

# 程序指令详细解析

后台指区分大小写，书写时注意大小写

## 目录

1 基础控制指令.....	6
1.1 IF.....	6
1.2 ELSEIF.....	6
1.3 ELSE.....	7
1.4 ENDIF.....	7
1.5 WHILE.....	7
1.6 ENDWHILE.....	7
1.7 FOR.....	8
1.8 ENDFOR.....	8
1.9 SWITCH.....	9
1.10 CASE.....	9
1.11 DEFAULT.....	9
1.12 ENDSWITCH.....	10
1.13 WAIT.....	10
1.14 GOTO.....	10
1.15 CALL.....	10
1.16 CONTINUE.....	11
1.17 BREAK.....	11
1.18 RETURN.....	11
1.19 EXIT.....	11
2 PLC 函数指令.....	12
2.1 GetMotionStatus.....	12
2.2 SetMotionStatus.....	12
2.3 MotionBegin.....	12
2.4 MotionContinus.....	13
2.5 MotionEnd.....	13
2.6 IsAlarmed.....	13
2.7 GetControlMode.....	14
2.8 SocketServerInit.....	14
2.9 SocketServerAccept.....	15
2.10 SocketClientInit.....	15
2.11 SocketClose.....	16
2.12 SocketWrite.....	16
2.13 SocketRead.....	17
2.14 SocketReadLine.....	17
2.15 RegScanfD, RegScanfS.....	18

2.16	GetMainMotionFile.....	18
2.17	GetExecMotionFile.....	18
2.18	GetExecMotionLine.....	19
2.19	IsMotionFinesshed.....	19
2.20	RemoveFile.....	19
2.21	WriteFile.....	20
2.22	ReadFile.....	20
2.23	LoadConfigs.....	20
3.	标准函数.....	21
3.1	SLEEP.....	21
3.2	PrintMsg.....	21
3.3	Sprintf.....	21
3.4	GetTime.....	22
3.5	IOIN.....	22
3.6	IOOUT.....	22
3.7	DAOUT.....	22
3.8	ReadIni.....	23
3.9	WriteIni.....	23
3.10	IniStringRead.....	24
3.11	IniStringWrite.....	24
3.12	IniDigitalRead.....	25
3.13	IniDigitalWrite.....	25
3.14	TcpServerControl.....	25
3.15	Replace.....	26
3.16	LoadUserConfig.....	26
3.17	SaveUserConfig.....	27
3.18	J1~J9,X,Y,Z,RX,RY,RZ.....	27
3.19	MaxJointPos.....	28
3.20	MinJointPos.....	28
3.21	JointToSpatial.....	29
3.22	BasePoint.....	29
3.23	StrLeft.....	29
3.24	StrRight.....	30
3.25	StrMid.....	30
3.26	StrToDig.....	30
3.27	DigToStr.....	31
3.28	ByteToDig.....	31
3.29	strcpy.....	31
3.30	strncpy.....	31
3.31	strcat.....	32
3.32	strncat.....	32
3.33	strlen.....	32
3.34	strcmp.....	33
3.35	strncmp.....	33

3.36	strcasecmp.....	33
3.37	strncasecmp.....	34
3.38	strchr.....	34
3.39	strrchr.....	34
3.40	strstr.....	35
3.41	strtol.....	35
3.42	strtoul.....	35
3.43	strtod.....	36
3.44	strat.....	36
3.45	strset.....	36
3.46	ConfigsRootPath.....	37
3.47	ConfigsPath.....	37
3.48	TempPath.....	37
3.49	sin.....	37
3.50	cos.....	38
3.51	tan.....	38
3.52	asin.....	38
3.53	acos.....	38
3.54	atan.....	38
3.55	atan2.....	39
3.56	sinh.....	39
3.57	cosh.....	39
3.58	tanh.....	39
3.59	sqrt.....	39
3.60	ExcelOpen.....	40
3.61	ExcelRead.....	40
3.62	ExcelClose.....	40
4	其他.....	41
5	运动指令.....	42
5.1	MoveL.....	42
5.2	MoveC.....	42
5.3	MoveFC.....	43
5.4	MoveJ.....	43
5.5	MoveO.....	44
5.6	MoveNurbs.....	44
5.7	SetSimultaneousMode.....	44
5.8	MoveS.....	45
5.9	SetOffset.....	45
5.10	SetToolParam.....	45
5.11	SetUserParam.....	46
5.12	GETCURPOINT.....	46
5.13	POINTCOPY.....	46
5.14	POINTRESET.....	46
5.15	POINTSET.....	47

5.16 POINTGET.....	47
5.17 POINTADDOFFSET.....	47
5.18 CONVEY.....	47
5.19 MovingTrigger.....	48
5.20 MovingUntil.....	48
5.21 PAUSE.....	48
6.焊接指令.....	49
6.1 ARCSTART.....	49
6.2 WeldS.....	49
6.3 ARCSET.....	49
6.4 ARCEnd.....	49
6.5 ArcWireFeed.....	50
6.6 ArcWireBack.....	50
6.7 WEAVESTART.....	50
6.8 WEAVEEND.....	50
6.9 WELDSPOTSTART.....	51
6.10 WELDSPOTEND.....	51
6.11 WeldTurnBackStart.....	51
6.12 WeldTurnBackEnd.....	51
6.13 SearchStart.....	52
6.14 SearchMoveL.....	52
6.15 SearchEnd.....	52
6.16 WasltSearchStart.....	52
6.17 WasltSearchMoveL.....	53
6.18 WasltSearchEnd.....	53
6.19 TouchOffsetStart.....	53
6.20 TouchOffsetEnd.....	53
6.21 WASLTFULLSTART.....	54
6.22 WASLTSTART.....	54
6.23 WASLTEND.....	54
6.24 WASLTOPERATION.....	55
6.25 WasltParameterSet.....	55
6.26 WASLTGETCURPOINT.....	55
6.27 WASLTGETINTERSECTION.....	56
6.28 ArcTrackStart.....	56
6.29 ArcTrackSample.....	56
6.30 ArcTrackDir.....	57
7.锁螺丝指令.....	57
7.1 LockScrewSetParam.....	57
7.2 LockScrewMove.....	58
8.焊锡指令.....	59
8.1 SolderingSetParam.....	59
8.2 SolderingMoveStart.....	59
8.3 SolderingMoveEnd.....	60

8.4 SolderingMovePoint.....	60
8.5 SolderingClearIron.....	61
9 冲压指令.....	61
9.1 PunchMoveL.....	61
9.2 PunchMoveJ.....	62
9.3 PunchAuxiliaryInstruction.....	63
10 视觉指令.....	63
10.1 VisualOpen.....	63
10.2 VisualClose.....	63
10.3 VisualRecvPoint.....	63
10.4 VisualRecv.....	64
10.5 VisualSend.....	64
10.6 VisualTrigger.....	64
11.7 VisualConversion.....	64
12 跟踪指令.....	65
12.1 FollowInit.....	65
12.2 FollowStartSync.....	65
12.3 FollowStopSync.....	65
12.4 FollowIsInSpace.....	65
12.5 FollowGetState.....	66
12.6 FollowSetPoint.....	66
12.7 FollowGetPoint.....	66
12.8 FollowUpdateUser.....	66
12.9 FollowUpdatePointPosition.....	67
12.10 SetCircleFollowStartPosition.....	67
13 码垛指令.....	67
13.1 PALLETINIT.....	67
13.2 PALLETGETCURPOS.....	67
14 点胶指令.....	68
14.1 GlueMoveNurbsL.....	68
14.2 GlueMoveNurbsC.....	68
14.3 GlueMoveFC.....	69
14.4 GlueMoveNurbs.....	69
14.5 GlueSetOCTime.....	69
14.6 GlueSetParamNum.....	70

# 后台指令篇

## 1 基础控制指令

### 1.1 IF

IF(digital|string)

IF 指令	指令功能	IF 指令和 C 语言的定义类似，语法定义 IF 和 ENDIF 成对出现，IF 指令只有一个参数。		
	参数项	参数 1	数字类型	一个 double 类型的数字，比如 1
			字符串类型	一个字符串类型“123”，或者使用 S1 变量
			表达式	可以使用机器人定义变量，比如 V1==0
使用举例	IF(V1 == 0) IF(S1 == “123”)			

### 1.2 ELSEIF

ELSEIF(digital|string)

ELSEIF 指令	指令功能	ELSEIF 指令和 C 语言的定义类似，语法定义 ELSEIF 必须在 IF 和 ENDIF 之间，ELSEIF 指令只有一个参数。		
	参数项	参数 1	数字类型	一个 double 类型的数字，比如 1
			字符串类型	一个字符串类型“123”，或者使用 S1 变量
			表达式	可以使用机器人定义变量，比如 V1==0
使用举例	ELSEIF(V1 == 0) ELSEIF(S1 == “123”)			

## 1.3 ELSE

ELSE 指令	指令功能	ELSE 指令和 C 语言的定义类似，语法定义 ELSE 必须在 IF 和 ENDIF 之间，ELSE 指令没有参数。
	参数项	无
	使用举例	ELSE

## 1.4 ENDIF

ENDIF 指令	指令功能	语法定义 IF 和 ENDIF 必须成对出现，ENDIF 指令没有参数。
	参数项	无
	使用举例	ELSE

## 1.5 WHILE

WHILE(digital|string)

WHILE 指令	指令功能	WHILE 指令和 C 语言的定义类似，语法定义 WHILE 和 ENDWHILE 成对出现，WHILE 指令只有一个参数。		
	参数项	参数 1	数字类型	一个 double 类型的数字，比如 1
			字符串类型	一个字符串类型“123”，或者使用 S1 变量
			表达式	可以使用机器人定义变量，比如 V1==0
使用举例	WHILE(V1 == 0) WHILE(S1 == “123”)			

## 1.6 ENDWHILE

ENDWHILE 指令	指令功能	语法定义 WHILE 和 ENDWHILE 必须成对出现，ENDWHILE 指令没有参数。
	参数项	无
	使用举例	ENDWHILE

## 1.7 FOR

FOR(digital|string|void;digital|string|void;digital|string|void)

FOR 指令	指令功能	FOR 指令和 C 语言的定义类似，语法定义 FOR 和 ENDFOR 必须成对出现，FOR 指令有 3 个参数，3 个参数都为表达式或空（空默认为真），第一个参数，一般是表达式的初始化。第二个参数，是是否执行 FOR 循环内语句，为真就继续执行，为假不执行。第 3 个参数，表达式在执行完 FOR 语句之后执行。		
	参数项	参数 1	数字类型	一个 double 类型的数字，比如 1
			字符串类型	一个字符串类型“123”，或者使用 S1 变量
			表达式	可以使用机器人定义变量，比如 V1==0
			空	无
		参数 2	数字类型	一个 double 类型的数字，比如 1
			字符串类型	一个字符串类型“123”，或者使用 S1 变量
			表达式	可以使用机器人定义变量，比如 V1==0
			空	表达式为真
		参数 3	数字类型	一个 double 类型的数字，比如 1
			字符串类型	一个字符串类型“123”，或者使用 S1 变量
			表达式	可以使用机器人定义变量，比如 V1==0
			空	无
	使用举例	FOR(V1=0; V1<10;V1=V1+1) FOR(;;)//死循环		

## 1.8 ENDFOR

ENDFOR 指令	指令功能	语法定义 ENDFOR 和 FOR 指令必须成对出现，ENDFOR 指令没有参数。
	参数项	无
	使用举例	ENDFOR



## 1.9 SWITCH

SWITCH(digital|string)

SWITCH 指令	指令功能	SWITCH 指令和 C 语言的定义类似， SWITCH 指令只有一个参数。		
	参数项	参数 1	数字类型	一个 double 类型的数字，比如 1
			字符串类型	一个字符串类型“123”，或者使用 S1 变量
使用举例	SWITCH(V1) SWITCH(S1)			

## 1.10 CASE

CASE string|digital

CASE 指令	指令功能	CASE 指令和 C 语言的定义类似，配合 SWITCH 指令使用， SWITCH 指令只有一个参数，参数必须为常量		
	参数项	参数 1	数字类型	一个 int 类型的数字，比如 1
			字符串类型	一个字符串类型“123”
使用举例	CASE 1: CASE '1'			

## 1.11 DEFAULT

DEFAULT 指令	指令功能	DEFAULT 指令和 C 语言的定义类似，配合 SWITCH 指令使用，一般是用于 CASE 无法匹配的情况使用 DEFAULT，DEFAULT 没有参数。		
	参数项	参数无		
	使用举例	DEFAULT		

## 1.12 ENDSWITCH

ENDSWITCH 指令	指令功能	ENDSWITCH 指令需要和 SWITCH 指令成对出现，ENDSWITCH 指令没有参数。		
	参数项	参数无		
	使用举例	ENDSWITCH		

## 1.13 WAIT

WAIT(digital|string)

WAIT 指令	指令功能	WAIT 等待指令，参数表达式为真，就跳出等待，参数表达式为假，就一直等待，指令有一个参数。		
	参数项	参数 1	数字类型	一个 double 类型的数字，比如 1
			字符串类型	一个字符串类型“123”，或者使用 S1 变量
			表达式	表达式判断，比如 V1==0
使用举例	WAIT(V1==2)//V1!=2 时，会一直等待，V1==2 时，会执行下一行语句。			

## 1.14 GOTO

GOTO loop

GOTO 指令	指令功能	GOTO 跳转语句，需要和标记 LXXX 配合使用，直接跳转到对应的 LXXX 语句位置执行语句。		
	参数项	参数 1	标记的序号	比如 L1，必须先定义 L1
	使用举例	GOTO L1		

## 1.15 CALL

CALL string

CALL 指令	指令功能	CALL 执行参数项文件名的文件，比如调用子程序		
	参数项	参数 1	执行的文件名	比如“test”，文件名称不含.txt
	使用举例	CALL “test”		

## 1.16 CONTINUE

CONTINUE 指令	指令功能	和 C 语言的 CONTINUE 类似，继续执行循环体，一般和 FOR 或 WHILE 循环体使用。
	参数项	参数无
	使用举例	CONTINUE

## 1.17 BREAK

BREAK 指令	指令功能	和 C 语言的 BREAK 类似，跳出本层循环体，一般和 FOR 或 WHILE 循环体使用。
	参数项	参数无
	使用举例	BREAK

## 1.18 RETURN

RETURN 指令	指令功能	返回之前的文件，通常配合 CALL 指令使用。
	参数项	参数无
	使用举例	RETURN

## 1.19 EXIT

EXIT 指令	指令功能	程序结束。
	参数项	参数无
	使用举例	EXIT

## 2 PLC 函数指令

### 2.1 GetMotionStatus

int GetMotionStatus()

GetMotionStatus	指令功能	获取当前机器人的运动状态		
	参数项	参数无		
	返回值	-1: 机器人当前为错误状态		
		0: 机器人已经停止		
		1: 机器人在停止中状态		
使用举例	V1= GetMotionStatus()//获取当前机器人的运动状态, 返回到 V1			

### 2.2 SetMotionStatus

int SetMotionStatus(digital)

SetMotionStatus	指令功能	设置机器人的运动状态		
	参数项	参数 1	数字类型	值只能设置成 0, 即停止机器人
	返回值	0: 设置机器人状态出错		
		1: 设置机器人状态成功		
使用举例	V1= SetMotionStatus(0)//停止机器人的运动, 结果返回 V1			

### 2.3 MotionBegin

int MotionBegin(string,digital)

MotionBegin	指令功能	启动指定的文件名, 参数 1 设置启动文件名, 参数 2 设置启动行, 必须先等待机器人的状态为停止状态, 再重新启动执行机器人程序		
	参数项	参数 1	字符串类型	选择要启动的文件名, 不需要带.txt 后缀。
		参数 2	启动行	-1: 表示从当前行启动
				0: 表示从首行启动
	>0: 表示从指定行启动, 从 0 开始计行数。			
	返回值	<0: 函数执行失败		
	0: 函数执行成功			
使用举例	V1=MotionBegin("test",0)//从首行启动文件 test, 结果返回 V1			

## 2.4 MotionContinue

int MotionContinue()

MotionContinue	指令功能	继续运行机器人
	参数项	参数 无
	返回值	负数： 机器人状态出错
		0： 机器人状态成功
使用举例	V1= MotionContinue()//继续运行机器人， 结果返回 V1	

## 2.5 MotionEnd

int MotionEnd()

MotionEnd	指令功能	停止机器人运动，注意本函数仅触发停止动作，并不会等到运动停止
	参数项	参数无
	返回值	0： 表示停止成功
	使用举例	V1= MotionEnd()//停止机器人运动， 结果返回 V1

## 2.6 IsAlarmed

int IsAlarmed()

IsAlarmed	指令功能	查询机器人是否报警
	参数项	参数无
	返回值	0： 表示机器人无报警
		1： 表示机器人处于报警状态
使用举例	V1= IsAlarmed ()//查询机器人是否报警， 结果返回 V1	

## 2.7 GetControlMode

int GetControlMode()

GetControlMode	指令功能	获取机器人的控制模式		
	参数项	参数无		
	返回值	0x000: 无		
		0x001: 运动异常		
		0x002: EtherCat 总线通讯异常		
		0x004: 伺服报警		
		0x008: 示教模式		
		0x010: 再现模式		
		0x020: 上使能状态		
		0x040: 急停状态		
		0x080: 安全门开启		
		0x100: 机器人碰撞		
		0x200: 软件版本激活失败		
	0x400: 驱动器电池电压低			
使用举例	V1= GetControlMode();//获取机器人的控制模式结果 返回 V1			

## 2.8 SocketServerInit

int SocketServerInit(int)

int SocketServerInit(int,int)

SocketServerInit	指令功能	初始化视觉服务器端口，该函数有两种类型，一个参数时，默认协议是 TCP。		
	参数项	参数 1	数字类型	设置服务器监听的端口号
		参数 2	数字类型	设置网络连接协议，值为 1 时，是 TCP 协议，值为 2 时，是 UDP 协议。
	返回值	<0: 返回的是负数，监听端口失败		
		>0: 返回的是正数，获取的是套接字。		
使用举例	V1= SocketServerInit(6000)//服务器监听端口 6000，默认是 TCP 协议 V1= SocketServerInit(6000,2)// 服务器监听端口 6000，UDP 协议通信			

## 2.9 SocketServerAccept

int SocketServerAccept(int)

SocketServerAccept	指令功能	等待客户端连接，没有客户端连接时，该函数会阻塞。		
	参数项	参数 1	数字类型	套接字，可以使用 SocketServerInit 函数获取的套接字作为参数。
	返回值	<0: 获取客户端连接失败		
		>0: 返回的是正数，获取的是套接字。		
使用举例	V1= SocketServerAccept(V2)//等待套接字 V2 的服务器，有客户端连接			

## 2.10 SocketClientInit

int SocketClientInit(string,int)

int SocketClientInit(string,int,int)

SocketClientInit	指令功能	初始化客户端，连接服务器		
	参数项	参数 1	字符串类型	输入连接服务器的 IP 地址，比如“192.168.1.10”，或者使用 S 变量，比如 S1，需要在其他地方将 IP 地址赋值到 S1
		参数 2	数字类型	连接服务器的端口
		参数 3	数字类型	设置网络连接协议，值为 1 时，是 TCP 协议，值为 2 时，是 UDP 协议。
	返回值	<0: 获取客户端连接失败		
>0: 返回的是正数，获取的是套接字。				
使用举例	V1= SocketClientInit("192.168.1.10",6000)//连接服务器，服务器地址为 192.168.1.10，端口号为 6000，使用的是 TCP 协议 V1= SocketClientInit("192.168.1.10",6000,2)//连接服务器，服务器地址为 192.168.1.10，端口号为 6000，使用的是 UDP 协议			

## 2.11 SocketClose

void SocketClose(int)

SocketClose	指令功能	关闭连接		
	参数项	参数 1	数字类型	套接字，关闭连接，既可以是关闭服务器也可以是客户端。
	返回值	无		
	使用举例	SocketClose(ClientFD1)//关闭服务器 ClientFD1		

## 2.12 SocketWrite

int SocketWrite(int,string)

int SocketWrite(int,string,int)

SocketWrite	指令功能	向服务器或客户端写入数据		
	参数项	参数 1	数字类型	套接字，写入的是服务器的套接字或者客户端的套接字。
		参数 2	字符串类型	发送的数据
		参数 3	数字类型	发送指定长度的字符串
	返回值	<0: 写入失败		
		>0: 返回成功发送的字符串长度。		
使用举例	V1= SocketWrite(V2," Successful connection")//向 V2 的套接字发送"Successful connection"			



## 2.13 SocketRead

int SocketRead(int,string|out,int)

int SocketRead(int,string|out,int,int)

SocketRead	指令功能	接收数据		
	参数项	参数 1	数字类型	套接字, socket id
		参数 2	字符串类型	接收字符串 buffer, 存储缓冲区
		参数 3	数字类型	需要接收的长度, 为 0 则有多少读多少 (最大为 string 大小), 之后立刻返回
		参数 4	数字类型	超时值, 不使用带超时的函数时, 函数阻塞, 读取长度达标才返回, 使用函数时, 函数待超时以后或读取长度达标时返回, 超时为 0 时, 立刻返回
	返回值	<0: 接收失败		
	>0: 返回成功读取的长度。			
使用举例	SocketRead(ClientFD,Buffer,1000)//从 ClientFD 中读取 1000 长度的字符串			

## 2.14 SocketReadLine

int SocketReadLine(int,string|out,int)

SocketReadLine	指令功能	接收一行数据		
	参数项	参数 1	数字类型	套接字
		参数 2	字符串类型	接收字符串 buffer
		参数 3	数字类型	超时时间, 超时为 0 时, 立即返回
	返回值	<0: 写入失败		
		>0: 返回成功读取的长度。		
使用举例	Size = SocketReadLine(ClientFD,Buffer,1000)//从 ClientFD 中读取一行 1000 长度的字符串			

## 2.15 RegScanfD, RegScanfS

int RegScanfD(string,string out|...)

int RegScanfS(string,string out|...)

RegScanfD RegScanfS	指令功能	分割字符串		
	参数项	参数 1	字符串类型	待处理字符串
		参数 2	字符串类型	正则表达式
		参数 3~n	数字类型	匹配后存放参数 1, 参数 4 ~ n, 依次存放正则切割结果。
	返回值	<0: 错误		
0: 执行成功				
使用举例	RegScanfD("4.5 -5.98 +7.69","(*) (*)(*),0,V1,V2,V3)// 把 "4.5 -5.98 +7.69"中的三个浮点数放到变量 V1, V2, V3 中			

## 2.16 GetMainMotionFile

string GetMainMotionFile()

GetMainMotionFile	指令功能	读取当前正在运行的文件的主文件名
	参数项	参数 无
	返回值	文件名 (不带后缀)
	使用举例	S1=GetMainMotionFile();//读取当前正在运行的文件的主文件名, 结果返回 S1

## 2.17 GetExecMotionFile

string GetExecMotionFile()

GetExecMotionFile	指令功能	读取当前正在运行的文件名
	参数项	参数 无
	返回值	文件名 (不带后缀)
	使用举例	S1=GetExecMotionFile();//读取当前正在运行的文件名, 结果返回 S1

## 2.18 GetExecMotionLine

int GetExecMotionLine()

GetExecMotionLine	指令功能	读取当前正在运行行号
	参数项	参数 无
	返回值	行号
	使用举例	V1=GetExecMotionLine()//读取当前正在运行行号，结果返回 V1

## 2.19 IsMotionFineshed

int IsMotionFineshed()

IsMotionFineshed	指令功能	当前 motion 文件是否运行完毕
	参数项	参数 无
	返回值	1: 完成 0: 未完成
	使用举例	V1=IsMotionFineshed()//当前 motion 文件是否运行完毕，结果返回 V1

## 2.20 RemoveFile

int RemoveFile(string)

RemoveFile	指令功能	删除文件
	参数项	参数 1   字符串类型   文件名
	返回值	非 0: 失败 0: 成功
	使用举例	V1=RemoveFile(S1)//删除文件 S1，结果返回 V1

## 2.21 WriteFile

int WriteFile(string, string)

WriteFile	指令功能	写文件		
	参数项	参数 1	字符串类型	函数名
		参数 2	字符串类型	文件内容，注意不是追加写入
	返回值	失败返回负数		
		成功返回写入长度		
使用举例	V1=WriteFile(S1, S2)//向 S1 文件中写入 S2 内容，结果返回 V1			

## 2.22 ReadFile

string ReadFile(string)

ReadFile	指令功能	读文件		
	参数项	参数	字符串类型	文件名
	返回值	成功返回读取内容，失败返回空，如果文件大小大于 string 类型长度，则读取 string 长度，因此本函数不适合读取较大文件		
	使用举例	S2=ReadFile(S1)//读取文件 S1，结果返回 S2		

## 2.23 LoadConfigs

void LoadConfigs()

LoadConfigs	指令功能	强制加载配置文件到内存		
	参数项	参数 无		
	使用举例	LoadConfigs()//强制加载配置文件到内存		

## 3.标准函数

### 3.1 SLEEP

int SLEEP(digital|reg{\d{1,}}exp)

SLEEP	指令功能	睡眠		
	参数项	参数	数字类型	睡眠时间, 单位毫秒
	使用举例	SLEEP(500)//延时 500 毫秒		

### 3.2 PrintMsg

void PrintMsg(digital,string out|...)

PrintMsg	指令功能	写入		
	参数项	参数 1	数字类型	一个 double 类型的数字, 比如 1 或者使用 V1 变量
		参数 2	字符串类型	一个字符串类型"123",或者使用 S1 变量
	使用举例	PrintMsg(ERR,"SocketClientInit 执行失败 %d\n",ClientFD1) //日志中打印"SocketClientInit 执行失败"错误提示		

### 3.3 Sprintf

int Sprintf(string|out,string out|...)

Sprintf	指令功能	把格式化数据写入字符串		
	参数项	参数 1	字符串类型	一个字符串类型"123",或者使用 S1 变量
		参数 2	字符串类型	一个字符串类型"123",或者使用 S1 变量
	使用举例	Sprintf(Buf, "MM")//将 MM 写入 Buf		

### 3.4 GetTime

digital GetTime()

GetTime	指令功能	取得系统时间		
	参数项	参数	数字类型	系统时间
	使用举例	V1=Gettime()//获取系统时间, 结果返回 V1		

### 3.5 IOIN

digital IOIN(digital)

IOIN	指令功能	读取 I/O 索引		
	参数项	参数	数字类型	I/O 索引,IO 值 (00~199) 或者 V 变量 (1~999)
	返回值	V 变量, IO 开为 1, 关为 0		
	使用举例	V1=IOIN(1)//读取 IO1 的值, 返回 V1		

### 3.6 IOOUT

void IOOUT(digital,digital)

void IOOUT(digital,digital,digital)

IOOUT	指令功能	设置 I/O 值		
	参数项	参数 1	数字类型	I/O 索引,IO 值 (00~199) 或者 V 变量 (1~999)
		参数 2	数字类型	值, 0 表示 off, 1 表示 on
	使用举例	IOOut(01,0)//I/O 输出 01=OFF (程序停止提示)		

### 3.7 DAOUT

void DAOUT(digital,digital)

DAOUT	指令功能	设置模拟量值		
	参数项	参数 1	数字类型	模拟量索引, 模拟量值 (00~199) 或者 V 变量 (1~999)
		参数 2	数字类型	值
	使用举例	DAOut(01,0)//DA 输出 01=0		

## 3.8 ReadIni

void ReadIni(string,string,string,string|out)

ReadIni	指令功能	读 ini 文件		
	参数项	参数 1	字符串类型	添加 config 后面的目录名, 字符串或 S/FS 变量 (0~999)
		参数 2	字符串类型	ini 文件的 section, 字符串或 S/FS 变量 (0~999)
		参数 3	字符串类型	ini 文件的 key, 字符串或 S/FS 变量 (0~999)
		参数 4	字符串类型	默认值, ini 文件没有时, 返回这个值, 否则返回读取的内容
使用举例	ReadINI("gui.ini","COORD","TOOL_NUM",S4)## 读取 gui.ini 中 COORD 下 TOOL_NUM 的键值, 返回到 S4 中			

## 3.9 WriteIni

void WriteIni(string,string,string,string)

WriteIni	指令功能	写 ini 文件		
	参数项	参数 1	字符串类型	添加 config 后面的目录名, 字符串或 S/FS 变量 (0~999)
		参数 2	字符串类型	ini 文件的 section, 字符串或 S/FS 变量 (0~999)
		参数 3	字符串类型	ini 文件的 key, 字符串或 S/FS 变量 (0~999)
		参数 4	字符串类型	默认值, ini 文件没有时, 返回这个值, 否则返回读取的内容
使用举例	WriteINI("gui.ini","COORD","TOOL_NUM",S1)## 将 S1 写入到 gui.ini 中 COORD 下 TOOL_NUM 的键值下			

## 3.10 IniStringRead

string IniStringRead(string,string,string,string)

IniStringRead	指令功能	读 ini 文件		
	参数项	参数 1	字符串类型	添加 config 后面的目录名, 字符串或 S/FS 变量 (0~999)
		参数 2	字符串类型	ini 文件的 section,字符串或 S/FS 变量 (0~999)
		参数 3	字符串类型	ini 文件的 key, 字符串或 S/FS 变量 (0~999)
		参数 4	字符串类型	默认值, ini 文件没有时, 返回这个值, 否则返回读取的内容
使用举例	IniStringRead("ethercat.ini","GLOBAL","enifile",S1)## 读取 ethercat.ini 中 GLOBAL 下 enifile 的键值, 返回到 S1 中			

## 3.11 IniStringWrite

void IniStringWrite(string,string,string,string)

IniStringWrite	指令功能	写 ini 文件		
	参数项	参数 1	字符串类型	添加 config 后面的目录名, 字符串或 S/FS 变量 (0~999)
		参数 2	字符串类型	ini 文件的 section,字符串或 S/FS 变量 (0~999)
		参数 3	字符串类型	ini 文件的 key, 字符串或 S/FS 变量 (0~999)
		参数 4	字符串类型	默认值, ini 文件没有时, 返回这个值, 否则返回读取的内容
使用举例	IniStringWrite("ethercat.ini","GLOBAL","enifile",S1)## 将 S1 中的值写入到 ethercat.ini 中 GLOBAL 下 enifile 的键值下			



## 3.12 IniDigitalRead

digital IniDigitalRead(string,string,string,digital)

IniDigitalRead	指令功能	读 ini 文件		
	参数项	参数 1	字符串类型	添加 config 后面的目录名，字符串或 S/FS 变量 (0~999)
		参数 2	字符串类型	ini 文件的 section,字符串或 S/FS 变量 (0~999)
		参数 3	字符串类型	ini 文件的 key, 字符串或 S/FS 变量 (0~999)
		参数 4	数字类型	默认值，ini 文件没有时，返回这个值，否则返回读取的内容
使用举例	IniDigitalRead("gui.ini","COORD","TOOL_NUM",V4)## 读取 gui.ini 中 COORD 下 TOOL_NUM 的键值，返回到 V4 中			

## 3.13 IniDigitalWrite

void IniDigitalWrite(string,string,string,digital)

IniDigitalWrite	指令功能	写 ini 文件		
	参数项	参数 1	字符串类型	添加 config 后面的目录名，字符串或 S/FS 变量 (0~999)
		参数 2	字符串类型	ini 文件的 section,字符串或 S/FS 变量 (0~999)
		参数 3	字符串类型	ini 文件的 key, 字符串或 S/FS 变量 (0~999)
		参数 4	数字类型	默认值，ini 文件没有时，返回这个值，否则返回读取的内容
使用举例	IniDigitalWrite("gui.ini","COORD","TOOL_NUM",V1)## 将 V1 写入到 gui.ini 中 COORD 下 TOOL_NUM 的键值下			

## 3.14 TcpServerControl

void TcpServerControl(string,string|out)

TcpServerControl	指令功能	TCP 服务器控制		
	参数项	参数 1	字符串类型	命令，字符串“123”或 S 变量 (0~999)
		参数 2	字符串类型	返回结果，S0~S999
使用举例	TcpServerControl(S1,S2)//发送命令 S1，结果返回 S2			

## 3.15 Replace

void Replace(string,string,string,string|out)

void Replace(string,string,string,digital|out)

Replace	指令功能	将一个输入字符串根据正则表达式来分割，并根据新的格式生成新的字符串		
	参数项	参数 1	字符串类型	输入字符串，即要分割的字符串
		参数 2	字符串类型	正则表达式，即分割规则
		参数 3	字符串类型	重组表达式：即重组规则
		参数 4	数字类型	存储处理结果，可以是字符串或数字变量，以后计划加入坐标点变量
	返回值	无，处理失败参数 4 得不到结果		
使用举例	Replace("4.5-5.98+7.69","(*) (*) (*),"X=\1 Y=-\2 Z=\3",S1)// 将字符串 "4.5 -5.98 +7.69" 改成 "X=4.5 Y=-5.98 Z=+7.69"			

## 3.16 LoadUserConfig

void LoadUserConfig(digital,digital,digital|out)

void LoadUserConfig(digital,digital,string|out)

void LoadUserConfig(digital,digital,point|out)

LoadUserConfig	指令功能	用户配置加载		
	参数项	参数 1	数字类型	用户配置组，01~20
		参数 2	数字类型	参数，01~10
		参数 3	数字类型	保存到数字类型 (V/FV 变量，0~999) 或字符串类型 (S/FS 变量，0~999) 或 (模拟) IO 输入/输出 (0~199)
			字符串类型 点位	
使用举例	LoadUserConfig(01,01,V0)//加载用户配置组 01 参数 01 保存到 V0			

## 3.17 SaveUserConfig

void SaveUserConfig(digital,digital,digital)

void SaveUserConfig(digital,digital,string)

void SaveUserConfig(digital,digital,point)

SaveUserConfig	指令功能	用户配置保存		
	参数项	参数 1	数字类型	用户配置组, 01~20
		参数 2	数字类型	参数, 01~10
		参数 3	数字类型	保存到数字类型 (V/FV 变量, 0~999) 或字符串类型 (S/FS 变量, 0~999) 或 (模拟) IO 输入/输出 (0~199)
	字符串类型 点位			
使用举例	SaveUserConfig(01,01,V0)//保存用户配置 组 01 参数 01 值 V0			

## 3.18 J1~J9,X,Y,Z,RX,RY,RZ

### 1.以 J1 为例

digital J1(point)			
参数项	参数 1	点位	P 点, 返回值为 P 点的 J1
使用举例	V1 = J1(P1)//将 P1 点的 J1 轴坐标值保存到 V1 中		

void J1(point out,digital)			
参数项	参数 1	点位	P 点
	参数 2	数字类型	设置的值
使用举例	J1(P1,1)//将 P1 点的 J1 轴坐标值设置成 1		

digital J1(digital)			
参数项	参数 1	数字类型	1 表示当前点的 J1 值, 2 表示目标点的 J1 值, 返回值 J1 的坐标值
返回值	J1 的坐标值		
使用举例	V1=J1(1)//返回当前点的 J1 轴的值		

digital J1 () 或 digital J1			
使用举例	V1=J1//返回值是返回当前点的 J1 轴的值		

## 2.以 X 为例

digital X(point)			
参数项	参数 1	点位	获取的 P 点, 返回值是 P 点的 X 坐标
使用举例	V1=X(P1) //将 P1 的直角坐标系的 X 值存在 V1 变量中		

void X(point[out,digital)			
参数项	参数 1	点位	要设置坐标值的 P 点
	参数 2	数字类型	要保存的值
使用举例	X(P1,1) //将 P1 的直角坐标系的 X 值设置成 1		

digital X (digital, digital, digital)			
参数项	参数 1	数字类型	表示当前点还是目标点, 1 为当前点, 2 为目标点
	参数 2	数字类型	工具号, 1~10
	参数 3	数字类型	用户号, 1~10
返回值	用户坐标的 X 值		
使用举例	V1=X(1,2,2) //返回当前点工具 2 用户 2 的用户坐标的 X 值		

digital X()或 digital X	
使用举例	V1=X1 //返回值是当前点的示教界面上面工具号的工具坐标的 X 值

## 3.19 MaxJointPos

digital MaxJointPos(digital)

MaxJointPos	指令功能	获取指定轴的最大关节角度		
	参数项	参数	数字类型	关节位置 (1~9)
	返回值	返回获取的最大关节角度		
	使用举例	V1=MaxJointPos(1)//获取 1 轴的最大关节角度到 V1		

## 3.20 MinJointPos

digital MinJointPos(digital)

MinJointPos	指令功能	获取指定轴的最小关节角度		
	参数项	参数	数字类型	关节位置 (1~9)
	返回值	返回获取的最小关节角度		
	使用举例	V1=MinJointPos(1)//获取 1 轴的最小关节角度到 V1		

## 3.21 JointToSpatial

point JointToSpatial(point,digital,digital)

JointToSpatial	指令功能	关节转空间		
	参数项	参数 1	点位	关节点, P0~P999
		参数 2	数字类型	工具,0~10
		参数 3	数字类型	用户,0~10
	返回值	返回已经转换的点 (关节和空间)		
使用举例	P1=JointToSpatial(P2,1,1)//获取工具 1, 用户 1 坐标下的 P2 关节点到 P1			

## 3.22 BasePoint

int BasePoint(digital,point[out])

BasePoint	指令功能	获取基准点		
	参数项	参数 1	数字类型	基准点索引, 01~10
		参数 2	点位	返回基准点 (关节和空间) P1~P999
	返回值	基准点已经开启返回 0, 否则返回-1		
	使用举例	BasePoint(1,P1)//将基准点序号 1 的基准点获取到 P1		

## 3.23 StrLeft

void StrLeft(string,digital,string[out]),  
string StrLeft(string,digital)

StrLeft	指令功能	从字符串左边取多少个字符		
	参数项	参数 1	字符串类型	字符串"123"或 S0~S999
		参数 2	数字类型	截取长度
		参数 3	字符串类型	获取到的位置, S0~S999
使用举例	StrLeft(S1,2,S2)//将 S1 中的字符左边两位截取到 S2 中			

## 3.24 StrRight

void StrRight(string,digital,string|out),  
string StrRight(string,digital)

StrRight	指令功能	从字符串右边取多少个字符		
	参数项	参数 1	字符串类型	字符串“123”或 S0~S999
		参数 2	数字类型	截取长度
		参数 3	字符串类型	获取到的位置, S0~S999
使用举例	StrRight(S1,2,S2)//将 S1 中的字符右边两位截取到 S2 中			

## 3.25 StrMid

void StrMid(string,digital,digital,string|out),  
string StrMid(string,digital,digital)

StrMid	指令功能	截取子串		
	参数项	参数 1	字符串类型	字符串“123”或 S0~S999
		参数 2	数字类型	截取起始点
		参数 3	数字类型	截取长度
		参数 4	字符串类型	获取到的位置, S0~S999
使用举例	StrMid (S1,2,3,S2) //将 S1 中的字符从第 3 位开始截取 3 个字符到 S2 中			

## 3.26 StrToDig

void StrToDig(string,digital|out),  
digital StrToDig(string)

StrToDig	指令功能	字符串转数字量		
	参数项	参数 1	字符串类型	一个字符串类型“123”,或者使用 S1 变量 (0~999)
		参数 2	数字类型	获取位置 V0~V999
使用举例	StrToDig("123",V1)//将字符“123”转到 V1 中			

## 3.27 DigToStr

```
void DigToStr(digital,string|out),  
string DigToStr(digital)
```

DigToStr	指令功能	数字量转字符		
	参数项	参数 1	数字类型	使用机器人定义变量, 比如 V1=0
		参数 2	字符串类型	S1 变量 (0~999)
使用举例	DigToStr (1, S1)//将数字量 1 转到字符 S1 变量中			

## 3.28 ByteToDig

```
void ByteToDig(string,digital|out),  
digital ByteToDig(string)
```

ByteToDig	指令功能	字节转数字量		
	参数项	参数 1	字符串类型	一个字符串类型"123",或者使用 S1 变量 (0~999)
		参数 2	数字类型	变量 V (0~999)
使用举例	ByteToDig (S1,V1)//将 S1 的字节转到 V1 中			

## 3.29 strcpy

```
int strcpy(string|out,string)
```

strcpy	指令功能	把指向的字符串拷贝到 S 变量中去		
	参数项	参数 1	字符串类型	字符串, S0~S999
		参数 2	字符串类型	字符串,S0~S999
	返回值	参数 1		
使用实例	Strcpy(S1,S2)//将 S2 中的字符串拷贝到 S1 中			

## 3.30 strncpy

```
int strncpy(string|out,string,digital)
```

strncpy	指令功能	把 S 变量指向的字符串中最多前 count 个字符拷贝到串 S 变量中去		
	参数项	参数 1	字符串类型	字符串 S0~S999
		参数 2	字符串类型	字符串 S0~S999
		参数 3	数字类型	字符长度
	返回值	第一个字符变量中		
使用实例	strncpy(S1,S2,2)//将 S2 中的前 2 个字符拷贝到 S1 中去			

### 3.31 strcat

int strcat(string|out,string)

strcat	指令功能	把字符 S 变量 2 接到 S 变量 1 后面，取消原来 S 变量 1 最后面的串结束符“\n”		
	参数项	参数 1	字符串类型	字符串 S0~S999
		参数 2	字符串类型	字符串 S0~S999
	返回值	第一个字符变量中		
使用实例	strcat(S1,S2)//将 S2 变量中的字符接到 S1 变量后面，取消 S1 最后的“\n”			

### 3.32 strncat

int strncat(string|out,string,digital)

strncat	指令功能	把 S 变量 2 指向的字符串中最多前 count 个字符连到串 S 变量 1 后，并以 NULL 结尾		
	参数项	参数 1	字符串类型	字符串 S0~S999
		参数 2	字符串类型	字符串 S0~S999
		参数 3	数字类型	字符长度
	返回值	第一个字符变量中		
使用实例	strncat(S1,S2,2)//将 S2 变量中的前 2 个字符接到 S1 变量后			

### 3.33 strlen

int strlen(string)

strlen	指令功能	统计字符串中字符的个数（不包括终止符“\n”）		
	参数项	参数	字符串类型	字符“123”或 S0~S999
	返回值	字符个数		
	使用实例	V1=Strlen(S1)//将 S1 变量中的字符个数统计到 V1 中		



### 3.34 strcmp

int strcmp(string,string)

strcmp	指令功能	比较字符串 S1 和 S2		
	参数项	参数 1	字符串类型	字符量为"123"或 S0~S999
		参数 2	字符串类型	字符量为"123"或 S0~S999
	返回值	负数: S1<S2		
		0: S1=S2		
正数: S1>S2				
使用实例	V1=Strcmp(S1,S2)//比较 S1 和 S2 的大小, 返回到 V1			

### 3.35 strncmp

int strncmp(string,string,digital)

strncmp	指令功能	比较字符串 S1 和 S2 中最多前 count 个字符		
	参数项	参数 1	字符串类型	字符量为"123"或 S0~S999
		参数 2	字符串类型	字符量为"123"或 S0~S999
		参数 3	数字类型	字符长度
	返回值	负数: S1<S2		
0: S1=S2				
正数: S1>S2				
使用实例	V1=Strncmp(S1,S2,3)//比较 S1 和 S2 前 3 位大小, 返回 V1			

### 3.36 strcasecmp

int strcasecmp(string,string)

strcasecmp	指令功能	忽略大小写比较字符串		
	参数项	参数 1	字符串类型	字符量为"123"或 S0~S999
		参数 2	字符串类型	字符量为"123"或 S0~S999
	返回值	0: S1=S2		
		正数: S1>S2		
负数: S1<S2				
使用实例	V1=Strcasecmp(S1,S2)//忽略大小比较 S1 和 S2, 返回到 V1			

### 3.37 strncasecmp

int strncasecmp(string,string,digital)

strncasecmp	指令功能	忽略大小写比较字符串前 n 个字符		
	参数项	参数 1	字符串类型	字符量为“123”或 S0~S999
		参数 2	字符串类型	字符量为“123”或 S0~S999
		参数 3	数字类型	字符长度
	返回值	0: S1=S2		
		正数: S1>S2		
负数: S1<S2				
使用实例	V1=Strncasecmp(S1,S2,2)//忽略大小比较 S1 和 S2 前 2 个字符, 返回到 V1			

### 3.38 strchr

int strchr(string,string)

strchr	指令功能	找出 S1 指向的字符串中第一次出现字符 S2 的位置		
	参数项	参数 1	字符串类型	字符量为“123”或 S0~S999
		参数 2	字符串类型	字符量为“123”或 S0~S999
	返回值	返回指向该位置的指针, 如找不到, 则返回 NULL		
	使用实例	V1=Strchr(S1,S2)//找出 S1 中第一次出现 S2 字符串的位置, 返回 V1		

### 3.39 strrchr

int strrchr(string,string)

strrchr	指令功能	查找字符在知道字符串中从后面开始的第一次出现的位置		
	参数项	参数 1	字符串类型	字符量为“123”或 S0~S999
		参数 2	字符串类型	字符量为“123”或 S0~S999
	返回值	成功返回指向该位置的指针		
		失败返回 false		
使用实例	V1=Strrchr(S1,S2)//找出 S1 中从后面开始第一次出现 S2 字符串的位置, 返回 V1			

### 3.40 strstr

int strstr(string,string)

strstr	指令功能	寻找 S2 指向的字符串在 S1 指向的字符串中首次出现的位置		
	参数项	参数 1	字符串类型	字符常量"123"或 S0~S999
		参数 2	字符串类型	字符常量"123"或 S0~S999
	返回值	返回 S2 指向的字符串首次出现的地址，否则返回 NULL		
使用实例	V1=Strstr(S1,S2)//找出 S1 中第一次出现 S2 字符串的位置，返回 V1			

### 3.41 strtol

digital strtol(string,string|out,digital)

strtol	指令功能	将参数 S1 字符串根据参数 V1 转换成长整型数，参数 V1 范围从 2 到 36		
	参数项	参数 1	字符串类型	字符常量"123"或 S0~S999
		参数 2	字符串类型	S0~S999
		参数 3	数字类型	数字常量或 V0~V999
返回值	若参数 S2 不为 NULL，则会将遇到不合条件而终止的 S1 中的字符指针由 S2 返回；若参数 S2 为 NULL，则会不返回非法字符串			
使用实例	Strtol(S1,S2,V1)//将 S1 根据 V1 转换成长整数，返回到 S2			

### 3.42 strtoul

digital strtoul(string,string|out,digital)

strtoul	指令功能	将参数 S1 字符串根据参数 V1 来转换成无符号的长整型数		
	参数项	参数 1	字符串类型	字符常量"123"或 S0~S999
		参数 2	字符串类型	S0~S999
		参数 3	数字类型	数字常量或 V0~V999
返回值	若参数 S2 部位 NULL，则会将遇到不合条件而终止的 S1 中的字符指针由 S2r 返回			
使用实例	Strtoul(S1,S2,V1)//将 S1 根据 V1 转换成无符号长整型数，返回 S2			

### 3.43 strtod

digital strtod(string,string|out)

strtod	指令功能	将字符串转换成浮点数		
	参数项	参数 1	字符串类型	字符"123"或 S0~S999
		参数 2	字符串类型	S0~S999
	返回值	返回转换后的浮点型数		
	使用实例	Strtod(S1,S2)//将字符串 S1 转换成浮点数, 返回 S2		

### 3.44 strat

digital strat(string,digital)

strat	指令功能	读取字符串索引字符, 并转成数字返回		
	参数项	参数 1	字符串类型	字符"123"或 S0~S999
		参数 2	数字类型	数字常量或 V0~V999
	返回值	返回转换成的数字		
	使用实例	V1=Strat(S1,2)//读取 S1 中的字符串索引字符, 转换成数字, 返回到 V1		

### 3.45 strset

string strset(string|out,digital,digital),

string strset(string|out,digital,string)

strset	指令功能	将字符串制定索引写入数字或字符		
	参数项	参数 1	字符串类型	字符"123"或 S0~S999
		参数 2	数字类型	索引, 数字常量或 V0~V999
		参数 3	数字类型	写入数字, 数字常量或 V0~V999
	字符串类型		写入字符, 字符"123"或 S0~S999	
返回值	操作后的字符			
使用实例	S1=Strset(S2,2,V2)//将 S2 制定索引 2 写入 V2 到 S1 中			

### 3.46 ConfigsRootPath

string ConfigsRootPath( )

string ConfigsRootPath

ConfigsRootPath	指令功能	配置根路径
	参数项	参数无
	使用举例	S1=ConfigsRootPath( )//配置根目录返回到 S1

### 3.47 ConfigsPath

string ConfigsPath()

string ConfigsPath

ConfigsPath	指令功能	配置路径
	参数项	参数无
	使用举例	S1=ConfigsPath()//配置路径目录返回到 S1

### 3.48 TempPath

string TempPath()

string TempPath

TempPath	指令功能	临时文件路径
	参数项	参数无
	使用举例	S1=TempPath( )//临时文件路径返回到 S1

### 3.49 sin

digital sin(digital)

sin	指令功能	返回某个角的正弦值		
	参数项	参数	数字类型	值,数字常量或 V0~V999
	返回值	计算结果		
	使用实例	V2=Sin(V1)//计算 V1 的正弦值, 返回到 V2		

## 3.50 cos

digital cos(digital)

cos	指令功能	返回某个角的余弦值		
	参数项	参数	数字类型	值,数字常量或 V0~V999
	返回值	计算结果		
	使用实例	V2=Cos(V1)//计算 V1 的余弦值返回到 V1		

## 3.51 tan

digital tan(digital)

Tan	指令功能	返回某个角的正切值		
	参数项	参数	数字类型	值, 数字常量或 V0~V999
	返回值	计算结果		
	使用实例	V2=Tan(V1)//计算 V1 的正切值返回到 V1		

## 3.52 asin

digital asin(digital)

asin	指令功能	求 arcsinx 的值, 其中 $-1 \leq x \leq 1$		
	参数项	参数	数字类型	值, 数字常量或 V0~V999
	返回值	计算结果		
	使用实例	V2=Asin(V1)//计算 V1 的反正弦值返回到 V2		

## 3.53 acos

digital acos(digital)

acos	指令功能	求 arccosx 的值, 其中 $-1 \leq x \leq 1$		
	参数项	参数	数字类型	值, 数字常量或 V0~V999
	返回值	计算结果		
	使用实例	V2=Acos(V1)//计算 V1 的反余弦值返回到 V2		

## 3.54 atan

digital atan(digital)

atan	指令功能	求 arctanx 的值		
	参数项	参数	数字类型	值,数字常量或 V0~V999
	返回值	计算结果		
	使用实例	Atan(V1) //计算 V1 的反正切值返回到 V2		

## 3.55 atan2

digital atan2(digital,digital)

Atan2	指令功能	求 $\arctan x/y$ 的值		
	参数项	参数 1	数字类型	值, 数字常量或 V0~V999
		参数 2	数字类型	值, 数字常量或 V0~V999
	返回值	计算结果		
使用实例	V3=Atan2 (V1, V2) //计算 V1/V2 的反正切值返回到 V3			

## 3.56 sinh

digital sinh(digital)

sinh	指令功能	计算 x 的双曲正弦函数 $\sinh x$ 的值		
	参数项	参数	数字类型	值, 数字常量或 V0~V999
		返回值	计算结果	
	使用实例	V2=Sinh (V1) //计算 V1 的双曲正弦函数返回到 V2		

## 3.57 cosh

digital cosh(digital)

cosh	指令功能	计算 x 的双曲余弦函数 $\cosh x$ 的值		
	参数项	参数	数字类型	值, 数字常量或 V0~V999
		返回值	计算结果	
	使用实例	V2=Cosh (V1) //计算 V1 的双曲余弦函数返回到 V2		

## 3.58 tanh

digital tanh(digital)

tanh	指令功能	计算 x 的双曲正切函数 $\tanh x$ 的值		
	参数项	参数	数字类型	值, 数字常量或 V0~V999
		返回值	计算结果	
	使用实例	V2=Tanh (V1) //计算 V1 的双曲正切函数返回到 V2		

## 3.59 sqrt

digital sqrt(digital)

sqrt	指令功能	平方根计算, 其中 $x \geq 0$		
	参数项	参数	数字类型	值, 数字常量或 V0~V999
		返回值	计算结果	
	使用实例	V2=.Sqrt (V1) //计算 V1 的平方根, 返回到 V2		

## 3.60 ExcelOpen

digital ExcelOpen(string)

ExcelOpen	指令功能	打开运行目录		
	参数项	参数	字符串类型	运行目录，字符串或 S/FS 变量 (0~999)
	使用实例	ExcelOpen("points.txt")		

## 3.61 ExcelRead

digital ExcelRead(digital,point|out),

void ExcelRead(digital,point|out,digital|out)

ExcelRead	指令功能	读取运行目录信息		
	参数项	参数 1	数字类型	运行目录信息，数字常量或 V/FV 变量 (0~999)
		参数 2	点位	点，P/FP 变量 (0~999)
		参数 3	数字类型	信息，V/FV 变量 (0~999)
使用实例	ExcelRead(V0, P0, V1)			

## 3.62 ExcelClose

void ExcelClose()

ExcelClose	指令功能	关闭运行目录
	参数项	参数无
	使用实例	ExcelClose()



## 4 其他

1. [\a](#)  
\a 响铃 (BEL), 007
2. [\b](#)  
\b 退格 (BS), 将当前位置移动到前一列, 008
3. [\f](#)  
\f 换页 (FF), 将当前位置移到下页开头, 012
4. [\n](#)  
\n 换行 (LF), 将当前位置移动到下一行开头, 010
5. [\t](#)  
\t 水平制表 (HT), 跳到下一个 TAB 位置, 009
6. [\v](#)  
\v 垂直制表 (VT), 011
7. [\\](#)  
\\ 代表一个反斜线字符, 092
8. [\'](#)  
' 代表一个单引号 (撇号) 字符, 039
9. [\"](#)  
" 代表一个双引号字符, 034
10. [\?](#)  
\? 代表一个问号, 063
11. [\0](#)  
\0 空字符 (NULL), 000
12. " "

# 前台指令篇

## 5.运动指令

### 5.1 MoveL

MoveL	指令功能	直线运动	
	参数项	参数 1~9	用户坐标点
		参数 10~18	关节坐标点
		参数 19	速度, 0~100
		参数 20	加速度, 0~100
		参数 21	单位, %为 1, mm 为 2
		参数 22	平滑, 0~100
		参数 23	工具坐标系序号, 0~10
		参数 24	用户坐标系序号, 0~10
使用实例	MoveL(883.908,116.618,534.591,-24.836,17.038,15.456,0.000,0.000,0.000, 7.880,-34.020,54.879,-22.191,0.000,-1.851,0.000,0.000,0.000,10,0,1,20,01,02)//直线运动 S=10% B=20 T=01 U=02		

### 5.2 MoveC

MoveC	指令功能	圆弧运动	
	参数项	参数 1~9	用户坐标点
		参数 10~18	关节坐标点
		参数 19	速度, 0~100
		参数 20	加速度, 0~100
		参数 21	单位, %为 1, mm 为 2
		参数 22	平滑, 0~100
		参数 23	工具坐标系序号, 0~10
		参数 24	用户坐标系序号, 0~10
使用实例	MoveC(848.000,0.000,1043.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,10,0,1,0,00,00)//圆弧运动 S=10%		

## 5.3 MoveFC

MoveFC	指令功能	整圆运动	
	参数项	参数 1~9	用户坐标点
		参数 10~18	关节坐标点
		参数 19	速度, 0~100
		参数 20	加速度, 0~100
		参数 21	单位, %为 1, mm 为 2
		参数 22	平滑, 0~100
		参数 23	工具坐标系序号, 0~10
		参数 24	用户坐标系序号, 0~10
使用实例	MoveFC(848.000,0.000,1043.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,10,0,1,0,00,00) )//整圆运动 S=10%		

## 5.4 MoveJ

MoveJ	指令功能	关节运动	
	参数项	参数 1~9	用户坐标点
		参数 10~18	关节坐标点
		参数 19	速度, 0~100
		参数 20	加速度, 0~100
		参数 21	单位, %为 1, mm 为 2
		参数 22	平滑, 0~100
		参数 23	工具坐标系序号, 0~10
		参数 24	用户坐标系序号, 0~10
使用实例	MoveJ(848.000,0.000,1043.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,10,0,1,0,00,00)/ //关节运动 S=10%		

## 5.5 MoveO

void MoveO(digital,digital,digital,digital,digital,digital,digital,digital,digital,  
digital,digital, digital,digital, digital, digital,digital)

MoveO	指令功能	增量		
	参数项	参数 1~9	数字类型	空间坐标或关节
		参数 10		增量类型，世界增量为 0，工具增量为 1，用户增量为 2
		参数 11		速度，0~100
		参数 12		加速度，0~100
		参数 13		单位，%为 1，mm 为 2
		参数 14		平滑，0~100
		参数 15		工具坐标系序号，0~10
		参数 16		用户坐标系序号，0~10
使用实例	MoveO(0,0,0,0,0,0,0,0,0,0,25,25,1,0,00,00)//世界增量运动 S=25% A=25%			

## 5.6 MoveNurbs

MoveNurbs	指令功能	曲线运动	
	参数项	参数 1~9	用户坐标点
		参数 10~18	关节坐标点
		参数 19	速度，0~100
		参数 20	加速度，0~100
		参数 21	单位，%为 1，mm 为 2
		参数 22	平滑，0~100
		参数 23	工具坐标系序号，0~10
		参数 24	用户坐标系序号，0~10
使用实例	MoveJ(848.000,0.000,1043.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,10,0,1,0,00,00)//曲线运动 S=10%		

## 5.7 SetSimultaneousMode

void SetSimultaneousMode(digital)

SetSimultaneousMode	指令功能	变位机协同		
	参数项	参数	数字常量	模式，关闭为 0，点动为 1，联动 7 轴为 2，联动 8 轴为 3，联动 9 轴为 4，协同 7 轴为 5，协同 8 轴为 6，协同 9 轴为 7
	使用实例	SetSimultaneousMode(1)//变位机协同 模式 点动		

## 5.8 MoveS

MoveS	指令功能	变位机运动	
	参数项	参数 1~9	用户坐标点
		参数 10~18	关节坐标点
		参数 19	速度, 0~100
		参数 20	加速度, 0~100
		参数 21	单位, %为 1, mm 为 2
		参数 22	工具坐标系序号, 0~10
	参数 23	用户坐标系序号, 0~10	
使用实例	MoveS(848.000,0.000,1043.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,100,0,1,0,05,04)//变位机运动 S=100% T=05 U=04		

## 5.9 SetOffset

void SetOffset(digital, digital, digital, digital)

SetOffset	指令功能	设置偏移		
	参数项	参数 1	数字类型	X 增量, 正负 999
		参数 2	数字类型	Y 增量, 正负 999
		参数 3	数字类型	Z 增量, 正负 999
		参数 4	数字类型	用户坐标系序号, 0~10
使用实例	SetOffset(20,10,10,1)//设置位置偏移 X=20 Y=10 Z=10 U=1			

## 5.10 SetToolParam

void SetToolParam(digital, digital, digital, digital, digital, digital, digital)

SetToolParam	指令功能	设置工具参数		
	参数项	参数 1	数字类型	工具坐标系序号, 1~10
		参数 2		X 值, 正负 10000
		参数 3		Y 值, 正负 10000
		参数 4		Z 值, 正负 10000
		参数 5		A 值, 正负 180
		参数 6		B 值, 正负 180
		参数 7		C 值, 正负 180
使用实例	SetToolParam(1,1,6,5,2,5,3)//设置工具 1 参数 X=1 Y=6 Z=5 A=2 B=5 C=3			

## 5.11 SetUserParam

void SeUserParam(digital, digital, digital, digital, digital, digital, digital)

SetUserParam	指令功能	设置用户参数		
	参数项	参数 1	数字类型	用户坐标系序号, 1~10
		参数 2		X 值, 正负 10000
		参数 3		Y 值, 正负 10000
		参数 4		Z 值, 正负 10000
		参数 5		A 值, 正负 180
		参数 6		B 值, 正负 180
		参数 7		C 值, 正负 180
使用实例	SetUserParam(1,1,2,3,4,5,6)//设置用户 1 参数 X=1 Y=2 Z=3 A=4 B=5 C=6			

## 5.12 GETCURPOINT

void GETCURPOINT(point|out)

GETCURPOINT	指令功能	获取当前位置		
	参数项	参数	点位	坐标点 P 变量 (0~999) , out 类型
	使用实例	GetCurPoint(P0)//获取当前位置到 P0		

## 5.13 POINTCOPY

void POINTCOPY(point|out,point)

POINTCOPY	指令功能	点复制		
	参数项	参数 1	点位	Dest, P 变量 (0~999)
		参数 2	点位	Src, P 变量 (0~999)
使用实例	PointCopy(P1,P0)//坐标点复制 P0 复制到 P1			

## 5.14 POINTRESET

void POINTRESET(point|out)

POINTRESET	指令功能	清空坐标点		
	参数项	参数	点位	坐标点 P 变量 (0~999) , out 类型
	使用实例	PointReset(P0)//清空坐标点 P0		

## 5.15 POINTSET

void POINTSET(point|out,digital,digital,digital)

POINTSET	指令功能	坐标点修改		
	参数项	参数 1	点位	点索引, P 变量 (0~999)
		参数 2	数字类型	坐标类型, 关节为 1, 直角为 2
		参数 3	数字类型	坐标系属性索引 (1~9)
		参数 4	数字类型	设置值, V/FV 变量 (0~999) 或数字常量
使用实例	PointSet(P0,1,1,2)//坐标点 P0 关节 1 改为 2			

## 5.16 POINTGET

void POINTGET(point,digital,digital,digital|out)

POINTGET	指令功能	坐标点读取		
	参数项	参数 1	点位	点索引, P 变量 (0~999)
		参数 2	数字类型	坐标类型, 关节为 1, 直角为 2
		参数 3	数字类型	坐标系属性索引 (1~9)
		参数 4	数字类型	读取值, V/FV 变量 (0~999)
使用实例	PointGet(P0,1,1,V1)//坐标点 P0 关节 1 读取到 V1			

## 5.17 POINTADDOFFSET

void POINTADDOFFSET(point|out,digital,digital,digital)

POINTADDOFFSET	指令功能	坐标点值加偏移		
	参数项	参数 1	点位	点索引, P 变量 (0~999)
		参数 2	数字类型	坐标类型, 关节为 1, 直角为 2
		参数 3	数字类型	坐标系属性索引 (1~9)
		参数 4	数字类型	偏移值, 可以为负数, V/FV 变量 (0~999) 或数字常量
使用实例	PointAddOffset(P0,1,1,1)//坐标点 P0 关节 1 增加 1			

## 5.18 CONVEY

void CONVEY(digital,digital)

CONVEY	指令功能	传送带状态控制		
	参数项	参数 1	数字类型	第几轴
		参数 2	数字类型	1 为使能, 0 为不使能
使用实例	CONVEY (2, 1) //第 2 轴使能			

## 5.19 MovingTrigger

void MovingTrigger(digital,digital,digital,digital,digital)

MovingTrigger	指令功能	运动中触发动作		
	参数项	参数 1	数字类型	触发 type, 开始距离触发为 1, 开始时间触发为 2, 停止距离触发为 3, 停止时间触发为 4
		参数 2	数字类型	触发值, 数字常量或 V/FV 变量 (0~999)
		参数 3	数字类型	指定 IO 模拟量 (DO/DA)或系统 V 变量类型, DO 为 2, DA 为 3, V 为 5
		参数 4	数字类型	参数 3 的索引, 数字常量 (0~999) 或 V/FV 变量 (0~999)
		参数 5	数字类型	指定参数 3 和 4 的值, 数字常量或 V/FV 变量 (0~999)
使用实例	MovingTrigger(1,5,5,2,V1)//运动开始延迟触发,延迟 5 毫米时,将 V[2]更改为 V1			

## 5.20 MovingUntil

void MovingUntil(digital,string)

MovingUntil	指令功能	运动中触发停止		
	参数项	参数 1	数字类型	触发类型, 程序减速停为 1, 程序急速停为 2, 指令减速停为 3, 指令急速停为 4
		参数 2	字符串类型	触发表达式
使用实例	MovingUntil(1,"V1==V2")//运动中触发,当 V1==V2 时,程序减速停			

## 5.21 PAUSE

void PAUSE()

PAUSE	指令功能	程序暂停
	参数项	参数无
	使用实例	PAUSE( )



## 6.焊接指令

### 6.1 ARCSTART

void ARCSTART(digital)

ARCSTART	指令功能	引弧		
	参数项	参数	数字类型	文件索引, 01~20
	使用实例	ArcStart(01)//引弧 参数 01		

### 6.2 WeldS

digital WeldS(), digital WeldS

WeldS	指令功能	焊接		
	参数项	参数无		
	使用实例	Welds ()		

### 6.3 ARCSET

void ARCSET(digital,digital)

ARCSET	指令功能	立即设置电压电流		
	参数项	参数 1	数字类型	电压值, 数字常量 (0~1000) 或 V/FV 变量 (0~999)
		参数 2	数字类型	电流值, 数字常量 (0~1000) 或 V/FV 变量 (0~999)
	使用实例	ArcSet(20,10)//立即设置 电压=20 电流=10		

### 6.4 ARCEAD

void ARCEAD(digital)

ARCEAD	指令功能	收弧		
	参数项	参数	数字类型	文件索引,01~20
	使用实例	ArcEnd(01)//收弧 参数 01		

## 6.5 ArcWireFeed

void ArcWireFeed(digital)

ArcWireFeed	指令功能	送丝		
	参数项	参数	数字类型	送丝时间 (0~99) , 单位: 秒
	使用实例	ArcWireFeed(1)//送丝 1 秒		

## 6.6 ArcWireBack

void ArcWireBack(digital)

ArcWireBack	指令功能	退丝		
	参数项	参数	数字类型	退丝时间 (0~99) , 单位: 秒
	使用实例	ArcWireBack(1)//退丝 1 秒		

## 6.7 WEAVESTART

void WEAVESTART(digital)

WEAVESTART	指令功能	摆弧开		
	参数项	参数	数字类型	文件索引
	使用实例	WeaveStart(01)//摆动开始 参数 01		

## 6.8 WEAVEEND

void WEAVEEND()

WEAVEEND	指令功能	摆弧关		
	参数项	参数无		
	使用实例	WeaveEnd()//摆动结束		

## 6.9 WELDSPOTSTART

void WELDSPOTSTART(digital)

WELDSPOTSTART	指令功能	点焊开		
	参数项	参数	数字类型	文件索引, 01~20
	使用实例	WeldSpotStart(01)//点焊开始 参数 01		

## 6.10 WELDSPOTEND

void WELDSPOTEND()

WELDSPOTEND	指令功能	点焊关		
	参数项	参数无		
	使用实例	WeldSpotEnd()//点焊结束		

## 6.11 WeldTurnBackStart

void WeldTurnBackStart(digital,digital)

WeldTurnBackStart	指令功能	折返运动开		
	参数项	参数 1	数字类型	前进距离, 数字常量 (0~100) 或 V/FV 变量 (0~999)
		参数 2	数字类型	后退距离, 数字常量 (0~100) 或 V/FV 变量 (0~999)
使用实例	WeldTurnBackStart(10,5)//折返开始 前进距离 10 毫米 后退距离 5 毫米			

## 6.12 WeldTurnBackEnd

void WeldTurnBackEnd()

WeldTurnBackEnd	指令功能	折返运动关		
	参数项	参数无		
	使用实例	WeldTurnBackEnd()//折返结束		

## 6.13 SearchStart

void SearchStart(digital)

SearchStart	指令功能	寻位运动开始		
	参数项	参数	数字类型	文件索引, 01~999
	使用实例	SearchStart(01)//寻位开始 参数 01		

## 6.14 SearchMoveL

SearchStart	指令功能	寻位运动		
	参数项	参数 1~9	用户坐标	
		参数 10~18	关节坐标	
		参数 19	X 方向为 1, Y 方向为 2, Z 方向为 3, R 方向为 4	
		参数 20	工具坐标系序号, 00~10	
		参数 21	用户坐标系序号, 00~10	
使用实例	SearchMoveL(848.000,0.000,1043.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,1,00,00)//寻位运动 X 方向			

## 6.15 SearchEnd

void SearchEnd(digital)

SearchEnd	指令功能	寻位运动结束		
	参数项	参数	数字类型	寄存器, 01~999
	使用实例	SearchEnd(01)//寻位结束 结果存入寄存器 01		

## 6.16 WasltSearchStart

void WasltSearchStart(digital)

WasltSearchStart	指令功能	激光寻位开始		
	参数项	参数	数字类型	文件索引, 01~999
	使用实例	WasltSearchStart(01)//激光寻位开始 参数 01		

## 6.17 WasItSearchMoveL

WasItSearchMoveL	指令功能	激光寻位运动		
	参数项	参数 1~9	用户坐标	
		参数 10~18	关节坐标	
		参数 19	X 方向为 1, Y 方向为 2, Z 方向为 3	
		参数 20	用户坐标系序号, 00~10	
		参数 21	工具坐标系序号, 00~10	
使用实例	WasItSearchMoveL(848.000,0.000,1043.000,0.000,1,00,00)//激光寻位运动 X 方向			

## 6.18 WasItSearchEnd

void WasItSearchEnd(digital)

WasItSearchEnd	指令功能	激光寻位结束		
	参数项	参数	数字类型	寄存器, 01~999
	使用实例	WasItSearchEnd(01)//激光寻位结束 结果存入寄存器 01		

## 6.19 TouchOffsetStart

void TouchOffsetStart(digital)

TouchOffsetStart	指令功能	开启寻位偏移		
	参数项	参数	数字类型	寄存器, 01~999
	使用实例	TouchOffsetStart(01)//开启寻位偏移 寄存器 01		

## 6.20 TouchOffsetEnd

void TouchOffsetEnd ( )

TouchOffsetEnd	指令功能	关闭寻位偏移		
	参数项	参数无		
	使用实例	TouchOffsetEnd()//关闭寻位偏移		

## 6.21 WASLTFULLSTART

void WASLTFULLSTART(digital,digital,digital,digital,digital)

WASLTFULLSTART	指令功能	激光跟踪开始		
	参数项	参数 1	数字类型	激光号, 数字常量 (1~10) 或 V/FV 变量 (0~999)
		参数 2	数字类型	横向补偿系数, 数字常量 (0~100) 或 V/FV 变量 (0~999)
		参数 3	数字类型	纵向补偿系数, 数字常量 (0~100) 或 V/FV 变量 (0~999)
		参数 4	数字类型	横向偏移, 数字常量 (正负 100) 或 V/FV 变量 (0~999)
		参数 5	数字类型	纵向偏移, 数字常量 (正负 100) 或 V/FV 变量 (0~999)
使用实例	WaslFullStart(1,1,2,10,5)//激光跟踪开始 激光号 1 横向补偿系数 1 纵向补偿系数 2 横向偏移 10 纵向偏移 5			

## 6.22 WASLTSTART

void WASLTSTART(digital,digital,digital,digital)

WASLTSTART	指令功能	激光趋近开始		
	参数项	参数 1	数字类型	补偿系数,数字常量 (0~1) 或 V/FV 变量 (0~999)
		参数 2	数字类型	跟踪保持距离, 数字常量 (0~100) 或 FV 变量 (0~999)
使用实例	WaslStart(0,20)//激光趋近开始 补偿系数 0 跟踪保持距离 20 毫米			

## 6.23 WASLTEND

void WASLTEND()

WASLTEND	指令功能	激光跟踪结束
	参数项	参数无
	使用实例	WaslEnd()//激光跟踪结束

## 6.24 WASLTOPERATION

void WASLTOPERATION(digital,digital)

WASLTOPERATION	指令功能	激光器操作		
	参数项	参数 1	数字类型	操作符, 数字常量 (1~9) 或 V/FV 变量 (0~999)
		参数 2	数字类型	操作码, 数字常量 (1~63)或 V/FV 变量 (0~999)
使用实例	WaslOperation(1,1)//激光跟踪器操作操作符 1 操作码 1			

## 6.25 WasltParameterSet

void WasltParameterSet(digital,digital)

WasltParameterSet	指令功能	激光跟踪器参数设置		
	参数项	参数 1	数字类型	激光跟踪器参数, 激光为 0, 激光视觉为 1, 激光和激光视觉为 2, 调用激光参数为 3
		参数 2	数字类型	激光开关, OFF 为 0, ON 为 1 激光参数, 数字常量 (1~63) 或 V/FV 变量 (0~999)
使用实例	WasltParameterSet(0,0)//激光跟踪器参数设置 激光 OFF			

## 6.26 WASLTGETCURPOINT

void WASLTGETCURPOINT(point|out)

WASLTGETCURPOINT	指令功能	激光寻点		
	参数项	参数	点位	坐标点 P/FP 变量 (0~999) , out 类型
	使用实例	WaslGetCurPoint(P0)//激光寻点 存入 P0		

## 6.27 WASLTGETINTERSECTION

void WASLTGETINTERSECTION(point|out,point,point,point,point)

WASLTGETINTERSECTION	指令功能	激光计算交点		
	参数项	参数 1	点位	交点存入点, P/FP 变量 (0~999)
		参数 2	点位	基准点 1, P/FP 变量 (0~999)
		参数 3	点位	基准点 2, P/FP 变量 (0~999)
		参数 4	点位	基准点 3, P/FP 变量 (0~999)
		参数 5	点位	基准点 4, P/FP 变量 (0~999)
使用实例	WasltGetIntersection(P0,P1,P2,P3,P4)// 激光计算交点 存入 P0 基准点 P1 P2 P3 P4			

## 6.28 ArcTrackStart

void ArcTrackStart(digital)

ArcTrackStart	指令功能	电弧跟踪开始		
	参数项	参数	数字类型	电弧跟踪参数, 01~99
	使用实例	ArcTrackStart(01)//电弧跟踪开始 参数 01		

## 6.29 ArcTrackSample

void ArcTrackSample(digital,digital,digital)

ArcTrackSample	指令功能	电弧跟踪采样		
	参数项	参数 1	数字类型	电弧跟踪参数, 01~99
		参数 2	数字类型	采样开始时间, 0~1000
		参数 3	数字类型	采样结束时间, 0~1000
使用实例	ArcTrackSample(01,20,10)//电弧跟踪采样 参数 01 开始周期 20 结束周期 10			



## 6.30 ArcTrackDir

Void ArcTrackDir(digital,digital,digital,digital,digital,digital,digital,digital,digital,digital,digital,digital,  
digital,digital,digital,digital,digital,digital,digital,digital)

ArcTrackDir	指令功能	电弧跟踪方向设置		
	参数项	参数 1~9	用户坐标	
		参数 10~18	关节坐标	
		参数 19	工具坐标系序号, 00~10	
		参数 20	用户坐标系序号, 00~10	
使用实例	ArcTrackDir(848.000,0.000,1043.000,01,00) )//电弧跟踪方向设置			

## 7 锁螺丝指令

### 7.1 LockScrewSetParam

void LockScrewSetParam(digital)

LockScrewSetParam	指令功能	设置锁螺丝参数		
	参数项	参数	数字类型	锁螺丝参数,01~10
	使用实例	LockScrewSetParam(01)//设置锁螺丝参数 01		

## 7.2 LockScrewMove

```
voidLockScrewMove(digital,digital,digital,digital,digital,digital,digital,digital,digital,digital,digital,digit  
al,digital,digital,digital,digital,digital,digital,digital,digital,digital,digital,digital,digital, digital,digital),  
voidLockScrewMove(digital,digital,digital,digital,digital,digital,digital,digital,digital,digital,digital,digit  
al,digital,digital,digital,digital),  
void LockScrewMove(point, digital,digital,digital,digital,digital,digital)
```

LockScrewMove	指令功能	锁螺丝运动	
	参数项	参数 1~9	用户坐标
		参数 10~18	关节坐标
		参数 19	速度, 0~100
		参数 20	加速度, 0~100
		参数 21	单位, %为 1, mm 为 2
		参数 22	平滑, 0~100
		参数 23	工具坐标系序号, 00~10
		参数 24	用户坐标系序号, 00~10
	使用实例	LockScrewMove(848.000,0.000,1043.000,0.000,0.000,0.000, 0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000, ,0.000,0.000,0,0,2,0,00,00)//锁螺丝运动	



### 8.3 SolderingMoveEnd

SolderingMoveEnd	指令功能	拖焊结束	
	参数项	参数 1~9	用户坐标
		参数 10~18	关节坐标
		参数 19	速度, 0~100
		参数 20	加速度, 0~100
		参数 21	单位, %为 1, mm 为 2
		参数 22	平滑, 0~100
		参数 23	工具坐标系序号, 00~10
		参数 24	用户坐标系序号, 00~10
使用实例	SolderingMoveEnd(848.000,0.000,1043.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0,0,2,0,01,02)// 拖焊结束 T=01 U=02		

### 8.4 SolderingMovePoint

SolderingMovePoint	指令功能	单点焊	
	参数项	参数 1~9	用户坐标
		参数 10~18	关节坐标
		参数 19	速度, 0~100
		参数 20	加速度, 0~100
		参数 21	单位, %为 1, mm 为 2
		参数 22	平滑, 0~100
		参数 23	工具坐标系序号, 00~10
		参数 24	用户坐标系序号, 00~10
使用实例	SolderingMovePoint(848.000,0.000,1043.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0,0,2,0,02,03)//单点焊 T=02 U=03		

## 8.5 SolderingClearIron

void SolderingClearIron()

SolderingClearIron	指令功能	焊接烙铁
	参数项	参数无
	使用实例	SolderingClearIron ()

## 9 冲压指令

### 9.1 PunchMoveL

PunchMoveL	指令功能	冲压直线运动	
	参数项	参数 1~9	用户坐标
		参数 10~18	关节坐标
		参数 19	速度, 0~100
		参数 20	加速度, 0~100
		参数 21	单位, %为 1, mm 为 2
		参数 22	平滑, 0~100
		参数 23	工具坐标系序号, 00~10
		参数 24	用户坐标系序号, 00~10
		参数 25	主指令, 不启用为 0, 等待允许取料为 1, 取料指令为 2, 取料完成检测为 3, 运动到抛料点为 4, 等待允许放料为 5, 放料指令为 6, 二次冲压指令为 7
参数 26	状态, 空为 0, 启用为 1		
使用实例	PunchMoveL(848.000,0.000,1043.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,10,0,1,100,01,02,0,1)//冲压直线运动 S=10% B=100 T=01 U=02 I=不启用 E=启用		

## 9.2 PunchMoveJ

PunchMoveJ	指令功能	冲压关节运动	
	参数项	参数 1~9	用户坐标
		参数 10~18	关节坐标
		参数 19	速度, 0~100
		参数 20	加速度, 0~100
		参数 21	单位, %为 1, mm 为 2
		参数 22	平滑, 0~100
		参数 23	工具坐标系序号, 00~10
		参数 24	用户坐标系序号, 00~10
		参数 25	主指令, 不启用为 0, 等待允许取料为 1, 取料指令为 2, 取料完成检测为 3, 运动到抛料点为 4, 等待允许放料为 5, 放料指令为 6, 二次冲压指令为 7
		参数 26	状态, 空为 0, 启用为 1
使用实例	PunchMoveJ(848.000,0.000,1043.000,0.000,10,0,1,0,01,00,0,0)//冲压关节运动 S=10% T=01 I=不启用 E=空		

## 9.3 PunchAuxiliaryInstruction

void PunchAuxiliaryInstruction(digital)

PunchAuxiliaryInstruction	指令功能	冲压辅助指令		
	参数项	参数	数字类型	冲压辅助指令，等待冲压完成为 0，冲压动作为 1，停止运行为 2，吸盘检测为 3，夹具 1 打开为 4，夹具 1 关闭为 5，夹具 2 打开为 6，夹具 2 关闭为 7
	使用实例	PunchAuxiliaryInstruction(0)//冲压辅助指令 1.等待冲压完成		

# 10 视觉指令

## 10.1 VisualOpen

void VisualOpen(digital)

VisualOpen	指令功能	开启视觉		
	参数项	参数	数字类型	视觉参数，01~10
	使用实例	VisualOpen(01)//开启视觉 参数 01		

## 10.2 VisualClose

void VisualClose(digital)

VisualClose	指令功能	关闭视觉		
	参数项	参数	数字类型	视觉参数，01~10
	使用实例	VisualClose(01)//关闭视觉 参数 01		

## 10.3 VisualRecvPoint

void VisualRecvPoint(digital,point[out])

VisualClose	指令功能	接收视觉坐标点		
	参数项	参数 1	数字类型	视觉参数，01~10
		参数 2	点位	存入指定的 P 点，P/FP 变量 (0~999)
使用实例	VisualRecvPoint(01,P0)//接收视觉坐标点 参数 01 保存到 P0			

## 10.4 VisualRecv

void VisualRecv(digital,string|out,int)

VisualRecv	指令功能	接收视觉数据		
	参数项	参数 1	数字类型	视觉参数, 01~10
		参数 2	字符串类型	存入指定的 S/FS 变量 (0~999)
		参数 3	数字类型	接收视觉数据的长度, 数字常量或 V/FV 变量 (0~999)
使用实例	VisualRecv(01,S1,1)//接收视觉数据 参数 01 保存到 S1 接收长度 1 字节			

## 10.5 VisualSend

void VisualSend(digital,string)

VisuaSend	指令功能	发送数据		
	参数项	参数 1	数字类型	视觉参数, 01~10
		参数 2	字符串常量	发送指令设置的数据, 字符串或 S/FS 变量 (0~999)
使用实例	VisualSend(01,"qwe")//发送视觉数据 参数 01 发送 "qwe"			

## 10.6 VisualTrigger

void VisualTrigger(digital)

VisualTrigger	指令功能	触发拍照		
	参数项	参数	数字类型	视觉参数, 01~10
	使用实例	VisualTrigger(01)//触发拍照 参数 01		

## 11.7 VisualConversion

void VisualConversion(digital,point|out)

VisualConversion	指令功能	视觉转换		
	参数项	参数 1	数字类型	视觉参数, 01~10
		参数 2	点位	点, P/FP 变量 (0~999)
使用实例	VisualConversion (01, P1)			



# 12 跟踪指令

## 12.1 FollowInit

void FollowInit(digital)

FollowInit	指令功能	初始化跟踪		
	参数项	参数	数字类型	视觉参数, 01~10
	使用实例	FollowInit(01)//初始化跟踪 参数 01		

## 12.2 FollowStartSync

void FollowStartSync()

FollowStartSync	指令功能	开始跟踪		
	参数项	参数无		
	使用实例	FollowStartSync()//开始跟踪		

## 12.3 FollowStopSync

void FollowStopSync()

FollowStaopSync	指令功能	结束跟踪		
	参数项	参数无		
	使用实例	FollowStopSync()//停止跟踪		

## 12.4 FollowIsInSpace

void FollowIsInSpace(digital|out)

FollowIsInSpace	指令功能	是否进入跟踪区域		
	参数项	参数	数字类型	将获取工件是否在跟踪范围内的结果, 存在指定的 V 变量中, V/FV 变量 (0~999)
	返回值	0: 表示工件还没有进入到跟踪范围。		
		1: 表示工件在跟踪范围内。 2: 表示工件超出跟踪范围。		
使用实例	FollowIsInSpace(V0)//是否进入到跟踪区域 结果存入 V0			

## 12.5 FollowGetState

void FollowGetState(digital|out)

FollowGetState	指令功能	获取跟踪状态		
	参数项	参数	数字类型	将获取工件是否在跟踪范围内的结果, 存在指定的 V 变量中。V/FV 变量 (0~999)
	返回值	-1: 表示机器人跟踪是出错状态。		
		0: 表示机器人跟踪在空状态。		
		1: 表示机器人跟踪在加速状态。		
2: 表示机器人跟踪在和传送带的同步状态。				
使用实例	FollowGetState(V0)//获取跟踪状态 结果存入 V0			

## 12.6 FollowSetPoint

void FollowSetPoint(point)

FollowSetPoint	指令功能	设置跟踪点位		
	参数项	参数	点位	跟踪点位, P/FP 变量 (0~999)
	使用实例	FollowSetPoint(P0)//设置跟踪点 P0		

## 12.7 FollowGetPoint

void FollowGetPoint(digital,point|out)

FollowGetPoint	指令功能	获取跟踪点位		
	参数项	参数 1	数字类型	视觉参数, 01~10
		参数 2	点位	将从跟踪队列中获取的点, 存在指定的 P 变量的空间坐标系中, P/FP 变量 (0~999)
使用实例	FollowGetPoint(01,P0)//获取跟踪点 参数 01 保存到 P0			

## 12.8 FollowUpdateUser

void FollowUpdateUser(point,digital)

FollowUpdateUser	指令功能	更新用户坐标		
	参数项	参数 1	点位	更改用户坐标的原始 P 点, P/FP 变量 (0~999)
		参数 2	数字类型	选择更新的用户号, 01~10
使用实例	FollowUpdateUser(P0,01)//更新用户坐标 P0 用户=01			

## 12.9 FollowUpdatePointPosition

void FollowUpdatePointPosition(digital|out)

FollowUpdatePointPosition	指令功能	跟新跟踪点位置		
	参数项	参数	数字类型	获取的结果值，存在 V 变量中，V/FV 变量 (0~999)
	使用实例	FollowUpdatePointPosition(V0)//更新跟踪点的位置 结果存入 V0		

## 12.10 SetCircleFollowStartPosition

void SetCircleFollowStartPosition(digital,digital|out)

SetCircleFollowStartPosition	指令功能	设置圆盘跟踪开始位置		
	参数项	参数 1	数字类型	视觉参数，01~10
		参数 2	数字类型	偏移量，数字常量 (0~999) 或 V/FV 变量 (0~999)
使用实例	SetCircleFollowStartPosition(01,0)//设置圆盘跟踪开始位置 参数 01 偏移 0			

# 13 码垛指令

## 13.1 PALLETINIT

void PALLETINIT(digital)

PALLETINIT	指令功能	码垛初始化		
	参数项	参数	数字类型	码垛参数，01~100
	使用实例	PalletInit(01)//码垛初始化 参数 01		

## 13.2 PALLETGETCURPOS

void PALLETGETCURPOS(digital,digital|out,digital|out,digital|out)

PALLETGETCURPOS	指令功能	码垛初始化		
	参数项	参数 1	数字类型	垛数，数字常量 0~999
		参数 2	数字类型	X 值，数字常量 0~999
		参数 3	数字类型	Y 值，数字常量 0~999
		参数 4	数字类型	Z 值，数字常量 0~999
使用实例	PalletGetCurPos(V1,V2,V3,V4)//计算码垛放件点坐标 垛数 V1 X 给 V2 Y 给 V3 Z 给 V4			

# 14 点胶指令

## 14.1 GlueMoveNurbsL

GlueMoveNurbsL	指令功能	点胶直线运动	
	参数项	参数 1~9	用户坐标
		参数 10~18	关节坐标
		参数 19	速度, 0~100
		参数 20	加速度, 0~100
		参数 21	单位, %为 1, mm 为 2
使用实例	GlueMoveNurbsL(848.000,0.000,1043.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,50,50,1,GlueBl,GlueTool,GlueUser)//点胶直线运动 S=50% A=50%		

## 14.2 GlueMoveNurbsC

GlueMoveNurbsC	指令功能	点胶圆弧运动	
	参数项	参数 1~9	用户坐标
		参数 10~18	关节坐标
		参数 19	速度, 0~100
		参数 20	加速度, 0~100
		参数 21	单位, %为 1, mm 为 2
使用实例	GlueMoveNurbsC(848.000,0.000,1043.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,50,50,1,GlueBl,GlueTool,GlueUser)//点胶圆弧运动 S=50% A=50%		

### 14.3 GlueMoveFC

GlueMoveFC	指令功能	点胶整圆运动	
	参数项	参数 1~9	用户坐标
		参数 10~18	关节坐标
		参数 19	速度, 0~100
		参数 20	加速度, 0~100
		参数 21	单位, %为 1, mm 为 2
使用实例	GlueMoveFC(848.000,0.000,1043.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,50,50,1,GlueBl,GlueTool,GlueUser)// 点胶整圆运动 S=50% A=50%		

### 14.4 GlueMoveNurbs

GlueMoveNurbs	指令功能	点胶曲线运动	
	参数项	参数 1~9	用户坐标
		参数 10~18	关节坐标
		参数 19	速度, 0~100
		参数 20	加速度, 0~100
		参数 21	单位, %为 1, mm 为 2
使用实例	GlueMoveNurbs(848.000,0.000,1043.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,50,50,1,GlueBl,GlueTool,GlueUser)// 点胶曲线运动 S=50% A=50%		

### 14.5 GlueSetOCTime

void GlueSetOCTime(digital,digital)

GlueSetOCTime	指令功能	设置点胶参数		
	参数项	参数 1	数字类型	提前开胶时间, 数字常量正负 10 或 V/FV 变量 (0~999)
		参数 2	数字类型	提前关胶时间, 数字变量正负 10 或 V/FV 变量 (0~999)
使用实例	GlueSetOCTime(10,5)//设置点胶参数 提前开胶 10 秒,提前关胶 5 秒			

## 14.6 GlueSetParamNum

void GlueSetParamNum(digital)

GlueSetParamNum	指令功能	选择胶枪		
	参数项	参数	数字类型	胶枪参数, 01~02
	使用实例	GlueSetParamNum(01)//选择胶枪 01		