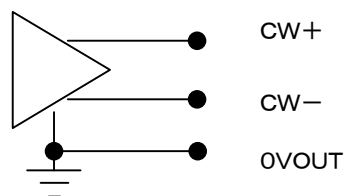
2.1.3. AMPコネクタ

0軸 ピン番号	1軸 ピン番号	2軸 ピン番号	3軸 ピン番号	信号名	説明
1	26	51	76	CW+ / PUL+	サーボアンプへの位置指令ラ インドライバ出力です。
2	27	52	77	CW- / PUL-	
3	28	53	78	CCW+ / DIR+	
4	29	54	79	CCW- / DIR-	
5	30	55	80	CWOC / PULOC	サーボアンプへの位置指令オ ープンコレクタ出力です。
6	31	56	81	CCWOC / DIROC	
7	32	57	82	0VOUT	電源出力(0V)
8	33	58	83	SON	サーボアンプへのサーボオン 出力です。
9	34	59	84	DEVCLR	サーボアンプへの偏差クリア 出力です。
10	35	60	85	ERRRST	サーボアンプへのエラーリセ ット出力です。
11	36	61	86	OUTC	上記3出力信号のコモンで す。
12	37	62	87	5VOUT	電源出力(5V)
13	38	63	88	A+	エンコーダ入力です。
14	39	64	89	A-	
15	40	65	90	B+	
16	41	66	91	B-	
17	42	67	92	Z+	
18	43	68	93	Z-	
19	44	69	94	END	サーボアンプのEND(位置決 め完了)信号の入力です。
20	45	70	95	ALARM	サーボアンプからのアラーム 入力です。
21	46	71	96	INC	上記2入力信号のコモンで す。
22	47	72	97	24VOUT	電源出力(24V)
23	48	73	98	N. C.	未接続端子です。 使用しないでください。
24	49	74	99	N. C.	
25	50	75	100	N. C.	

コネクタ型式 8840-100-174BD シェルカバー型式 8840S-100-1-CVR
メーカー KEL

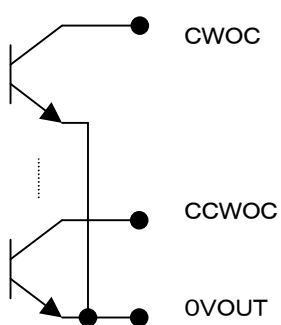
SC2008(8軸)の場合、2枚目のボードが4軸から7軸に対応します。

<CW+、CW-、CCW+、CCW-出力回路>



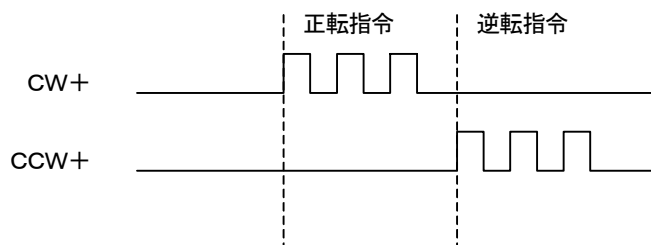
OVOUTは全軸共通です。

<CWOC、CCWOC出力回路>

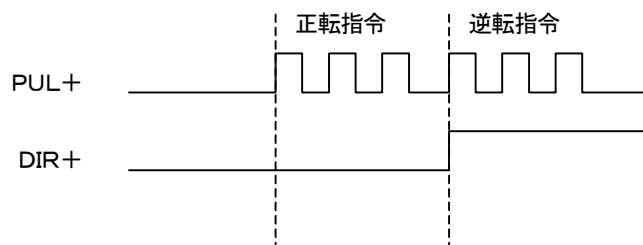


OVOUTは全軸共通です。

<2パルス出力信号波形>

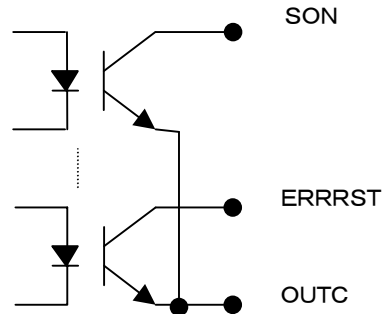


<1パルス出力信号波形>



1パルス、2パルスの切り替えはパラメータで行えます。

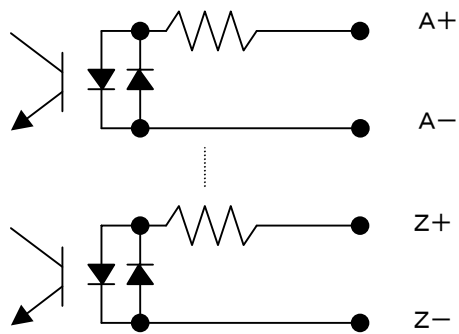
<SON、DEVCLR、ERRRST入力回路>



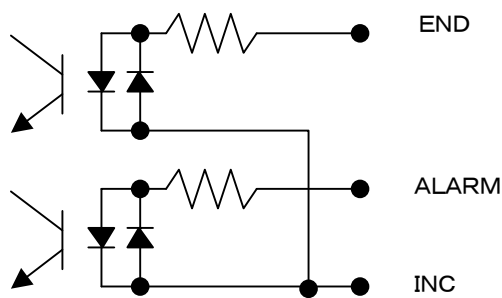
OUTCは各軸独立しています。

SON、ERRRST出力はパラメータによって出力論理を変えられますが、SON出力は電源投入時、本機の異常時にはパラメータの論理設定に関係なく常にオフします。

<A+、A-、B+、B-、Z+、Z-入力回路>



<END、ALARM入力回路>



INCは各軸独立しています。

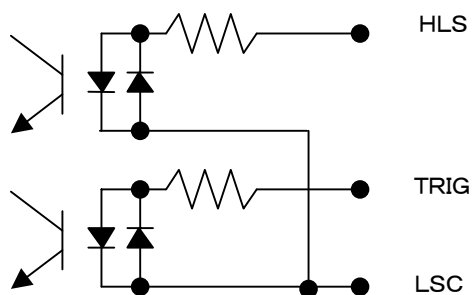
END、ALARM入力はパラメータによって入力論理を変えられます。

2.1.4. LSコネクタ

0軸 ピン番号	1軸 ピン番号	2軸 ピン番号	3軸 ピン番号	信号名	説明
1	7	13	19	HLS	原点センサの入力です。
2	8	14	20	NLS	原点近傍センサの入力です。
3	9	15	21	PLS	+限界センサーの入力です。
4	10	16	22	MLS	-限界センサーの入力です。
5	11	17	23	TRIG	現在値取り込みセンサの入力です。
6	12	18	24	LSC	入力コモンです。
25		26		24VOUT	電源(24V)の出力です。
27		28		0VOUT	

コネクタ型式 8840-028-174BD シェルカバー型式 8840S-028-CVR
 メーカー KEL

<HLS、NLS、PLS、MLS、TRIG入力回路>



LSCは各軸独立しています。

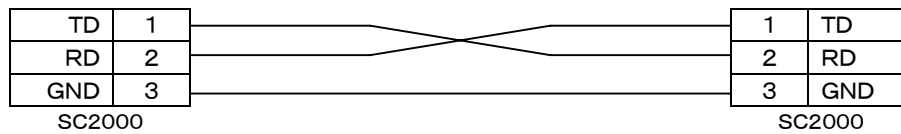
HLS、NLS、PLS、MLS、TRIG入力はパラメータによって入力論理を変えることができます。

NLS入力はシステム上汎用入力として扱われます。原点復帰時にNLS入力によって減速処理を行う場合はプログラムで記述する必要があります。

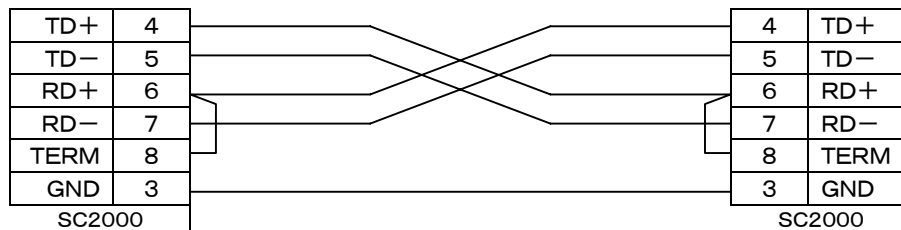
2.1.5. 通信コネクタ

ピン番号	信号名	説明
1	TD	送信データ(RS-232C)
2	RD	受信データ(RS-232C)
3	GND	回路グランド
4	TD+	送信データ+(RS-422A/RS-485)
5	TD-	送信データ-(RS-422A/RS-485)
6	RD+	受信データ+(RS-422A/RS-485)
7	RD-	受信データ-(RS-422A/RS-485)
8	TERM	ターミネータ(6-8ピンを短絡すると終端が行えます)
9		未使用端子です。接続しないでください。
10		
11		
12	TOOL	ツール選択(タッチパネル、SCLinkを使用する場合、開放で使用します。GNDと接続することによって専用パソコンツールの入出力となります。)
13	VCCOUT	電源出力(SC2000専用機器用の電源出力です。使用しないでください。)
14	GND	グランド
15	CASE	ケースグランド

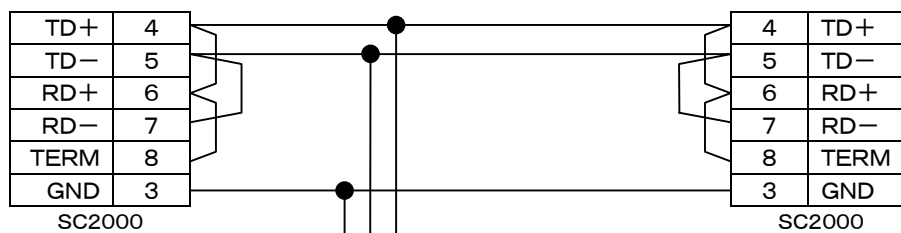
RS-232Cの接続例



RS-422Aの接続例



RS-485の接続例



マルチドロップの場合中間の配置される機器のターミネータ(TERM)は使用しません。

2.2. 入出力仕様

2.2.1. 電源入力

信号名	回路形式	項目	最小	最大	備考
24VIN		電圧	21.6V	26.4V	
		電流		1A	

2.2.2. 電源出力

信号名	回路形式	項目	最小	最大	備考
24VOUT		電流		0.4A	出力電流の 合計値
5VOUT		電流		0.2A	

2.2.3. 入力信号

信号名	回路形式	項目	最小	最大	備考
IN0~31、 END、ALARM、 PLS、MLS、 HLS、NLS、TRIG	双方向 フォトカプラ	印加電圧		26.4V	コモンを+電圧、 または-電圧で 使用することが可 能です。
		ON電圧	8V		
		OFF電圧		2V	
		ON電流		10mA	

信号名	回路形式	項目	最小	最大	備考
A、B、Z	高速 フォトカプラ	印加電圧		6V	
		ON電圧	2.5V		
		OFF電圧		1V	
		ON電流		16mA	
		周波数		1M/S	4通倍
				500K/S	2低倍
	250K/S		1低倍		

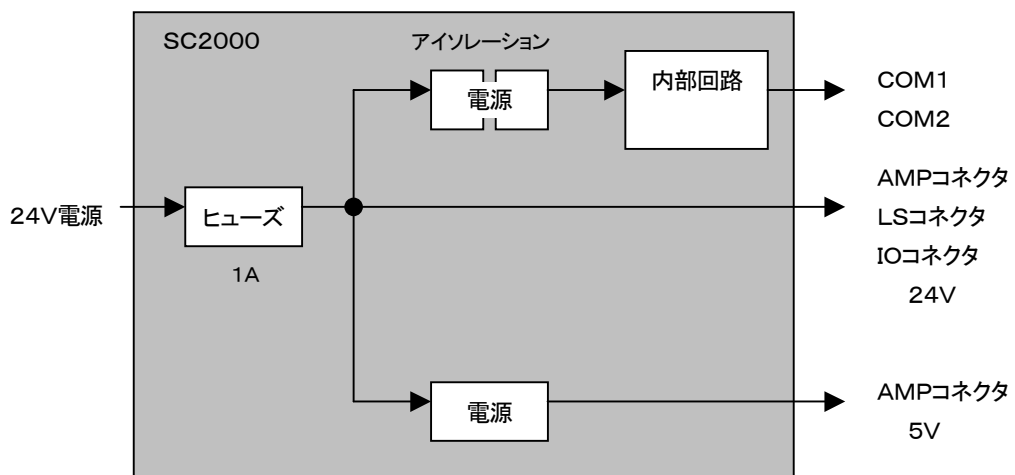
2.2.4. 出力信号

信号名	回路形式	項目	最小	最大	備考
RDY、 OUT0~31、 SON、DEVCLR、 ERRRST	オープン コレクタ	印加電圧		26.4V	
		ON電流		40mA	
		ON電圧		2V	
		OFF電流		0.5mA	

信号名	回路形式	項目	最小	最大	備考
CW、CCW	ラインド ライバ	出力電流	20mA		
		差動電圧	2V		
		周波数		1M/S	

信号名	回路形式	項目	最小	最大	備考
CWOC、CCWOC	オープン コレクタ	印加電圧		26.4V	
		ON電流		40mA	
		ON電圧		1V	
		OFF電流		0.1mA	
		周波数		250K/S	

2.3. 電源系統図



24V電源入力はヒューズを介して各回路に供給され、同時に外部コネクタにも接続されています。ヒューズは自己復帰型で、動作しても5分以内に復帰します。

2.4. R D Y出力について

電源コネクタのRDY出力は、本機が正常に動作していることを確認するための重要な信号です。システムの安全上、機器の異常時に遮断しなくてはならない回路がある場合、この信号を使用してインターロックを行ってください。

RDY出力は本機が正常に動作しているときはONしていますが、以下の場合OFFとなります。

電源が印加されていないとき。

ハード上の異常を検出したとき。

CPUが正常に動作しなくなったとき。

異常を検出した場合、IOコネクタの出力信号は全てOFF、モータの指令パルスは停止となります。

2.5. サーボアンプについて

パルス列指令の位置制御のサーボアンプが使用可能です。

サーボアンプに以下の入出力があれば、本機の専用入出力により制御が可能です。

サーボアンプの	アラーム出力
	位置決め完了出力
	サーボON入力
	偏差クリア入力
	エラーリセット入力
	エンコーダ出力

2.6. エンコーダのフィードバックについて

エンコーダフィードバック信号の接続は任意です。

エンコーダフィードバックがない場合、位置決め完了判定はサーボアンプの完了信号によって行います。

エンコーダフィードバックがある場合、位置決め完了判定は本機の判定機能、サーボアンプの完了信号から選択することが可能です。

2.7. T R I G入力について

LSコネクタのTRIG入力によって各軸別に現在値の取り込みが可能です。外部のリミットスイッチを使用して位置決めを行う場合に使用します。TRIG入力によって現在値がハードウェアによってラッチされ軸レジスタに書き込まれますので、この値をもとに位置決めを行うことができます。

2.8. 駆動限界リミットスイッチの検出

LSコネクタのPLS及びMLSはモータの駆動範囲を制限するために使用します。

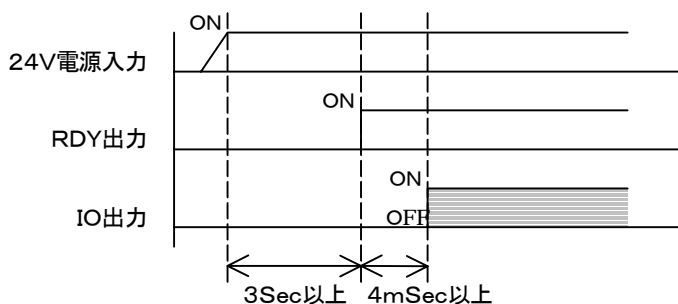
検出はモータの回転方向がプラスの場合PLSが、マイナスの場合MLSが監視され検出すると運転を停止しエラーをセットします。稼動範囲の内側に向かったの運転にはLS検出を行いませんので、LS上から稼動範囲内に復帰することが可能です。

2.9. サーボアンプのアラーム入力

サーボアンプのアラーム入力(AMPコネクタのALARM)は、サーボオン状態のときだけ監視されます。サーボオン状態かつ、アラーム入力が入っていた場合、サーボアンプのエラーとしてA□. 7. 6がオンします。アラーム入力の入力論理はパラメータで設定が行えます。

2.10. 電源投入時の信号タイミング

電源投入から各信号の出力タイミングは以下のとおりです。



3. プログラミング概要

本機には、モーションとラダーの2つのプログラム言語が搭載されています。それぞれの言語の特徴を生かすことによって短時間に目的の機能を作成することができます。

3.1. モーション

モータ駆動、計算式、はモーションで記述することができます。モーションはラダーから起動され、同時に256個のプログラムを起動することができます。

プログラムメモリーは64Kバイトあり、最大16000ステップが格納できます。

モーションプログラムはフラッシュメモリーに保存されます。

3.2. ラダー

ラダーは電源が投入されている間、常に実行されていますので、サーボアーム時のブレーキ制御などに使用することが可能です。

常時実行しなくてはならない処理、単純な処理はラダーで、複雑な処理は条件を満たしたときにモーションを起動し処理をします。プログラムメモリーには1000ステップが格納できます。

ラダープログラムはフラッシュメモリーに保存されます。

3.3. 処理の流れ

実行は4mSの周期で行われます。

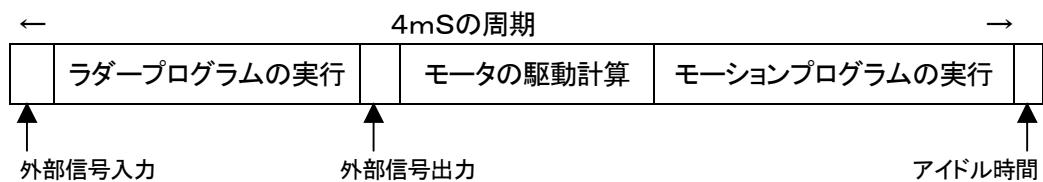
まずラダープログラムを実行します。ラダーの内容によって実行時間が変化します。

次にモータの運転指令があればそのための計算を行います。

最後にモーションプログラムを実行します。モーションプログラムはその周期内に終わる必要はなく、常に周期の最後を使用して実行されます。

外部信号の入力はラダーの実行の直前、出力はラダー実行の直後にそれぞれ行われます。

4mS内にすべての処理が終了したら、残りの時間をアイドル状態で待ちます。このアイドル時間はシステムレジスタS15によって確認することができます。



3.4. プログラミングの要点

ラダーはステップ数に比例して実行時間が長くなります。従って常時監視しなくてはならない信号処理だけをプログラムするようにします。このことによりモーションの実行時間が多く取れモーションの実行速度が向上します。

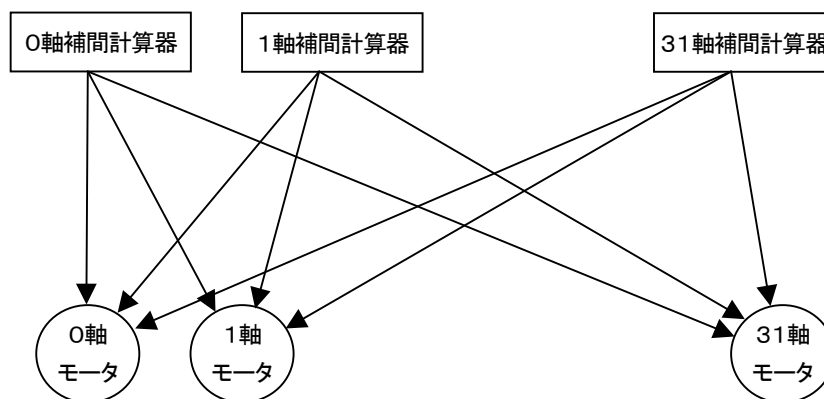
モーションは、起動しているプログラムを時分割で実行するため、同時に実行するプログラム数が少ないほど実行速度は向上します。

また、プログラム中条件待ちでループを作るとそのプログラムに多くの実行時間が集中するためほかのモーションの実行速度が低下します。この場合wait命令をループ中に記述することによってほかのプログラムの実行を妨げないプログラムとなります。以下に例を示します。

LOOP:	
if R0. 1=0	入力スイッチがOFFだったら
wait 0	CPUを開放する
goto LOOP	
endif	
	入力スイッチがONになったときの処理

4. モータの運転

モータの運転は32個の補間計算器によって行われます。この補間計算器は1軸～32軸の任意の軸を補間運転することができます。例えば32軸を非同期に制御すること、軸を何グループ化に分割してグループごとに補間を行うなど、柔軟な制御が行えます。



4.1. モータの運転方法の説明

目的に合わせてさまざまな運転方法を選択することができます。

各運転の指令ビットをONにすることによって運転が開始されます。指令ビットはOFFからONのエッジで検出されます。指令を行うとき、A□. 0. 0で補間演算器がレディー(ON)であることを確認してから行います。レディーでない場合指令は無視されます。以下に運転手順を説明します。

4.1.1. ジョグ運転

軸番号(A□の番号)の1軸をジョグ運転します。運転速度および加速度は速度番号で指定しますが、S字加減速は行われません。

原点復帰後は±ソフトリミットの範囲で運転が制限されます。

指令実行中はA□. 0. 0はOFF、終了するとONになります。

実行中オーバーライドの変更で速度の調整が行えます。

以下は+方向のジョグ運転の指令手順です。-方向の場合はA□. 16. 1をONします。

レジスタ	説明
A□. 18	速度番号を指定します。
A□. 16. 0	ONしている間、+方向にジョグ運転を行います。必要な項目が設定し終わってからONにします。
A□. 0. 0	モータの運転中はOFF、終了するとONになります。

4.1.2. 原点復帰

運転の前にパラメータC7で原点復帰方法、C2で原点復帰位置を設定します。

軸番号(A□の番号)の1軸を原点復帰します。運転速度は速度番号の指定が第1速度、速度番号+1が第2速度となります。第2速度では加減速は行われませんので速度は充分低い速度で運転してください。また第1、第2速度のSじ加減速は行われません。

原点復帰完了時、AMPコネクタのDEVCLR信号を出力してサーボアンプの偏差をクリアします。これによって本機とサーボアンプの間の位置合わせを行います。原点復帰完了時の位置はパラメータのC2の原点復帰位置となります。

指令実行中はA□. 0. 0はOFF、終了するとONになります。

正常終了するとA□. 0. 4がONし原点復帰の完了を表わします。

レジスタ	説明
C7	原点復帰方法。
C2	原点復帰位置。
A□. 18	速度番号を指定します。指定した番号と、その次の番号が使用されます。
A□. 16. 7	立ち上がりエッジで原点復帰を開始します。必要な項目が設定し終わってから最後にONにします。

各原点復帰の動作を説明します。[]内の数字はパラメータC7(原点復帰方法)の設定値を示します。

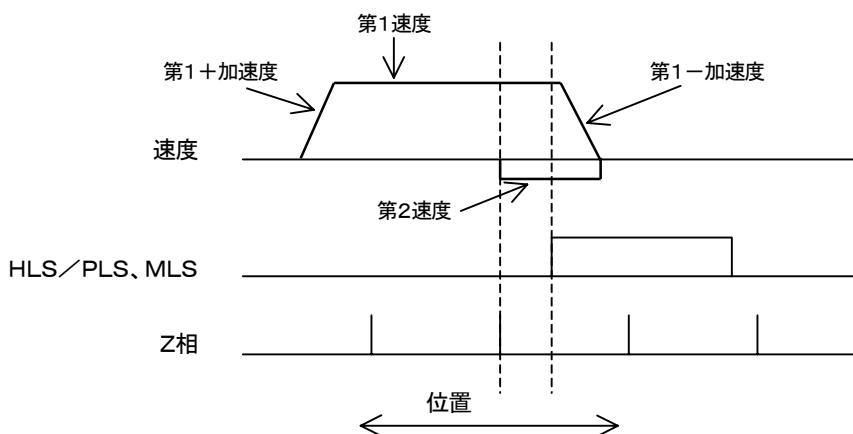
[0]原点なし

モータの運転を行わず現在値を設定のみ行って終了します。

- [2] プラス方向 HLS、反転、Z相
- [3] マイナス方向 HLS、反転、Z相
- [12] プラス方向 PLS、反転、Z相
- [13] マイナス方向 MLS、反転、Z相

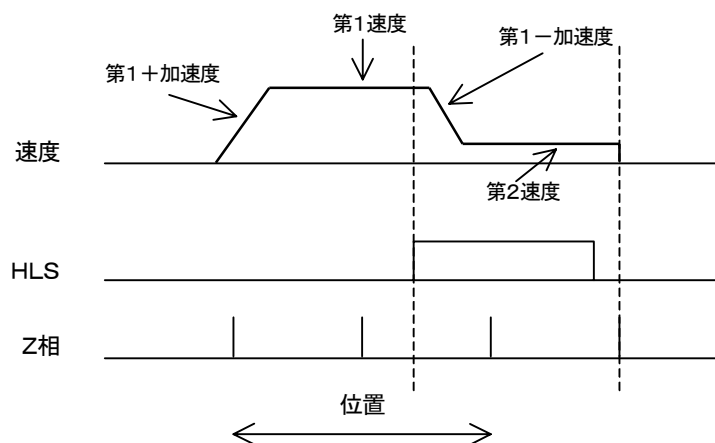
[2][3]はプラスまたはマイナス方向に運転してHLSを検出し、方向反転してHLSを脱出してから最初のエンコーダZ相信号で原点復帰を終了します。

[12]はPLS、[13]はMLSを検出し、方向反転してPLS/MLSを脱出してから最初のエンコーダZ相信号で原点復帰を終了します。



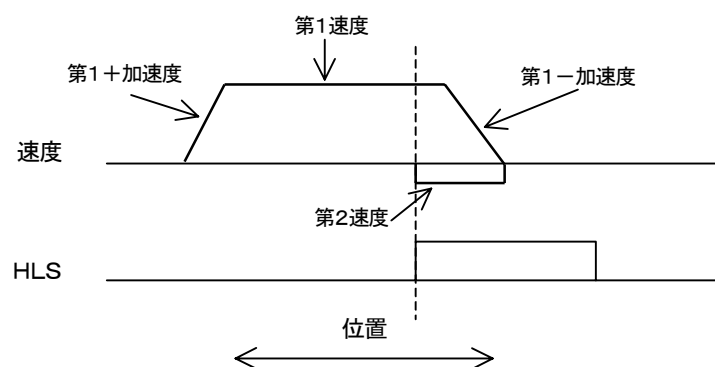
- [4] プラス方向 HLS、Z相
- [5] マイナス方向 HLS、Z相

プラスまたはマイナス方向に運転してHLSを検出し、方向反転せずにHLSを脱出してから最初のエンコーダのZ相で原点復帰を終了します。



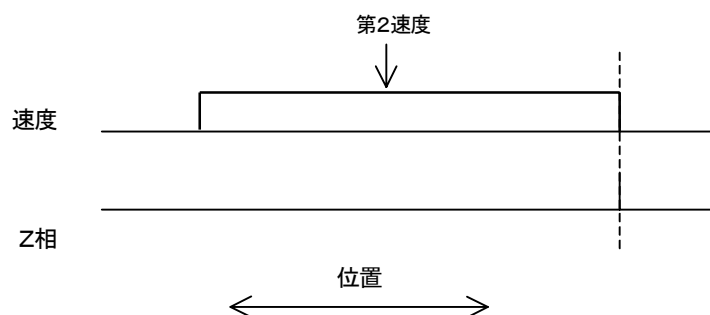
[6]プラス方向 HLS、反転
[7]マイナス方向 HLS、反転

プラスまたはマイナス方向に運転してHLSを検出し、方向反転してHLSを脱出するエッジで原点復帰を終了します。



[8]プラス方向 Z相
[9]マイナス方向 Z相

[8]プラス方向、[9]マイナス方向の最初のエンコーダZ相信号で原点復帰を終了します。



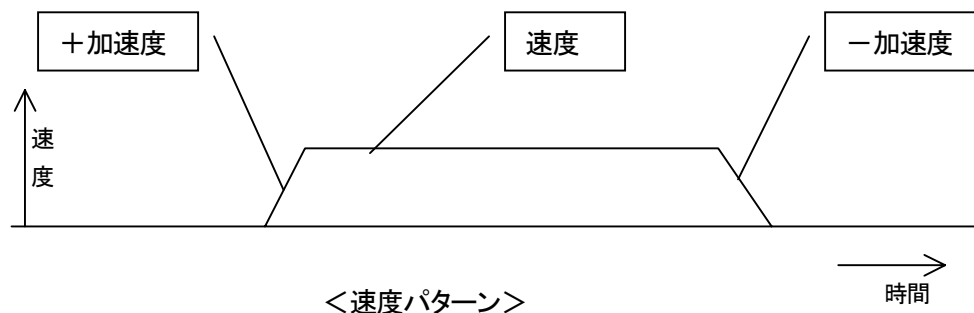
4.1.3. 単軸運転

軸番号(A□の番号)の1軸を目標位置に位置決めします。このとき速度、加速度およびS字時間の指定が行えます。

指令実行中はA□. 0. 0はOFF、終了するとONになります。

実行中オーバーライドの変更で速度の調整が行えます。

レジスタ	説明
A□. 16. 8	OFFで絶対位置、ONで相対位置指令を行います。
A□. 17	目標の位置番号を指定します。
P□. □	目標位置はここに設定します。 □には上で設定した位置番号と軸番号が入ります。
A□. 18	速度番号を指定します。
V□. 0~3	速度はここに設定します。 □には上で設定した速度番号が入ります。
A□. 16. 2	単軸運転開始。立ち上がりエッジで動作を開始します。必要な項目が設定し終わってから最後にONにします。



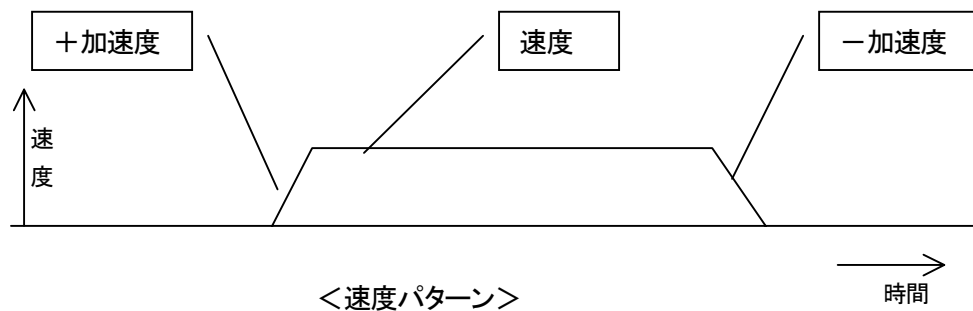
4.1.4. 多軸運転

A□. 22の軸指定で指定される、多軸の補間を行います。このとき速度、加速度およびS字時間の指定が行えます。軸指定はビット対応でONになっているビットの軸が駆動されます。

b31	b30	...	b7	b6	b5	b4	b3	b2	b1	b0
31軸	30軸	...	7軸	6軸	5軸	4軸	3軸	2軸	1軸	0軸

指令実行中はA□. 0. 0はOFF、終了するとONになります。
 実行中オーバーライドの変更で速度の調整が行えます。

レジスタ	説明
A□. 16. 8	OFFで絶対位置、ONで相対位置指令を行います。
A□. 17	目標位置の位置番号を指定します。
P□. □	目標位置はここに設定します。 □には上で設定した位置番号と軸番号が入ります。
A□. 18	速度番号を指定します。
V□. 0~3	速度はここに設定します。 □には上で設定した速度番号が入ります。
A□. 21	速度モデルを設定します。
A□. 22	どの軸を運転するかビット対応で設定します。
A□. 16. 3	多軸運転開始。立ち上がりエッジで動作を開始します。必要な項目が設定し終わってから最後にONにします。



4.1.5. 円弧運転

任意の2軸または3軸の円弧補間を行います。このとき速度、加速度およびS字時間の指定が行えます。

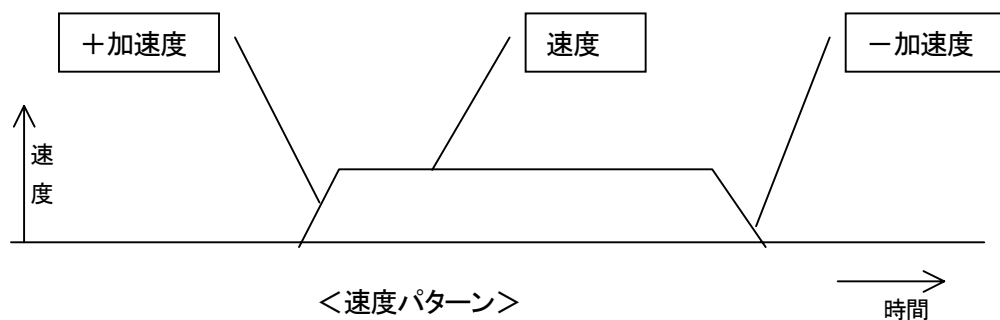
2軸を指定した場合任意2次元上、3軸を指定した場合は任意3次元空間上の円弧補間を行います。円弧補間する軸以外の軸を同時に運転することが可能です。

円弧補間の開始転は現在位置で、そこから通過点を通り終点に向かう円弧を作ります。これらの3点が同一点や直線上の点である場合、円弧の半径が大きすぎる場合、中心が座標外である場合、エラーとなります。

速度モデルで円弧補間を行う軸を選んだ場合円弧の周速度、円弧以外の軸を選んだ場合それらの軸の合成速度、円弧補間を行う軸とそれ以外を選んだ場合には周速度と円弧以外の軸の合成速度となります。

相対位置指定を行った場合、通過点および終点は現在位置からの相対値で扱われます。

レジスタ	説明
A□. 26	円弧通過点の位置番号を指定します。
A□. 27	どの軸を円弧補間するかビット対応で指定します。32ビット中の2ビットまたは3ビットをONします。
P□. □	目標位置はここに設定します。 □には上で設定した位置番号と軸番号が入ります。 円弧軌跡上の通過点と終点を指定する2組を使用します。
A□. 18	速度番号を指定します。
V□. 0~3	速度はここに設定します。 □には上で設定した速度番号が入ります。
A□. 21	速度モデルを設定します。
A□. 22	円弧を含む運転する軸をビット対応で設定します。
A□. 26	円弧通過点の位置番号を指定します。
A□. 27	円弧補完する軸を指定します。2ビットまたは3ビットをONします。
A□. 16. 6	円弧運転開始。立ち上がりエッジで動作を開始します。必要な項目が設定し終わってから最後にONIにします。



4.1.6. オーダー運転

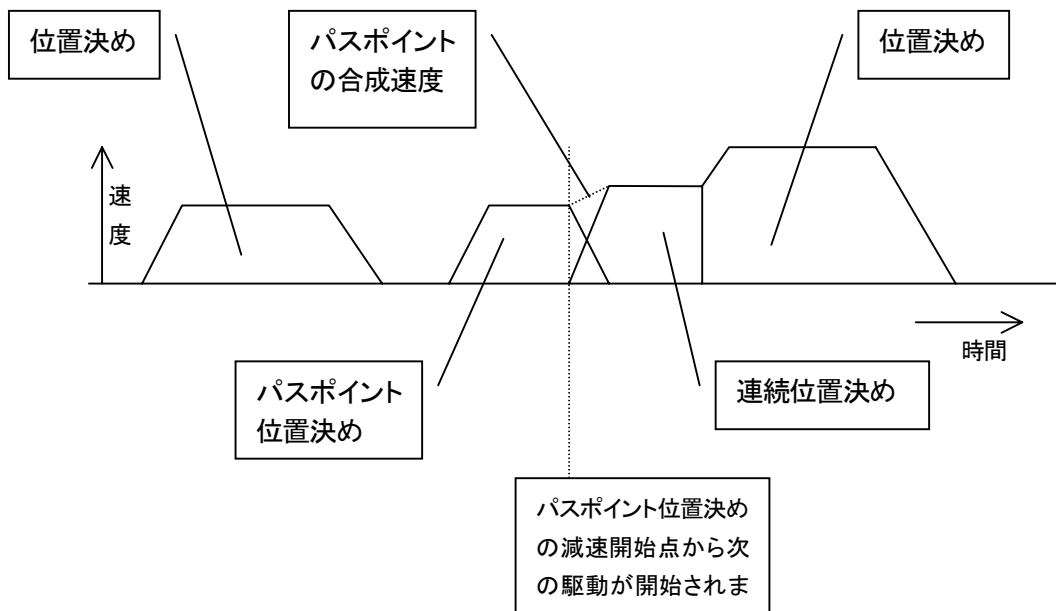
オーダー運転を行います。

指令実行中はA□. 0. 0はOFF、終了するとONになります。

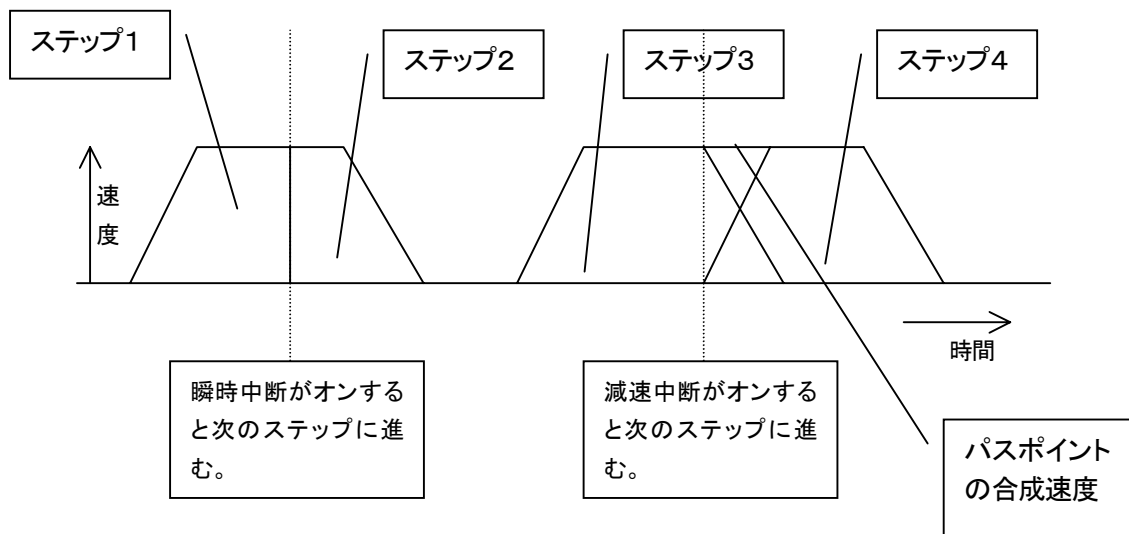
実行中オーバーライドの変更で速度の調整が行えます。

オーダー運転中にA□. 16. 4をOFFすると現在実行中のオーダー運転のステップを終了した後停止します。また場合A□. 16. 16(減速中断)、A□. 16. 17(瞬時中断)が入力された場合、現在実行中のオーダー運転のステップを中断して次のステップに進みます。

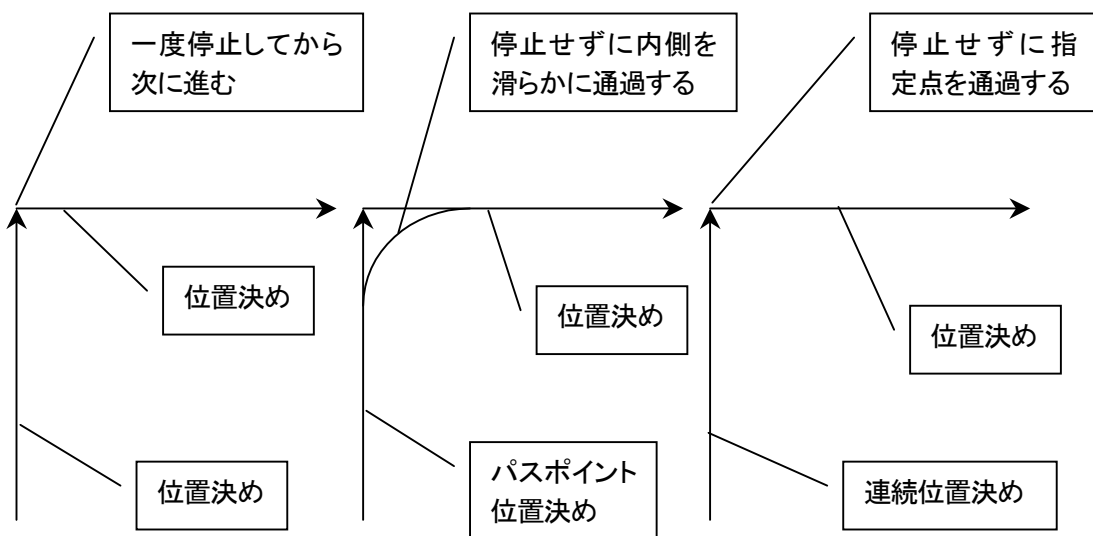
レジスタ	説明
A□. 19	オーダー番号を設定します。
A□. 16. 4	オーダー運転開始。立ち上がりエッジで動作を開始します。必要な項目が設定し終わってから最後にONにします。



<各命令の運転パターン例>



<瞬時中断、減速中断時の運転パターン例>



<各命令の軌跡例>

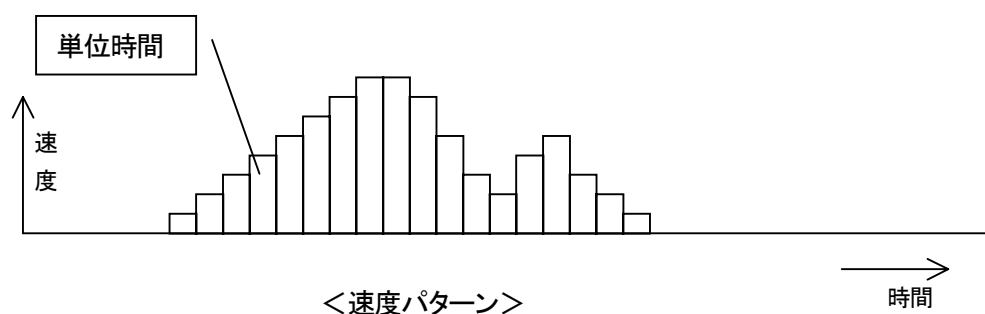
4.1.7. 単位時間運転

時間と目標位置で運転を行います。設定時間ごとにデータを与えることにより任意の速度パターンを作り出すことができます。

時間は計算周期(4mS)の整数倍で設定します。1は4mS、2は8mSとなります。目標位置は上記の時間に駆動する、絶対値または相対値の指定を行います。速度、加速度、S字の設定は行えません。

指令実行中はA□. 0. 0はOFF、終了するとONになります。

レジスタ	説明
A□. 16. 8	OFFで絶対位置、ONで相対位置指令を行います。
A□. 17	目標の位置番号を指定します。
P□. □	目標位置はここに設定します。 □には上で設定した位置番号と軸番号が入ります。
A□. 20	時間を指定します。指定値×4mSで駆動量を消化します。
A□. 16. 5	単位時間運転開始。立ち上がりエッジで動作を開始します。必要な項目が設定し終わってから最後にONにします。



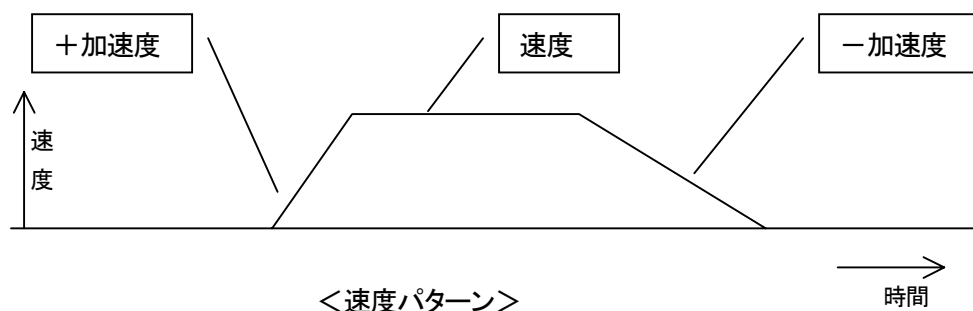
4.2. ユーザー単位

プログラム上の位置、速度、加速度を扱う単位をユーザー単位と呼び、とモータに出力するパルスとの関係はこの係数によって定義します。M係数、D係数ともに1とした場合、指令値＝パルスとなります。また機械座標とモータの回転方向が合わない場合、M係数に－符号を付けることによってモータの回転方向を反転させることができます。

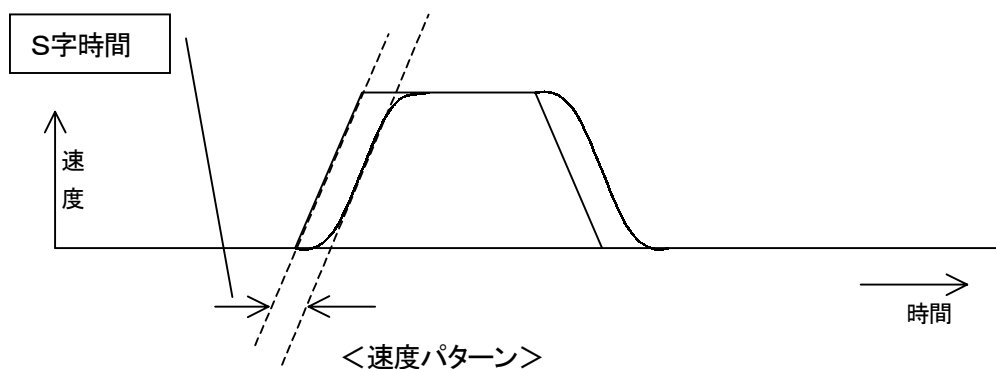
$$\text{出力パルス} = \text{指令値} \times \frac{\text{M係数}}{\text{D係数}}$$

4.3. 速度設定

各運転の速度、±加速度およびS字時間はVレジスタの設定によって制御が行えます。速度はユーザー単位/Sec、加速度はユーザー単位/S²で与えます。



S字時間は速度、加速度で作られる台形パターンの加減速部分をS字の形に変形させます。設定は時間(mSec)で行います。この設定値だけ位置決め時間は増加します。



4.4. 非常停止

IO入力から任意の1ビットを選択して非常停止入力とすることができます。非常停止信号が入力されると以下を実行します。

全軸の補間演算器に瞬時停止を送りパルス指令出力を停止します。

パラメータのサーボOFF時間経過後、AMPコネクタSON出力をOFFします。

非常停止入力はソフトウェアによって常時監視しています。またソフトウェアの動作状況はウォッチドッグタイマーで監視されていますが安全上、非常停止信号で動力部等の電源供給を遮断するように本機の外部で回路構成を行ってください。

4.5. ソフトリミット

+ソフトリミット<−ソフトリミットの関係で設定を行った場合、ソフトリミットは無効となります。通常は両方に0を設定します。ソフトリミットを無効とした場合、ソフトリミットによる制限は行われずに運転が行えます。

+ソフトリミット>−ソフトリミットの関係で設定を行った場合、ソフトリミットは有効となります。原点復帰が終了するとソフトリミット判定が開始され、±ソフトリミット範囲を超えて駆

動を行った場合エラーとなり、モータは停止します。

各軸ごとにモータの運転する方向のリミット値によって判定されます。モータが＋方向に運転する時には＋ソフトリミットの判定だけが行われます。従って稼動範囲から外側に向かっての運転はソフトリミットによって禁止されますが内側に向かっての運転は制限を受けません。

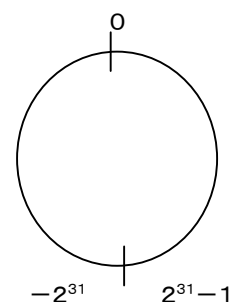
4.6. 無限送りについて

一方向に回転し続ける動作を無限送りと言います。内部の指令値は32ビット長で計算されていますから -2^{31} から $2^{31}-1$ の範囲が正規の範囲です。本機ではこの範囲を超えた場合にも正しく運転ができるよう -2^{31} 、 $2^{31}-1$ を連続した座標として扱います。

運転を行うには絶対値、相対値どちらでも行えます。絶対値の場合、現在値と目標値が近い方向に運転を行います。

無限送りを行う場合、ソフトリミットは無効にして使用します。

M係数、D係数が1対1以外の関係で使用した場合、座標境界を超えた位置でのフィードバック値からの現在値の調整は正しく行えませんので注意してください。これはM係数、D係数の影響で指令側とパルス側で座標境界が異なってしまうためで、調整を行うとフィードバックから換算した値が指令側に設定されます。



4.7. 速度モデル

モータ駆動を行うとき速度をどの軸に適用するか設定します。ビット対応でビット0から0軸、1軸...に対応します。補間を行うときこのビットがONしている軸の合成速度で補間を行います。すべてのビットがOFFの場合、駆動距離の最も長い軸を指定速度で運転します。合成速度の求め方を以下に示します。

$$L = \sqrt{D_0^2 + D_1^2 + D_2^2}$$

$$S_0 = D_0 / L$$

$$S_1 = D_1 / L$$

$$S_2 = D_2 / L$$

0、1、2軸を運転する場合

D_0 、 D_1 、 D_2 は駆動距離

0軸の速度

1軸の速度

2軸の速度

4.8. 現在値の取り込み機能

外部のセンサによって位置決めを行う場合、TRIG入力を使用して現在値を取得することができます。センサの入力は各軸ごとにあり、専用のハードウェアによって実現されていますので正確な位置を読み取ることが可能です。

レジスタ	説明
A□. 0. 21	TRIG信号が入力された1サイクル間、ONします。
A□. 3	読み取った現在値が書き込まれます。

4.9. エラーからの復帰について

各軸のエラーはレジスタA□. 7、その他のエラーはS10、S11、S12で調べることができます。

各軸のエラーを解除するには、A□. 16. 15をONにすることで解除が行えます。

その他のエラーはS10、S11、S12に0を書き込むことで行えます。

エラーが発生してサーボOFFになった場合、本機の指令現在値とフィードバック値がずれてしまう場合があります。この状態で運転を行った場合位置決め終了判定ができずにエラー終了する場合があります。

パラメータによってエンコーダフィードバックを行う設定にしている場合、フィードバック値に指令現在値を合わせるによって調整が行えます。調整はA□. 16. 14をONにすることによって行えます。またエンコーダフィードバックを行わないに設定している場合、モータの位置がわかりませんので再度原点復帰からやり直す必要があります。

4.10. オーバーライド

運転中の速度をオーバーライド設定によって即座に変更することができます。オーバーライド値は100(%)で設定された速度となり0~500(%)の範囲で調整することができます。

0(%)の設定をした場合運転は一時停止され、0(%)以上を設定することによって運転再開されます。

オーバーライド変更時の加減速は、運転指令の±加速度値が使用されます。速度を上げるときは+加速度、下げるときは-加速度が使用されます。

設定レジスタはS8で、電源投入時は100(%)に初期化されます。範囲外の設定を行った場合は範囲内の値に書き換えられます。

5. パラメータ

システムのさまざまな設定を行うレジスタの集まりです。設定値はフラッシュメモリーに記憶されます。

5.1. 共通パラメータ

番号	名称	説明
C512	軸数	制御する軸数を設定します。初期値は32軸です。
C513	非常停止入力	非常停止信号の入力番号を設定します。0から31がIOコネクタの入力に対応し、入力がOFFしたとき非常停止状態になります。入力の論理変更はできません。使用しない場合は-1を設定します。
C514	サーボOFF時間	非常停止検出後AMPコネクタのSON出力はこの時間を待ってからOFFします。運転中に非常停止がかかった場合、サーボでモータを停止させたいときに使用します。設定は1mS単位です。
C515	言語の選択	パソコンツールを使用するときの言語を選択します。0で英語 1で日本語の設定となります。
C516	オーバーライドの基底	この値を基準にオーバーライド値を設定します。実際の速度は以下の式で求められます。 $A \square . 25 / C516$ デフォルト値は100です。0以下を設定した場合100として扱われます。

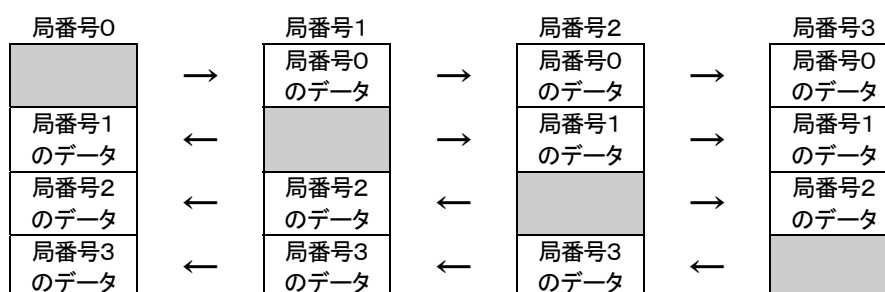
5.2. 通信パラメータ

COM1 番号	COM2 番号	名称	説明
C544	C560	通信仕様	通信仕様を設定します。 0 RS-232C(全二重) 1 RS-422(全二重 送信バッファは常にON) 2 RS-485(半二重 送信バッファは送信するときのみON)
C545	C561	ボーレート	ボーレートを設定します。 9600 9600ボー 19200 19200ボー 38400 38400ボー 57600 57600ボー 115200 115200ボー
C546	C562	ビット数	ビット数を設定します。 7 7ビット 8 8ビット
C547	C563	パリティ	パリティビットを設定します。 0 なし 1 奇数 2 偶数
C548	C564	ストップビット	ストップビット数を設定します。 0 1ビット 1 2ビット
C549	C565	接続機器	接続する機器を設定します。 0 なし 10 SCLink 11 タッチパネル
C550	C566	自局番号	SCLinkのとき自局番号(0~31)
C551	C567	接続局数	SCLinkで自局番号が0のとき接続台数
C552	C568	開始アドレス	SCLinkのときリンクレジスタの開始アドレス。
C553	C569	ワード数	SCLinkのとき転送ワード数。
C554	C570	送信遅延	RS-485接続のとき送信バッファをONにする遅延時間を設定します。mS単位で、0を設定すると最適な値を選択します。
C555	C571	タイムアウト	応答が無い場合、タイムアウト判定を行う時間を設定します。mS単位で、0を設定すると最適な値を選択します。

5.2.1. SCLinkについて

SCLinkを使用すると、複数台のSC2000シリーズのデータを交換することができます。シリアルポートを使用して最大32台の接続が行えます。通信は局番号0の親局と最大31台の子局でリンクレジスタのデータをやり取りします。各局は自分が出力するリンクレジスタの開始番地とデータ数をパラメータで設置します。各局は1度の送信で最大64ワード(1ワードは32ビット)を送信しますが、設定がそれより多い場合分割して送信されますので転送周期が遅くなります。

2台を接続する場合にはRS-232C、RS-422A、RS-485が選択できます。3台以上の場合、RS-485により接続を行います。



5.2.2. タッチパネルの接続について

タッチパネルは発紘電機製 MONITOUCH VSeries が使用できます。PLCの機種選択は三菱電機・AnA/N/Uシリーズを選択します。COM2使用時のパラメータの設定は以下の通りです。

パラメータ番号	名称	設定値
C576	通信仕様	タッチパネルの設定にあわせてます。
C577	ボーレート	
C578	ビット数	
C579	パリティ	
C580	ストップビット	
C581	接続機器	

5.3. 各軸パラメータ

各軸ごとの設定を行います。1軸のパラメータ数は16個で各軸ごと以下の番号が割り当てられています。

軸番号	パラメータ番号	軸番号	パラメータ番号
0	C0~C15	16	C256~C271
1	C16~C31	17	C272~C287
2	C32~C47	18	C288~C303
3	C48~C63	19	C304~C319
4	C64~C79	20	C320~C335
5	C80~C95	21	C336~C351
6	C96~C111	22	C352~C367
7	C112~C127	23	C368~C383
8	C128~C143	24	C384~C399
9	C144~C159	25	C400~C415
10	C160~C175	26	C416~C431
11	C176~C191	27	C432~C447
12	C192~C207	28	C448~C463
13	C208~C223	29	C464~C479
14	C224~C239	30	C480~C495
15	C240~C255	31	C496~C511

パラメータの内容は以下の通りです。

番号	名称	説明
C0	M係数	運転指令値とモータのパルス数の変換を行うための係数です。詳細は【4.2ユーザー単位】を参照してください。
C1	D係数	
C2	原点復帰位置	原点復帰が完了したときの位置を設定します。詳細は【4.1.2原点復帰】を参照してください。
C3	+ソフトリミット	+ソフトリミット値を設定します。詳細は【4.5ソフトリミット】を参照してください。±ソフトリミットに0を設定するとソフトリミットは無効となります。
C4	-ソフトリミット	-ソフトリミット値を設定します。
C5	最大パルス速度	使用するモータの最大回転数を逡倍後のエンコーダパルス数にして設定します。モータの指令パルスはこの速度で制限されます。
C6	速度モデル	速度基準をどの軸にするか設定します。詳細は【4.7速度モデル】を参照してください。

C7	原点復帰方法	<p>原点復帰の方向、方法を選択します。詳細は【4.1.2原点復帰】を参照してください。</p> <p>2 プラス方向 HLS、反転、Z相。 3 マイナス方向 HLS、反転、Z相。 4 プラス方向 HLS、非反転、Z相。 5 マイナス方向 HLS、非反転、Z相。 6 プラス方向 HLS、反転。 7 マイナス方向 HLS、反転。 8 プラス方向Z相。 9 マイナス方向 Z相。 12 プラス方向 PLS、反転、Z相。 13 マイナス方向 MLS、反転、Z相。</p>
C8	入力論理	<p>各軸の入力信号の論理を設定します。ビット対応で、該当ビット0のときノーマルOFF(A接点)、1のときノーマルON(B接点)となります。</p> <p>ビット0 LSコネクタPLS、MLS入力。 ビット1 LSコネクタHLS入力。 ビット2 LSコネクタNLS入力。 ビット3 LSコネクタTRIG入力。 ビット4 AMPコネクタEND入力。 ビット5 AMPコネクタALARM入力。 ビット7 HLS、TRIG入力のチャタリング防止フィルター。 これらの信号はエッジ検出しますがスイッチのチャタリングで反対エッジを検出してしまう場合に有効にすると、約4mSecのチャタリング除去フィルターが働きます。</p>
C9	出力論理	<p>各軸の出力信号の論理を設定します。ビット対応で、該当ビット0のときノーマルOFF(A接点)、1のときノーマルON(B接点)となります。</p> <p>ビット0 AMPコネクタSON出力。 ビット1 AMPコネクタERRRST出力。</p>
C10	エンコーダ入力形式	<p>エンコーダフィードバック信号を設定します。通倍設定はサーボアンプが指令1パルスに対してエンコーダの帰還量が等しくなるように設定します。またカウント方向も+指令を出した時にエンコーダが+カウントされるように設定します。</p> <p>0 エンコーダフィードバックなし。現在値はパルス出力をカウントした値となります。 1 エンコーダのフィードバック信号を1通倍でカウントします。 2 エンコーダのフィードバック信号を2通倍でカウントします。 3 エンコーダのフィードバック信号を4通倍でカウントします。設定1、2、3はA相進みで+カウントします。 5 エンコーダのフィードバック信号を1通倍でカウントします。 6 エンコーダのフィードバック信号を2通倍でカウントします。 7 エンコーダのフィードバック信号を4通倍でカウントします。 設定5、6、7はA相進みで-カウントします。</p>
C11	指令パルス形式	<p>サーボアンプの位置指令出力の形式を選択します。</p> <p>0 2パルス形式。CW、CCWの出力を行います。 1 1パルス形式。パルス+ディレクションの出力を行います。</p>
C12	位置決めパルス数	<p>位置決め完了の判定方法を設定します。</p> <p>-1 判定なし。 0 END信号による判定。 1~ フィードバックパルスによる判定。エンコーダのパルス数で設定します。</p>
C13	位置決め時間	<p>位置決めパルス数による判定を行う最大時間を設定します。時間内に範囲内に入らない場合エラー出力を行います。設定単位は1mSです。</p>

6. レジスタ

レジスタは実数レジスタを除いてモーションとラダー両方から共通に参照が行えます。16ビットまたは32ビット長でレジスタにより異なります。各レジスタはワード単位またはビット単位でのアクセスが行えます。特殊な機能を持つレジスタを操作することによってさまざまな操作を行うことができます。

実数レジスタはモーションだけ参照が行えます。

6.1. 入出力レジスタ

入力ポートの状態、出力の制御を行うためのレジスタです。電源投入時、入力は入力信号の状態、その他は0となります。表のRWはRWで読み書き可能、Rは読み出し専用のレジスタ、ビットを示します。

ワード参照	ビット参照	RW	説明
R0	R0. 0~R0. 31	R	IOコネクタからの32ビット入力です。 入力回路が通電状態のときONとなります。
R1	R1. 0~R1. 31	RW	IOコネクタへの32ビット出力です。 ONにすると出力が導通状態となります。
R2	R2. 0~R2. 31	RW	増設IOボードが実装されている場合、IOコネクタからの32ビットの入力となります。
R3	R3. 0~R3. 31	RW	増設IOボードが実装されている場合、IOコネクタへの32ビットの出力となります。
R4~R255	R4. 0 ~R255. 31	RW	汎用レジスタとして使用できます。

6.2. 軸レジスタ

軸制御を行うためのレジスタの集まりです。表中の□記号には軸番号(0~31)が入ります。表のRWはRWで読み書き可能、Rは読み出し専用のレジスタ、ビットを示します。

ワード参照	ビット参照	RW	説明	
A□. 0	A□. 0. 0	R	ONしていると駆動指令を受け付けられます。	
	A□. 0. 1	R	モータが加速中ONします。 駆動距離が短い場合など、ONしない場合があります。	
	A□. 0. 2	R	モータが指定速度に到達したときONします。 駆動距離が短い場合など、ONしない場合があります。	
	A□. 0. 3	R	モータが減速中にONします。	
	A□. 0. 4	R	原点復帰が完了するとONします。	
	A□. 0. 5	R	位置決め完了でONします。	
	A□. 0. 11	R	サーボON(A□. 24. 0)がONするとONになります。	
	A□. 0. 12	R	TRIG入力によって現在値がA□. 3に取り込まれたことを示します。 1スキャンだけオンします。	
	A□. 0. 16	R	LSコネクタPLS入力の状態です。	パラメータで入力論理の設定が行え、変換後の状態が読み出されます。
	A□. 0. 17	R	LSコネクタMLS入力の状態です。	
	A□. 0. 18	R	LSコネクタHLS入力の状態です。	
	A□. 0. 19	R	LSコネクタNLS入力の状態です。	
	A□. 0. 20	R	LSコネクタTRIG入力の状態です。	
	A□. 0. 21	R	エンコーダのZ相入力の状態です。	パラメータで入力論理の設定が行え、変換後の状態が読み出されます。
A□. 0. 22	R	AMPコネクタのEND入力の状態です。		
A□. 0. 23	R	AMPコネクタのALARM入力の状態です。		
A□. 1		R	指令現在値です。	
A□. 2		R	エンコーダからのフィードバック現在値です。 パルスからMD係数によって変換された値が入ります。	
A□. 3		R	LSコネクタTRIG信号の入った時点のフィードバック現在値です。 パルスからMD係数によって変換された値が入ります。	
A□. 4		R	駆動目標位置です。	
A□. 5		R	オーダー運転時、終了コードが出力されます。	
A□. 6		R	オーダー運転時、終了したステップ番号が出力されます。	

A□. 7	A□. 7. 0	R	減速停止指令により減速停止を行いました。
	A□. 7. 1	R	瞬時停止指令により瞬時停止を行いました。
	A□. 7. 2	R	+ソフトリミットを越えて駆動しようとした。
	A□. 7. 3	R	-ソフトリミットを越えて駆動しようとした。
	A□. 7. 4	R	LSコネクタPLS信号が入力されました。
	A□. 7. 5	R	LSコネクタMLS信号が入力されました。
	A□. 7. 6	R	AMPコネクタALARM信号が入力されました。
	A□. 7. 7	R	位置決めが完了しませんでした。
	A□. 7. 8	R	時間指定運転で速度が最高速度を超えました。
	A□. 7. 9	R	駆動距離が長すぎます。
	A□. 7. 10	R	円弧補間の位置指定が異常です。
	A□. 7. 15	R	非常停止が入力されました。
	A□. 7. 16	R	補間している他の軸にエラーが検出されました。
A□. 8	A□. 8. 0	R	補間中の軸がパラメータの最高速度により制限されました。
A□. 16	A□. 16. 0 ※1	RW	+ジョグ運転開始指令。詳細は【4.1.1ジョグ運転】を参照してください。
	A□. 16. 1 ※1	RW	-ジョグ運転開始指令。詳細は【4.1.1ジョグ運転】を参照してください。
	A□. 16. 2 ※1	RW	単軸運転開始指令。詳細は【4.1.3単軸運転】を参照してください。
	A□. 16. 3 ※1	RW	多軸運転開始指令。詳細は【4.1.4多軸運転】を参照してください。
	A□. 16. 4 ※1	RW	オーダー運転開始指令。オーダー運転実行中にここをOFFすると実行中のステップ終了後に停止します。詳細は【4.3オーダー運転】を参照してください。
	A□. 16. 5 ※1	RW	時間指定運転開始指令。詳細は【4.1.7単位時間運転】を参照してください。
	A□. 16. 6 ※1	RW	円弧運転開始指令。詳細は【4.1.5円弧運転】を参照してください。
	A□. 16. 7 ※1	RW	原点復帰運転開始指令。詳細は【4.1.2原点復帰】を参照してください。
	A□. 16. 8	RW	OFFの場合、位置指令は絶対値として扱います。ONの場合、位置指令は相対値として扱います。
	A□. 16. 10	RW	ONにすると運転中のモータを減速停止させます。A□. 7. 0のエラー状態で停止します。
	A□. 16. 11	RW	ONにすると運転中のモータを瞬時停止させます。A□. 7. 1のエラー状態で停止します。
	A□. 16. 12 ※1	RW	現在値設定指令。
	A□. 16. 14 ※1	RW	現在値調整指令。現在値をフィードバック値に合わせます。
	A□. 16. 15	RW	エラーリセット指令。
	A□. 16. 16	RW	ONにすると運転中のモータを減速中断させます。オーダー運転中にここをONすると現在実行中のステップを終了して次のオーダー運転のステップに進みます。立ち上がりエッジで動作を行います。
	A□. 16. 17	RW	ONにすると運転中のモータを瞬時中断させます。オーダー運転中にここをONすると現在実行中のステップを終了して次のオーダー運転のステップに進みます。立ち上がりエッジで動作を行います。
	A□. 17		RW
A□. 18		RW	速度番号指定。
A□. 19		RW	ステップ番号指定。
A□. 20		RW	時間指定。
A□. 21		RW	速度モデル指定。

A□. 22		RW	駆動軸指定。	
A□. 24	A□. 24. 0	RW	サーボアンプのサーボON(AMPコネクタSON)出力の制御です。	
	A□. 24. 1	RW	サーボアンプの偏差クリア(AMPコネクタDEV CLR)出力制御です。	パラメータで出力論理の設定が行えます。
	A□. 24. 2	RW	サーボアンプのエラークリア(AMPコネクタER RRST)出力の制御です。	
A□. 25		RW	オーバーライド値を設定します。電源投入時はパラメータ(C516)の値に初期化されます。数値を書き込むことによってモータの運転速度を調節できます。数値は0~C516の5倍の範囲で設定が行えます。 0を設定すると運転速度も0となり、途中停止を行うことができます。0以上に設定することによって運転は再開されます。	
A□. 26		RW	円弧通過点の位置番号を指定します。	
A□. 27		RW	円弧補間を行う軸をビット対応で指定します。2軸または3軸を指定(ON)します。	

※1 同時に複数ビットONすることはできません。またA□. 0. 0がONしていないと指令は受け付けられません。A□. 7のエラー発生中は受け付けられません。

6.3. システムレジスタ

特殊な機能を持つレジスタの集まりです。表のRWはRWで読み書き可能、Rは読み出し専用のレジスタ、ビットを示します。

ワード参照	ビット参照	RW	説明
S0	S0. 0	R	常時OFFです。
	S0. 1	R	常時ONです。
	S0. 2	R	電源投入直後の1サイクルだけONします。
	S0. 5	R	0. 01Sごとに1サイクルだけONします。
	S0. 6	R	0. 1Sごとに1サイクルだけONします。
	S0. 7	R	1Sごとに1サイクルだけONします。
S1		R	軸ボードの実装軸数が読み取れます。(4、8、12、...)
S2		RW	1秒で1加算されるのカウンタです。電源投入時からの秒数をカウントします。
S3		R	全ビットOFF(0)の固定値です。
S4		R	全ビットON(-1)の固定値です。
S5	S5. 0	RW	ONすると非常停止がかかります。 (共通パラメータの非常停止入力と同様の動作を行います)
	S5. 13	R	軸のエラー(A□. 7)のうちいずれかが発生している場合ONします。
	S5. 14	R	システムのエラー(S7、10、S11、S12)のうちいずれかが発生している場合ONします。
	S5. 15	R	全てのエラーのうちいずれかが発生している場合ONします。
S6		RW	ラダーで実行したDIV命令のあまりが書き込まれます。

S7	S7. 0	RW	ファイルレジスタのフラッシュメモリへの書き込み指令です。ONにすると書き込みが開始し、終了するとOFFになります。	
	S7. 4	RW	駆動レジスタ(位置、速度、オーダー)のフラッシュメモリへの書き込み指令です。ONにすると書き込みが開始し、終了するとOFFになります。	
	S7. 7	RW	パラメータレジスタのフラッシュメモリへの書き込み指令です。ONにすると書き込みが開始し、書き込み完了後ハードリセットが行われます。従って動作中のプログラムは全て停止状態となりますので、 <u>装置が動作中及びモータ運転中には行わないでください。</u>	
S10	S10. 0	RW	ラダーのフラッシュメモリへの保存のエラー。	
	S10. 1	RW	ラダーのフラッシュメモリからの読み出しのエラー。	
	S10. 2	RW	モーションのフラッシュメモリへの保存のエラー。	
	S10. 3	RW	モーションのフラッシュメモリからの読み出しのエラー。	
	S10. 4	RW	パラメータのフラッシュメモリへの保存のエラー。	
	S10. 5	RW	パラメータのフラッシュメモリからの読み出しのエラー。	
	S10. 6	RW	駆動レジスタのフラッシュメモリへの保存のエラー。	
	S10. 7	RW	駆動レジスタのフラッシュメモリからの読み出しのエラー。	
	S10. 8	RW	ファイルレジスタのフラッシュメモリへの保存のエラー。	
	S10. 9	RW	ファイルレジスタのフラッシュメモリからの読み出しのエラー。	
	S10. 16	RW	キープレジスタのエラー。	
S11	S11. 0	RW	ラダーのレジスタ番号が範囲外。	
S12	S12. 0	RW	モーションのレジスタ番号が範囲外。	
	S12. 1	RW	モーションスタックのオーバーフロー。	
S15		R	制御周期4mSecの内のアイドル時間。単位は μ Secでマイナス表示の場合は4mSec以内に収まっていないことを示します。	
S16	S16. 0~S16. 31	RW	モーション0~31番の起動。	ONにすることによってモーションプログラムが記述されていれば実行を開始します。モーションが終了するとOFFに戻ります。 またモーション実行中にOFFするとプログラムを強制的に終了させます。mov、wait命令実行の場合、その命令を実行終了してから停止します。※1
S17	S17. 0~S17. 31	RW	モーション32~63番の起動。	
S18	S18. 0~S18. 31	RW	モーション64~95番の起動。	
S19	S19. 0~S19. 31	RW	モーション96~127番の起動。	
S20	S20. 0~S20. 31	RW	モーション128~159番の起動。	
S21	S21. 0~S21. 31	RW	モーション160~191番の起動。	
S22	S22. 0~S22. 31	RW	モーション192~223番の起動。	
S23	S23. 0~S23. 31	RW	モーション224~255番の起動。	
S28	S28. 0~S28. 31	RW	SCLink使用時の接続確認ビット。各ビットはSCLinkの局番号に対応し、ONで通信は正常を示します。	
S29	S29. 0	RW	通信のチェックサムエラーが発生しました。	COM1、2 共通です。
	S29. 1	RW	通信の未定義コードを検出しました。	

※1 モーションの実行が1サイクル以内で終了した場合、ラダーでの立ち上がりエッジ検出は使用できません。この場合ON時実行命令を使用してください。

6.4. 駆動レジスタ

モータを駆動するための位置、速度およびオーダーを記憶するレジスタです。位置、速度及びオーダーはまとめてフラッシュメモリーに記憶されます。

6.4.1. 位置レジスタ

各軸の位置を記憶するレジスタです。□記号には位置番号(0~255)が、続0~31は軸番号が入ります。

位置の単位はユーザー単位です。ユーザー単位については【4.2ユーザー単位】を参照してください。

ワード参照	説明
P□. 0 ~P□. 31	255点31軸の位置データを設定します。 □に続0~31は軸番号です。

6.4.2. 速度レジスタ

速度、加速度及びS字時間を設定するレジスタです。□記号には速度番号(0~511)が入ります。

速度の単位はユーザー単位/S、加速度の単位はユーザー単位/S²です。ユーザー単位については【4.2ユーザー単位】を参照してください。

S字時間は加減速をS字パターンにするときの時間(mS)で設定します。この設定時間分だけ加速時間が長くなります。

ワード参照	説明
V□. 0	速度を設定します。単位はユーザー単位/Sです。
V□. 1	+加速度を設定します。単位はユーザー単位/S ² です。
V□. 2	-加速度を設定します。単位はユーザー単位/S ² です。
V□. 3	S字時間を設定します。単位はmSです。

6.4.3. オーダー

駆動ポイントを連結して駆動するオーダー運転の場合、ここに順番を設定します。□記号にはオーダー番号(0~2047)が入ります。

ワード参照	ビット参照	説明
○□. 0		命令コードを設定する8ビットのレジスタです。詳細は【4.1.6オーダー運転】を参照してください。
○□. 1		位置番号を設定する12ビットのレジスタです。
○□. 2		速度番号を設定する12ビットのレジスタです。
○□. 3	○□. 3. 0~ ○□. 3. 31	32ビットの拡張コードです。詳細は【4.1.6オーダー運転】を参照してください。

拡張コード領域の内容は以下のとおりです。

命令コード	位置番号	速度番号	拡張コード
0 終了	未使用	未使用	未使用
1 位置決め(絶対値)	位置番号	速度番号	軸マスク ビット対応でONの軸を駆動対象とします。
2 位置決め(相対値)			
3 パスポイント位置決め(絶対値)			
4 パスポイント位置決め(相対値)			
5 連続位置決め(絶対値)			
6 連続位置決め(相対値)			
7 円弧補間通過点	未使用	円弧補間を行う2軸または3軸を指定します。	
8 ドウエルタイム	未使用	未使用	ドウエル時間(mS)
9 コード出力	未使用	未使用	出力コード
10 ジャンプ	未使用	未使用	ジャンプ先ステップ番号

円弧補間を行う場合、最初のステップで通過点と円弧補間を行う軸を指定し、次に終点を位置決め/パスポイント位置決め/連続位置決めで指定します。絶対値/相対値の指定は終点で行い円弧補間通過点もこの設定に従います。

6.5. キープレジスタ

電源をOFFにしても値を保持しているレジスタです。□記号にはレジスタ番号(0~511)が入ります。

ワード参照	ビット参照	説明
K□	K□. 0~K□. 31	32ビットのレジスタです。

6.6. ファイルレジスタ

フラッシュメモリーによって値を保持しているレジスタです。□記号にはレジスタ番号(0～16382)が入ります。

ワード参照	ビット参照	説明
F□	F□. 0 ～F□. 16382	32ビットのレジスタです。

6.7. リンクレジスタ

外部機器と通信をするためのレジスタです。電源投入時に0クリアーされます。□記号にはレジスタ番号(0～2047)が入ります。

ワード参照	ビット参照	説明
L□	L□. 0～L□. 32	32ビットのレジスタです。

6.8. 拡張レジスタ

タッチパネル等と通信を行うためのレジスタです。電源投入時に0クリアーされます。□記号にはレジスタ番号(0～2047)が入ります。

ワード参照	ビット参照	説明
E□	E□. 0～E□. 15	16ビットのレジスタです。

6.9. パラメータレジスタ

パラメータを操作するのに使用します。パラメータはフラッシュメモリーに記憶されます。□記号にはパラメータ番号(0～1023)が入ります。パラメータ番号は【5パラメータ】を参照してください。

ワード参照	ビット参照	説明
C□	C□. 0～C□. 31	パラメータの値を読み書きします。

7. モーシヨンの説明

7.1. 文法

1行には命令語から始まる文または、ラベル文が記述できます。どちらもオプションで;に続けてコメントを書くことができます。コメントは命令の実行に影響を与えません。以下に例を示します。

```
命令語 オペランド ... ;コメント
ラベル: ;コメント
```

7.2. 数値

整数と実数が扱えます。

整数の数値範囲は-2147483648~+2147483647です。

実数の数値範囲は $-3.4 \times 10^{38} \sim 3.4 \times 10^{38}$ (有効桁数は7桁)です。

また、整数の表記方法として16進数が使用できます。先頭に '\$'、続けて0~9、A~Fで記述します。

7.3. レジスタの参照

モーシヨンからレジスタを参照する場合、即値とインデックスでの参照ができます。インデックスには数式が使用できます。以下に例を示します。

即値参照	A0. 1
インデックス参照	A(F1. 0). 16
	A(F23 * 3). (F14)

7.4. モーシヨン専用のレジスタ

汎用のレジスタのほかにモーシヨン専用のレジスタがあります。

Qレジスタは全プログラムから共通に使用できる実数のレジスタです。

YレジスタとZレジスタはプログラム番号ごとのレジスタです。例えばプログラム1番のY0とプログラム2番のY0の実体は別のものとなります。異なるプログラムから呼び出される共通のサブルーチンで、どのタイミングでレジスタが使用されるかわからない場合、YレジスタとZレジスタを使用することによって混乱を回避することができます。

これらのレジスタは電源投入時に0クリアされます。

ワード参照	説明
Q0～Q255	32ビットの実数レジスタです。 ワード参照のみ行えます。
Y0～Y15	プログラム番号毎の32ビットの整数レジスタです。ワード参照、ビット参照が行えます。
Z0～Z15	プログラム番号毎の32ビットの実数レジスタです。ワード参照のみ行えます。

7.5. 演算子

演算子は以下のものが使用できます。また演算子の優先順位を変えるのに()が使用できます。

演算子	動作	優先度
()	括弧	高 ↑
—	単項演算子 負符号	
～	単項演算子 ビットの反転	
* /	乗算 除算	
+ —	加算 減算	
&	ビットごとのアンド	↓ 低
^	ビットごとの排他的論理和	
	ビットごとのオア	
= <> < > <= >=	比較演算子	
>> <<	右シフト 左シフト	

7.6. 数式

数値または変数を演算子で結合したものを数式と呼びます。数式は演算子の優先順序に従い計算されます。括弧を用いて計算順序を入れ替えることができます。

数式中に整数と実数が混在するときは以下のルールで型変換が行われ実行されます。

	整数	実数
整数	整数演算を行い結果も整数	整数を実数に変換して実数演算を行い結果は実数
実数	整数を実数に変換して実数演算を行い結果は実数	実数演算を行い結果も実数

7.7. プログラム

モーションには複数のプログラムを書くことができます。

プログラムはprgまたはsub命令から始まり、次のprg、sub命令で終了します。

7.8. ラベル

ラベルはgoto、gosubの飛び先を定義します。アルファベットから始まるアルファベット、

数字で構成され、で終わる最大16文字の文字列です。大文字、小文字はそのまま記憶されますが比較時には大文字、小文字の区別をしません。

例

TEST1:	test1はgoto命令から参照されます。
prg 3, mainprg	mainprgはgosub命令から参照されます。
sub subroutine	subroutineはgosub命令から参照されます。

7.9. スタック

gosub命令でサブルーチンを呼び出すとき、戻り番地を記憶するためにスタックを使用します。gosub命令で書き込みが行われ、return命令で読み出しが行われ、最大10回分の戻り番地が記憶できます。

ラダーからプログラムが起動された直後、スタックは空の状態となります。

スタックの状態で動作の変化する命令を以下に述べます。

命令語	スタックが空のとき	スタックが空でないとき
prg	プログラムを停止します。	呼び出したgosub命令の次に戻ります。
sub		呼び出したgosub命令の次に戻ります。
stop		プログラムを停止します。
return		呼び出したgosub命令の次に戻ります。

7.10. 命令語

7.10.1.prg

書式

prg <no>、<label>

説明

ラダーから起動するモーシヨンプログラムの先頭に記述し、プログラム番号<no>とプログラム名<label>から構成されます。<no>は0から255の範囲で指定します。<label>はgosubから参照する場合に必要ですが、省略も可能です。

実行中にprg命令を見つけた場合、スタックが空でないなら呼び出したgosub命令の次に戻り、スタックが空ならプログラムは終了します。

7.10.2.sub

書式

sub <label>

説明

サブルーチンの先頭に記述します。<label>はgosub命令から参照されます。サブルーチン中のreturn命令を実行すると、呼ばれたgosub命令の直後に復帰します。

実行中にsub命令を見つけた場合、スタックが空でないなら呼び出したgosub命令の次に戻り、スタックが空ならプログラムは終了します。

7.10.3.if、elseif、else、endif

書式

```
if <expr>
elseif <expr>
else
endif
```

説明

ifから始まりendifで終了する条件命令です。間にelseif、elseをはさむことによって条件から1つを選択することができます。

例

```
if <expr1>
....
elseif <expr2>
....
elseif <expr3>
....
else
....
endif
```

<expr1>が真なら実行します。
 <expr2>が真なら実行します。
 <expr3>が真なら実行します。
 <expr1><expr2><expr3>
 いずれも真でないなら実行します。

7.10.4.goto

書式

```
goto <label>
```

説明

現在のプログラム中のラベルへ分岐します。ラベルを探す範囲はprgまたはsubで始まる1つのプログラム内で、他のプログラム中に同一のラベルがあっても参照されません。

7.10.5.gosub

書式

```
gosub <label>
```

説明

prgまたはsubのラベルに分岐します。分岐先でreturn命令を実行すると呼び出したgosub命令の次の行に戻ります。
スタックがいっぱいになるとS12. 0がONしエラーを知らせます。

7.10.6.return

書式

return

説明

呼び出しもとのgosub命令の次に戻ります。return命令実行時にスタックが空の場合プログラムは終了します。このときエラーにはなりません。

7.10.7.wait

書式

wait <expr>

説明

<expr>の値(1mS単位)プログラムを停止します。実際の実行時間は4mSの処理周期で判定されるため4mSの整数倍となります。またモーションの実行中のプログラム数が多い場合に実行時間に影響を受ける場合があります。

7.10.8.stop

書式

stop

説明

プログラムを終了します。

7.10.9.movh

書式

movh <expr:ax>

説明

パラメータに指定された原点復帰方法で<expr:ax>で指定する1軸を原点復帰し、動作終了後次のステップへ進みます。

命令の先駆けて速度(A□. 18)の指定を行います。

実行開始するためにmovh命令はA□. 16. 7をONにしますが、すでに命令実行中の場合、実行終了を待ってから行います。A□. 16. 7がONのままだと命令は無視されますので注意が必要です。

実行終了後、A□. 16の全ビットはOFFされます。は詳細は【4.1.2原点復帰】を参照してください。

7.10.10. movs、movsi

書式

```
movs <expr:ax>
```

説明

movsは絶対値、movsiは相対値で<expr:ax>で指定される1軸の単軸運転を行います。単軸運転が終了するとプログラムは次のステップへ進みます。

命令の先駆けて位置(A□. 17)、速度(A□. 18)の指定を行います。

実行開始するためにmovs、movsi命令はA□. 16. 2をONにしますが、すでに命令実行中の場合、実行終了を待ってから行います。A□. 16. 2がONのままだと命令は無視されますので注意が必要です。

実行終了後、A□. 16の全ビットはOFFされます。詳細は【4.1.3単軸運転】を参照してください。

7.10.11. movm、movmi

書式

```
movm <expr:ax>
```

説明

movmiは絶対値、movmiは相対値多軸運転の起動を行います。多軸運転が終了するとプログラムは次のステップへ進みます。

命令の先駆けて位置(A□. 17)、速度(A□. 18)、駆動軸指定(A□. 22)の指定を行います。

実行開始するためにA□. 16. 3をONにしますが、すでに命令実行中の場合、実行終了を待ってから行います。A□. 16. 3がONのままだと命令は無視されますので注意が必要です。

実行終了後、A□. 16の全ビットはOFFされます。詳細は【4.1.4多軸運転】を参照してください。

7.10.12. movc、movci

書式

```
movc<expr:ax>
```

説明

movciは絶対値、movciは相対値円弧運転の起動を行います。円弧運転が終了するとプログラムは次のステップへ進みます。

命令の先駆けて位置(A□. 17)、速度(A□. 18)、駆動軸指定(A□. 22)、円弧通過点の位置(A□. 26)、円弧補間を行う軸(A□. 27)の指定を行います。

実行開始するためにA□. 16. 6をONにしますが、すでに命令実行中の場合、実行終了を待ってから行います。A□. 16. 6がONのままだと命令は無視され

ますので注意が必要です。

実行終了後、A□. 16の全ビットはOFFされます。詳細は【4.1.5円弧運転】を参照してください。

7.10.13. **movo**

書式

movo <expr:ax>

説明

オーダー運転の起動を行います。オーダー運転が終了するとプログラムは次のステップへ進みます。

命令の先駆けてステップ番号(A□. 19)の指定を行います。

実行開始するためにA□. 16. 4をONにしますが、すでに命令実行中の場合、実行終了を待ってから行います。A□. 16. 4がONのままだと命令は無視されますので注意が必要です。

実行終了後、A□. 16の全ビットはOFFされます。詳細は【4.1.6オーダー運転】を参照してください。

7.10.14. **movt、movti**

書式

movt <expr:ax>

説明

movtは絶対値、movtiは相対値単位時間運転の起動を行います。単位時間運転が終了するとプログラムは次のステップへ進みます。

命令の先駆けて位置(A□. 17)、時間指定(A□. 20)の指定を行います。

実行開始するためにA□. 16. 5をONにしますが、すでに命令実行中の場合、実行終了を待ってから行います。A□. 16. 5がONのままだと命令は無視されますので注意が必要です。

実行終了後、A□. 16の全ビットはOFFされます。詳細は【4.1.7単位時間運転】を参照してください。

7.10.15. **pset**

書式

pset <expr:ax>

説明

psetは<expr:ax>で指定される1軸の現在値設定を行います。

命令の先駆けて位置(A□. 17)の指定を行います。

実行開始するためにpset命令はA□. 16. 12をONにしますが、すでに命令実行中の場合、実行終了を待ってから行います。A□. 16. 12がONのままだと命令は無視されますので注意が必要です。

実行終了後、A□. 16の全ビットはOFFされます。

7.10.16. padj

書式

padj <expr:ax>

説明

padjは<expr:ax>で指定される1軸の現在値調整を行います。

実行開始するためにpadj命令はA□. 16. 14をONにしますが、すでに命令実行中の場合、実行終了を待ってから行います。A□. 16. 14がONのままだと命令は無視されますので注意が必要です。

実行終了後、A□. 16の全ビットはOFFされます。

7.10.17. reset

書式

reset

説明

エラー状態から復帰するために必要な処理を行い、エラーを解除します。

7.10.18. 代入文

書式

<reg> = <expr>

説明

右辺の計算式の結果を左辺のレジスタに代入します。

左辺、右辺ともワード指定、ビット指定が可能で、左辺がビット指定、右辺がワード指定の場合右辺の最下位ビットが代入されます。

7.10.19. コメント文

書式

; <remarks>

説明

; の後に任意の文字列を書くことができます。この行は実行時には何もしません。

7.11. 関数

関数は計算式の中で使用されます。

7.11.1.sin()、cos()、tan()

書式

```
sin(<expr>)  
cos(<expr>)  
tan(<expr>)
```

説明

<expr>のsin、cos、tanの計算を行います。角度の表現はradです。結果は実数を返します。

7.11.2.asin()、acos()、atan()、atan2()

書式

```
asin(<expr>)  
acos(<expr>)  
atan(<expr>)  
atan2(<expr:y>、<expr:x>)
```

説明

<expr>のasin、acos、atanの計算を行います。角度の表現はradです。atan2は<expr:y>、<expr:x>の2つの引数から計算され、結果は $\pm\pi$ で得られます。結果は実数を返します。

7.11.3.sqrt()

書式

```
sqrt(<expr>)
```

説明

<expr>の平方根を計算します。結果は実数を返します。

7.11.4.pow()

書式

```
pow(<expr:x>、<expr:y>)
```

説明

<expr:x>の<expr:y>乗を計算します。結果は実数を返します。

7.11.5.ceil()、floor()

書式

ceil(<expr>)
floor(<expr>)

説明

<expr>の小数点以下の、ceil切り捨て、floorは切り上げを行います。結果は実数を返します。

7.11.6.float()、int()

書式

float(<expr>)
int(<expr>)

説明

floatは<expr>を実数に型変換します。
intは<expr>を整数に型変換します。

7.11.7.abs()

書式

abs(<expr>)

説明

<expr>の絶対値を計算します。<expr>が整数の場合結果は整数、実数の場合実数を返します。

7.11.8.PI

書式

PI

説明

π の値を実数で返します。

7.11.9.PI2

書式

PI2

説明

$\pi \times 2$ の値を実数で返します。

8. ラダーの説明

8.1. 文法

命令語は以下の構造をもちます。オペランドの数は命令語によって異なります。

<命令語> <オペランド1>、<オペランド2>、...

命令語の最後が“l”で終わるとオペランドに即値指定の命令語、“r”で終わるとOFFからONの立ち上がり時に実行される命令語となります。

8.2. アキュムレータとスタック

ビット単位の演算結果を格納する場所をアキュムレータ(以降Acc)と呼び、同時に次の命令の実行条件を決定します。Accは1ビットのレジスタです。

LD命令、比較命令などを実行したときAccの値は自動的にスタックにプッシュされます。スタックは31ビットの記憶領域で最初にプッシュされたものが最後にポップされる構造をもちます。31回までプッシュした状態を記憶できます。

8.3. 命令一覧表

	基本	反転	立ち上がり	即値	即値、立ち上がり
基本命令	LD	LDN			
	LDA				
	OUT				
	AND	ANDN			
	ANDS				
	OR	ORN			
	ORS				
	LDSA				
	LDS				
	XOR	XORN			
	CPL				
	RAISE				
	FALL				
	MC				
	MCR				
	TMR			TMRr	
	TMRI			TMRIr	
	TMRRST			TMRRSTr	
	EQ				EQI
	NE				NEI
GE				GEI	
GT				GTI	
LE				LEI	
LT				LTI	
応用命令	SET		SETr		
	RES		RESr		
	MOV		MOVr	MOVI	MOVIr
	ADD		ADDR	ADDI	ADDIr
	SUB		SUBr	SUBI	SUBIr
	MUL		MULr	MULI	MULIr
	DIV		DIVr	DIVI	DIVIr
	NRG		NEGr		
	SFT		SFTr		
	LDBCD		LDBCDr		
	OUTBCD		OUTBCDr		
	XTRACT		XTRACTr		
	MERGE		MERGEr		
	MOVB		MOVBr		
	MOVBw		MOVBrw		
	MOVBL		MOVBLr		
	SIX				
DIX					
IXR					
NOP					

8.4. 基本命令

基本命令語の説明を行います。
基本命令は、Accの状態を制御します。

8.4.1. LD、LDN

書式

LD <bit>
LDN <bit>

実行条件

常時。

説明

LDは<bit>の状態をAccにロードします。
LDNは<bit>の反転状態をAccにロードします。
直前のAccの状態はスタックにプッシュされ後で使用することができます。

8.4.2. LDA

書式

LDA

実行条件

常時。

説明

Accの状態をスタックにプッシュします。この命令は現在のAccの状態を別の回路で使用するためにスタックに保存するために使用します。

8.4.3. OUT

書式

OUT <bit>

実行条件

常時。

説明

Accの状態を<bit>に出力します。

8.4.4. AND、ANDN

書式

AND <bit>
ANDN <bit>

実行条件

常時。

説明

ANDはAccと<bit>の積をとり、Accの格納します。
ANDNはAccと<bit>の反転との積をとり、Accの格納します。

8.4.5. ANDS

書式

ANDS

実行条件

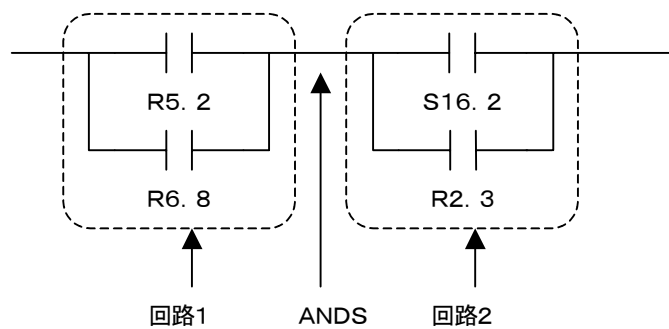
常時。

説明

Accの状態とスタックの状態の論理積を取りAccに格納します。実行後スタックは1段ポップされます。

例

LD R5. 2	回路1
OR R6. 8	
LD S16. 2	回路2
OR R2. 3	
ANDS	結合



8.4.6. OR、ORN

書式

OR <bit>
ORN <bit>

実行条件

常時。

説明

ORはAccと<bit>の論理和をとりAcc1に格納します。
ORNはAccと<bit>の反転との論理和をとりAccに格納します。

8.4.7. ORS

書式

ORS

実行条件

常時。

説明

Accの状態とスタックの状態の論理和を取りAccに格納します。実行後スタックは1段ポップされます。

8.4.8. LDSA

書式

LDSA

実行条件

常時。

説明

スタックの先頭をAccにロードします。実行後スタックはポップされず、以前のままの状態です。

8.4.9. LDS

書式

LDS

実行条件

常時。

説明

スタックの先頭をAccにロードします。実行後スタックは1段ポップされます。

8.4.10.XOR、XORN

書式

XOR <bit>
XORN <bit>

実行条件

常時。

説明

XORはAccと<bit>との排他的論理和をとりAccに格納します。
XORNはAccと<bit>の反転との排他的論理和をとりAccに格納します。

8.4.11.CPL

書式

CPL

実行条件

常時。

説明

Accの状態を反転させます。OFFならONに、ONならOFFになります。

8.4.12.RAISE

書式

RAISE

実行条件

常時。

説明

AccのOFFからONになった1周期だけAccをONにします。
立ち上がりエッジを検出します。

8.4.13.FALL

書式

FALL

実行条件

常時。

説明

AccのONからOFFになった1周期だけAccをONにします。
たち下がりエッジを検出します。

8.4.14.MC

書式

MC

実行条件

常時。

説明

MCとMCR命令に囲まれたプログラムの実行を制御することができます。制御する命令はLD、LDN、EQ、EQI、NE、NEI、GE、GEI、LE、LEIでこれらの命令実行後のAccをOFFとします。

MCとMCR命令のペアは入れ子で使用することが可能です。つまりMCとMCR命令間のプログラムにさらにMCとMCR命令を記述することができ、複雑な回路を整理することが可能です。入れ子は31回まで可能です。

MCR命令はAccの状態にかかわらず常に実行を行います。

例

```
LD R5. 2
MC
LD S16. 2           R5. 2がONならAccはS16. 2、OFFならAccはOFF
OR R2. 3
OUT R1. 7
MCR
```

8.4.15.MCR

書式

MCR

実行条件

常時。

説明

MC命令を参照してください。

8.4.16.TMR、TMRr、TMRI、TMRIr

書式

TMR <reg>
 TMRr <reg>
 TMRI <imm>
 TMRIr <imm>

実行条件

TMR、TMRIはAccがONのとき実行。
 TMRr、TMRIrはAccがOFFからONになるとき実行。

説明

TMR、TMRrは<reg>の時間(1mS単位)だけAccをONにします。
 TMRI、TMRIrは<imm>の時間(1mS単位)だけAccをONにします。
 TMR、TMRIは入力がONのとき出力もON、ONからOFFになるとそこから設定時間後に出力をOFFします。
 途中で出力をOFFするには、TMRRST命令を使用します。

8.4.17.TMRRST、TMRRSTr

書式

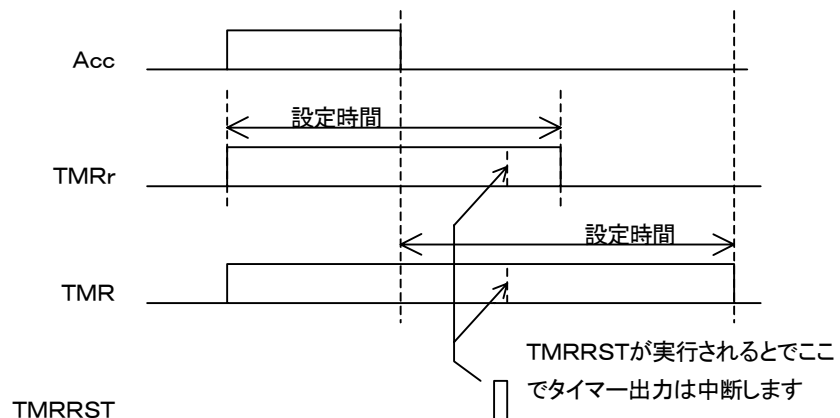
TMRRST
 TMRRSTr

実行条件

TMRRSTはAccがONのとき実行。
 TMRRSTrはAccがOFFからONになるとき実行。

説明

この命令の直後のTMR、TMRr、TMRI、TMRIr命令の出力をOFFします。直後のタイマー命令1つに対して有効です。



8.4.18.EQ、NE、GE、GT、LE、LT、EQI、NEI、GEI、GTI、LEI、LTI

書式

EQ <reg1>, <reg2>
 NE <reg1>, <reg2>
 GE <reg1>, <reg2>
 GT <reg1>, <reg2>
 LE <reg1>, <reg2>
 LT <reg1>, <reg2>
 EQI <imm>, <reg2>
 NEI <imm>, <reg2>
 GEI <imm>, <reg2>
 GTI <imm>, <reg2>
 LEI <imm>, <reg2>

実行条件

常時。

説明

比較を行い結果をAccに格納します。実行時現在のAccの状態をスタックに格納してから行います。比較前のAccと演算する場合はANDS、ORS命令を使用します。

比較の結果AccがONする条件は以下のとおりです。

命令	結果が真になる条件		
EQ EQI	<reg1> <imm>	=	<reg2>
NE NEI	<reg1> <imm>	<>	<reg2>
GE GEI	<reg1> <imm>	<=	<reg2>
GT GTI	<reg1> <imm>	<	<reg2>
LE LEI	<reg1> <imm>	>=	<reg2>
LT LTI	<reg1> <imm>	>	<reg2>

例

LD R5, 2
 EQ R10, F25
 ANDS

R5. 2がONかつ、R10とF25が等しいときON

8.5. 応用命令

応用命令の説明を行います。

応用命令はAccの結果によって動作を制御される命令です。

8.5.1. SET、SETr

書式

SET <bit>
SETr <bit>

実行条件

SETはAccがONのとき実行。
SETrはAccがOFFからONになるとき実行。

説明

<bit>をONします。OUT命令とは異なり実行時だけ出力を行いますのでSET命令でONさせたビットはほかの回路でリセットすることができます。

8.5.2. RES、RESr

書式

RES <bit>
RESr <bit>

実行条件

RESはAccがONのとき実行。
RESrはAccがOFFからONになるとき実行。

説明

<bit>をOFFします。OUT命令とは異なり実行時だけ出力を行いますのでRES命令でOFFさせたビットはほかの回路でセットすることができます。

8.5.3. MOV、MOVr、MOVI、MOVIr

書式

MOV <reg:src>, <reg:dist>
MOVr <reg:src>, <reg:dist>
MOVI <imm>, <reg:dist>
MOVIr <imm>, <reg:dist>

実行条件

MOV、MOVIはAccがONのとき実行。
MOVr、MOVIrはAccがOFFからONになるとき実行。

説明

MOV、MOVrは<reg:src>を<reg:dist>にコピーします。
MOVI、MOVIrは<imm>を<reg:dist>にコピーします。

8.5.4. ADD、ADDr、ADDI、ADDIr

書式

ADD <reg:src>, <reg:dist>
ADDr <reg:src>, <reg:dist>
ADDI <imm>, <reg:dist>
ADDIr <imm>, <reg:dist>

実行条件

ADD、ADDIはAccがONのとき実行。
ADDr、ADDIrはAccがOFFからONになるとき実行。

説明

ADD、ADDrは<reg:dist>と<reg:src>を加算し<reg:dist>に格納します。<reg:src>の値は変化しません。

ADDI、ADDIrは<reg:dist>と<imm>を加算し<reg:dist>に格納します。

8.5.5. SUB、SUBr、SUBI、SUBIr

書式

SUB <reg:src>, <reg:dist>
SUBr <reg:src>, <reg:dist>
SUBI <imm>, <reg:dist>
SUBIr <imm>, <reg:dist>

実行条件

SUB、SUBIはAccがONのとき実行。
SUBr、SUBIrはAccがOFFからONになるとき実行。

説明

SUB、SUBrは<reg:dist>から<reg:src>を減算し<reg:dist>に格納します。<reg:src>の値は変化しません。

SUBI、SUBIrは<reg:dist>から<imm>を減算し<reg:dist>に格納します。

8.5.6. MUL、MULr、MULI、MULIr

書式

MUL <reg:src>, <reg:dist>
MULr <reg:src>, <reg:dist>
MULI <imm>, <reg:dist>
MULIr <imm>, <reg:dist>

実行条件

MUL、MULIはAccがONのとき実行。
MULr、MULIrはAccがOFFからONになるとき実行。

説明

MUL、MULrは<reg:dist>と<reg:src>をかけて<reg:dist>に格納します。<reg:src>の値は変化しません。

MULI、MULIrは<reg:dist>と<imm>をかけて<reg:dist>に格納します。

8.5.7. DIV、DIVr、DIVI、DIVIr

書式

DIV <reg:src>, <reg:dist>
DIVr <reg:src>, <reg:dist>
DIVI <imm>, <reg:dist>
DIVIr <imm>, <reg:dist>

実行条件

DIV、DIVIはAccがONのとき実行。
DIVr、DIVIrはAccがOFFからONになるとき実行。

説明

DIV、DIVrは<reg:dist>を<reg:src>で割って<reg:dist>に格納します。
<reg:src>の値は変化しません。
DIVI、DIVIrは<reg:dist>を<imm>で割って<reg:dist>に格納します。
割り算のあまりはS6レジスタに書き込まれます。

8.5.8. NEG、NEGr

書式

NEG <reg:src>, <reg:dist>
NEGr <reg:src>, <reg:dist>

実行条件

NEGはAccがONのとき実行。
NEGrはAccがOFFからONになるとき実行。

説明

<reg:src>を符号反転して<reg:dist>に格納します。<reg:src>の値は変化しません。

8.5.9. SFT、SFTTr

書式

SFT <reg:src>, <reg:dist>, <imm:nbit>
SFTTr <reg:src>, <reg:dist>, <imm:nbit>

実行条件

SFTはAccがONのとき実行。
SFTTrはAccがOFFからONになるとき実行。

説明

<reg:src>を<imm:nbit>ビットシフトして<reg:dist>に格納します。<imm:nbit>が正の場合左シフト、負の場合右シフトを行います。<reg:src>の値は変化しません。

8.5.10.LDBCD、LDBCDr

書式

LDBCD <reg:src>, <reg:dist>

LDBCDr <reg:src>, <reg:dist>

実行条件

LDBCDはAccがONのとき実行。

LDBCDrはAccがOFFからONになるとき実行。

説明

<reg:src>のBCDコードをバイナリーに変換して<reg:dist>に格納します。
<reg:src>の値は変化しません。

8.5.11.OUTBCD、OUTBCDr

書式

OUTBCD <reg:src>, <reg:dist>

OUTBCDr <reg:src>, <reg:dist>

実行条件

OUTBCDはAccがONのとき実行。

OUTBCDrはAccがOFFからONになるとき実行。

説明

<reg:src>のバイナリーをBCDコードに変換して<reg:dist>に格納します。
<reg:src>の値は変化しません。

8.5.12.XTRACT, XTRACTr

書式

XTRACT <reg:src>, <imm:bpos>, <reg:dist>, <imm:nbit>
 XTRACTr <reg:src>, <imm:bpos>, <reg:dist>, <imm:nbit>

実行条件

XTRACTはAccがONのとき実行。
 XTRACTrはAccがOFFからONになるとき実行。

説明

<reg:src>の<imm:bpos>の示すビット番号から<imm:nbit>の示すビット数だけ<reg:dist>の0ビット目にコピーします。<reg:src>は変化しません。<reg:dist>は下位ビットから<imm:nbit>だけ変化します。

例

XTRACT F0, 4, F1, 6

転送元		転送先
F0. 4	→	F1. 0
F0. 5	→	F1. 1
F0. 6	→	F1. 2
F0. 7	→	F1. 3
F1. 4~F1. 31	→	F1. 4~F1. 31

8.5.13.MERGE、MERGEr

書式

MERGE <reg:src>, <imm:nbit>, <reg:dist>, <imm:bpos>
 MERGEr <reg:src>, <imm:nbit>, <reg:dist>, <imm:bpos>

実行条件

MERGEはAccがONのとき実行。
 MERGErはAccがOFFからONになるとき実行。

説明

<reg:src>の0ビット目から<imm:nbit>の示すビット数だけ<reg:dist>の<imm:bpos>の示すビット番号にコピーします。<reg:src>は変化しません。<imm:nbit>から<reg:dist>ビットだけ変化します。

例

MERGE F0, 4, F1, 6

転送元		転送先
F1. 0~F1. 5	→	F1. 0~F1. 5
F0. 0	→	F1. 6
F0. 1	→	F1. 7
F0. 2	→	F1. 8
F0. 3	→	F1. 9
F1. 10~F1. 31	→	F1. 10~F1. 31

8.5.14.MOVB、MOVBr

書式

MOVB <reg:src>, <reg:dist>, <imm:n>

MOVBr <reg:src>, <reg:dist>, <imm:n>

実行条件

MOVBはAccがONのとき実行。

MOVBrはAccがOFFからONになるとき実行。

説明

<reg:src>から<reg:dist>へ<imm:n>の示すワード数だけコピーします。ビット長の異なるレジスタを指定した場合、下位ビットのみコピーされます。<reg:src>の値は変化しません。<imm:n>の範囲は1から63です。

SIX、DIX命令によるインデックス修飾が行えます。

例

MOVB F0, F10, 4

転送元		転送先
F0. 0~F0. 31	→	F10. 0~F10. 31
F1. 0~F1. 31	→	F11. 0~F11. 31
F2. 0~F2. 31	→	F12. 0~F12. 31
F3. 0~F3. 31	→	F13. 0~F13. 31

8.5.15.MOVBW、MOVW_r

書式

MOVBW <reg:src16>, <reg:dist32>, <imm:n>
 MOVW_r <reg:src16>, <reg:dist32>, <imm:n>

実行条件

MOVBWはAccがONのとき実行。
 MOVW_rはAccがOFFからONになるとき実行。

説明

<reg:src16>から<reg:dist32>へ<imm:n>の示すワード数だけコピーします。このとき16ビットレジスタ2個を32ビットレジスタ1個にコピーします。コピーするワード数は32ビットレジスタの数で指定します。<reg:src>の値は変化しません。<imm:n>の範囲は1から63です。

SIX、DIX命令によるインデックス修飾が行えます。

例

MOVBW E0, F10, 3

転送元		→	転送先	
E0. 0~E0. 15		→	F10. 0~F10. 15	
E1. 0~E1. 15		→	F10. 16~F10. 31	
E2. 0~E2. 15		→	F11. 0~F11. 15	
E3. 0~E3. 15		→	F11. 16~F11. 31	
E4. 0~E4. 15		→	F12. 0~F12. 15	
E5. 0~E5. 15		→	F12. 16~F12. 31	

8.5.16.MOVBL、MOVBLr

書式

```
MOVBL <reg:src32>, <reg:dist16>, <imm:n>
MOVBLr <reg:src32>, <reg:dist16>, <imm:n>
```

実行条件

MOVBLはAccがONのとき実行。
MOVBLrはAccがOFFからONになるとき実行。

説明

<reg:src32>から<reg:dist16>へ<imm:n>の示すワード数だけコピーします。このとき32ビットレジスタ1個を16ビットレジスタ2個にコピーします。コピーするワード数は32ビットレジスタの数で指定します。<reg:src>の値は変化しません。<imm:n>の範囲は1から63です。

SIX、DIX命令によるインデックス修飾が行えます。

例

```
MOVBL F0, E10, 3
```

転送元		→	転送先	
F10. 0	~F10. 15		E0. 0	~E0. 15
F10. 16	~F10. 31		E1. 0	~E1. 15
F11. 0	~F11. 15		E2. 0	~E2. 15
F11. 16	~F11. 31		E3. 0	~E3. 15
F12. 0	~F12. 15		E4. 0	~E4. 15
F12. 16	~F12. 31		E5. 0	~E5. 15

8.5.17.SIX、DIX

書式

```
SIX <reg>
DIX <reg>
```

実行条件

常時。

説明

SIXは<reg>の値をソースインデックスに設定します。

DIXは<reg>の値をディスティネーションインデックスに設定します。

8.5.18.IXR

書式

IXR

実行条件

常時。

説明

ソースインデックスおよびディスティネーションインデックスの値を0クリアします。

8.5.19.NOP

書式

NOP

実行条件

常時。

説明

何もせずに次に進みます。

9. プログラム例

9.1. ラダーによるプログラム

9.1.1. ジョグ運転

LD R0. 0	マニュアルモード入力
AND R0. 2	+ジョグスイッチ
MOVI 10, A0. 18	速度番号10設定
OUT A0. 16. 0	+ジョグ開始

マニュアルモード入力がONで+JOGスイッチがONするとジョグ運転を開始します。速度は10番を使用しています。

9.1.2. モーションの開始

LDN R0. 0	オートモード入力
AND R0. 4	モーション開始スイッチ
SETr S16. 3	モーションプログラム3の開始

オートモードでモーション開始スイッチがONするとモーションプログラム3が開始します。

9.2. モーションによるプログラム

9.2.1. 原点復帰

prg 3	モーションプログラム3開始
A3. 18=14	速度番号14設定
H10:	
if A3. 0. 19=0	もしNLSがOFFだったら
A3. 16. 0=1	3軸ジョグON
wait 0	ほかのプログラムにCPUを開放
goto H10	
endif	
A3. 16. 0=0	3軸ジョグOFF
;	
A3. 18=15	速度番号15設定
movh 3	3軸の原点復帰開始
stop	終了

3軸の原点復帰を行います。原点復帰終了後プログラムは停止します。

9.2.2. インデックスを使った全軸一括位置決め

prg 20	モーションプログラム20開始
R100=0	インデックスの初期化
loop:	ラベル
A(R100). 17=18	位置番号18設定
A(R100). 18=22	速度番号22設定
A(R100). 22=1<<R100	駆動軸指定
A(R100. 16. 3=1	多軸運転開始指令ON
R100=R100+1	インデックス+1
if R100<32	32軸以下なら
goto loop	ラベルloopへ分岐
endif	if文終了
stop	モーションプログラム終了

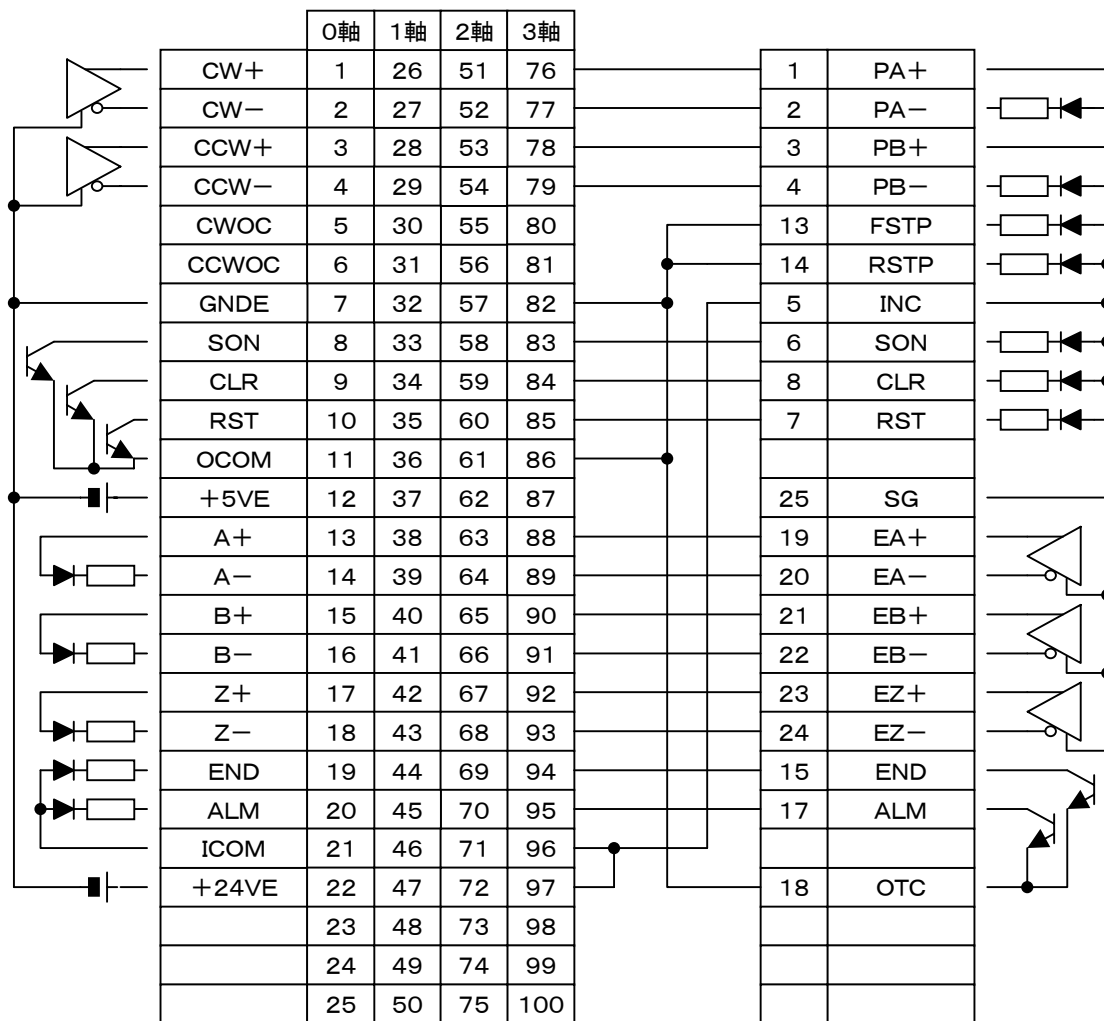
32軸すべて位置決めを開始します。各補間演算器で1軸を駆動します。直接多軸位置決め指令をONしていますので運転の終了を待たずにモーションプログラムは終了します。

10. サーボアンプとの接続例

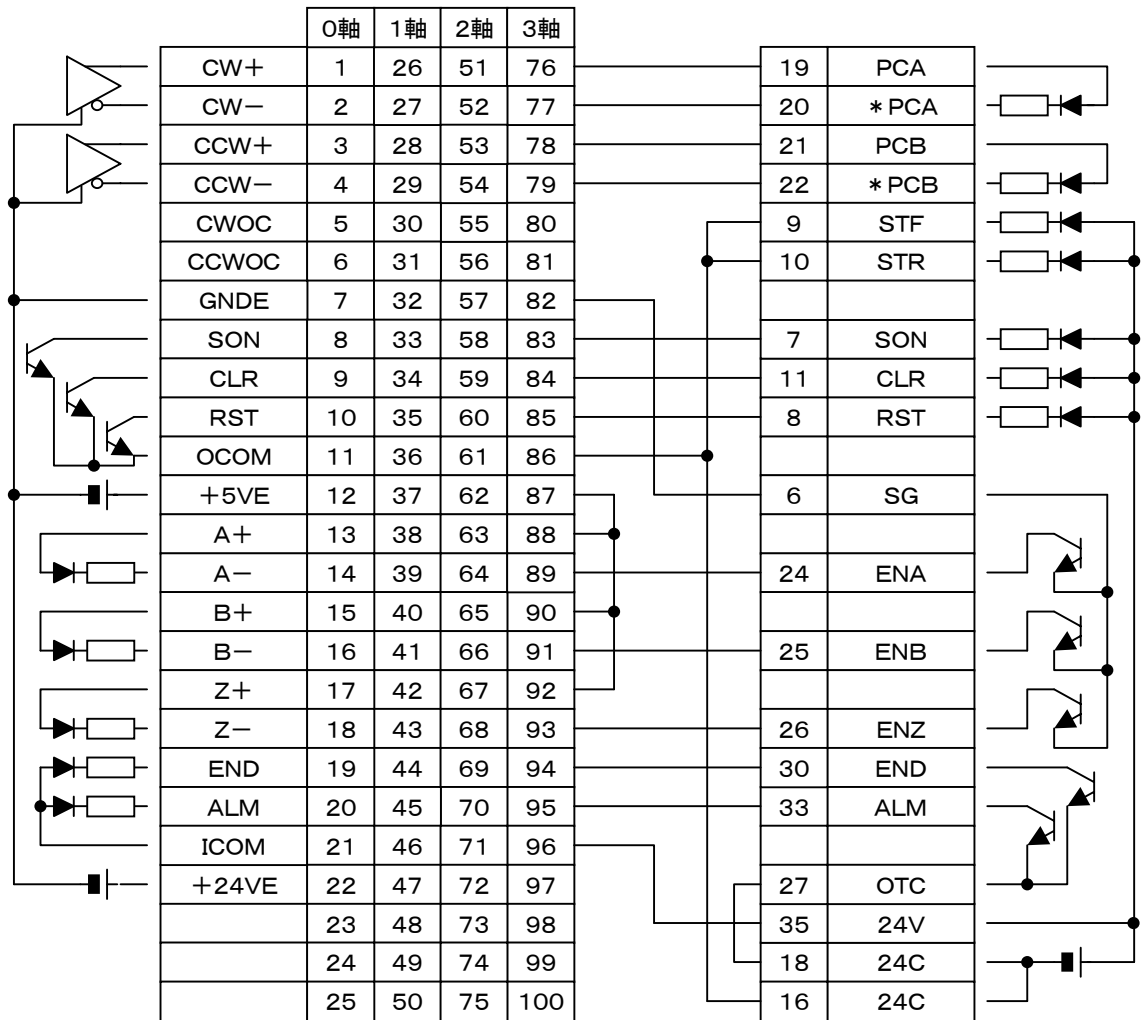
10.1. G P P型サーボアンプとの接続

コントローラ

GPP型サーボアンプ



10.2. A F S 型サーボアンプとの接続



索引

索引

2

24VIN..... 30
24VOUT..... 30

5

5VOUT..... 30

A

A..... 27, 30
abs()..... 71
acos()..... 70
ADD..... 82
ADDI..... 82
ALARM..... 27, 30, 33
AMPコネクタ..... 25
AND..... 75
ANDN..... 75
ANDS..... 75
asin()..... 70
atan()..... 70
atan2()..... 70

B

B..... 27, 30

C

CCW..... 26, 31
CCWOC..... 26, 31
ceil()..... 70
cos()..... 70
CPL..... 77
CPU LED..... 19, 20
CW..... 26, 31
CWOC..... 26, 31

D

DEVCLR..... 27, 31
DIV..... 84
DIVI..... 84
DIX..... 90

D係数..... 45

E

else..... 65
elseif..... 65
END..... 27, 30
endif..... 65
EQ..... 80
EQI..... 80
ERRRST..... 27, 31

F

FALL..... 77
FG端子..... 19, 20
float()..... 71
floor()..... 70

G

GE..... 80
GEI..... 80
GTI..... 80
gosub..... 65
goto..... 65
GT..... 80

H

HLS..... 28, 30

I

if..... 65
INO..... 30
INO~31..... 24
int()..... 71
IOコネクタ..... 19, 20
IXR..... 91

L

LD..... 74
LDA..... 74
LDBCD..... 85
LDN..... 74

LDS.....	76	OUT0~31	24
LDSA.....	76	OUTBCD.....	85
LE	80	P	
LEI	80	padj.....	69
LSコネクタ	19, 20, 28	PI	71
LT	80	PI2.....	71
LTI.....	80	PLS	28, 30, 33
M		pow()	70
MC	78	POWER LED	19, 20
MCR	78	prg	64
MERGE	87	pset.....	68
MLS.....	28, 30, 33	R	
MOV	82	RAISE	77
MOVB	88	RDY	22, 31
MOVBL.....	90	RES.....	81
MOVBW.....	89	reset	69
movc.....	67	return.....	66
movci.....	67	RS-232C.....	29, 51
movh.....	66	RS-422A.....	29, 51
MOVI	82	RS-485	29, 51
movm	67	S	
movmi.....	67	SC2004.....	14
movo.....	68	SC2008.....	14
movs.....	67	SCLink.....	51
movsi.....	67	SET	81
movt.....	68	SFT.....	84
movti.....	68	sin()	70
MUL.....	83	SIX.....	90
MULI.....	83	SON	27, 31
M係数.....	45	sqrt().....	70
N		stop	66
NE.....	80	sub	64
NEI.....	80	SUB.....	83
NEG.....	84	SUBI.....	83
NLS	28, 30	T	
NOP	91	tan()	70
O		TMR.....	79
OR.....	76	TMRI.....	79
ORN.....	76	TMRRST.....	79
ORS.....	76	TRIG	28, 30, 32, 47
OUT	74		
OUT0.....	31		

索引

W

wait 66

X

XOR 77

XORN 77

XTRACT 86

Z

Z 27, 30

あ

アイドル時間 34, 58

アキュムレータ 72

アンプコネクタ 19, 20

い

位置レジスタ 59

一般仕様 14

え

S字時間 45

エラー 48

エラーリセット 56

円弧運転 42

演算子(モーション) 63

お

応用命令 81

オーダー 60

オーダー運転 43

オーバーライド 48, 57

か

各軸パラメータ 52

拡張レジスタ 61

加速度 45

関数 69

き

キープレジスタ 60

基本命令 74

共通パラメータ 49

く

駆動レジスタ 59

け

減速中断 56

減速停止 56

原点復帰 37, 93

こ

コメント文 69

し

軸レジスタ 55

軸指定 41

システムレジスタ 57

瞬時中断 56

瞬時停止 56

ジョグ運転 36, 93

す

数式(モーション) 63

数値(モーション) 62

スタック(モーション) 64

スタック(ラダー) 72

せ

設置 15

全二重 50

そ

速度 45

速度モデル 47

速度レジスタ 59

速度設定 45

ソフトリミット 46

た

代入文 69

多軸運転 41

タッチパネル 51

単位時間運転 45

単軸運転 40

つ

通信コネクタ 19, 20, 29

通信パラメータ 50

て

点検 14

電源コネクタ 19, 20, 22
 電源系統図 31

に

入出力レジスタ 54
 入出力仕様 30

は

パラメータ 15, 61
 半二重 50

ひ

非常停止 46

ふ

ファイルレジスタ 61
 フラッシュメモリー 15
 プログラム (モーション) 63
 文法 (モーション) 62
 文法 (ラダー) 72

ほ

保守 14

も

モーション 34, 93

ゆ

ユーザー単位 45, 59

ら

ラダー 34
 ラベル 63

り

リンクレジスタ 61

れ

レジスタ 54
 レジスタの参照 (モーション) 62

株式会社 ワコー技研

本社 〒230-0051 横浜市鶴見区鶴見中央3-27-2
TEL(045)502-4441
大阪営業所 〒532-0011 大阪市淀川区西中島5-7-19
TEL(06)6390-6251
名古屋出張所 〒464-0851 名古屋市千種区今池南29-24
TEL(052)735-5989

URL <http://www.wacogiken.co.jp/>

SC2000アナログボード

仕様

項目	仕様
アナログ出力	4チャンネル、12ビット、シングルエンド±10V出力
アナログ入力	4チャンネル、12ビット ディファレンシャル±1V、±3V、±10V入力(スイッチにより切替) 入力抵抗10KΩ以上 0、1と2、3チャンネルはアナログ入力を同時サンプルする
レゾルバ励磁出力	2チャンネル、8KHz正弦波出力 シングルエンド±8～±16V(p-p)ボリュームにて連続可変 位相調整ボリューム(アナログ入力サンプリングタイミングに対して0～-90度調整可能)

ADC入力レンジ切り替えディップスイッチ

スイッチ	±1V	使用不可	±3V	±10V	
S1	OFF	OFF	ON	ON	CH0
S2	OFF	ON	OFF	ON	
S3	OFF	OFF	ON	ON	CH1
S4	OFF	ON	OFF	ON	
S5	OFF	OFF	ON	ON	CH2
S6	OFF	ON	OFF	ON	
S7	OFF	OFF	ON	ON	CH3
S8	OFF	ON	OFF	ON	

レゾルバ励磁出力調整ボリューム(上から1、2、3、4の順番です)

ボリューム	
VR1	レゾルバ出力チャンネル0振幅調整
VR2	レゾルバ出力チャンネル0位相調整
VR3	レゾルバ出力チャンネル1振幅調整
VR4	レゾルバ出力チャンネル1位相調整

コネクタピン番表

ピン番	記号	信号	内部回路	
1	FG	フレームグラウンド		
15	FG			
2	DA0	DAC出力 チャンネル0		
3	DA1	DAC出力 チャンネル1		
4	DA2	DAC出力 チャンネル2		
5	DA3	DAC出力 チャンネル3		
6	+AD0	ADC入力 チャンネル0		
7	-AD0			
8	+AD1	ADC入力 チャンネル1		
9	-AD1			
10	+AD2	ADC入力 チャンネル2		
11	-AD2			
12	+AD3	ADC入力 チャンネル3		
13	-AD3			
12	R0	レゾルバ励磁出力 チャンネル0		
13	R1	レゾルバ励磁出力 チャンネル1		
14	+12V	+12V電源出力	<p>+12V、-12Vとも100Ωの抵抗が入っています</p>	
28	-12V	-12V電源出力		

注意

- ・レゾルバ入力はADCチャンネル0、1のペアまたはチャンネル2、3のペアを使用します。このペアはアナログ信号を同時にサンプルします。
- ・ADC入力をシングルエンドで使用する場合+側に信号を接続し、-側はGNDと接続します。

レジスタの割り振り

入出力レジスタ	チャンネル	ボード
R80	ADCチャンネル0	1枚目
R81	ADCチャンネル1	
R82	ADCチャンネル2	
R83	ADCチャンネル3	
R84	DACチャンネル0	
R85	DACチャンネル1	
R86	DACチャンネル2	
R87	DACチャンネル3	
R88	ADCチャンネル0	2枚目
R89	ADCチャンネル1	
R90	ADCチャンネル2	
R91	ADCチャンネル3	
R92	DACチャンネル0	
R93	DACチャンネル1	
R94	DACチャンネル2	
R95	DACチャンネル3	
R96	ADCチャンネル0	3枚目
R97	ADCチャンネル1	
R98	ADCチャンネル2	
R99	ADCチャンネル3	
R100	DACチャンネル0	
R101	DACチャンネル1	
R102	DACチャンネル2	
R103	DACチャンネル3	
R104	ADCチャンネル0	4枚目
R105	ADCチャンネル1	
R106	ADCチャンネル2	
R107	ADCチャンネル3	
R108	DACチャンネル0	
R109	DACチャンネル1	
R110	DACチャンネル2	
R111	DACチャンネル3	