

9. 資料

9. 1 プログラム例

- (1) 1回目のプログラム実行時は原点復帰、2回目以降はポジション1、2へ移動
電源投入後の1回目の実行時は、原点復帰のみを行いプログラムを終了します。
2回目以降は、原点復帰を行わず、ポジション1、2へ移動します。

ステップ No.	命令	操作1	操作2	条件	結果	コメント
1	JUMP	1		F100		原点復帰済の場合、LB1へ
2	HOME	3			F100	原点復帰 完了後フラグ ON
3	END					
4	LB	1				1-----
5	VEL	100				速度設定
6	ACC	100				加減速設定
7	MVP	1				ポジション1へ移動
8	MVP	2				ポジション2へ移動
9	END					

- (2) 原点復帰未完了の場合、原点復帰後、ポジション1、2へ移動
原点復帰未完了の場合、原点復帰を行い、ポジション1、2へ移動します。
原点復帰済の場合、原点復帰を行わず、ポジション1、2へ移動します。

ステップ No.	命令	操作1	操作2	条件	結果	コメント
1	JUMP	1		F100		原点復帰済の場合、LB1へ
2	HOME	3			F100	原点復帰 完了後フラグ ON
3	LB	1				1-----
4	VEL	100				速度設定
5	ACC	100				加減速設定
6	MVP	1				ポジション1へ移動
7	MVP	2				ポジション2へ移動
8	END					

(3) 原点復帰（2軸同時）後、ポジション1～5に移動

原点復帰を行い、その後ポジション1から5を繰り返し動作します。

ステップ No.	命令	操作1	操作2	条件	結果	コメント
1	HOME	3				原点復帰
2	LB	1				
3	VEL	100				速度設定
4	ACC	100				加減速設定
5	MVP	1				ポジション1へ移動
6	MVP	2				ポジション2へ移動
7	MVP	3				ポジション3へ移動
8	MVP	4				ポジション4へ移動
9	MVP	5				ポジション5へ移動
10	JUMP	1				繰り返し

ステップ5～9は1ステップで設定することも可能です。

ステップ No.	命令	操作1	操作2	条件	結果	コメント
5	MVP	1	5			ポジション1～5へ移動

(4) 原点復帰（順序付き）後、ポジション1～5に移動

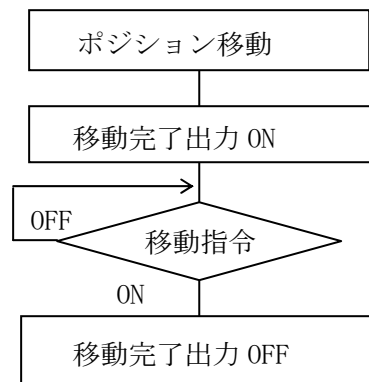
前述のプログラムの、原点復帰を1軸ずつ行なうように変更したプログラムです。

原点復帰を同時に行なうと干渉する場合などに使用します。

ステップ No.	命令	操作1	操作2	条件	結果	コメント
1	HOME	1				原点復帰 1軸目
2	HOME	2				原点復帰 2軸目
3	LB	1				
4	VEL	100				速度設定
5	ACC	100				加減速設定
6	MVP	1				ポジション1へ移動
7	MVP	2				ポジション2へ移動
8	MVP	3				ポジション3へ移動
9	MVP	4				ポジション4へ移動
10	MVP	5				ポジション5へ移動
11	JUMP	1				繰り返し

(5) 各ポジションで、外部との入出力を行なう

各ポジションへ移動後、移動完了をONします。
外部からの移動指令を受けて、移動完了をOFFし
次の移動を実行します。

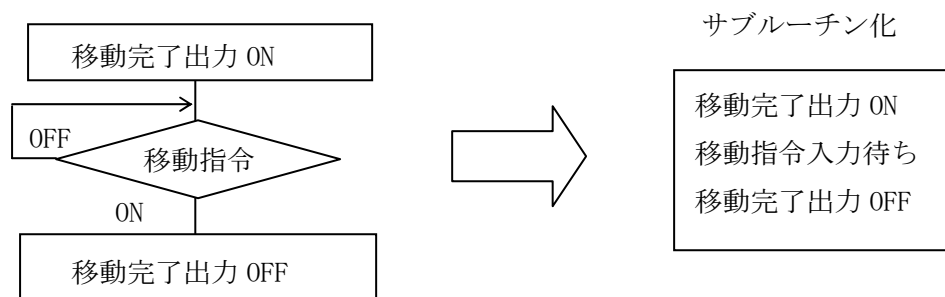


ステップ No.	命令	操作 1	操作 2	条件	結果	コメント
1	HOME	3				原点復帰
2	LB	1				
3	VEL	100				速度設定
4	ACC	100				加減速設定
5	MVP	1				ポジション 1 へ移動
6	ON	01				移動完了出力 ON
7	WTON	I1				移動指令入力待ち
8	OFF	01				移動完了出力 OFF
9	MVP	2				ポジション 2 へ移動
10	ON	01				移動完了出力 ON
11	WTON	I1				移動指令入力待ち
12	OFF	01				移動完了出力 OFF
13	MVP	3				ポジション 3 へ移動
14	ON	01				移動完了出力 ON
15	WTON	I1				移動指令入力待ち
16	OFF	01				移動完了出力 OFF
17	MVP	4				ポジション 4 へ移動
18	ON	01				移動完了出力 ON
19	WTON	I1				移動指令入力待ち
20	OFF	01				移動完了出力 OFF
21	MVP	5				ポジション 5 へ移動
22	ON	01				移動完了出力 ON
23	WTON	I1				移動指令入力待ち
24	OFF	01				移動完了出力 OFF
25	JUMP	1				繰り返し

(6) 各ポジションで外部との入出力を行なう (サブルーチン使用)

前述のプログラムでは、外部との入出力を行なうプログラムを、各ポジションで記述していますが、毎回同じプログラムを記述しています。

本例では、共通の部分をサブルーチン化し、ステップ数を削減したものです。

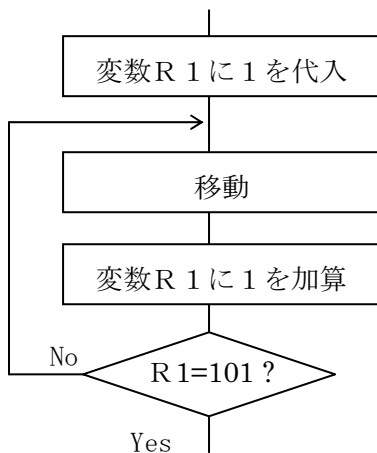


ステップ No.	命令	操作 1	操作 2	条件	結果	コメント
1	HOME	3				原点復帰
2	LB	1				
3	VEL	100				速度設定
4	ACC	100				加減速設定
5	MVP	1				ポジション 1 へ移動
6	CALL	1				サブルーチンコール
7	MVP	2				ポジション 2 へ移動
8	CALL	1				サブルーチンコール
9	MVP	3				ポジション 3 へ移動
10	CALL	1				サブルーチンコール
11	MVP	4				ポジション 4 へ移動
12	CALL	1				サブルーチンコール
13	MVP	5				ポジション 5 へ移動
14	CALL	1				サブルーチンコール
15	JUMP	1				繰り返し
16	SR	1				---サブルーチン ---
17	ON	01				移動完了出力 ON
18	WTON	I1				移動指令入力待ち
19	OFF	01				移動完了出力 OFF
20	SRET					---サブルーチン終了 ---

(7) ポジション1～100を連続で動作

ポジション1から100までを連続して動作させます。

移動命令を100ステップ使用するのではなく、変数を使用して、ポジション番号を+1ずつ加算して移動を行なうプログラムです。



ステップ No.	命令	操作1	操作2	条件	結果	コメント
1	HOME	3				原点復帰
2	VEL	100				速度設定
3	ACC	100				加減速設定
4	=	R1	1			変数 R1 に 1 を代入
5	LB	1				1-----
6	MVP	R1				変数 R1 のポジションに移動
7	+	R1	1			変数 R1+1
8	==	R1	101		F1	R1=101 なら F1 ON
9	JUMP	1		/F1		R1<101 なら JUMP
10	END					

(8) 外部からBCDでポジションを指定して動作

外部入力からBCD 2桁で、ポジション番号1～99を指定して動作させます。

BCD入力 : 1-I4 2-I5 4-I6 8-I7
10-I8 20-I9 40-I10 80-I11

移動開始入力 : I17

ステップ No.	命令	操作1	操作2	条件	結果	コメント
1	HOME	7				原点復帰
2	LB	1				
3	WTON	I17				移動開始指令 ON 待ち
4	INB	R1	I4			BCD データ読み取り
5	VEL	100				速度設定
6	ACC	100				加減速設定
7	MVA	R1				動作
8	WTOF	I17				移動開始指令 OFF 待ち
9	JUMP	1				

(9) 外部入力により、移動位置を変える

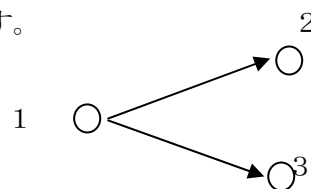
外部入力によって、移動するポジション番号を変えます。

移動開始入力 : I17

条件A入力 : I4 が ON でポジション 2 へ移動

条件B入力 : I5 が ON でポジション 3 へ移動

I4 と I5 はどちらかのみ ON します。



ステップ No.	命令	操作1	操作2	条件	結果	コメント
1	HOME	7				原点復帰
2	VEL	100				速度設定
3	ACC	100				加減速設定
4	LB	1				1-----
5	WTON	I17				移動開始指令 ON 待ち
6	MVA	2		I4		条件A入力? ポジション2へ
7	MVA	3		I5		条件B入力? ポジション3へ
8	WTOF	I17				移動開始指令 OFF 待ち
9	MVA	1				ポジション1へ戻る
10	JUMP	1				

(10) 外部入力により、JOG移動を行なう

外部入力によって、入力信号がONの間だけ動作を行ないます。

1軸目+JOG入力：I4 - JOG入力：I5

2軸目+JOG入力：I6 - JOG入力：I7

このプログラムでは、どれか1軸のみのJOG動作になります。

2軸同時に行ないたい場合は、(9)を参照ください。

ステップ No.	命令	操作1	操作2	条件	結果	コメント
1	HOME	3				原点復帰
2	VEL	50				速度設定
3	LB	1				1-----
4	JOG+	1	I4			1軸目+JOG動作
5	JOG-	1	I5			1軸目-JOG動作
6	JOG+	2	I6			2軸目+JOG動作
7	JOG-	2	I7			2軸目-JOG動作
8	JUMP	1				

(11) 外部入力により、JOG移動を行なう。

外部入力によって、入力信号がONの間だけ動作を行ないます。

1軸目+JOG入力：I4 - JOG入力：I5

2軸目+JOG入力：I6 - JOG入力：I7

このプログラムでは、2軸同時にJOGが行なえます。

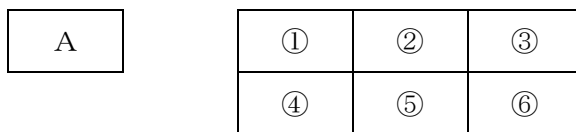
ステップ No.	命令	操作1	操作2	条件	結果	コメント
1	HOME	3				原点復帰
2	VEL	50				速度設定
3	STPG	2				プログラム2起動
4	LB	1				1-----
5	JOG+	1	I4			1軸目+JOG動作
6	JOG-	1	I5			1軸目-JOG動作
7	JUMP	1				

プログラム2

ステップ No.	命令	操作1	操作2	条件	結果	コメント
1	VEL	50				速度設定
2	LB	1				1-----
3	JOG+	2	I6			2軸目+JOG動作
4	JOG-	2	I7			2軸目-JOG動作
5	JUMP	1				

(12) パレタイズ動作を行なう

① → A → ② → A → ③ → A → ④ → A → ⑤ → A → ⑥ → A
 の順に動作を行います。



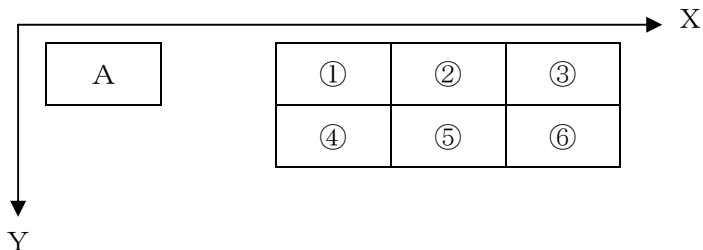
① : Pos. 1 ② : Pos. 2 ③ : Pos. 3 ④ : Pos. 4
 ⑤ : Pos. 5 ⑥ : Pos. 6 A : Pos. 10

ステップ No.	命令	操作1	操作2	条件	結果	コメント
1	HOME	3				原点復帰
2	VEL	100				速度設定
3	ACC	100				加減速設定
4	MVP	1				①へ移動
5	MVP	10				Aへ移動
6	MVP	2				②へ移動
7	MVP	10				Aへ移動
8	MVP	3				③へ移動
9	MVP	10				Aへ移動
10	MVP	4				④へ移動
11	MVP	10				Aへ移動
12	MVP	5				⑤へ移動
13	MVP	10				Aへ移動
14	MVP	6				⑥へ移動
15	MVP	10				Aへ移動
16	END					

ポジションNo. に変数を使用する方法もあります。

ステップ No.	命令	操作1	操作2	条件	結果	コメント
1	HOME	3				原点復帰
2	VEL	100				速度設定
3	ACC	100				加減速設定
4	=	R1	1			
5	LB	1				
6	MVP	R1				変数 R1 のポジションへ移動
7	MVP	10				Aへ移動
8	+	R1	1			変数 R1+1
9	>	R1	6		F1	R1>6 なら F1 ON
10	JUMP	1		/F1		R1<=6 なら JUMP
11	END					

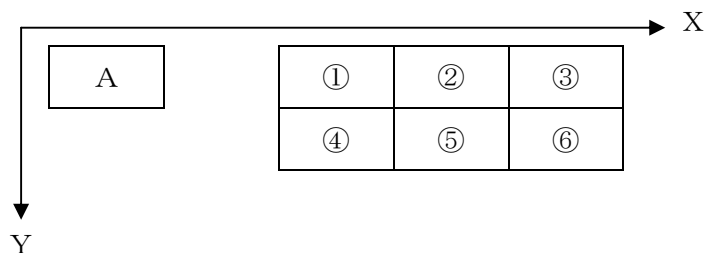
パレタイズが多い場合に、設定するポジション数が多くなります。
 次の例では、少ないポジション数の設定で同様の動作を行なうことができます。



ポジション① : P o s . 1
 A : P o s . 1 0

P o s . 1 0 0を一時的な位置データとして、パレットデータを
 代入して使用します。

ステップ No.	命令	操作1	操作2	条件	結果	コメント
1	HOME	3				原点復帰
2	VEL	100				速度設定
3	ACC	100				加減速設定
4	=	R1	2			Y軸移動回数セット
5	PGET	1	1			①のX軸データ→PR1
6	PGET	2	1			①のY軸データ→PR2
7	LB	1				1-----
8	=	R2	3			X軸移動回数セット
9	LB	2				2-----
10	PPUT	1	100			Pos. 100 X軸データ←PR1
11	PPUT	2	100			Pos. 100 Y軸データ←PR1
12	MVP	100				パレットへ移動
13	MVP	10				A点へ移動
14	+	PR1	10.000			X軸ピッチ+
15	-	R2	1		F1	X軸繰り返し回数-1
16	JUMP	2		/F1		繰り返し回数=0なら次へ
17	PGET	1	1			①のX軸データ→PR1
18	+	PR2	20.000			Y軸ピッチ+
19	-	R1	1		F2	Y軸繰り返し回数-1
20	JUMP	1		/F2		繰り返し回数=0なら次へ
21	END					



① : P o s . 1 A : P o s . 1 0

ポジション変数の値に直接移動することもできます。

ステップ No.	命令	操作 1	操作 2	条件	結果	コメント
1	HOME	3				原点復帰
2	VEL	100				速度設定
3	ACC	100				加減速設定
4	=	R1	2			Y 軸移動回数セット
5	PGET	1	1			①の X 軸データ→PR1
6	PGET	2	1			①の Y 軸データ→PR2
7	LB	1				1-----
8	=	R2	3			X 軸移動回数セット
9	LB	2				2-----
10	MVD	3				パレットへ移動
11	MVP	10				A 点へ移動
12	+	PR1	10.000			X 軸ピッチ+
13	-	R2	1		F1	X 軸繰り返し回数-1
14	JUMP	2		/F1		繰り返し回数=0 なら次へ
15	PGET	1	1			①の X 軸データ→PR1
16	+	PR1	20.000			Y 軸ピッチ+
17	-	R1	1		F2	Y 軸繰り返し回数-1
18	JUMP	1		/F2		繰り返し回数=0 なら次へ
19	END					

(13) 各ポジションで外部出力を行う（他プログラム起動を使用）

ポジション1、ポジション2に移動後、出力1をON・OFFします。

共通の部分を別プログラムで実行し、ステップ数を削減したものです。

PRG. 1

ステップ No.	命令	操作1	操作2	条件	結果	コメント
1	HOME	7				原点復帰
2	LB	1				1-----
3	MVP	1				ポジション1へ移動
4	ON	F1				起動フラグ ON
5	STPG	2				PRG.2を起動
6	WTOF	F1				起動フラグ OFF 待ち
7	MVP	2				ポジション2へ移動
8	ON	F1				起動フラグ ON
9	STPG	2				PRG.2を起動
10	WTOF	F1				起動フラグ OFF 待ち
11	JUMP	1				

PRG. 2

ステップ No.	命令	操作1	操作2	条件	結果	コメント
1	ON	01				Out 1 を ON
2	TIM	1.000				タイマー 1 秒
3	OFF	01				Out 1 を OFF
4	OFF	F1				起動フラグを OFF
5	END					

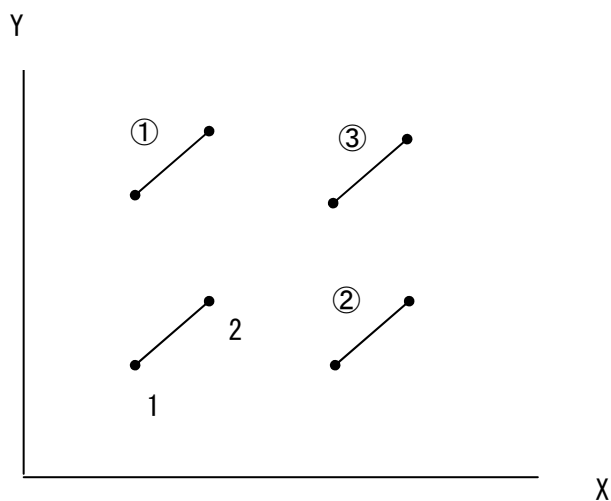
(14) オフセットを利用した移動

ポジションNo. はそのまま、位置をオフセットして移動させることができ、
 ティーチングする点数を減らすことができます。

ステップ No.	命令	操作1	操作2	条件	結果	コメント
1	HOME	7				原点復帰
2	LB	1				1-----
3	MVP	1	2			ポジション1~2へ移動
4	OFST	4				オフセット設定 Y軸+20mm
5	MVP	1	2			ポジション1~2へ移動 ①
6	OFST	5				オフセット設定 X軸+20mm
7	MVP	1	2			ポジション1~2へ移動 ②
8	OFST	6				オフセット設定 X・Y軸+20mm
9	MVP	1	2			ポジション1~2へ移動 ③
10	OFST	3				オフセット設定 解除
11	JUMP	1				

ポジションデータ

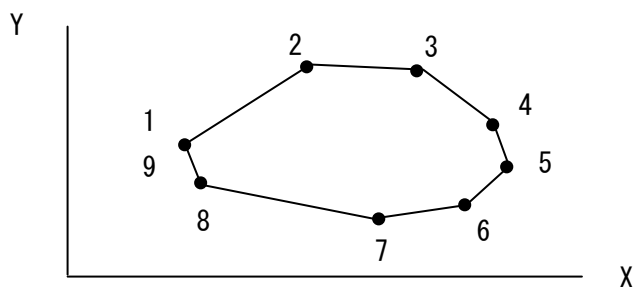
POS No.	X軸	Y軸	Z軸	S軸	コメント
1	10 mm	10 mm	0 mm	0 mm	動作位置1
2	15 mm	15 mm	0 mm	0 mm	動作位置2
3	0 mm	0 mm	0 mm	0 mm	
4	0 mm	20 mm	0 mm	0 mm	Y軸オフセット値
5	20 mm	0 mm	0 mm	0 mm	X軸オフセット値
6	20 mm	20 mm	0 mm	0 mm	X・Y軸オフセット値



(15) パス移動

Pos. 1からPos. 9まで各ポジションで停止せずに連続動作します。

ステップ No.	命令	操作1	操作2	条件	結果	コメント
1	HOME	7				原点復帰
2	VEL	20				速度設定
3	PATH	1	9			1～9でパス移動
4	END					



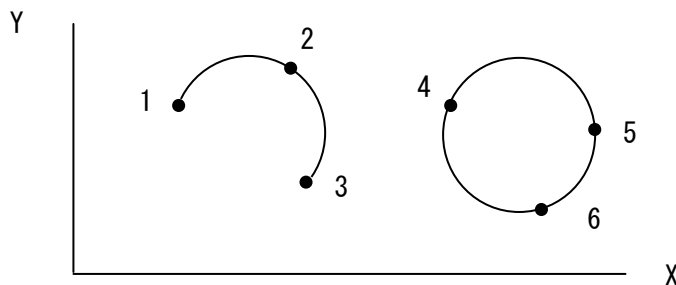
Pos. 1とPos. 9は、同じ位置です

(16) 円弧・円移動

円弧：Pos. 1からPos. 2を通り、Pos. 3までの円弧

円：Pos. 4からPos. 5、6を通り、Pos. 4までの円

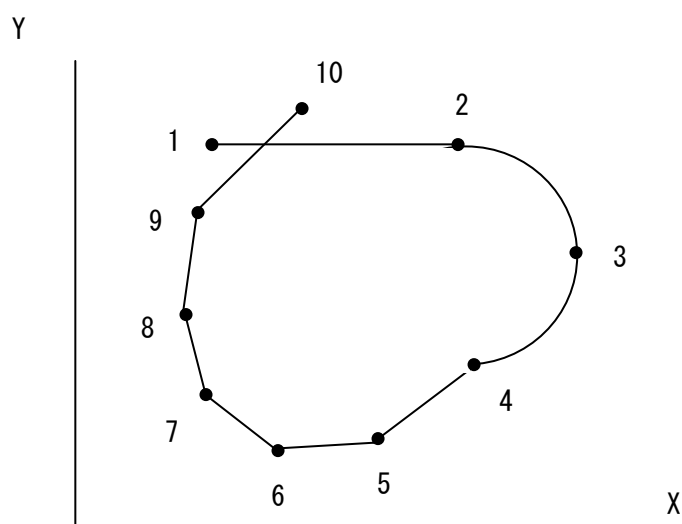
ステップ No.	命令	操作1	操作2	条件	結果	コメント
1	HOME	7				原点復帰
2	VEL	20				速度設定
3	ARAX	1	2			円弧軸指定 X・Y軸
4	MVP	1				Pos. 1へ移動
5	ARC	2	3			円弧移動
6	MVP	4				Pos. 4へ移動
7	CIR	5	6			円移動
8	END					



(17) パス移動と円弧移動の組み合わせ

パス移動と円弧移動をステップ間で停止せずに連続動作します。

ステップ No.	命令	操作1	操作2	条件	結果	コメント
1	HOME	7				原点復帰
2	VEL	20				速度設定
3	AXAR	1	2			円弧軸指定 X・Y 軸
4	PATH	1	2			パス移動
5	ARC	3	4			円弧移動
6	PATH	5	10			パス移動
7	END					



■ ■ 9. 2 プログラムシート ■ ■

ステップ No.	命令	操作1	操作2	条件	結果	コメント
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						
38						
39						

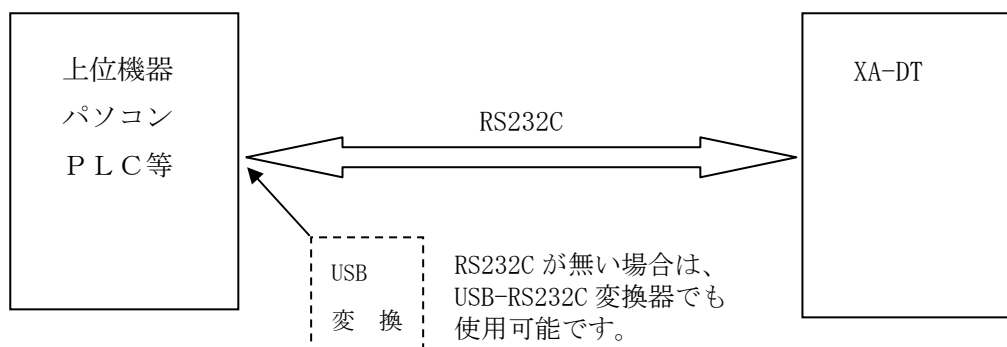
■ ■ 9. 3 ポジションデータシート ■ ■

PoS No.	X軸位置	Y軸位置	Z軸位置	S軸位置	コメント
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					
29					
30					
31					
32					
33					
34					
35					
36					
37					
38					
39					

■ ■ 9. 4 通信プロトコル資料 ■ ■

9. 4. 1 概要

通信プロトコルは、RS232C通信を使用し上位機器とコントローラとの通信を行なうためのフォーマットです。以下のような構成でパソコンもしくはPLC等と、RS232Cを介して通信を行い、ダイレクトにコントローラに指令を送り動作させることが可能です。



(1) RS232Cの設定

上位機器のRS232C設定は、下表のとおりを設定してください。

設定値があっていない場合は、通信が正常に行なわれません。

項目	設定値
ボーレート	38400
データ長	8
ストップビット	1
パリティ	なし

(2) 用語の定義

コマンド : 上位機器から、コントローラに対して送られるデータ

アンサー : コマンドを受けたコントローラが、上位機器に送るデータ

CR (キャリッジリターン) : ASCII 0Dh

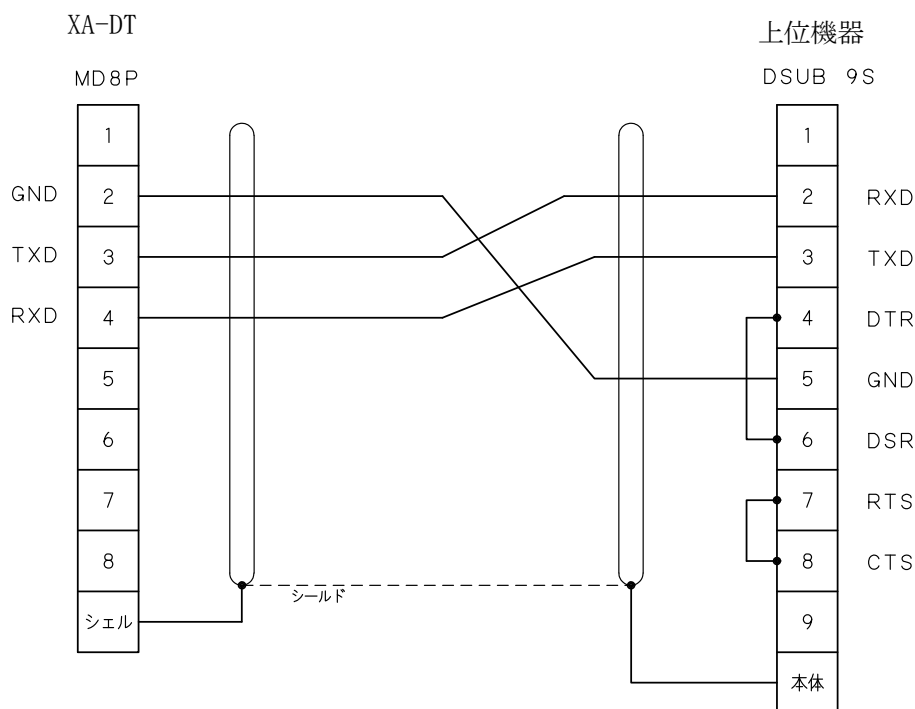
LF (ラインフィード) : ASCII 0Ah

(3) 通信用ケーブル

通信ケーブルは、オプションにてご用意しております。

型式：PC232-8-CAB ケーブル長2m

お客様にてケーブルを製作される場合は、下図によって製作してください。
また、ノイズ等のない環境での使用で、ケーブル長は最大10mまでとしてください。
環境により、ケーブルが長いと正常に動作出来ない場合があります。



(4) USB-RS232C変換器

上位機器にRS232Cポートを有していない場合は、USBポートを使用して、RS232Cに変換することが可能です。

変換機は、オプションにてご用意しております。

型式：USB-RS232C (サンワサプライ株)

9. 4. 2 コマンド一覧

コマンドは次の通りで、コマンドの最終データはCR・LFです。
通信からの命令でエラーが発生したときはエラーコードで応答します。

	コマンド	内容	送信バイト数	受信バイト数
1	OMP	ポイント移動	9	5
2	OMI	ポイント補間移動	9	5
3	OMV	ファイル移動	50	5
4	OJR	JOG早送り	10	5
5	OSP	ストップ	5	5
6	ORA	移動完了確認	5	6
7	ORH	原点復帰完了確認	5	6
8	ORY	入力状態読出	5	12
9	ORB	出力状態読出	5	12
10	OWB	出力状態変更	12	5
11	ORP	ポジションデータ読出	8	68
12	OWP	ポジションデータ書込	68	8
13	OWA	ポジションデータEEPROM書込	11	11
14	ORC	現在値読出	6	26
15	OWC	位置更新	9	5
16	ORV	バージョン照会	5	11
17	OAR	アラームリセット	5	5

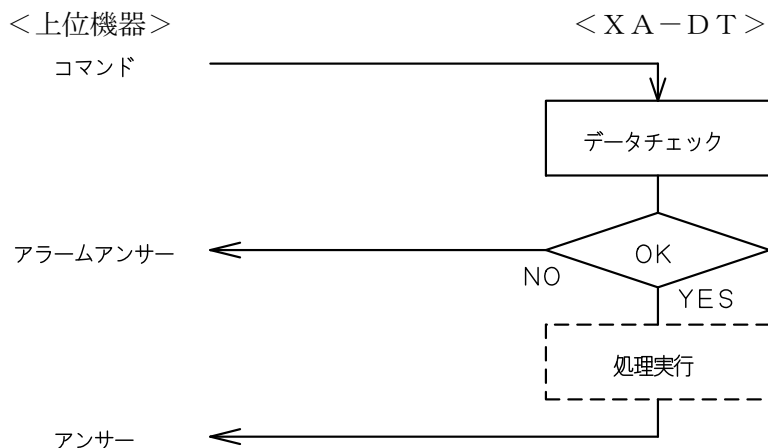
各コマンドの先頭の文字は“ゼロ”です。

送信・受信のバイト数は、CR・LFも含まれます。

各コマンドの詳細は 別冊の「[プロトコル説明書](#)」をご覧ください。
(HPからダウンロードできます。)

9. 4. 3 通信の手順

通信は、上位機器（パソコン等）から、XA-DTへコマンドを送信し、その返信をアンサーとして上位機器へ送ります。



【通信例】 ORV : バージョン照会 を行います。

以下のコマンドをコントローラに送信します。

【 コマンド 】

1	2	3	4	5
0	R	V	C R	L F

通信が正常に行われれば、以下のアンサーが返信されます。

【 アンサー 】

1	2	3	4	5	6	7	8	9	10	11
0	R	V	v	e	r	D	T	3	C R	L F

(注) v e r にはバージョン番号が入ります。

8文字目は ロボットの軸数によって入る文字が変わります。

4 : 4軸 3 : 3軸 2 : 2軸 1 : 1軸



アンサーが正しく返信されない場合、次のような問題がある可能性があります。

- ①上位機器の、RS232Cの設定値をご確認ください。
- ②コマンドをご確認ください。1文字目が0”ゼロ“になっていますか。

改版履歴

版	年月日	内 容	変更ページ
1.0	'08/10/07	初版 新規作成	
1.1	'12/10/1	外形図訂正	3-3、3-4
1.2	'15/5/27	外形図訂正	3-3、3-4
1.3	'17/8/24	取扱説明書 CD の付属を廃止。	1-1