Robot Language User Manual

www.deltaww.com

Smarter. Greener. Together.

Robot Language Manual

1. Basic introduction

1.1 Basic introduction

Table 1-1 Robot language overview

Function items	Instruction symbols	Description
Arithmetic, Logical and Comparison operators	+	Add
	-	Subtract
	*	Multiply
	/	Divide
	^	Exponential
	and	Logic operator: AND
	or	Logic operator: OR
	>	Greater than
	>=	Greater than or equal to
	<	Less than
	<=	Less than or equal to
	==	Equal to
	–=	Not equal to
Mathematic and Trigonometric instructions	ABS	Absolute value
	ACOS	Arccosine function
	ASIN	Arcsine function
	ATAN	Arctangent function
	ATAN2	ATAN2 function
	CEIL	Ceiling function; largest integer that is not less than the input value
	COS	Cosine function
	COSH	Hyperbolic cosine function
	DEG	Arc angle
	EXP	Exponential e-based value
	FLOOR	Floor function; largest integer that is not greater than the input value
	FMOD	Remainder
	LOG10	Logarithm base 10

	LOG	Logarithm base e
	MAX	Maximum value in list
	MIN	Minimum value in list
	MODF	Separate value into integer and decimal parts
	POW	Power of a value
	RAD	Angle rotation
	SIN	Sine function
	SINH	Hyperbolic sine function
	SQRT	Square root
	TAN	Tangent function
	TANH	Hyperbolic tangent function
	ROUND	Rounding function
Basic instructions	DELAY	Set delay time
	TimerOn	Enable timer
	TimerRead	Read time since last timer was enabled
Point Management instructions	SetGlobalPoint	Set global point
	CopyPoint	Copy point data
	ReadPoint	Read point data
	WritePoint	Write temporary value to point data
	RobotX	Current X-direction coordinate
	RobotY	Current Y-direction coordinate
	RobotZ	Current Z-direction coordinate
	RobotRX	Current RX-direction coordinate
	RobotRY	Current RY-direction coordinate
	RobotRZ	Current RZ-direction coordinate
	RobotHand	Current robot hand status
	RobotElbow	Current robot shoulder status
	RobotShoulder	Current robot elbow status
	RobotFlip	Current robot wrist status
	RobotJRC	Current robot joint index status
	RobotJ1	Current robot first axis angle
	RobotJ2	Current robot second axis angle
	RobotJ3	Current robot third axis angle or Z-axis position
	RobotJ4	Current robot fourth axis angle

	RobotJ5	Current robot fifth axis angle
	RobotJ6	Current robot sixth axis angle
Motion Parameter instructions	AccJ	Acceleration setting for PTP motion instructions
	DecJ	Deceleration setting for PTP motion instructions
	SpdJ	Maximum speed setting for PTP motion instructions
	AccL	Acceleration setting for linear interpolation and arc interpolation motion instructions
	DecL	Deceleration setting for linear interpolation and arc interpolation motion instructions
	SpdL	Maximum speed setting for linear interpolation and arc interpolation motion instructions
	Accur	All axes passed positioning accuracy; valid even for motion instructions without PASS parameter set
	SetAccur	Single axis passed position accuracy; valid even for motion instructions without PASS parameter set
	SetPayload	Set automatic acceleration and deceleration payload value
	PassMode	Switch the PASS mode for motion control instructions
	SetOverlapDistance	Set the distance value for distance interruption mode; must be used in distance interruption mode
	SetOverlapTime	Set the distance value for time interruption mode; must be used in deceleration interruption mode
	JerkL	Add acceleration speed setting
Motion Control instructions	RobotServoOn	Turn on all robot axes motor servo
	RobotServoOff	Turn off all robot axes motor servos
	ExtServoOn	Turn on single external axis motor servo
	ExtServoOff	Turn off single external axis motor servo
	MovJ	Move the set axis to the target position

	ExtMovJ	Move the external motor axis to the target position
	MovP	Specify target point for PTP motion
	MovPR	Use relative distance for PTP motion
	MovL	Specify target point for linear interpolation motion
	MovLR	Use relative distance for linear interpolation motion
	MArchP	Perform arch motion along the Z-axis in PTP motion mode
	MArchL	Perform arch motion along the Z-axis in linear interpolation motion mode
	MArc	Specify target position and points for arc interpolation arc motion
	MCircle	Specify target position and points for arc interpolation circular motion; three-point circle
	Lift	Use absolute coordinate to move to the relative reference point position
	MotionStop	Stop robot motion
	ExtMotionStop	Stop external axis motion
	ContinueCartesianJOG	Use linear interpolation to move continuously in a specific direction
Coordinate instructions	SetUF	Set user coordinates
	SetTF	Set tool coordinates
	ChangeUF	Switch user coordinates
	ChangeTF	Switch tool coordinates
	GetTF	Get the width, height, angle, pitch, roll and yaw setting values for the tool coordinates
Process control instructions	if…then…elseif…then…else…end	if-then-else-elseif conditional statement
	break	exit the current loop
	while…do…end	while loop
	for…do…end	for loop
	repeat…until	repeat-until loop
	function…end	User defined sub-function
Input/Output instructions	DI	Return digital input status (ON or OFF)
	DO	Read or write digital output status

	ExtDI	Read external digital input status
	ExtDO	Read or set external digital output status
	ReadModbus	Read values at a Modbus memory address
	WriteModbus	Write values to a Modbus memory address
	ConnectMBC	Connect to Modbus client
	CloseMBC	Close Modbus client connection
	SetSlaverMBC	Set Modbus client station number
	SetTimeOutMBC	Set connection timeout time
	WriteMBC	Write values to a Modbus client address
	ReadMBC	Read values from a Modbus client address
Program Execution instructions	QUIT	Stop executing program
	PAUSE	Pause current motion and program execution; must use external software or equipment to restart the program to continue execution.
Application Function instructions	SafetyMode	Switch function pause mode
	SafetyStatus	Read the function pause triggering status
	MultiTask	Multi-threading function for collaborative motion instructions

1.2 Syntax definition

Table 1-2 Syntax definitions

Notes	Description
Case sensitive	The robot language is case sensitive; a and A are not the same
Statement separator	Robot language statements can be separated using commas (,), or you can use a blank space. For example: a1=0 a2=1 a3=2 is the same as a1, a2, a3 = 0, 1, 2
Number of variables＞ number of values in an assignment statement	Fill in nil according to the number of variables; for example: a1, a2, a3 = 0, 1, then the value of a3 is equal to nil
Number of variables＜ number of values in an assignment statement	Excess values are ignored; for example: a1, a2 = 0, 1, 2, then 2 is ignored

1.3 Declaring variables

In the robot language, all variables are global unless you add the keyword “local” to declare a local variable. The following table lists examples of global and local variables.

Examples	a=1 (Sets variable a as global variable) if a==1 then local b=2 (Sets variable b as local variable in the if-the-end statement) end if b==2 then (False; the value of b here is nil because b is local to the if-then-end statement) c=1 end

1.4 Reserved keywords

(1) Do not use keywords as variable names. Be careful when naming variables.

(2) The robot language keywords are case sensitive: and, And, AND are not the same.

(3) Do not use the following reserved words as variable names: and, break, do, else, elseif, end, false, for, function, if, in, local, global, nil, not, or, repeat, return, then, true, until, while, P, p, table, boolean, number, string, thread, goto, in, ON, OFF.

1.5 Point definition P

Points are represented in two ways in the robot language:

(1) Use text inside double quotes for point names; example: MovP ("FirstPoint").

(2) Use point number representation; example: MovP (1).

2. Arithmetic, Logical and Comparison Operators

Table 2-1 Operators

Symbol	Description
+	Add
-	Subtract
*	Multiply
/	Divide
^	Exponential
and	Logic operator: AND
or	Logic operator: OR
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
==	Equal to
–=	Not equal to

3. Mathematic and Trigonometric Instructions

Instructions for tri-axis robot: Tri-axis

Instructions for four-axis robot: Four-axis

Instructions for five-axis robot: Five-axis

Instructions for six-axis robot: Six-axis

Instructions for all robots: All

ABS				**(All)**
Usage	Calculate the absolute value of the input value			
Syntax	ret = ABS(a)			
Input	Parameter	Type	Name	Description
	a	number	Value to be calculated	Value to be calculated
Output	Parameter	Type	Name	Description

	ret	number	Calculated value	Calculated value
Example	ret = ABS(-10) -- ret is 10			

ACOS				**(All)**
Usage	Calculate the arccosine of the input value			
Syntax	ret = ACOS(a)			
Input	Parameter	Type	Name	Description
	a	number	Value to be calculated	Value to be calculated
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value; output unit: degrees
Example	ret = ACOS (0.5) -- ret is 60			

ASIN				**(All)**
Usage	Calculate the arcsine of the input value			
Syntax	ret= ASIN(a)			
Input	Parameter	Type	Name	Description
	a	number	Value to be calculated	Value to be calculated
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value; output unit: degrees
Example	ret = ASIN (1) -- ret is 90			

ATAN				**(All)**
Usage	Calculate the arctangent of the input value			
Syntax	ret = ATAN(a)			
Input	Parameter	Type	Name	Description
	a	number	Value to be calculated	Value to be calculated
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value; output unit: degrees
Example	ret = ATAN (1) -- ret is 45			

ATAN2				(All)
Usage	Calculate the arctangent of the input values			
Syntax	ret = ATAN2(a,b)			
Input	Parameter	Type	Name	Description
	a	number	Numerator	Numerator of value to be calculated
	b	number	Denominator	Denominator of value to be calculated
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value; output unit: radian
Example	ret = ATAN2 (1,1) -- ret is 0.785			

CEIL				(All)
Usage	Calculate the largest integer that is not less than the input value			
Syntax	ret = CEIL(a)			
Input	Parameter	Type	Name	Description
	a	number	Value to be calculated	Value to be calculated
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value; output unit: degrees
Example	ret = CEIL(1.234) -- ret is 2			

COS				(All)
Usage	Calculate the cosine of the input value			
Syntax	ret = COS(a)			
Input	Parameter	Type	Name	Description
	a	number	Value to be calculated	Value to be calculated; input unit: degrees
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value
Example	ret = COS(60) -- ret is 0.5			

COSH				(All)
Usage	Calculate the hyperbolic cosine of the input value			

Syntax	ret = COSH(a)			
Input	Parameter	Type	Name	Description
	a	number	Value to be calculated	Value to be calculated; input unit: radian
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value
Example	ret = COSH(1) -- ret is 1.543			

DEG				**(All)**
Usage	Calculate the arc angle of the input value			
Syntax	ret = DEG(a)			
Input	Parameter	Type	Name	Description
	a	number	Value to be calculated	Value to be calculated; input unit: radian
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value
Example	ret = DEG(1.047) -- ret is 59.988			

EXP				**(All)**
Usage	Calculate the e-based exponential of the input value			
Syntax	ret = EXP(a)			
Input	Parameter	Type	Name	Description
	a	number	Power value	Power value
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value
Example	ret = EXP(1) -- ret is 2.718			

FLOOR				**(All)**
Usage	Calculate the largest integer that is not greater than the input value			
Syntax	ret = FLOOR(a)			
Input	Parameter	Type	Name	Description
	a	number	Value to be calculated	Value to be calculated
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value
Example	ret = FLOOR(1.543) -- ret is 1			

FMOD				**(All)**
Usage	Calculate the remainder of the input values			
Syntax	ret = FMOD(a,b)			
Input	Parameter	Type	Name	Description
	a	number	Numerator	Numerator of value to be calculated
	b	number	Denominator	Denominator of value to be calculated
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value
Example	ret = FMOD(2,3) -- ret is 2			

LOG10				**(All)**
Usage	Calculate the logarithm base 10 of the input value			
Syntax	ret = LOG10(a)			
Input	Parameter	Type	Name	Description
	a	number	Value to be calculated	Value to be calculated
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value
Example	ret = LOG10(100) -- ret is 2			

LOG				**(All)**
Usage	Calculate the logarithm base e as of the input value			
Syntax	ret = LOG(a)			
Input	Parameter	Type	Name	Description
	a	number	Value to be calculated	Value to be calculated
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value
Example	ret = LOG(10) -- ret is 2.302			

MAX		**(All)**
Usage	Find the maximum value in input list	

Syntax	ret = MAX(…)			
Input	Parameter	Type	Name	Description
	…	number	Value to be calculated	Value to be calculated; any value can be input for calculation
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value
Example	ret = MAX(11,22,25,99,7,9) -- ret is 99			

MIN				**(All)**
Usage	Get the minimum value in the input list			
Syntax	ret = MIN(…)			
Input	Parameter	Type	Name	Description
	…	number	Value to be calculated	Value to be calculated; can input any value for calculation
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value
Example	ret = MIN(11,22,25,99,7,9) -- r7			

MODF				**(All)**
Usage	Separate the input value into integer and decimal parts			
Syntax	ret1,ret2 = MODF(a)			
Input	Parameter	Type	Name	Description
	a	number	Value to be calculated	Value to be calculated
Output	Parameter	Type	Name	Description
	ret1	number	Integer	Integer of calculated value
	ret2	number	Decimal number	Decimal number of calculated value
Example	ret1,ret2 = MODF(1.5) – ret1 is 1, ret2 is 0.5			

POW				**(All)**
Usage	Calculate the first input raised to the second input			
Syntax	ret = POW(a,b)			
Input	Parameter	Type	Name	Description
	a	number	Base number	Base number of value to be

				calculated
	b	number	Power	Power number of value to be calculated
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value
Example	ret = POW(10,2) -- ret is 100			

RAD				**(All)**
Usage	Calculate the angle rotation of the input value			
Syntax	ret = RAD(a)			
Input	Parameter	Type	Name	Description
	a	number	Value to be calculated	Value to be calculated; unit: degrees
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value
Example	ret = RAD(60) -- ret is 1.047			

SIN				**(All)**
Usage	Calculate the sine of the input value			
Syntax	ret = SIN(a)			
Input	Parameter	Type	Name	Description
	a	number	Value to be calculated	Value to be calculated; input unit: degrees
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value
Example	ret = SIN(30) -- ret is 0.5			

SINH				**(All)**
Usage	Calculate the hyperbolic sine of the input value			
Syntax	ret = SINH(a)			
Input	Parameter	Type	Name	Description
	a	number	Value to be calculated	Value to be calculated; input unit: radian
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value
Example	ret = SINH(1) -- ret is 1.175			

SQRT				**(All)**
Usage	Calculate the square root of the input value			
Syntax	ret = SQRT(a)			
Input	Parameter	Type	Name	Description
	a	number	Value to be calculated	Value to be calculated
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value
Example	ret = SQRT(4) -- ret is 2			

TAN				**(All)**
Usage	Calculate tangent function			
Syntax	ret = TAN(a)			
Input	Parameter	Type	Name	Description
	a	number	Value to be calculated	Value to be calculated; input unit: degrees
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value
Example	ret = TAN(45) -- ret is 1			

TANH				**(All)**
Usage	Calculate the hyperbolic tangent of the input value			
Syntax	ret = TANH(a)			
Input	Parameter	Type	Name	Description
	a	number	Value to be calculated	Value to be calculated; input unit: radian
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value
Example	ret = TANH(1) -- ret is 0.761			

ROUND				**(All)**
Usage	Calculate the rounded value of the input value			
Syntax	ret = ROUND(a,b)			
Input	Parameter	Type	Name	Description
	a	number	Value to be	Value to be calculated

			calculated	
	b	number	Selection of number of digits	Select the number of digits for rounding operation
Output	Parameter	Type	Name	Description
	ret	number	Calculated value	Calculated value
Example	ret = ROUND(123.45 , 2) -- ret is 120 ret = ROUND(123.45 , -1) -- ret is 123.5			

4. Basic Instructions

DELAY				**(All)**
Usage	Set delay time			
Syntax	DELAY(a)			
Input	Parameter	Type	Name	Description
	a	number	Delay time	Unit is second; minimum accuracy is millisecond
Output	Parameter	Type	Name	Description
	None			
Example	DELAY(0.5) -- delay 0.5 seconds Time=5 -- Set the value of the variable Time to 5 DELAY (Time) --- delay 5 seconds			

TimerOn				**(All)**
Usage	Enable timer			
Syntax	TimerOn()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	None			
Example	TimerOn() -- Start timer for i = 1,10000 do i = i+1 end t = TimerRead() – t is the time passed since the timer started			

TimerRead				**(All)**
Usage	Read the time passed after the previous timer has started to time the program execution time; use the TimerOn function to start the timer.			
Syntax	ret = TimerRead()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	ret	number	Time passed	Time passed after the return timer has started; the unit is millisecond
Example	TimerOn() -- Start timer for i = 1,10000 do i = i+1 end t = TimerRead()-- t is the time passed since the timer started			

5. Point Management Instructions

SetGlobalPoint				**(All)**
Usage	Set global point			
Syntax	Tri-axis robot SetGlobalPoint(a,b,c,d,h,l,m,n) SetGlobalPoint(a,b,c,d,h,l,m) Four-axis robot SetGlobalPoint(a,b,c,d,g,h,l,m,n) SetGlobalPoint(a,b,c,d,g,h,l,m) Five-axis robot SetGlobalPoint(a,b,c,d,f,g,h,l,m,n) SetGlobalPoint(a,b,c,d,f,g,h,l,m) Six-axis robot SetGlobalPoint(a,b,c,d,e,f,g,i,j,k,l,m,n) SetGlobalPoint(a,b,c,d,e,f,g,i,j,k,l,m)			
Input	Parameter	Type	Name	Description
	a	number	Point number	Point number, range: 1–1000

	b	number	X	Space coordinate X; unit is millimeter
	c	number	Y	Space coordinate Y; unit is millimeter
	d	number	Z	Space coordinate Z; unit is millimeter
	e	number	RX	Space coordinate RX; unit is millimeter
	f	number	RY	Space coordinate RY; unit is millimeter
	g	number	RZ	Space coordinate RZ; unit is millimeter
	h	Number or string	HAND	SCARA: 0 or "R" (right hand) 1 or " L" (left hand)
	i	number or string	ELBOW	VA: 0 or "D" (bottom elbow) 1 or " U" (top elbow)
	j	number or string	SHOULDER	VA: 0 or "R" (right shoulder) 1 or " L" (left shoulder)
	k	number or string	FLIP	VA: 0 or "N" (wrist with no flip) 1 or " H" (wrist with flip)
	l	number	UF	User coordinates, provide 1–9 sets, 0 is the earth coordinates
	m	number	TF	Tool coordinates, provide 1–9 sets, 0 is the earth coordinates
	n	table	JRC	Joint Index table; if not input, the default is {0,0,0,0,0,0,0,0}
Output	Parameter	Type	Name	Description
	None			
Example	Tri-axis robot SetGlobalPoint(1,10,20,30,1,9,8,{0,0,0,0,0,0,0,0}) SetGlobalPoint(1,10,20,30,1,9,8) Four-axis robot SetGlobalPoint(1,10,20,30,40,1,9,8,{0,0,0,0,0,0,0,0}) SetGlobalPoint(1,10,20,30,40,1,9,8) Five-axis robot			

	SetGlobalPoint(1,10,20,30,40,50,1,9,8,{0,0,0,0,0,0,0,0})
	SetGlobalPoint(1,10,20,30,40,50,1,9,8)
	Six-axis robot
	SetGlobalPoint(1,10,20,30,40,50,60,1,1,1,9,8,{0,0,0,0,0,0,0,0})
	SetGlobalPoint(1,10,20,30,40,50,60,"U","L","H",9,8)

CopyPoint				**(All)**
Usage	Copy point data			
Syntax	CopyPoint(a,b)			
Input	Parameter	Type	Name	Description
	a	number or string	Point to be copied	Point to be copied, point number or point name
	b	number or string	Copied point	Copied point, point number or point name
Output	Parameter	Type	Name	Description
	None			
Example	CopyPoint(1,2) CopyPoint("P1","P2")			

ReadPoint				**(All)**
Usage	Read point data			
Syntax	ret = ReadPoint(a,b)			
Input	Parameter	Type	Name	Description
	a	number or string	Point to be read	Point to be read, point number or point name
	b	string	Item to read	"X": Coordinate value in the X direction (unit: millimeter) "Y": Coordinate value in the Y direction (unit: millimeter) "Z": Coordinate value in the Z direction (unit: millimeter) "A": Coordinate value in the RX direction (unit: degree) "B": Coordinate value in the RY

				direction (unit: degree) "C": Coordinate value in the RZ direction (unit: degree) "RX": Coordinate value in the RX direction (unit: degree) "RY": Coordinate value in the RY direction (unit: degree) "RZ": Coordinate value in the RZ direction (unit: degree) "UF": User coordinates of the point "TF": Tool coordinates of the point "H": Hand information (0: right hand; 1: left hand) "E": Elbow information (0: bottom elbow; 1: top elbow) "S": Shoulder information (0: right shoulder; 1: left shoulder) "F": Hand information (0: wrist with no flip; 1: wrist with flip) "COORD": Coordinates ("WCS": earth coordinates; "PCS": user coordinates; "TCS": tool coordinates; "ACS": axis coordinates) "JRC" :Joint Index table, 8 element table for example: {0,0,0,0,0,0,0,0}
Output	Parameter	Type	Name	Description
	ret	number or string or table	Point information read from the point	Mode "X", allows reading of X value Mode "Y", allows reading of Y value

		Mode "Z", allows reading of Z value Mode "A", allows reading of A value Mode "B", allows reading of B value Mode "C", allows reading of C value Mode "RX", allows reading of RX value Mode "RY", allows reading of RY value Mode "RZ", allows reading of RZ value Mode "H": hand information (read 0 or "R": right hand; read 1 or "L": left hand) Mode "E": elbow information (read 0 or "D": bottom elbow; read 1 or "U": top elbow) Mode "S": shoulder information (read 0 or "R": right shoulder; read 1 or "L": left shoulder) Mode "F": wrist information (read 0 or "N": wrist with no flip; read 1 or "H": wrist with flip) Mode "UF": user coordinates; allows reading of range 1–9 Mode "TF": tool coordinates; allows reading of range 1–9 Mode "COORD": Coordinates (read "WCS": earth coordinates; read "PCS": user coordinates; read "TCS": tool coordinates; read "ACS": axis coordinates) Mode "JRC" :Joint Index table, 8 element table for example:

				{0,0,0,0,0,0,0,0}
Example	PostionX = ReadPoint(1001,"X") PostionY = ReadPoint(1001,"Y") PostionZ = ReadPoint(1001,"Z") PostionRZ = ReadPoint(1001,"C") PostionRZ = ReadPoint(1001,"RZ") PostionUF = ReadPoint("P1","UF") PostionTF=ReadPoint("P1","TF") PositionJRC{}=ReadPoint(1001,"JRC")			

WritePoint				**(All)**
Usage	Write temporary value to point data			
Syntax	WritePoint(a,b,c)			
Input	Parameter	Type	Name	Description
	a	number or string	Point to be written	Point to write to, input point number or point name
	b	string	Data to write	"X": Coordinate value in the X direction (unit: millimeter) "Y": Coordinate value in the Y direction (unit: millimeter) "Z": Coordinate value in the Z direction (unit: millimeter) "A": Coordinate value in the RX direction (unit: degree) "B": Coordinate value in the RY direction (unit: degree) "C": Coordinate value in the RZ direction (unit: degree) "RX": Coordinate value in the RX direction (unit: degree) "RY": Coordinate value in the RY direction (unit: degree) "RZ": Coordinate value in the RZ direction (unit: degree) "UF": User coordinates of the

				point "TF": Tool coordinates of the point "H": Hand information (0: right hand; 1: left hand) "E": Elbow information (0: bottom elbow; 1: top elbow) "S": Shoulder information (0: right shoulder; 1: left shoulder) "F": Hand information (0: wrist with no flip; 1: wrist with flip) "COORD": Coordinates ("WCS": earth coordinates; "PCS": user coordinates; "TCS": tool coordinates; "ACS": axis coordinates) "JRC": Joint Index table, 8 element table for example: {0,0,0,0,0,0,0,0}
	c	number or string or table	Input value	Mode "X", to write X value Mode "X", to write Y value Mode "X", to write Z value Mode "X", to write RX value Mode "X", to write RY value Mode "X", to write RZ value Mode "RX", to write RX value Mode "RY", to write RY value Mode "RZ", to write RZ value Mode "H": hand information (enter 0 or "R": right hand; enter 1 or "L": left hand) Mode "E": elbow information (enter 0 or "D": bottom elbow; enter 1 or "U": top elbow) Mode "S": shoulder information

				(enter 0 or "R": right shoulder; enter 1 or "L": left shoulder) Mode "F": wrist information (enter 0 or "N": wrist with no flip; enter 1 or "H": wrist with flip) Mode "UF": user coordinates; to write range 1–9 Mode "TF": tool coordinates; to write range 1–9 Mode "COORD": Coordinates (enter "WCS": earth coordinates; enter "PCS": user coordinates; enter "TCS": tool coordinates; enter "ACS": axis coordinates) Mode "JRC": Joint Index table, 8 element table for example: {0,0,0,0,0,0,0,0}
Output	Parameter	Type	Name	Description
	None			
Example	WritePoint(1001,"X",300) WritePoint(1001,"Y",50) WritePoint("P1","RZ",30) WritePoint("P1","H",1) WritePoint(1002,"RX",10) WritePoint("P2","COORD","PCS") WritePoint(1001,"JRC",{0,0,0,0,0,0,0,0})			

RobotX				**(All)**
Usage	Get the current X-direction coordinate			
Syntax	ret = RobotX()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	ret	number	X-direction coordinates	Current X-direction coordinate values; to read the X-direction coordinate values of the tool

				coordinates or user coordinates, switch to the corresponding coordinate status in order to read the corresponding coordinates information; unit: millimeter
Example	NowPosition_X = RobotX()			

RobotY				**(All)**
Usage	Get the current Y-direction coordinate			
Syntax	ret = RobotY()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	ret	number	Y-direction coordinates	Current Y-direction coordinate values; to read the Y-direction coordinate values of the tool coordinates or user coordinates, switch to the corresponding coordinate status in order to read the corresponding coordinates information; unit: millimeter
Example	NowPosition_Y = RobotY()			

RobotZ				**(All)**
Usage	Get the current Z-direction coordinate			
Syntax	ret = RobotZ()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	ret	number	Z-direction coordinates	Current Z-direction coordinate values; to read the Z-direction coordinate values of the tool coordinates or user coordinates, switch to the corresponding

				coordinate status in order to read the corresponding coordinates information; unit: millimeter
Example	NowPosition_Z = RobotZ()			

RobotRX				**(Six axis)**
Usage	Get the current RX-direction coordinate			
Syntax	ret = RobotRX()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	ret	number	RX-direction coordinates	Current RX-direction coordinate values; to read the RX-direction coordinate values of the tool coordinates or user coordinates, switch to the corresponding coordinate status in order to read the corresponding coordinates information; unit: degree
Example	NowPosition_RX = RobotRX()			

RobotRY				**(Five axes and six axes)**
Usage	Get the current RY-direction coordinate			
Syntax	ret = RobotRY()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	ret	number	RY-direction coordinates	Current RY-direction coordinate values; to read the RY-direction coordinate values of the tool coordinates or user coordinates, switch to the corresponding coordinate status in order to read the corresponding

				coordinates information; unit: degree
Example	NowPosition_RY = RobotRY()			

RobotRZ				**(Four axes, five axes and six axes)**
Usage	Get the current RZ-direction coordinate			
Syntax	ret = RobotRZ()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	ret	number	RZ-direction coordinates	Current RZ-direction coordinate values; to read the RZ-direction coordinate values of the tool coordinates or user coordinates, switch to the corresponding coordinate status in order to read the corresponding coordinates information; unit: degree
Example	NowPosition_RZ = RobotRZ()			

RobotHand				**(Tri-axes, four axes and five axes)**
Usage	Get the current robot hand status			
Syntax	ret = RobotHand()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	ret	number	Hand value	Ret is the hand status of the current robot; read value 0 for right hand and read value 1 for left hand
Example	NowPosition_Hand = RobotHand() – NowPosition_Hand is 0 (right hand) or 1 (left hand)			

RobotElbow		**(Six axis)**
Usage	Get the current robot elbow status	

Syntax	ret = RobotElbow()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	ret	number	Elbow value	Ret is the elbow status of the current robot; read value 0 for bottom elbow and read value 1 for top elbow
Example	NowPosition_Elbow = RobotElbow()			

RobotShoulder				**(Six axis)**
Usage	Get the current robot shoulder status			
Syntax	ret = RobotShoulder ()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	ret	number	Shoulder value	Ret is the shoulder status of the current robot; read value 0 for the right shoulder and read value 1 for the left shoulder
Example	NowPosition_Shoulder = RobotShoulder()			

RobotFlip				**(Six axis)**
Usage	Get the current robot wrist status			
Syntax	ret = RobotFlip()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	ret	number	Wrist value	Ret is the wrist status of the current robot; read value 0 for the wrist with no flip and read value 1 for the wrist with a flip
Example	NowPosition_Flip = RobotFlip ()			

RobotJRC		**(All)**
Usage	Get the current robot joint index status	

Syntax	ret = RobotJRC ()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	ret	table	Joint index value	Ret is the joint index status of the current robot; it is an 8 element table. Elements 1–6 corresponds to the J1–J6 joint index statuses, elements 7–8 are currently not used.
Example	NowPosition_JRC = RobotJRC () J1_index = NowPosition_JRC[1] -- read J1 index J2_index = NowPosition_JRC[2] -- read J2 index J3_index = NowPosition_JRC[3] -- read J3 index			

RobotJ1				**(All)**
Usage	Get the current robot first axis angle			
Syntax	ret = RobotJ1()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	ret	number	J1 value	Ret is the first axis angle of the current robot; unit: degree
Example	NowPosition_J1 = RobotJ1()			

RobotJ2				**(All)**
Usage	Get the current robot second axis angle			
Syntax	ret = RobotJ2 ()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	ret	number	J2 value	Ret is the second axis angle of the current robot; unit: degree
Example	NowPosition_J2 = RobotJ2()			

RobotJ3	**(All)**

Usage	Get the current robot third axis angle or Z-axis position			
Syntax	ret = RobotJ3 ()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	ret	number	J3 value	Ret is the third axis angle or Z-axis position of the current robot; for tri-axis, four-axis and five-axis robots, unit: millimeters; for six-axis robots, unit: degree
Example	NowPosition_J3 = RobotJ3()			

RobotJ4				**(Four axes, five axes and six axes)**
Usage	Get the current robot fourth axis angle			
Syntax	ret = RobotJ4()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	ret	number	J4 value	Ret is the fourth axis angle of the current robot; unit: degree
Example	NowPosition_J4 = RobotJ4()			

RobotJ5				**(Five axes and six axes)**
Usage	Get the current robot fifth axis angle			
Syntax	ret = RobotJ5 ()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	ret	number	J5 value	Ret is the fifth axis angle of the current robot; unit: degree
Example	NowPosition_J5 = RobotJ5()			

RobotJ6		**(Six axis)**
Usage	Get the current robot sixth axis angle	
Syntax	ret = RobotJ6 ()	

Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	ret	number	J6 value	Ret is the sixth axis angle of the current robot; unit: degree
Example	NowPosition_J6 = RobotJ6()			

6. Motion Parameter Instructions

AccJ				**(All)**
Usage	Acceleration setting for PTP motion instructions			
Syntax	AccJ (a)			
Input	Parameter	Type	Name	Description
	a	number	Acceleration value	Unit is percentage; allowed input range 1–100
Output	Parameter	Type	Name	Description
	None			
Example	SpdJ(50) AccJ(50) DecJ(50) MovP(1001)			

DecJ				**(All)**
Usage	Deceleration setting for PTP motion instructions			
Syntax	DecJ (a)			
Input	Parameter	Type	Name	Description
	a	number	Deceleration value	Unit is percentage; allowed input range 1–100
Output	Parameter	Type	Name	Description
	None			
Example	SpdJ(50) AccJ(50) DecJ(50) MovP(1001)			

SpdJ				**(All)**
Usage	Maximum speed setting for PTP motion instructions			
Syntax	SpdJ (a)			
Input	Parameter	Type	Name	Description
	a	number	Maximum speed value	Unit is percentage; allowed input range 1–100
Output	Parameter	Type	Name	Description
	None			
Example	SpdJ(50) AccJ(50) DecJ(50) MovP(1001)			

AccL				**(All)**
Usage	Acceleration setting for linear interpolation and arc interpolation motion instructions			
Syntax	AccL(a)			
Input	Parameter	Type	Name	Description
	a	number	Acceleration value	Unit is millimeter/second squared; allowed input range 1–25000
Output	Parameter	Type	Name	Description
	None			
Example	SpdL(500) AccL(500) DecL(500) MovL(1001)			

DecL				**(All)**
Usage	Deceleration setting for linear interpolation and arc interpolation motion instructions			
Syntax	DecL(a)			
Input	Parameter	Type	Name	Description
	a	number	Deceleration value	Unit is millimeter/second squared; allowed input range 1–25000

Output	Parameter	Type	Name	Description
	None			
Example	SpdL(500) AccL(500) DecL(500) MovL(1001)			

SpdL				**(All)**
Usage	Maximum speed setting for linear interpolation and arc interpolation motion instructions			
Syntax	SpdL(a)			
Input	Parameter	Type	Name	Description
	a	number	Maximum speed value	Actual speed millimeter/second; allowed input range 1–2000
Output	Parameter	Type	Name	Description
	None			
Example	SpdL(500) AccL(500) DecL(500) MovL(1001)			

Accur				**(All)**
Usage	All axes passed positioning accuracy; valid even for motion instructions without the PASS parameter set			
Syntax	Accur (a)			
Input	Parameter	Type	Name	Description
	a	string	Accuracy mode	"HIGH" highest position accuracy "STANDARD" standard position accuracy "MEDIUM" medium position accuracy "ROUGH" low position accuracy "MAXROUGH" lowest position accuracy
Output	Parameter	Type	Name	Description

	None
Example	Accur("HIGH") MovL("P1") Accur("ROUGH") MovL("P3")

SetAccur				**(All)**
Usage	Single axis passed position accuracy; valid even for motion instructions without the PASS parameter set. A smaller position accuracy value is more accurate.			
Syntax	SetAccur (a,b)			
Input	Parameter	Type	Name	Description
	a	number	Motor axis number	Motor axis number; number range: 1–10
	b	number	Position accuracy value	Position accuracy value; unit is PUU
Output	Parameter	Type	Name	Description
	None			
Example	This example uses timers to record the motion time, showing that setting the position accuracy to a larger value results in a shorter motion time (t1 < t2). TimerOn() SetAccur(1,100000) MovJ(1,0,"PUU") MovJ(1,100000,"PUU") MovJ(1,200000,"PUU") t1 = TimerRead() TimerOn() SetAccur(1,1) MovJ(1,0,"PUU") MovJ(1,100000,"PUU") MovJ(1,200000,"PUU") t2 = TimerRead()			

SetPayload		**(All)**
Usage	Set automatic acceleration and deceleration payload value	

Syntax	SetPayload(a,b,c,d,e,f,g)			
Input	Parameter	Type	Name	Description
	a	number	Object weight	Object weight; unit is kilograms
	b	number	Distance between the X direction and the center of mass	Distance between the X direction and the center of mass; unit is meter
	c	number	Distance between the Y direction and the center of mass	Distance between the Y direction and the center of mass; unit is meter
	d	number	Distance between the Z direction and the center of mass	Distance between the Z direction and the center of mass; unit is meter
	e	number	Inertial tensor in the X direction	Inertial tensor in the X direction; unit is kilograms * meter squared
	f	number	Inertial tensor in the Y direction	Inertial tensor in the Y direction; unit is kilograms * meter squared
	g	number	Inertial tensor in the Z direction	Inertial tensor in the Z direction; unit is kilograms * meter squared
Output	Parameter	Type	Name	Description
	None			
Example	SetPayload(3,0,0,0,0,0,0.0091)			

PassMode				**(All)**
Usage	Switch the PASS mode for motion control instructions			
Syntax	PassMode (a)			
Input	Parameter	Type	Name	Description
	a	string	Interrupt mode	Input “DISTANT” to switch to distance interrupt mode. Input “DEC” to switch to deceleration interrupt mode.
Output	Parameter	Type	Name	Description

	None
Example	PassMode("DISTANT") SetOverlapDistance (50) MovP(1001, "PASS") MovP(1002) PassMode(" DEC") SetOverlapTime (50) MovP(1001,"PASS") MovP(1002)

SetOverlapDistance				**(All)**
Usage	Set the distance value for distance interruption mode; must be used in distance interruption mode			
Syntax	SetOverlapDistance (a)			
Input	Parameter	Type	Name	Description
	a	number	Distance interrupt value	Distance interrupt value; unit is millimeter
Output	Parameter	Type	Name	Description
	None			
Example	PassMode(" DISTANT") SetOverlapDistance (50) MovP(1001,"PASS") MovP(1002)			

SetOverlapTime				**(All)**
Usage	Set the distance value for time interruption mode; must be used in deceleration interruption mode			
Syntax	SetOverlapTime (a)			
Input	Parameter	Type	Name	Description
	a	number	Time interrupt value	Time interrupt value; unit is percentage
Output	Parameter	Type	Name	Description
	None			
Example	PassMode("DEC") SetOverlapTime(50) MovP(1001,"PASS")			

	MovP(1002)

JerkL				**(All)**
Usage	Add acceleration speed setting			
Syntax	JerkL (a)			
Input	Parameter	Type	Name	Description
	a	number	Add acceleration value	Unit is percentage; allowed input range 0–100%
Output	Parameter	Type	Name	Description
	None			
Example	SpdL(500) AccL(500) DecL(500) JerkL(100) MovL(1001)			

7. Motion Control Instructions

RobotServoOn				**(All)**
Usage	Turn on all robot axes motor servos			
Syntax	RobotServoOn ()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	None			
Example	RobotServoOn()			

RobotServoOff				**(All)**
Usage	Turn off all robot axes motor servos			
Syntax	RobotServoOff ()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description

	None			
Example	RobotServoOff ()			

ExtServoOn				**(All)**
Usage	Turn on a single external axis motor servo			
Syntax	ExtServoOn (a)			
Input	Parameter	Type	Name	Description
	a	number	External motor axis number	External motor axis number
Output	Parameter	Type	Name	Description
	None			
Example	ExtServoOn (1)			

ExtServoOff				**(All)**
Usage	Turn off a single external axis motor servo			
Syntax	ExtServoOff (a)			
Input	Parameter	Type	Name	Description
	a	number	External motor axis number	External motor axis number
Output	Parameter	Type	Name	Description
	None			
Example	ExtServoOff (1)			

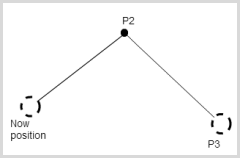
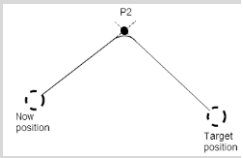
MovJ				**(All)**
Usage	Move the set axis to the target position			
Syntax	MovJ(a,b) MovJ(a,b,d) MovJ(a,b,d,e,f) MovJ(a,b,c) MovJ(a,b,c,d) MovJ(a,b,c,d,e,f)			
Input	Parameter	Type	Name	Description
	a	number	Motor axis number	Motor axis number:
	b	number	Moving value	Tri-axis robot When the input is axis 1, b is

		the absolute position angle; unit is degree. When the input is axis 2, b is the absolute position angle; unit is degree. When the input is axis 3, b is the absolute position; unit is millimeter. Four-axis robot When the input is axis 1, b is the absolute position angle; unit is degree. When the input is axis 2, b is the absolute position angle; unit is degree. When the input is axis 3, b is the absolute position; unit is millimeter When the input is axis 4, b is the absolute position angle; unit is degree. Five-axis robot When the input is axis 1, b is the absolute position angle; unit is degree. When the input is axis 2, b is the absolute position angle; unit is degree. When the input is axis 3, b is the absolute position; unit is millimeter When the input is axis 4, b is the absolute position angle; unit is degree. When the input is axis 5, b is the absolute position angle; unit is degree.

				Six-axis robot When the input is axis 1, b is the absolute position angle; unit is degree. When the input is axis 2, b is the absolute position angle; unit is degree. When the input is axis 3, b is the absolute position angle; unit is degree. When the input is axis 4, b is the absolute position angle; unit is degree. When the input is axis 5, b is the absolute position angle; unit is degree. When the input is axis 6, b is the absolute position angle; unit is degree.
	c	string	PUU mode	Input parameter is “PUU”; the input unit of the input parameter b is PUU
	d	number	Maximum speed	Unit is percentage; allowed input range 1–100
	f	number	Acceleration	Unit is percentage; allowed input range 1–100
	g	number	Deceleration	Unit is percentage; allowed input range 1–100
Output	Parameter	Type	Name	Description
	None			
Example	MovJ(4,180) MovJ(4,180,50) MovJ(4,-180,100,10,10) MovJ(4,18000,"PUU") MovJ(1,2000,"PUU",50)			

ExtMovJ	**(All)**

Usage	Move the external motor axis to the target position			
Syntax	ExtMovJ(a,b) ExtMovJ(a,b,e) ExtMovJ(a,b,e,f,g) ExtMovJ(a,b,c) ExtMovJ(a,b,c,e) ExtMovJ(a,b,c,e,f,g) ExtMovJ(a,b,d) ExtMovJ(a,b,d,e) ExtMovJ(a,b,d,e,f,g) ExtMovJ(a,b,d,c) ExtMovJ(a,b,d,c,e) ExtMovJ(a,b,d,c,e,f,g)			
Input	Parameter	Type	Name	Description
	a	number	External motor axis number	External motor axis number
	b	Number or string	Target point or PUU value	Input point number or point name; it is the PUU value in PUU mode
	c	string	PASS mode	Input is “PASS”
	d	string	PUU mode	Input is “PUU”, which means the input position unit is PUU
	e	number	Maximum speed	Unit is percentage; allowed input range 1–100
	f	number	Acceleration	Unit is percentage; allowed input range 1-100
	g	number	Deceleration	Unit is percentage; allowed input range 1–100
Output	Parameter	Type	Name	Description
	None			
Example	ExtMovJ(1,1001) ExtMovJ(4,1002, "PASS",50) ExtMovJ(1, "p1001",100,10,10) ExtMovJ(4,18000, "PUU") ExtMovJ(1,2000, "PUU","PASS",50,40,30)			

<table>
<tr><th colspan="4">MovP</th><th>(All)</th></tr>
<tr><td>Usage</td><td colspan="4">Specify target point for perform PTP motion</td></tr>
<tr><td>Syntax</td><td colspan="4">MovP(a)
MovP(a,c)
MovP(a,b)
MovP(a,b,c)
MovP(a,c,d,e)
MovP(a,b,c,d,e)</td></tr>
<tr><td>Input</td><td>Parameter</td><td>Type</td><td>Name</td><td>Description</td></tr>
<tr><td></td><td>a</td><td>Number or string</td><td>Target point</td><td>Input point number or point name</td></tr>
<tr><td></td><td>b</td><td>string</td><td>PASS mode</td><td>input “PASS” to switch the motion mode to PASS mode<table><tr><td>No PASS</td><td>PASS</td></tr><tr><td>When PASS keyword is not added, the robot fully reaches the P2 position and then moves to P3.</td><td>When PASS keyword is added, the robot moves towards P3 before reaching the P2 position; this function can help skip points that are not important positions.</td></tr><tr><td> 
</td><td> 
</td></tr></table></td></tr>
<tr><td></td><td>c</td><td>number</td><td>Maximum speed</td><td>Unit is percentage; allowed input range 1–100</td></tr>
<tr><td></td><td>d</td><td>number</td><td>Acceleration</td><td>Unit is percentage; allowed input range 1–100</td></tr>
<tr><td></td><td>e</td><td>number</td><td>Deceleratio</td><td>Unit is percentage; allowed input range 1–</td></tr>
</table>

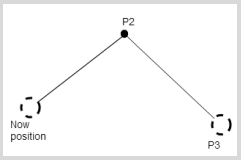
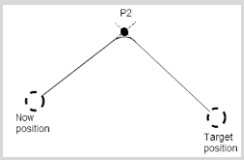
	n	100		
New syntax	MovP(a) -- See Chapter 13 “2.0 Revision Description” MovP(a,b)			
New input	Parameter	Type	Name	Description
	a	number+function or string+function	Target point related information setting	Can input the point number or point name plus the setting function; can also enter the setting function directly. Available setting functions include: X(a1), Y(a2), Z(a3), RX(a4), RY(a5) and RZ(a6). a1: the amount of change in X; unit is millimeter. a2: the amount of change in Y; unit is millimeter. a3: the amount of change in Z; unit is

				millimeter. a4: the amount of change in RX; unit is degree; use for six-axis robots. a5: the amount of change in RY; unit is degree; use for robots with five or more axes. a6: the amount of change in RZ; unit is degree.
	b	function	Speed related information setting and PASS setting	Can input the setting function directly; uses the default values for unset parameters. Available setting functions include: SPD(b1), ACC(b2), DEC(b3) and PASS(b4).

		b1: maximum speed %; allowed input range 1–100. b2: acceleration %; allowed input range 1–100. b3: deceleration %; allowed input range 1–100. b4: does not need parameter input; it is set to PASS mode when the function is called.
Output	Parameter Type Name Description	
	None	
Example	MovP(1) MovP(2,"PASS") MovP(3,100,50,50) MovP(" P0",100,50,50) MovP("P1", "PASS",1000,500,500) MovP("P1"+Z(10)+RZ(20)) MovP("P1"+X(15)+RX(10),SPD(200)+ACC(200)+PASS())	

MovPR	**(All)**
Usage	Use relative distance for PTP motion
Syntax	MovPR(a,b) MovPR(a,b,c)

	MovPR(a,b,c,d)			
Input	Parameter	Type	Name	Description
	a	number	Moving distance	Input a positive value to move in the positive direction. Input a negative value to move in the negative direction. When moving in the X, Y and Z coordinate directions, the unit is millimeter. When moving in the RX, RY and RZ coordinate directions, the unit is degree.
	b	string	Moving direction	"X": X coordinates direction "Y": Y coordinates direction "Z": Z coordinates direction "RX": RX coordinates direction "RY": RY coordinates direction "RZ": RZ coordinates direction
	c	number	Maximum speed	Unit is percentage; allowed input range 1–100
	d	string	Coordinates	"TOOL": movement relative to TOOL "USER": movement relative to USER "WORLD": movement relative to WORLD
Output	Parameter	Type	Name	Description
	None			
Example	MovPR(10,"X") MovPR(-10,"X") MovPR(-10,"Z") MovPR(10,"RX") MovPR(-10,"RZ") MovPR(10,"RX","TOOL") MovPR(10,"RX",10,"TOOL")			

<table>
<tr><th colspan="4">MovL</th><th>(All)</th></tr>
<tr><td>Usage</td><td colspan="4">Specify target point for linear interpolation motion</td></tr>
<tr><td>Syntax</td><td colspan="4">MovL(a)
MovL(a,c)
MovL(a,b)
MovL(a,b,c)
MovL(a,c,d,e)
MovL(a,b,c,d,e)</td></tr>
<tr><td>Input</td><td>Parameter</td><td>Type</td><td>Name</td><td>Description</td></tr>
<tr><td></td><td>a</td><td>number or string</td><td>Target point</td><td>Input point number or point name</td></tr>
<tr><td></td><td>b</td><td>string</td><td>PASS mode</td><td>Input “PASS” to switch the motion mode to PASS mode<table><tr><td>No PASS</td><td>PASS</td></tr><tr><td>When PASS keyword is not added, the robot fully reaches the P2 position and then moves to P3.</td><td>When PASS keyword is added, the robot moves towards P3 before reaching the P2 position; this function can help skip points that are not important positions.</td></tr><tr><td> 
</td><td> 
</td></tr></table></td></tr>
<tr><td></td><td>c</td><td>number</td><td>Maximum speed</td><td>Unit millimeter/second; allowed input range is 1-2000</td></tr>
<tr><td></td><td>d</td><td>number</td><td>Acceleration</td><td>Unit millimeter/second; allowed input range is 1-25000</td></tr>
</table>

	e	number	Deceleration	Unit millimeter/second; allowed input range is 1–25000
New syntax	MovL(a) -- See Chapter 13 “2.0 Revision Description” MovL(a,b)			
New input	Parameter	Type	Name	Description
	a	Number + function or string + function	Target point related information setting	Input point number or point name plus the setting function; can also input setting function. Available setting functions include: X(a1), Y(a2), Z(a3), RX(a4), RY(a5) and RZ(a6). a1: the amount of change in X; unit is millimeter. a2: the amount of change in Y; unit is millimeter. a3: the amount of change in Z; unit is millimeter. a4: the amount of change in RX; unit is degree. Use for six-axis robots. a5: the amount of change in RY; unit is degree; use for robots with five or more axes. a6: value for the amount of change

				in RZ; unit is degree.
	b	function	Speed related information setting and PASS setting	Can input setting function directly; uses the default values for unset parameters. Available setting functions include: SPD(b1), ACC(b2), DEC(b3) and PASS(b4). b1: maximum speed; unit millimeter/second; allowed input range is 1–2000. b2: acceleration; unit millimeter/second; allowed input range is 1–25000. b3: deceleration; unit millimeter/second; allowed input range is 1–25000. b4: does not need parameter input; it is set to PASS mode when the function is called.
Output	Parameter	Type	Name	Description
	Non			

	e
Example	MovL(1) MovL(2, "PASS") MovL(3,100,50,50) MovL("P0",100,50,50) MovL("P1", "PASS",1000,500,500) MovL("P1"+Z(10)+RZ(20)) MovL("P1"+X(15)+RX(10),SPD(200)+ACC(200)+PASS())

MovLR				**(All)**
Usage	Use relative distance for linear interpolation motion			
Syntax	MovLR(a,b) MovLR(a,b,c) MovLR(a,b,c,d)			
Input	Parameter	Type	Name	Description
	a	number	Moving distance	Input a positive value to move in the positive direction. Input a negative value to move in the negative direction. When moving in the X, Y and Z coordinate directions, the unit is millimeter. When moving in the RX, RY and RZ coordinate directions, the unit is degree.
	b	string	Moving direction	"X": X coordinates direction "Y": Y coordinates direction "Z": Z coordinates direction "RX": RX coordinates direction "RY": RY coordinates direction "RZ": RZ coordinates direction
	c	number	Maximum speed	Unit is percentage; allowed input range 1-100
	d	string	Coordinates	"TOOL": movement relative to TOOL "USER": movement relative to

				USER "WORLD": movement relative to WORLD
Output	Parameter	Type	Name	Description
	None			
Example	MovLR(10,"X") MovLR(-10,"X") MovLR(-10,"Z") MovLR(10,"RX") MovLR(-10,"RZ") MovLR(10,"RX","TOOL") MovLR(10,"RX",10,"TOOL")			

MArchP				**(All)**
Usage	Performs arch motion along the Z-axis in PTP motion mode			
Syntax	MArchP(a,b,c,d) MArchP(a,b,c,d,e) MArchP(a,b,c,d,e,f,g)			
Input	Parameter	Type	Name	Description
	a	number or string	Target point	Input point number or point name
	b	number	Z-axis vertex height	Absolute position; unit is millimeter
	c	number	Z-axis rise height	Maximum safety height; it is a relative position, unit: millimeter
	d	number	Z-axis fall height	Minimum safety height; it is a relative position, unit: millimeter
	e	number	Maximum speed	Unit is percentage; allowed input range 1–100
	f	number	Acceleration	Unit is percentage; allowed input range 1–100
	g	number	Deceleration	Unit is percentage; allowed input range 1-100

New syntax	MArchP(a,b,c,d) -- See Chapter 13 “2.0 Revision Description” MArchP(a,b,c,d,e)			
New input	Parameter	Type	Name	Description
	a	number+function or string+function	Target point related information setting	Input point number or point name plus setting function; can also input setting function directly. Available setting functions include: X(a1), Y(a2), Z(a3), RX(a4), RY(a5) and RZ(a6). a1: the amount of change in X; unit is millimeter. a2: the amount of change in Y; unit is millimeter. a3: the amount of change in Z; unit is millimeter. a4: the amount of change in RX; unit is degree; use for six-axis robots. a5: the amount of change in

				RY; unit is degree; use for robots with five or more axes. a6: the amount of change in RZ; unit is degree.
	b	number	Z-axis vertex height	The absolute position; unit is millimeter
	c	number	Z-axis rise height	Maximum safety height; it is a relative position, unit: millimeter
	d	number	Z-axis fall height	Minimum safety height; it is a relative position, unit: millimeter
	e	function	Speed related information setting and PASS setting	Can input the setting function directly; uses the default values for unset parameters. Available setting functions include: SPD(e1), ACC(e2), DEC(e3) and PASS(e4). e1: maximum speed %;

				allowed input range 1–100. e2: acceleration %; allowed input range 1-100. e3: deceleration %; allowed input range 1–100. e4: does not need parameter input; it is set to PASS mode when the function is called.
Output	Parameter	Type	Name	Description
	None			
Example	MArchP("P1",0,50,40) MArchP("P2",-10,50,40,10,5,5) MArchP ("P1"+X(10),-20,30,20, SPD(30))			

MArchL				**(All)**
Usage	Perform arch motion along the Z-axis in linear interpolation motion mode			
Syntax	MArchL(a,b,c,d) MArchL(a,b,c,d,e) MArchL(a,b,c,d,e,f,g)			
Input	Parameter	Type	Name	Description
	a	number or string	Target point	Input point number or point name
	b	number	Z-axis vertex height	The absolute position; unit is millimeter

	c	number	Z-axis rise height	Maximum safety height; it is a relative position, unit: millimeter
	d	number	Z-axis fall height	Minimum safety height; it is a relative position, unit: millimeter
	e	number	Maximum speed	Unit millimeter/second; allowed input range is 1–2000
	f	number	Acceleration	Unit millimeter/second; allowed input range is 1–25000
	g	number	Deceleration	Unit millimeter/second; allowed input range is 1–25000
New syntax	MArchL(a,b,c,d) -- See Chapter 13 “2.0 Revision Description” MArchL(a,b,c,d,e)			
New input	Parameter	Type	Name	Description
	a	number+function or string+function	Target point related information setting	Can input point number or point name plus setting function; can also input setting function. Available setting functions include: X(a1), Y(a2), Z(a3), RX(a4), RY(a5) and RZ(a6). a1: the amount of change in X; unit is millimeter. a2: the amount of change in Y; unit is millimeter. a3: the amount of change in Z; unit is millimeter. a4: the amount of change in RX; unit is degree;

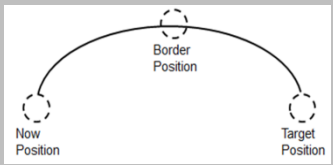
				use for six-axis robots. a5: the amount of change in RY; unit is degree; use for robots with five or more axes. a6: the amount of change in RZ; unit is degree.
	b	number	Z-axis vertex height	The absolute position; unit is millimeter
	c	number	Z-axis rise height	Maximum safety height; it is a relative position, unit: millimeter
	d	number	Z-axis fall height	Minimum safety height; it is a relative position, unit: millimeter
	e	function	Speed related information setting and PASS setting	Can input the setting function directly; uses the default values for unset parameters. Available setting functions include: SPD(e1), ACC(e2), DEC(e3) and PASS(e4). e1: maximum speed; unit millimeter/second; allowed input

				range is 1–2000. e2: acceleration; unit millimeter/second; allowed input range is 1–25000. e3: deceleration; unit millimeter/second; allowed input range is 1–25000. e4: does not need parameter input; it is set to PASS mode when the function is called.
Output	Parameter	Type	Name	Description
	None			
Example	MArchL("P1",0,50,40) MArchL("P2",-10,50,40,10,5,5) MArchL ("P1"+X(10),-20,30,20, SPD(30))			

MArc				**(All)**
Usage	Specify target position and points for arc interpolation arc motion			
Syntax	MArc(a,b,c) MArc(a,b,c,e) MArc(a,b,c,e,f,g) MArc(a,b,c,d) MArc(a,b,c,d,e) MArc(a,b,c,d,e,f,g)			
Input	Parameter	Type	Name	Description
	a	number or	Passing	Input point number or point name

		string	point	
	b	number or string r	Target point	Input Point number or point name
	c	string	Function setting	Input “BORDER” to not switch to three-point circle function; this mode is not limited to the X-Y plane motion. Input “BORDER_TAN_FOR” for tangent three-point circle function; this mode is limited to the X-Y plane motion. Input “BORDER_TAN_REV” for reverse tangent three-point circle function; this mode is limited to X-Y plane motion. Input “BORDER_CENTRIPETAL” for centripetal three-point circle function; this mode is limited to the X-Y plane motion. Input “BORDER_CENTRIFUGAL” for centrifugal three-point circle function; this mode is limited to the X-Y plane motion.
	d	string	PASS mode	Input is “PASS”
	e	number	Maximum speed	Unit millimeter/second; allowed input range is 1–2000
	f	number	Acceleration	Unit millimeter/second; allowed input range is 1–25000
	g	number	Deceleration	Unit millimeter/second; allowed input range is 1–25000
New syntax	MArc (a,b,c) -- See Chapter 13 “2.0 Revision Description” MArc (a,b,c,d)			
New input	Parameter	Type	Name	Description
	a	number+function or string+function	Passing point related information setting	Can input point number or point name plus setting function; can also input setting function directly. Available setting functions include: X(a1), Y(a2), Z(a3), RX(a4), RY(a5) and RZ(a6).

				a1: the amount of change in X; unit is millimeter. a2: the amount of change in Y; unit is millimeter. a3: the amount of change in Z; unit is millimeter. a4: the amount of change in RX; unit is degree; use for six-axis robots. a5: the amount of change in RY; unit is degree; use for robots with five or more axes. a6: value for the amount of change in RZ; unit is degree.
	b	number+function or string+function	Target point related information setting	Input point number or point name plus setting function; can also input setting function directly. Available setting functions include: X(b1), Y(b2), Z(b3), RX(b4), RY(b5) and RZ(b6). b1: the amount of change in X; unit is millimeter. b2: the amount of change in Y; unit is millimeter. b3: the amount of change in Z; unit is millimeter.

				b4: the amount of change in RX; unit is degree. Use for six-axis robots. b5: the amount of change in RY; unit is degree. Use for robots with five or more axes. b6: value for the amount of change in RZ; unit is degree.
	c	string	Function setting	 Input “BORDER” to not switch to three-point circle function; this mode is not limited to the X-Y plane motion. Input “BORDER_TAN_FOR” for tangent three-point circle function; this mode is limited to the X-Y plane motion. Input “BORDER_TAN_REV” for reverse tangent three-point circle function; this mode is limited to X-Y plane motion. Input “BORDER_CENTRIPETAL” for centripetal three-point circle function; this mode is limited to the X-

				Y plane motion. Input “BORDER_CENTRIFUGAL” for centrifugal three-point circle function; this mode is limited to the X-Y plane motion.
	d	function	Speed related information setting and PASS setting	Can input the setting function directly; uses the default value for unset parameters. Available setting functions include: SPD(d1), ACC(d2), DEC(d3) and PASS(d4). d1: maximum speed; unit millimeter/second; allowed input range is 1–2000. d2: acceleration; unit millimeter/second; allowed input range is 1–25000. d3: deceleration; unit millimeter/second; allowed input range is 1–25000. d4: does not need parameter input; it is set to PASS mode when the function is called.
Output	Paramete r	Type	Name	Description
	None			
Example	MArc("P1","P2","BORDER") MArc ("P1","P2","BORDER","PASS") MArc ("P1","P2","BORDER",100)			

	MArc ("P1","P2","BORDER","PASS",100) MArc ("P1","P2","BORDER","PASS",100,100,100) MArc ("P1"+X(10), "P2"+Y(10), "BORDER",SPD(30)+ACC(40)+DEC(50)) MArc ("P1"+X(10), "P2"+Y(10), "BORDER", SPD(30)+PASS())

MCircle				**(All)**
Usage	Specify target position and points for arc interpolation circular motion; three-point circle			
Syntax	MCircle (a,b,c) MCircle (a,b,c,e) MCircle (a,b,c,e,f,g) MCircle (a,b,c,d) MCircle (a,b,c,d,e) MCircle (a,b,c,d,e,f,g)			
Input	Parameter	Type	Name	Description
	a	number or string	Passing point	Input point number or point name
	b	number or string	Target point	Input point number or point name
	c	string	Function setting	Input “BORDER” to not switch to three-point circle function; this mode is not limited to the X-Y plane motion. Input “BORDER_TAN_FOR” for tangent three-point circle function; this mode is limited to the X-Y plane motion. Input “BORDER_TAN_REV” for reverse tangent three-point circle function; this mode is limited to X-Y plane motion. Input “BORDER_CENTRIPETAL” for centripetal three-point circle function; this mode is limited to the X-Y plane motion. Input “BORDER_CENTRIFUGAL” for centrifugal three-point circle function; this mode is limited to the X-Y plane motion.
	d	string	PASS mode	Input is “PASS”

	e	number	Maximum speed	Unit millimeter/second; allowed input range is 1–2000
	f	number	Acceleration	Unit millimeter/second; allowed input range is 1–25000
	g	number	Deceleration	Unit millimeter/second; allowed input range is 1–25000
New syntax	MCircle (a,b,c) -- See Chapter 13 “2.0 Revision Description” MCircle (a,b,c,d)			
New input	Parameter	Type	Name	Description
	a	number+function or string+function	Passing point related information setting	Input point number or point name plus setting function; can also input setting function directly. Available setting functions include: X(a1), Y(a2), Z(a3), RX(a4), RY(a5) and RZ(a6). a1: the amount of change in X; unit is millimeter. a2: the amount of change in Y; unit is millimeter. a3: the amount of change in Z; unit is millimeter. a4: the amount of change in RX; unit is degree; use for six-axis robots. a5: value for the amount of change in RY; unit is degree; use for robots with five or more axes

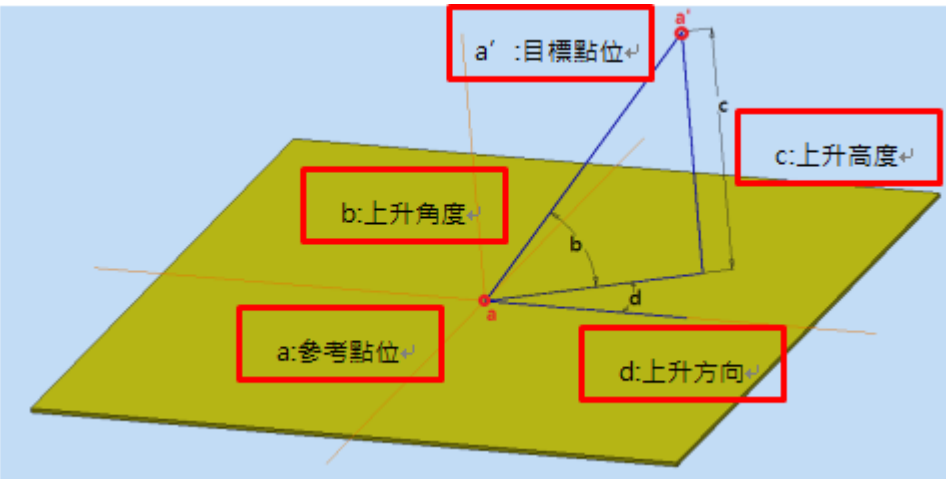
				a6: the amount of change in RZ; unit is degree.
	b	number+function or string+function	Target point related information setting	Input Point number or point name plus setting function; can also input setting function directly. Available setting functions include: X(b1), Y(b2), Z(b3), RX(b4), RY(b5) and RZ(b6) b1: the amount of change in X; unit is millimeter. b2: the amount of change in Y; unit is millimeter. b3: the amount of change in Z; unit is millimeter. b4: the amount of change in RX; unit is degree; use for six-axis robots. b5: the amount of change in RY; unit is degree; use for robots with five or more axes. b6: the amount of change in RZ; unit is degree.
	c	string	Function setting	Input “BORDER” to not switch to three-point circle function;

				this mode is not limited to the X-Y plane motion. Input "BORDER_TAN_FOR" for tangent three-point circle function; this mode is limited to the X-Y plane motion. Input "BORDER_TAN_REV" for reverse tangent three-point circle function; this mode is limited to X-Y plane motion. Input "BORDER_CENTRIPETAL" for centripetal three-point circle function; this mode is limited to the X-Y plane motion. Input "BORDER_CENTRIFUGAL" for centrifugal three-point circle function; this mode is limited to the X-Y plane motion.
	d	function	Speed related information setting and PASS setting	Can input the setting function directly; uses the default value for unset parameters. Available setting functions include: SPD(d1), ACC(d2),

		DEC(d3) and PASS(d4). d1: maximum speed; unit millimeter/second; allowed input range is 1–2000. d2: acceleration; unit millimeter/second; allowed input range is 1–25000. d3: deceleration; unit millimeter/second; allowed input range is 1–25000. d4: does not need parameter input; it is set to PASS mode when the function is called.
Output	Paramete r Type Name Description	
	None	
Example	MCircle ("P1","P2","BORDER") MCircle ("P1","P2","BORDER","PASS") MCircle ("P1","P2","BORDER",100) MCircle ("P1","P2","BORDER","PASS",100) MCircle ("P1","P2","BORDER","PASS",100,100,100) MCircle ("P1"+X(10), "P2"+Y(10), "BORDER",SPD(30)+ACC(40)+DEC(50)) MCircle ("P1"+X(10), "P2"+Y(10), "BORDER", SPD(30)+PASS())	

Lift	**(All)**
Usage	Use absolute coordinate to move to the relative reference point position
Syntax	Lift(a,b,c,d) Lift(a,b,c,d,e)

Lift(a,b,c,d,e,f,g)



Input	Parameter	Type	Name	Description
	a	number or string	Reference point	Input point number or point name
	b	number	Rise angle	Rise angle; unit is degree
	c	number	Rise height	Rise height; unit is millimeter
	d	number	Rise direction	Rise direction; unit is degree
	e	number	Maximum speed	Unit millimeter/second; allowed input range is 1–2000
	f	number	Acceleration	Unit millimeter/second; allowed input range is 1–25000
	g	number	Deceleration	Unit millimeter/second; allowed input range is 1–25000
New syntax	Lift(a,b,c,d) -- See Chapter 13 “2.0 Revision Description” Lift(a,b,c,d,e)			
New input	Parameter	Type	Name	Description
	a	number+function or string+function	Reference point related information setting	Input point number or point name plus setting function; also can input setting

		function directly. Available setting functions include: X(a1), Y(a2), Z(a3), RX(a4), RY(a5) and RZ(a6). a1: the amount of change in X; unit is millimeter. a2: the amount of change in Y; unit is millimeter. a3: the amount of change in Z; unit is millimeter. a4: the amount of change in RX; unit is degree; use for six-axis robots. a5: the amount of change in

				RY; unit is degree; use for robots with five or more axes. a6: the amount of change in RZ; unit is degree.
	b	number	Rise angle	Rise angle; unit is degree
	c	number	Rise height	Rise height; unit is millimeter
	d	number	Rise direction	Rise direction; unit is degree
	e	function	Speed related information setting and PASS setting	Can input setting function directly; uses the default values for unset parameters. Available setting functions include: SPD(e1), ACC(e2), DEC(e3)

				and PASS(e4). e1: maximum speed; unit millimeter/second; allowed input range is 1–2000. e2: acceleration; unit millimeter/second; allowed input range is 1–25000. e3: deceleration; unit millimeter/second; allowed input range is 1–25000. e4: does not need parameter input; it is set to PASS mode when the function is called.
Output	Parameter	Type	Name	Description

	None
Example	Lift("P0",45,10,90) Lift("P1"+Z(10)+RZ(20),45,10,90) Lift("P1"+X(15),45,10,90,SPD(200)+ACC(200)

MotionStop				**(All)**
Usage	Stop robot motion			
Syntax	MotionStop ()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	None			
Example	MovP(1001,"PASS") MotionStop() -- Read the current MovP motion speed and decelerate to stop MovL(1001,"PASS") MotionStop() -- Read the current MovL motion speed and decelerate to stop			

ExtMotionStop				**(All)**
Usage	Stop external axis motion			
Syntax	ExtMotionStop (a)			
Input	Parameter	Type	Name	Description
	a	number	External motor axis number	External motor axis number
Output	Parameter	Type	Name	Description
	None			
Example	ExtMovJ(1,300000, "PASS") DELAY(1) ExtMotionStop (1) -- Stop motion after one second after the external axis has moved			

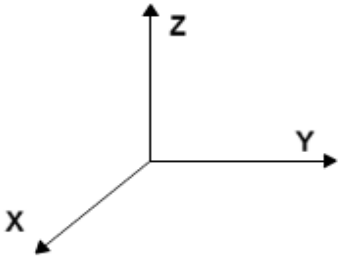
ContinueCartesianJOG		**(All)**
Usage	Use linear interpolation to move continuously in a specific direction	
Syntax	ContinueCartesianJOG(a)	

	ContinueCartesianJOG(a,b)			
Input	Parameter	Type	Name	Description
	a	string	Select moving direction	Input "X+" to move in the positive X direction. Input "X-" to move in the negative X direction. Input "Y+" to move in the positive Y direction. Input "Y-" to move in the negative Y direction. Input "Z+" to move in the positive Z direction. Input "Z-" to move in the negative Z direction.
	b	number	Maximum speed	Unit millimeter/second; allowed input range is 1–2000
Output	Parameter	Type	Name	Description
	None			
Example	ContinueCartesianJOG("X+") DELAY(2) ContinueCartesianJOG("Y+",500) DELAY(2)			

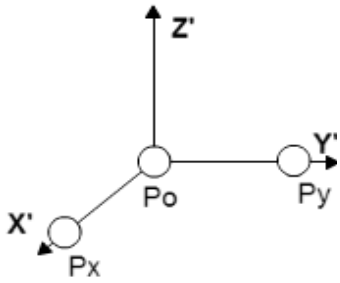
8. Coordinate instructions

SetUF	**(All)**
Usage	Set user coordinates
Syntax	SetUF(a,b,c,d)

SetUF(a,b,c,d,e)



World Coordinate



User Coordinate

Input	Parameter	Type	Name	Description
	a	number	Coordinate index	Provides nine sets of user coordinates for use; allowed input range is 1–9
	b	number or string	User coordinate origin	Set the origin coordinate direction position point of the user coordinates (point Po in the figure above)
	c	number or string	X coordinate direction position point of the user coordinates	Set the X coordinate direction position point of the user coordinates (point Px in the figure above)
	d	number or string	Y coordinate direction position point of the user coordinates	Set the Y coordinate direction position point of the user coordinates (point Py in the figure above)
	e	number	User coordinate types	Set user coordinate type; there are four modes ranging from 0–3. The default is mode 0. 0: Orthogonal no incline mode 1: Orthogonal incline mode 2: Non-orthogonal no incline 3: Non-orthogonal incline
Output	Parameter	Type	Name	Description

	None
Example	SetUF(1,"P0","P1","P2")

SetTF				**(All)**
Usage	Set tool coordinates			
Syntax	SetTF(a,b,c,d,e,f,g)			
Input	Parameter	Type	Name	Description
	a	number	Coordinate index	Provides nine sets of tool coordinates for use; allowed input range is 1–9
	b	number or string	Tool width	Set tool width; unit is millimeter When input is “PASS” or no input, skip input width
	c	number or string	Tool height	Set tool height; unit is millimeter When input is “PASS” or no input, skip input height
	d	number or string	Tool angle	Set tool angle; unit is degree When input is “PASS” or no input, skip input angle
	e	number or string	Tool pitch	Set tool pitch; unit is degree When input is “PASS” or no input, skip input pitch
	f	number or string	Tool roll	Set tool roll; unit is degree When input is “PASS” or no input, skip input roll
	g	number or string	Tool yaw	Set tool yaw; unit is degree When input is “PASS” or no input, skip input yaw
Output	Parameter	Type	Name	Description
	None			
Example	SetTF(1,10,20,30) SetTF(1,10,"PASS",30,40,"PASS",60)			

ChangeUF		**(All)**
Usage	Switch user coordinates	
Syntax	ChangeUF(a)	

Input	Parameter	Type	Name	Description
	a	number	Coordinate index	User coordinates: 1–9. Earth coordinate: 0
Output	Parameter	Type	Name	Description
	None			
Example	ChangeUF(1) ChangeUF(0)			

ChangeTF				**(All)**
Usage	Switch tool coordinates			
Syntax	ChangeTF(a)			
Input	Parameter	Type	Name	Description
	a	number	Coordinate index	Tool coordinates: 1–9. Earth coordinate: 0
Output	Parameter	Type	Name	Description
	None			
Example	ChangeTF(1) ChangeTF(0)			

GetTF				**(All)**
Usage	Get the width, height, angle, pitch, roll and yaw setting values for the tool coordinates			
Syntax	ret1,ret2,ret3,ret4,ret5,ret6 = GetTF(a)			
Input	Parameter	Type	Name	Description
	a	number	Coordinate index	Tool coordinates: 1–9. Earth coordinate: 0
Output	Parameter	Type	Name	Description
	ret1	number	Tool width	Tool width value
	ret2	number	Tool height	Tool height value
	ret3	number	Tool angle	Tool angle value
	ret4	number	Tool pitch	Tool pitch value
	ret5	number	Tool roll	Tool roll value
	ret6	number	Tool yaw	Tool yaw value
Example	Width,Height,Angle,Pitch,Roll,Yaw = GetTF(1)			

9. Process control instructions

if…then…elseif…then…else…end				**(All)**
Usage	If-then-else-elseif conditional statement: use for conditional execution of a program code section			

Syntax	if a then
	Execute program one
	elseif b then
	Execute program two
	else
	Execute program three
	end			

Input	Parameter	Type	Name	Description
	a	boolean	Determine condition one	Execute when evaluates to true
	b	boolean	Determine condition two	Execute when a evaluates to false and b evaluates to true
Output	Parameter	Type	Name	Description
	None			

Example	if DI(1) == "ON" then
	  MovP("P1")
	elseif DI(2) == "ON" then
	  MovL("P2")
	else
	  MovP(3)
	end			

Break	**(All)**
Usage	Use the break instruction to exit loops (while…do…end, for…do…end, repeat…until). See the description and example for “while…do…end” loop.

while…do…end	**(All)**
Usage	while loop: Use to repeatedly execute a program code section. Use the break instruction to exit the loop. You can also use the break instruction to exit other types of loops.

Syntax	while a do Loop execution program end			
Input	Parameter	Type	Name	Description
	a	boolean	Determining condition	If condition evaluates to false end loop
Output	Parameter	Type	Name	Description
	None			

Example	i = 1
	while true do
	  i = i + 1
	  if i==100 then
	  break
	  end
	End			

for…do…end				**(All)**
Usage	for loop: Use to repeatedly execute a program code section.			
Syntax	for a=b,c do Loop execution program end			
Input	Parameter	Type	Name	Description
	a		Loop variable	Set loop variable
	b	number	Loop variable initial value	Set loop variable initial value
	c	number	Determining condition	Perform last iteration of loop when b=c, then end loop;
Output	Parameter	Type	Name	Description
	None			
Example	a = {5, 4, 3, 2, 1} i = 1 sum = 0 for i = 1, 5 do sum = sum + a[i] end			

repeat…until				**(All)**
Usage	repeat loop: Use to repeatedly execute a program code section. Note: when you use the until keyword, you must increment a counter in the loop			
Syntax	repeat Loop execution program until a			
Input	Parameter	Type	Name	Description
	a	boolean	Determining condition	End loop when a evaluates to true
Output	Parameter	Type	Name	Description
	None			

Example	a = {5, 4, 3, 2, 1}
	i = 1
	sum = 0
	repeat
	  sum = sum + a[i] -- sum = 15
	  i = i + 1
	until i > #a -- #a is the table size			

function…end				**(All)**
Usage	User defined sub-function: When using a sub-function, the sub-function definition must be declared before the first time the sub-function is called in the program			
Syntax	function a() Execute program end			
Input	Parameter	Type	Name	Description
	a		Sub-function name	Must be English letter or number
Output	Parameter	Type	Name	Description
	None			

Example	function MyFunc1()
	  MovP(1)
	  MovP("P2")
	end
	MovL(3)			

	MyFunc1()

10. Input/output instructions

DI				**(All)**
Usage	Return digital input status; return “ON” or “OFF”			
Syntax	ret = DI(a)			
Input	Parameter	Type	Name	Description
	a	number	Pin number	Allowed input range is 1–24
Output	Parameter	Type	Name	Description
	ret	string	DI status	Return “ON” or “OFF” status
Example	if DI(1)== "ON" then MovL("P1") end			

DO				**(All)**
Usage	Read or set digital output status			
Syntax	ret = DO(a) · Read digital output DO(a,b) · Set digital output DO(a,b,c) · Set digital output			
Input	Parameter	Type	Name	Description
	a	number	Pin number	Allowed input range is 1–12
	b	string	Set DO status	Allowed input “ON” or “OFF”
	c	number	Delay time	After the delay time, switch to status opposite to the value set in b; unit is second
Output	Parameter	Type	Name	Description
	ret	string	DO status	Return “ON” or “OFF” status

Example	if DO(1) == "ON" then
	DO(1,"OFF") --Set DO for Pin 1 to Off
	end
	if DO(1) == "OFF" then
	DO(1,"ON") --Set DO for Pin 1 to On
	end
	DO(1,"ON",1) --After one second, switch DO for Pin 1 to "OFF"			

ExtDI				**(All)**
Usage	Read external digital input status			
Syntax	ret = ExtDI(a,b)			
Input	Parameter	Type	Name	Description
	a	number	External station number	Input external station number
	b	number	Pin number	Input pin number
Output	Parameter	Type	Name	Description
	ret	string	External DI status	Return “ON” or “OFF” status
Example	if ExtDI(3,1)== "ON" then MovL("P1") end			

ExtDO				**(All)**
Usage	Read or set external digital output status			
Syntax	ret = ExtDO(a,b) --Read external digital output ExtDO(a,b,c) --Write external digital output ExtDO(a,b,c,d) --Write external digital output			
Input	Parameter	Type	Name	Description
	a	number	External station number	Input external station number
	b	number	Pin number	Input pin number
	c	string	Set external DO status	Allowed input “ON” or “OFF”
	d	number	Delay time	After the delay time, switch to status opposite to the value set in c; unit is second
Output	Parameter	Type	Name	Description
	ret	string	External DO status	Return “ON” or “OFF” status

Example	if ExtDO(3,1) == "ON" then
	ExtDO(3,1,"OFF")
	end
	if ExtDO(3,1) == "OFF" then
	ExtDO(3,1,"ON") end			

	ExtDO(3,1,"ON",1) --After one second, switch external station 3 DO for Pin 1 to "OFF"

ReadModbus				**(All)**
Usage	Read the value at a Modbus memory address			
Syntax	ret = ReadModbus(a,b)			
Input	Parameter	Type	Name	Description
	a	number	Modbus address	Input Modbus address to read
	b	string	Data length mode	Input the data length to read; the input value is “W” or “DW” (data length is Word or DWord)
Output	Parameter	Type	Name	Description
	ret	number	Value read from Modbus address	Return the value read from the Modbus address

Example	WriteModbus(0x1000,"W",1)
	readModbus_0x1000=ReadModbus(0x1000,"W")
	if readModbus_0x1000 == 1 then
	  WriteModbus(0x1F00,"DW",2)
	  DELAY(0.1)
	end
	readModbus_0x1F00=ReadModbus(0x1F00,"DW")			

WriteModbus				**(All)**
Usage	Write a value to a Modbus memory address			
Syntax	WriteModbus(a,b,c)			
Input	Parameter	Type	Name	Description
	a	number	Modbus address	Input Modbus address to write
	b	string	Data length mode	Input the data length to write; the input value is “W” or “DW” (data length is Word or DWord)
	c	number	Value to write at the Modbus address	Input the Modbus value to write at the Modbus address
Output	Parameter	Type	Name	Description
	None			
Example	WriteModbus(0x1000,"W",1)			

```
readModbus_0x1000=ReadModbus(0x1000,"W")
if readModbus_0x1000 == 1 then
   WriteModbus(0x1F00,"DW",2)
   DELAY(0.1)
end
readModbus_0x1F00=ReadModbus(0x1F00,"DW")
```

ConnectMBC				**(All)**
Usage	Connect to Modbus Client			
Syntax	ret = ConnectMBC(a) ret = ConnectMBC(a,b) ret = ConnectMBC(a,b,c)			
Input	Parameter	Type	Name	Description
	a	string	Client terminal IP	Input the client terminal IP address
	b	number	Port	Input the port number; when not set, the default value is 502
	c	number	Client group	Input the connection group; when not set, the default value is 1
Output	Parameter	Type	Name	Description
	ret	number	Whether or not it is connected successfully	Return connection status: ret is 0 if successful ret is -1 if fails
Example	valid1 = ConnectMBC("192.168.1.33") valid2 = SetSlaverMBC(3) valid3 = SetTimeOutMBC (10) valid4= WriteMBC (10, "W",{11,22,33}) valid5,Data1 = ReadMBC (10, "W", 3) D11 = Data1[1] D12 = Data1[2] D13 = Data1[3] CloseMBC() valid6= ConnectMBC("192.168.1.99",502,2) valid7 = SetSlaverMBC(4,2)			

valid8 = SetTimeOutMBC (20,2)

valid9 = WriteMBC (10, "DW",{11,22,33},2)

valid10,Data2= ReadMBC (10, "DW", 3,2)

D21 = Data2[1]

D22 = Data2[2]

D23 = Data2[3]

CloseMBC(2)

CloseMBC				**(All)**
Usage	Close Modbus Client connection			
Syntax	CloseMBC(a)			
Input	Parameter	Type	Name	Description
	a	number	Client group	Group to close connection; when not set, the default value is 1
Output	Parameter	Type	Name	Description
	None			

Example

valid1 = ConnectMBC("192.168.1.33")

valid2 = SetSlaverMBC(3)

valid3 = SetTimeOutMBC (10)

valid4= WriteMBC (10, "W",{11,22,33})

valid5,Data1 = ReadMBC (10, "W", 3)

D11 = Data1[1]

D12 = Data1[2]

D13 = Data1[3]

CloseMBC()

valid6= ConnectMBC("192.168.1.99",502,2)

valid7 = SetSlaverMBC(4,2)

valid8 = SetTimeOutMBC (20,2)

valid9 = WriteMBC (10, "DW",{11,22,33},2)

valid10,Data2= ReadMBC (10, "DW", 3,2)

D21 = Data2[1]

D22 = Data2[2]

D23 = Data2[3]

CloseMBC(2)

<table>
<tr><th colspan="4">SetSlaverMBC</th><th>(All)</th></tr>
<tr><td>Usage</td><td colspan="4">Set Modbus Client station number</td></tr>
<tr><td>Syntax</td><td colspan="4">ret = SetSlaverMBC (a)
ret = SetSlaverMBC (a,b)</td></tr>
<tr><td>Input</td><td>Parameter</td><td>Type</td><td>Name</td><td>Description</td></tr>
<tr><td></td><td>a</td><td>number</td><td>Slave number</td><td>The Modbus client slave number</td></tr>
<tr><td></td><td>b</td><td>number</td><td>Client group</td><td>The slave number group; when not set, the default value is 1</td></tr>
<tr><td>Output</td><td>Parameter</td><td>Type</td><td>Name</td><td>Description</td></tr>
<tr><td></td><td>ret</td><td>number</td><td>Whether or not the slave was set successfully</td><td>Return the slave number setting status:
ret is 0 if successful
ret is -1 if fails</td></tr>
<tr><td>Example</td><td colspan="4">valid1 = ConnectMBC("192.168.1.33")
valid2 = SetSlaverMBC(3)
valid3 = SetTimeOutMBC (10)
valid1 = ConnectMBC("192.168.1.33")
valid2 = SetSlaverMBC(3)
valid3 = SetTimeOutMBC (10)
valid4= WriteMBC (10, "W",{11,22,33})
valid5,Data1 = ReadMBC (10, "W", 3)
D11 = Data1[1]
D12 = Data1[2]
D13 = Data1[3]
CloseMBC()

valid6= ConnectMBC("192.168.1.99",502,2)
valid7 = SetSlaverMBC(4,2)
valid8 = SetTimeOutMBC (20,2)
valid9 = WriteMBC (10, "DW",{11,22,33},2)
valid10,Data2= ReadMBC (10, "DW", 3,2)
D21 = Data2[1]
D22 = Data2[2]
D23 = Data2[3]
CloseMBC(2)</td></tr>
</table>

SetTimeOutMBC				**(All)**
Usage	Set connection timeout time			
Syntax	ret = SetTimeOutMBC (a) ret = SetTimeOutMBC (a,b)			
Input	Parameter	Type	Name	Description
	a	number	Set timeout time	Timeout time; unit is milliseconds
	b	number	Client group	Timeout group; when not set, the default value is 1
Output	Parameter	Type	Name	Description
	ret	number	Whether or not the timeout was set successfully	Return the timeout setting status: ret is 0 if successful ret is -1 if fails
Example	valid1 = ConnectMBC("192.168.1.33") valid2 = SetSlaverMBC(3) valid3 = SetTimeOutMBC (10) valid4= WriteMBC (10, "W",{11,22,33}) valid5,Data1 = ReadMBC (10, "W", 3) D11 = Data1[1] D12 = Data1[2] D13 = Data1[3] CloseMBC() valid6= ConnectMBC("192.168.1.99",502,2) valid7 = SetSlaverMBC(4,2) valid8 = SetTimeOutMBC (20,2) valid9 = WriteMBC (10, "DW",{11,22,33},2) valid10,Data2= ReadMBC (10, "DW", 3,2) D21 = Data2[1] D22 = Data2[2] D23 = Data2[3] CloseMBC(2)			

WriteMBC	**(All)**

Usage	Write values to a Modbus Client address			
Syntax	ret = WriteMBC(a,b,c) ret = WriteMBC(a,b,c,d)			
Input	Parameter	Type	Name	Description
	a	number	Modbus Client address	Input Modbus Client address to write
	b	string	Data length mode	Input = the data length to write; the input value is “W” or “DW” (data length is WORD or DWORD).
	c	table	Write Modbus Client address value	Input the value to write to the Modbus Client address; it is in table form
	d	number	Client group	Set the client group for the value to write; when not set, the default value is 1
Output	Parameter	Type	Name	Description
	ret	number	Whether or not it is written successfully	Return writing status: ret is the number of values written into the table if successful ret is -1 if fails
Example	valid1 = ConnectMBC("192.168.1.33") valid2 = SetSlaverMBC(3) valid3 = SetTimeOutMBC (10) valid4= WriteMBC (10, "W",{11,22,33}) valid5,Data1 = ReadMBC (10, "W", 3) D11 = Data1[1] D12 = Data1[2] D13 = Data1[3] CloseMBC() valid6= ConnectMBC("192.168.1.99",502,2) valid7 = SetSlaverMBC(4,2) valid8 = SetTimeOutMBC (20,2) valid9 = WriteMBC (10, "DW",{11,22,33},2) valid10,Data2= ReadMBC (10, "DW", 3,2)			

	D21 = Data2[1] D22 = Data2[2] D23 = Data2[3] CloseMBC(2)

ReadMBC				**(All)**
Usage	Read values from a Modbus Client address			
Syntax	ret1,ret2 = ReadMBC(a,b,c) ret1,ret2 = ReadMBC(a,b,c,d)			
Input	Parameter	Type	Name	Description
	a	number	Modbus Client address	Input Modbus client address to read
	b	string	Data length mode	Input the data length to write; the input value is “W” or “DW” (data length is WORD or DWORD).
	c	number	Read quantity	Input the number of data entries to read
	d	number	Client group	Set the client group for the read value; when not set, the default value is 1
Output	Parameter	Type	Name	Description
	ret1	number	Whether or not it was read successfully	Return reading status: ret is the number of values read if successful ret is -1 if fails
	ret2	table	Read address value	The Modbus client table data that was read
Example	valid1 = ConnectMBC("192.168.1.33") valid2 = SetSlaverMBC(3) valid3 = SetTimeOutMBC (10) valid4= WriteMBC (10, "W",{11,22,33}) valid5,Data1 = ReadMBC (10, "W", 3) D11 = Data1[1] D12 = Data1[2] D13 = Data1[3] CloseMBC()			

```
valid6= ConnectMBC("192.168.1.99",502,2)
valid7 = SetSlaverMBC(4,2)
valid8 = SetTimeOutMBC (20,2)
valid9 = WriteMBC (10, "DW",{11,22,33},2)
valid10,Data2= ReadMBC (10, "DW", 3,2)
D21 = Data2[1]
D22 = Data2[2]
D23 = Data2[3]
CloseMBC(2)
```

11. Program Execution Instructions

QUIT				**(All)**
Usage	Stop executing the program			
Syntax	QUIT ()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	None			
Example	IOStatus=DI(1) DELAY(1) IF IOStatus–="ON" then QUIT() end			

PAUSE				
Usage	Pause the current execution of motion and program; Must use external software or equipment to restart the program to continue execution			
Syntax	PAUSE ()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	None			
Example	IOStatus=DI(1)			

```
    DELAY(1)
    IF IOStatus–="ON" then
    PAUSE ()
end
```

12. Application Function Instructions

SafetyMode				**(All)**
Usage	Switch the function pause mode			
Syntax	SafetyMode (a)			
Input	Parameter	Type	Name	Description
	a	number	Mode selection	There are five modes ranging from 1–5; the default status is mode 1. 1: Motion completes after function pause is triggered and IO maintains the current status; after triggering the restore system DI (system DI 3), it continues running. 2: Motion completes after function pause is triggered and IO returns to the OFF status; after triggering the restore system DI (system DI 3), it continues running. 3: Turn off the function pause function. 4: Motion stops after function pause is triggered and IO maintains the current status; after triggering the restore system DI (system DI 3), it continues running. 5: Motion stops after function pause is triggered and IO

				returns to the OFF status; after triggering the restore system DI (system DI 3), it continues running.
Output	Parameter	Type	Name	Description
	None			
Example	SafetyMode(4) MovP(1) SafetyMode(1) MovP(2)			

SafetyStatus				**(All)**
Usage	Read the triggering status of function pause			
Syntax	ret = SafetyStatus ()			
Input	Parameter	Type	Name	Description
	None			
Output	Parameter	Type	Name	Description
	ret	number	Function pause trigger status	Status read can be 0, 2 or 3 0: Function pause not triggered status, which is the normal operating status. 2: Function pause was triggered when the program was not running; the motor servo status is OFF. 3: Function pause was triggered while the program was running.
Example	if SafetyStatus(a)== 3 then PAUSE() end			

MultiTask				**(All)**
Usage	Multi-threading function for collaborative motion instructions			
Syntax	MultiTask(a,b,c,d,e,f)			
Input	Parameter	Type	Name	Description
	a	function	Thread one	Thread one functions can

				include motion related instructions: MovP, MovPR, MovL, MovLR, MovJ, MArc, MCircle, Lift, MArchP, MArchL, and ContinueCartesianJOG; can be switched to other threads while motion instruction is being executed; can also use other functions in this function parameter.
	b	function	Thread two	Thread two functions cannot include motion related instructions: MovP, MovPR, MovL, MovLR, MovJ, MArc, MCircle, Lift, MArchP, MArchL, ContinueCartesianJOG; can use other functions in this function parameter. No input means this thread is not turned on.
	c	function	Thread three	Thread three functions cannot include motion related instructions: MovP, MovPR, MovL, MovLR, MovJ, MArc, MCircle, Lift, MArchP, MArchL, ContinueCartesianJOG; can use other functions in this function parameter. No input means this thread is not turned on.
	d	function	Thread four	Thread four functions cannot include motion related instructions: MovP, MovPR, MovL, MovLR, MovJ, MArc, MCircle, Lift, MArchP, MArchL, ContinueCartesianJOG; can use other functions in this

				function parameter. No input means this thread is not turned on.
	e	function	Thread five	Thread five functions cannot include motion related instructions: MovP, MovPR, MovL, MovLR, MovJ, MArc, MCircle, Lift, MArchP, MArchL, ContinueCartesianJOG; can use other functions in this function parameter. No input means this thread is not turned on.
	f	function	Thread six	Thread six functions cannot include motion related instructions: MovP, MovPR, MovL, MovLR, MovJ, MArc, MCircle, Lift, MArchP, MArchL, ContinueCartesianJOG; can use other functions in this function parameter. No input means this thread is not turned on.
Output	Parameter	Type	Name	Description
	None			

Example

```
function Task1()
  MovL(1001)
  MovP(1002)
end

function Task2()
   DO(1,"ON")
   DO(1,"OFF")
end

function Task3()
   DO(2,"ON")
```

```
  DO(2,"OFF")
end

function Task4()
  DO(3,"ON")
  DO(3,"OFF")
end

function Task5()
  DO(4,"ON")
  DO(4,"OFF")
end

function Task6()
  DO(5,"ON")
  DO(5,"OFF")
end

MultiTask(Task1,Task2,Task3) -- Collaborative execution of three threads
MultiTask(Task1,Task2,Task3,Task4,Task5,Task6) -- Collaborative
execution of six threads
```

13.2.0 Revision Description

Explanation of the differences between the new and old syntaxes

Motion instructions that can use the new syntax

MovP, MovL, MArc, MCircle, MArchP, MArchL, Lift

2.0 syntax related setting functions

X(a1), Y(a2), Z(a3), RX(a4), RY(a5) and RZ(a6) are point setting related functions.

SPD(b1), ACC(b2), DEC(b3) and PASS(b4) are speed related and continuous motion setting functions.

- **a1**, **a2** and **a3** are amounts of change in position; unit is millimeter.
- **a4**, **a5** and **a6** are amounts of change in angle; unit is degree.
- **b1**, **b2** and **b3** are speed, acceleration and deceleration; the units are millimeter/second, millimeter/second squared and millimeter/second squared respectively. PASS() does not require input; the continuous motion function is enabled when this function is called.

Concepts for converting from 1.0 syntax to 2.0 syntax

MovP (a,b,c,d,e): In the old syntax **a** is the target point information and is a required input parameter, **b**, **c**, **d** and **e** are speed, acceleration, deceleration and PASS information, and are optional input parameters
In the new MovP syntax, the point setting related function can use “+” to set the point information and change the position of that point. Speed related and continuous motion setting functions can use “+” to perform related settings to other parameters. The default values are used for parameters that are not set.

MArc (a,b,c,d,e,f,g): In the old syntax **a** is the passing point information, **b** is the target point information, **c** is the BORDER function setting, and these are required input parameters. Parameters **d**, **e**, **f** and **g** are speed, acceleration, deceleration and PASS information, and are optional input parameters.

In the new MArc syntax, the point setting related function can use “+” to set the point information and change the position of that point. Speed related and continuous motion setting functions can use “+” to perform related settings to other parameters. The default values are used for parameters that are not set.

Example description

The following table shows examples of the new (version 2) and old (version 1) syntax to accomplish the same tasks. These examples show that you can use one command in the new syntax to accomplish a task that uses four or more commands in the old syntax.

Note: You cannot mix new and old speed related parameters and continuous motion mode settings; do not use them together.

Instructions version/parameter setting	For MovP moving mode X direction of point P1 relative movement: 10, speed: 5, acceleration: 10, deceleration: 15, continuous motion mode
New version MovP	MovP("P1"+X(10),SPD(5)+ACC(10)+DEC(15)+PASS())
Old version MovP	P1x = ReadPoint("P1","X"); sP1x = P1x+10 ; WritePoint("P1","X",sP1x); MovP("P1","PASS",5,10,15)
Instructions version/parameter setting	For MArc moving mode Z displacement of point P1 10, Z displacement of point P2 10
New version MArc	MArc("P1"+Z(10),"P2"+Z(10),"BORDER")
Old version MArc	P1z = ReadPoint("P1","Z"); sP1z = P1z+10 ; WritePoint("P1","Z",sP1z); P2z = ReadPoint("P2","Z"); sP2z = P2z+10 ; WritePoint("P2","Z",sP2z); MArc("P1","P2","BORDER")