

# *MOS*

ランゲージ・リファレンス



1.0.1.1 版 2007/04/25

Copyright © 2003-2007



## 変更履歴

Rev	日付	担当	内容
1.0.0.0	'03.03.29	片桐	新規作成
1.0.0.1	'04.01.10	片桐	RobSetAiStopMove 等 AD 連携コマンド追加
1.0.0.2	'04.10.28	片桐	使用例追加
1.0.0.3 3	'04.11.20	松田	3.72 項 ~ 3.82 項追加、一部使用例追加、4 項一部変更
1.0.0.4 4	'04.12.10	松田	goto 文追加 (2.4.4 項)
1.0.0.5 5	'04.12.20	松田	continue 文、break 文追加 (2.4.4 項)
1.0.0.6 6	'05.01.04	松田	プロセス制御関数 (3.83 項 ~ 3.88 項) 追加、4 項変更 3.56 項、3.57 項変更
1.0.0.7 7	'05.04.06	松田	ポート連動モーション / オーバラップ移動サポート関数 (3.89 項 ~ 3.94 項) 追加、4 項変更
1.0.0.8 8	'05.04.11	松田	2.4.4 項の if 文の説明一部変更 3.93 項、3.94 項の説明一部変更
1.0.0.9 9	'05.08.30	松田	2.4.4 項に for 文を追加、if 文、while 文、continue 文、break 文の説明一部変更、goto 文の誤記訂正。 3.20 項、3.21 項、3.25 項 ~ 3.27 項、3.37 項、3.39 項、3.44 項、 3.46 項、3.48 項、3.50 項 ~ 3.53 項、3.55 項、3.65 項、3.83 項 ~ 3.88 項の説明一部変更、 4 項にて RobResetAiStopMove と RobSetAiStopMove をアナ ログ入力連動モーションサポート関数に分類 LockMutex 関数、UnlockMutex 関数削除
1.0.0.10 10	'06.01.16	松田	3.95 項 ~ 3.112 項追加 (関数追加)、RobPtpPulseMove 関数 削除、4 章変更、関連する関数の説明変更、誤記訂正
1.0.0.11 11	'06.01.24	松田	3.113 項 ~ 3.115 項追加 (関数追加)、Printf1 関数、Printf2 関 数の説明変更
1.0.1.0 12	'06.10.13	松田	2.2.10 項追加 (構造体サポート)、3.116 項 ~ 3.125 項追加 (関 数追加)、3.37 項、3.67 項、3.107 項説明変更
1.0.1.1 13	'07.04.25	松田	3.72 ~ 3.76 項の説明変更 . 3.126 項、3.127 項追加 (関数追加)

## 目次

変更履歴 .....	2
ご注意 .....	3
1. はじめに .....	9
1.1. テクニカルサポート .....	9
2. MOS言語仕様 .....	10
2.1. 概要 .....	10
2.2. 構成要素 .....	10
2.2.1. 文字・記号 .....	10
2.2.2. トークンと空白文字 .....	10
2.2.3. コメント .....	10
2.2.4. 識別子 .....	11
2.2.5. キーワード .....	11
2.2.6. 変数と変数宣言、データ型 .....	11
2.2.7. 配列変数 .....	12
2.2.8. 定数 .....	12
2.2.9. 演算子 .....	13
2.2.10. 構造体 .....	14
2.3. 式 .....	17
2.4. 文・ブロック .....	18
2.4.1. 定義文 .....	18
2.4.2. 代入文 .....	20
2.4.3. 関数呼び出し文 .....	21
2.4.4. プログラム制御文 .....	22
2.5. プログラムの構造 .....	27
2.5.1. ソース ファイルとソース プログラム .....	27
2.5.2. メイン関数とプログラムの実行 .....	27
3. システム関数 .....	28
3.1. abs .....	28
3.2. acos .....	29
3.3. atan .....	30
3.4. atan2 .....	31
3.5. ceil .....	32
3.6. cos .....	33
3.7. fabs .....	34

---

3.8.	floor .....	35
3.9.	log .....	36
3.10.	log10 .....	37
3.11.	pow .....	38
3.12.	sin .....	39
3.13.	sqrt .....	40
3.14.	tan .....	41
3.15.	RadToDeg.....	42
3.16.	DegToRad.....	43
3.17.	pi .....	44
3.18.	ReadSharedVariable .....	45
3.19.	WriteSharedVariable .....	46
3.20.	StartTimer .....	47
3.21.	GetTimerValue .....	48
3.22.	LockMutex.....	49
3.23.	UnlockMutex.....	50
3.24.	RobPtpMove .....	51
3.25.	RobLinearMove.....	53
3.26.	RobArcMove .....	55
3.27.	RobCircularMove .....	57
3.28.	RobPtpJointMove .....	59
3.29.	RobPtpPulseMove .....	60
3.30.	RobReturnHome.....	61
3.31.	RobGetCarteGoal.....	62
3.32.	RobGetCartePosition .....	63
3.33.	RobGetJointGoal .....	64
3.34.	RobGetJointPosition .....	65
3.35.	RobGetPulseGoal .....	66
3.36.	RobGetPulsePosition .....	67
3.37.	RobGetMoveStatus.....	68
3.38.	RobStopMove .....	69
3.39.	RobWaitForMoveDone .....	70
3.40.	RobSetArm .....	71
3.41.	RobSetElbow .....	72
3.42.	RobSetWrist.....	73
3.43.	RobSetPtpSpeed.....	74
3.44.	RobSetPtpAccelerations .....	75
3.45.	RobSetCpSpeed .....	76

---

---

3.46.	RobSetCpAccelerations .....	77
3.47.	RobSetSettle .....	78
3.48.	RobSetSpeedOverride .....	79
3.49.	RobSetServoPower .....	80
3.50.	ReadPort .....	81
3.51.	ReadPorts .....	82
3.52.	WritePort .....	83
3.53.	WritePorts .....	84
3.54.	Sleep.....	85
3.55.	FireEvent .....	86
3.56.	Cycle .....	87
3.57.	Pause.....	88
3.58.	Stop.....	89
3.59.	Printf0.....	90
3.60.	Printf1.....	91
3.61.	Printf2.....	92
3.62.	strcpy .....	93
3.63.	strcat .....	94
3.64.	strcmp.....	95
3.65.	RobSetAiStopMove.....	96
3.66.	RobResetAiStopMove.....	97
3.67.	RobSetManipServoPower .....	98
3.68.	RobGetManipServoPowerStatus .....	99
3.69.	RobGetServoPowerStatus .....	100
3.70.	RobGetReturnHomeStatus .....	101
3.71.	ReadAi .....	102
3.72.	CommOpen 3 .....	103
3.73.	CommClose 3 .....	104
3.74.	CommOutput 3 .....	105
3.75.	CommInput 3 .....	106
3.76.	CommOutputString 3.....	107
3.77.	strlen 3 .....	108
3.78.	Sprintf1 3 .....	109
3.79.	Sprintf2 3 .....	110
3.80.	atoi 3.....	111
3.81.	atol 3.....	112
3.82.	atof 3.....	113
3.83.	ProcStartInitial 6 .....	114

---

3.84.	ProcStartResume	6	115
3.85.	ProcStopStep	6	116
3.86.	ProcStopPause	6	117
3.87.	ProcStopCycle	6	118
3.88.	QueryProcStatus	6	119
3.89.	RobSetMoveProgressDo	7	120
3.90.	RobResetMoveProgressDo	7	122
3.91.	RobSetOverlap	7	123
3.92.	RobResetOverlap	7	128
3.93.	RobSetDiStartMove	7	129
3.94.	RobResetDiStartMove	7	131
3.95.	RobEllipseMove	10	132
3.96.	RobEllipseMove2	10	134
3.97.	RobStopEllipseMove	10	136
3.98.	ProcCancelStopStep	10	137
3.99.	ProcCancelStopPause	10	138
3.100.	ProcCancelStopCycle	10	139
3.101.	TestSetSharedVariable	10	140
3.102.	ReadAo	10	142
3.103.	WriteAo	10	143
3.104.	QueryProcRuntimeError	10	144
3.105.	ProcResetRuntimeError	10	146
3.106.	RobSetOverTravelResetMode	10	147
3.107.	RobSetManipServoPower2	10	148
3.108.	RobReturnHomeStart	10	150
3.109.	RobReturnHomeComplete	10	152
3.110.	RobResetError	10	153
3.111.	RobGetServoOffPortStatus	10	154
3.112.	EnablePrintf	10	155
3.113.	RobCheckErrorExist	11	156
3.114.	MwCheckFatalErrorExist	11	157
3.115.	MwCheckSystemErrorExist	11	158
3.116.	GetProcNumber	12	159
3.117.	RobGetHomeSensorStatus	12	160
3.118.	RobGetTravelLimitSensorStatus	12	161
3.119.	RobJogMove	12	162
3.120.	RobJogJointMove	12	165
3.121.	RobGetTravelLimit	12	167

3.122.	RobGetMaxVel	12	.....	168
3.123.	RobGetScale	12	.....	169
3.124.	SetAutomaticSleep	12	.....	170
3.125.	RobGetReturnHomeSequence	12	.....	171
3.126.	CommSetBreak	13	.....	172
3.127.	CommClearBreak	13	.....	173
4.	システム関数一覧		.....	174

### 3.24. RobPtpMove

```
short RobPtpMove(  
    short idx,  
    double Goal[MaxRobAxes],  
    long MoveType  
);
```

#### 戻り値

0 が返却されます。

#### 引数

*idx*

ロボットを指定するインデックス

*Goal*

動作目標位置を XY 座標系で指定します。

配列のサイズは、MaxRobAxes で定義されている必要があります。

*MoveType*

動作の種類。以下の値が有効です。

- 1: 絶対位置指定、同期
- 2: 相対位置指定、同期
- 3: 絶対位置指定、非同期
- 4: 相対位置指定、非同期

#### 解説

PTP 動作にて指定位置へ移動します。

ロボットが複数軸で構成されている場合、移動開始タイミング、および停止タイミングは軸間で同期動作となります。

#### 備考

目標位置を設定する double 型の配列は配列要素[0]から順番に使用されます。例えば、2軸ロボットであれば1軸目の目標位置は配列要素[0]に、2軸目の目標位置は配列要素[1]に設定しておきます。(1軸目、2軸目、…の定義については、別途「MOTIWAREConfig.ini Reference」の Axis パラメタを参照してください。)

加減速時間の設定は RobSetPtpAccelerations で行います。移動速度の設定は RobSetPtpSpeed で行います。

目標位置の単位は[ユーザ単位]です。[ユーザ単位]はコンフィギュレーションファイル(MotiWareConfig.ini)の Scale パラメタで定義します。例えば Scale の単位を[パルス/mm]で定義してあったとして、RobPtpMove の相対目標位置を +30 と設定して発行すれば、現在の位置から +30mm の位置へ移動します。詳細については、別途「MOTIWAREConfig.ini Reference」を参照してください。



**使用例**

次のプログラムは、2軸で構成された「ロボット1」を2次元平面上の座標(0, 0)と(-100, -55)との間で無限回数移動する例です。

```
void main() {
    ....
    // 目標位置を格納する配列の要素は必ず MaxRobAxes としてください
    // MaxRobAxes の値はシステム内部で定義されており、ユーザが変更することはできません。
    double pos[MaxRobAxes]; // 目標位置を格納する配列を宣言
    ....
    RobSetPtpAccelerations(1, 50, 50, 0, 0); // 「ロボット1」加減速時間設定
    RobSetPtpSpeed(1, 100); // 「ロボット1」移動速度設定
    RobSetManipServoPower(1, 1); // 「ロボット1」全軸サーボオン
    RobReturnHome(1); // 「ロボット1」原点復帰
    while(1) {
        pos[0] = -100; // 1軸目標位置を設定
        pos[1] = -55; // 2軸目標位置を設定
        RobPtpMove(1, pos, 1); // 絶対位置(-100, -55)へ移動
        Sleep(500); // 500ms 処理遅延
        pos[0] = 0;
        pos[1] = 0;
        RobPtpMove(1, pos, 1); // 絶対位置(0, 0)へ移動
        Sleep(500); // 500ms 処理遅延
        .....
    }
    .....
}
```

**参照**

RobSetPtpAccelerations, RobSetPtpSpeed, RobLinearMove

### 3.55. FireEvent

```
short FireEvent(  
    long val  
);
```

#### 戻り値

VisualBasic などからの返却値が返却されます。

#### 引数

*val*

VisualBasic などへ通知する値を指定します。

#### 解説

VisualBasic などへイベントを通知し、処理依頼をするのに使用します。

本関数を使用することで、VisualBasic などのアプリケーションで事象待ち受けをポーリングにて監視する必要がなくなります。

#### 使用例

次の例では、RobPtpMove で指定位置へ移動後、VisualBasic イベントを通知します。そして FireEvent の戻り値を参照して良品と不良品の判定をしています。(例えばポート出力に良/不良の表示等が可能です。)

```
void main() {  
    ....  
    // 測定位置へ移動  
    RobPtpMove( 0, Pos1, 1);  
    rc = FireEvent( 1/*測定開始*/);  
    if( rc == 0 ) {  
        // 測定結果:良品  
    }  
    else {  
        // 測定結果:不良  
    }  
}
```

#### メモ

アプリケーション側では、MwController コンポーネントの ApplEvent イベントで本関数による通知を受け取ることができます。例えば VisualBasic では、ApplEvent イベントプロシージャに通知に対する処理を記述してイベントをハンドリングします。ApplEvent イベントには EventID というパラメータがあり、ここに本関数で指定した *val* が渡されてきますので、アプリケーション側では *val* に応じた処理を行うことができます。 <sup>9</sup>

#### 参照

## 4. システム関数一覧

### ロボット制御関数(1) <sup>10</sup>

関数名	機能
<b>直交座標系</b>	
RobPtpMove	指定ロボットをPTP動作にて指定位置へ移動(絶対位置、相対位置)する
RobLinearMove	指定ロボットを直線補間動作にて指定された目標位置(直交座標系)へ直線補間で移動(絶対位置、相対位置)する
RobArcMove	指定ロボットを円弧補間動作にて指定された目標位置(直交座標系)へ、指定された中継点を經由してで移動(絶対位置、相対位置)する
RobCircularMove	指定ロボットを現在位置からスタートし、中継点1、2(直交座標系)を經由して再び現在位置まで真円移動(絶対位置、相対位置)する
RobEllipseMove <sup>10</sup>	指定ロボットを指定された中心点と半径および周期で指定回数分の連続楕円移動を行う
RobEllipseMove2 <sup>10</sup>	指定ロボットを指定された中心点と半径および周期で指定回数分の連続8の字移動を行う
RobGetCarteGoal	指定の配列に、指定ロボットの最終目標位置を取得する
RobGetCartePosition	指定の配列に、指定ロボットの現在位置を取得する
RobJogMove <sup>12</sup>	指定ロボットの JOG パルス移動を許可する
<b>論理軸座標系</b>	
RobPtpJointMove	指定ロボットをPTP動作にて指定位置へ移動(絶対位置、相対位置)する
RobGetJointGoal	指定の配列に、指定ロボットの最終目標位置を取得する
RobGetJointPosition	指定の配列に、指定ロボットの現在位置を取得する
RobJogJointMove <sup>12</sup>	指定ロボットの JOG パルス移動を許可する
<b>動作パラメタ設定</b>	
RobSetArm	指定ロボットの左腕系右腕系形態を設定
RobSetElbow	指定ロボットのひじ形態(上・下)を設定
RobSetWrist	指定ロボットの手首形態(フリップ・ノンフリップ)を設定
RobSetPtpSpeed	指定ロボットの、PTP動作の移動速度を設定
RobSetPtpAccelerations	指定ロボットの、PTP動作の加減速時間・Sカーブ区間を設定
RobSetCpSpeed	指定ロボットの、補間動作の移動速度を設定
RobSetCpAccelerations	指定ロボットの、補間動作の加減速時間・Sカーブ区間を設定
RobSetSettle	指定ロボットの位置決め精度を設定
<b>動作状態の取得</b>	
RobGetPulseGoal	指定ロボットの、論理パルスの最終目標位置を取得する
RobGetPulsePosition	指定ロボットの、論理パルスの現在位置を取得する
RobGetMoveStatus	指定ロボットの動作残数を取得する
RobGetHomeSensorStatus <sup>12</sup>	指定ロボットの原点センサ状態を取得する
RobGetTravelLimitSensorStatus <sup>12</sup>	指定ロボットのオーバートラベルセンサ状態を取得する
RobGetReturnHomeSequence <sup>12</sup>	指定ロボット原点復帰シーケンス値を取得する

ロボット制御関数(2) <sup>10</sup>

関数名	機能
原点復帰関係	
RobReturnHome	指定ロボットの原点復帰動作を行う
RobGetReturnHomeStatus	指定ロボットの原点復帰完了状態を取得する
RobSetOverTravelResetMode <sup>10</sup>	指定ロボットのオーバトラベルリセットモードの設定 / 解除
RobReturnHomeStart <sup>10</sup>	指定ロボットの原点復帰動作の開始 (非同期動作)
RobReturnHomeComplete <sup>10</sup>	指定ロボットの原点復帰動作の完了処理
サーボ電源制御	
RobSetServoPower	指定ロボットの各軸のサーボON、またはサーボOFFを設定する
RobGetServoPowerStatus	指定ロボットの各軸のサーボ状態 (ONまたはOFF) を取得する
RobSetManipServoPower	指定ロボットの全軸のサーボON、またはサーボOFFを設定する
RobSetManipServoPower2 <sup>10</sup>	指定ロボットの全軸の即時サーボON、またはサーボOFFを設定する
RobGetManipServoPowerStatus	指定ロボットの全軸がサーボON状態か調査する
RobGetServoOffPortStatus <sup>10</sup>	指定ロボットのサーボOFFポートの状態を取得する
動作制御	
RobStopMove	指定ロボットの動作を中止し、即時停止する
RobStopEllipseMove <sup>10</sup>	指定ロボットの連続楕円移動および連続8の字移動動作を中止し、移動開始点に戻って減速停止する
RobWaitForMoveDone	指定ロボットの移動が指定された動作残数以下になるまで動作の完了を待つ
RobSetSpeedOverride	移動動作中の指定ロボットの、速度オーバーライド値を設定する
その他 <sup>10</sup>	
RobResetError <sup>10</sup>	指定ロボットのロボットエラーをリセット
RobCheckErrorExist <sup>11</sup>	指定ロボットのロボットエラーの発生状態を調べる
MwCheckFatalErrorExist <sup>11</sup>	MOTIWARE の致命的エラーの発生状態を調べる
MwCheckSystemErrorExist <sup>11</sup>	MOTIWARE のシステムエラーの発生状態を調べる
RobGetTravelLimit <sup>12</sup>	指定ロボットの TravelLimitU, TravelLimitL パラメータの値をコンフィギュレーションファイルから取得する
RobGetMaxVel <sup>12</sup>	指定ロボットの MaxVel パラメータの値をコンフィギュレーションファイルから取得する
RobGetScale <sup>12</sup>	指定ロボットの Scale パラメータの値をコンフィギュレーションファイルから取得する

## ポート制御関数

関数名	機能
ReadPort	指定したポートの値を取得
ReadPorts <sup>10</sup>	指定した複数の連続したポートの値を取得
WritePort	指定したポートに値を設定
WritePorts <sup>10</sup>	指定した複数の連続したポートに値を設定
ReadAi	指定したアナログ入力チャンネルの値を取得
ReadAo <sup>10</sup>	指定したアナログ出力チャンネルの出力設定値を取得
WriteAo <sup>10</sup>	指定したアナログ出力チャンネルに値を設定

## 数学処理関数

関数名	機能
abs	整数の絶対値を計算
acos	アークコサイン(逆余弦)を計算
atan	アークタンジェント(逆正接)を計算
atan2	アークタンジェント(逆正接)を計算(範囲 - ~ )
sin	サイン(正弦)を計算
cos	コサイン(余弦)を計算
tan	タンジェント(正接)を計算
fabs	浮動小数点の絶対値を計算
ceil	天井関数を計算
floor	床関数計算(小数点以下を切り捨て)
log	自然対数を計算
log10	常用対数を計算
pow	べき乗を計算
sqrt	平方根を計算
RadToDeg	ラジアン °(度)変換
DegToRad	°(度) ラジアン変換
pi	(円周率)を返す

## 文字列処理関数

関数名	機能
strcpy	文字列の複写
strcat	文字列の連結
strcmp	文字列の比較
strlen <sup>3</sup>	文字列の文字数取得

## コンソール・ログ出力処理関数

関数名	機能
Printf0	標準出力への文字列出力
Printf1	標準出力への文字列出力(書式付き)
Printf2	標準出力への文字列出力(書式付き)
EnablePrintf <sup>10</sup>	標準出力への文字列出力抑制制御

## プログラム制御関数

関数名	機能
Sleep	指定された時間だけ処理を遅延する(プロセス停止)
FireEvent	VisualBasicなどイベントを通知する
Cycle	プログラムのサイクル停止(デバッグで使用)
Pause	プログラムのポーズ停止(デバッグで使用)
Stop	プログラムのストップ停止
SetAutomaticSleep <sup>12</sup>	指定された行数実行後 1us 間 Sleep する(リアルタイムインタプリタにおいて)

## タイマ制御関数

関数名	機能
StartTimer	指定のタイマを起動する
GetTimerValue	指定のタイマの値を取得する

## プロセス間通信関数

関数名	機能
ReadSharedVariable	指定の共有変数の値を取得する
WriteSharedVariable	指定の共有変数の値を更新する
<sup>9</sup>	
<sup>9</sup>	
TestSetSharedVariable <sup>10</sup>	指定の共有変数の値をテスト&セット

シリアル通信ポート入出力処理関数 <sup>3</sup>

関数名	機能
CommOpen	シリアル通信ポートのオープン
CommClose	シリアル通信ポートのクローズ
CommOutput	シリアル通信ポートからデータ送信
CommInput	シリアル通信ポートでデータ受信
CommOutputString	シリアル通信ポートから文字列を送信
CommSetBreak <sup>13</sup>	シリアル通信ポートからブレイク信号送出開始
CommClearBreak <sup>13</sup>	シリアル通信ポートからブレイク信号送出停止

ストリーム出力処理関数 <sup>3</sup>

関数名	機能
Sprintf1	書式付きデータを文字列へ出力
Sprintf2	書式付きデータを文字列へ出力

文字列変換処理関数 <sup>3</sup>

関数名	機能
atoi	文字列を int に変換
atol	文字列を long に変換
atof	文字列を double に変換

プロセス制御関数 <sup>6</sup>

関数名	機能
ProcStartInitial	指定プロセスを初期起動
ProcStartResume	指定プロセスを継続起動
ProcStopStep	指定プロセスをステップ停止
ProcStopPause	指定プロセスをポーズ停止
ProcStopCycle	指定プロセスをサイクル停止
QueryProcStatus	指定プロセスの状態を取得
ProcCancelStopStep <sup>10</sup>	指定プロセスのステップ停止指令をキャンセル
ProcCancelStopPause <sup>10</sup>	指定プロセスのポーズ停止指令をキャンセル
ProcCancelStopCycle <sup>10</sup>	指定プロセスのサイクル停止指令をキャンセル
QueryProcRuntimeError <sup>10</sup>	指定プロセスのランタイムエラーに関するデータを取得
ProcResetRuntimeError <sup>10</sup>	指定プロセスのランタイムエラー状態をリセット
GetProcNumber <sup>12</sup>	自プロセス番号を取得

ポート連動モーション / オーバラップ移動 サポート関数 <sup>7</sup>

関数名	機能
RobSetMoveProgressDo	移動動作進捗度による出力ポートのリアルタイム制御有効化
RobResetMoveProgressDo	移動動作進捗度による出力ポートのリアルタイム制御無効化
RobSetOverlap	移動動作のオーバラップ設定
RobResetOverlap	移動動作のオーバラップ設定解除
RobSetDiStartMove	ポート状態によるリアルタイムな移動動作開始の設定
RobResetDiStartMove	ポート状態によるリアルタイムな移動動作開始の設定解除

アナログ入力連動モーションサポート関数 <sup>9</sup>

関数名	機能
RobSetAiStopMove	指定ロボットを、指定のアナログチャンネルの指定の値で途中停止するよう設定
RobResetAiStopMove	RobSetAiStopMove で設定した途中停止指示を解除する

以上