

COMANDOS E INSTRUCCIONES V +

Nota: aunque a continuación, los comandos e instrucciones van escritos con minúsculas, al programar el editor los escribe a casi todos directamente con mayúsculas.

PUESTA EN MARCHA Y PARADAS DEL ROBOT

<i>Comando-instrucción</i>	<i>Sintaxis - ejemplo</i>	<i>Efecto</i>
Enable power	Enable power (en po)	Habilitar potencia; da corriente al robot
Disable power	Disable power (dis po)	Quita la corriente al brazo robot
Execute	Execute ETI (ex ETI)	Ejecuta, pone en marcha al programa de nombre ETI. El programa a ejecutar debe estar en la RAM; si lo tenemos en un disco, con Load A:ETI lo cargamos en la memoria
	Execute ETI,2,3	Ejecuta ETI 2 veces, la 1ª empezando de la línea o paso 3º
Xstep	Xstep ETI - xstep ETI,2,9	Ejecuta ETI línea a línea de programa. Pulsar X y ↵ para cada línea
En trace	En trace	Aparecen en pantalla las líneas o instrucciones que se están ejecutando en ese momento.
Dis trace	Dis trace	Regresar al modo normal.
Abort	Abort	Tecleando abort + ↵, se detiene el robot al acabar la instrucción en curso
Panic	Panic	Panic + ↵, se detiene el robot inmediatamente.
Proceed	Proceed	Continúa el robot en la instrucción siguiente a la que se estaba ejecutando al producirse el paro. También se emplea para seguir, si hay una instrucción "pause" en el programa
Retry	Retry	Continúa la ejecución del programa, repitiendo la última instrucción.

El panic de teclado, equivale a los pulsadores RUN/HOLD y MAN/HALT del mando manual de aprendizaje. Los P.E. del mando manual y del armario, el selector de frenos del brazo-robot, y el pulsador de "hombre muerto", también hacen el efecto del panic; pero además, activan los frenos y cortan la potencia del brazo-robot.

OPERACIONES CON LOS PROGRAMAS EXISTENTES EN LA RAM

<i>Comando- instrucción</i>	<i>Sintaxis - ejemplo</i>	<i>Efecto</i>
Dir	Dir	Se visualizan en pantalla los títulos de los programas que tenga la RAM.
Listp	Listp ETI	Se visualiza el programa ETI.
Listl	Listl ETI	Se visualizan las coordenadas de los puntos que tiene el programa ETI.
^ C	^ C	Pulsando Ctrl C se anulan las instrucciones listp, listl, dir, store,
^S	^S	Detiene la visualización de las líneas del programa.
^Q	^Q	Anula a ^S
^W	^W	Ralentiza el listp
Copy	Copy Mto=ETI	Hace una copia del programa ETI llamada Mto Copy prog.nuevo=prog.existente.
Rename	Rename Mto=ETI	Mto, será el nuevo nombre de ETI
Delete	Delete ETI	Borra el programa ETI y todos los subprogramas, variables y puntos que dependan de él
Deletep	Deletep ETI	Borra solo el programa ETI
Deletel	Deletel ETI	Borra solo los puntos del programa ETI
Deleter	Deleter ETI	Borra las variables reales de ETI
Testp	Testp ETI	Verifica si existe el programa ETI
Zero	Zero	Esta instrucción, borra todo lo que en ese momento tenga la RAM

OPERACIONES-COMANDOS ASOCIADOS A LOS DISCOS, C: y A:

<i>Comando-instrucción</i>	<i>Sintaxis - ejemplo</i>	<i>Efecto</i>
CD	CD A: CD C: CD C:\ETI\Mto	Cambia al disco A: Cambia al C: También, podemos cambiar con DEF D=. Hacemos: DEF D=A:\ETI\Mto, nos lleva al subdirectorio Mto. Con DEF D= a solas, ascendemos por los subdirectorios
Fdir	Fdir A: (solo fdir si ya estamos en el A:)	Se visualizan los directorios o programas que contiene el disco A:
Fdir/C	Fdir/C A:\ETI\Mto	Se crea el subdirectorio o fichero Mto del directorio de programas ETI
Fdir/D	Fdir/D a:\ETI\Mto	Se suprime el fichero Mto
Flist	Flist a:ETI (solo flist ETI, si ya estamos en el A:)	Se visualiza el contenido del programa ETI
Fdelete	Fdelete ETI	Borra el fichero o programa ETI
Fcopy	Fcopy Mto=ETI	Hace una copia de ETI de nombre Mto
Frename	Frename Mto=ETI	ETI, pasa a llamarse Mto.
Store	Store A:ETI (store ETI, si ya estamos en el A:)	Guarda, almacena en A con nombre ETI todo lo que contenga la RAM, y le pone la extensión .V2 (ETI.V2)
Storep	Storep A:ETI	Guarda, almacena, solo él o los programas que contenga la RAM, y le pone la extensión .PG.
Storel	Storel A:ETI	Guarda, almacena solo los puntos o localizaciones que contenga la RAM, y le pone la extensión .LC.
Load	Load A:ETI (load ETI si ya estamos en el A:)	Carga, copia todo el programa ETI.V2 en la RAM. (Para poder ejecutar un programa, debe estar en la RAM).
	Load ETI.PG	Solo se carga el programa
	Load ETI.LC	Solo se cargan los puntos o localizaciones
Execute	Execute ETI (ex ETI) (ex)	Ejecuta; pone en marcha el programa ETI

ASIGNAR PUNTOS Y VARIABLES

<i>Comando-instrucción</i>	<i>Sintaxis - ejemplo</i>	<i>Efecto</i>
Here	Here a, here b, here pto5	Se mueve el robot en manual, al punto deseado, y se tecldea here a, seguimos instrucciones y el robot asigna esa posición al punto a..
	Here #a	El punto con # es un punto de precisión, el robot asume ese punto con la misma "pose" siempre en sus ejes y articulaciones; a y #a, pueden ser dos puntos distintos
Point	Point a	Point a ↵, tecldeamos las 6 coordenadas (X,Y,Z,y,p,r), seguimos instrucciones, y el robot asigna esas coordenadas al punto a.
	Point # a	Las coordenadas para puntos de precisión, son polares, deberemos tecldear los 6 ángulos solicitados por el robot
Variable real	I =4 n = 32 Pi = (I+n) × 2	El robot le asigna el valor 4 a la variable I, 32 a la n Esto se emplea en programación, y se pueden hacer operaciones con las variables.
Where	Where	Nos da la posición del robot y el estado de la pinza. Where 1, nos actualiza la posición cuando el robot, se esta moviendo; se sale de where 1 con Ctrol+C .

EDITOR

<i>Comando-instrucción</i>	<i>Sintaxis - ejemplo</i>	<i>Efecto</i>
See	See ETI	El control pregunta si quieres crear el programa ETI. (Y/N) ↵. Si el programa ya existe, se entra directamente en él.
Edit	Edit ETI	Se hacen líneas de programa moviendo el robot y grabando cada una con el pulsador REC/DONE (Record) del mando manual de aprendizaje.

Una vez dentro del editor, estamos en modo Comando; para pasar a insertar o a Reemplazar, pulsamos I o R. Para volver al modo comando, pulsamos Esc, y para salir del editor, Ctrl+E. (si hay líneas *cogidas*, y no se puede salir, pulsar Esc y después K). Podemos editar nuevos programas o subprogramas, pulsando F2.

INSTRUCCIONES DEL LENGUAJE V+

<i>Comando-instrucción</i>	<i>Sintaxis - ejemplo</i>	<i>Efecto</i>
Do	Do ready do move a	El robot ejecuta la instrucción seguida del do (haz); en los ejemplos: el robot va a la posición ready o se mueve al punto a
;	;comentario	Cuando estamos programando, lo que escribamos después de ; (punto y coma), no lo tiene en cuenta el robot, nos sirve de comentario o aclaración.
Move	Move A	El robot se mueve al punto A según la trayectoria que él determine. A la vez también puede hacer otras cosas programadas como: cálculos, poner a "1" una salida,.....
Moves	Moves A	El robot va al punto A en línea recta.
Break	break	El robot mientras se mueve, hace cálculos y otras acciones de las siguientes instrucciones, por ejemplo, si va de a hacia b, y de b hacia c, puede que no pase por b, pues antes de llegar ya esta haciendo cálculos para ir a c. Para evitar esto, se intercala un break y el robot ejecutará instrucción por instrucción.
Appro	Appro A,40	El robot, se aproxima a una distancia de 40 mm de A medidos siguiendo el eje Z tool.
Appros	Appros A,40	Idem a appro pero desplazándose en línea recta.
Depart	Depart 30	El robot, se aleja 30 mm del punto en el que se encuentra, medidos en la dirección del eje Z tool.
Departs	Departs 30	Idem a depart pero en línea recta.
speed	Speed 60 sp 60	Al empezar a trabajar, el robot asume una velocidad relativa de 10 sobre 100. Tecleando sp 70 iría a 70 sobre 100. Estas velocidades se llaman velocidad de monitor.

Speed	Speed 40 Always	Speed 40 always dentro del programa, indica que el robot, a partir de esa instrucción, ira a un 40 % de la velocidad de monitor. Si no se especifica, ira a un 100% de la de monitor.
Speed	Speed 38	Si se omite always, el 38% sobre la velocidad de monitor, se aplica solo al movimiento siguiente
Speed	Speed 80 MMPS Always	El robot irá a 80 mm/s si la velocidad de monitor es 100; si por ejemplo es 30, irá a 24 mm/s. Puede ir con Always o sin él
Accel	Accel (7) 50, 60 7 = Perfil 50 = rampa de aceleración. 70 = rampa de frenado	Para suavizar los movimientos de robot, tenemos los perfiles de movimiento: 8, 7, 6 y 5, con rampas de aceleración y de frenado que van del 1% al 100%. Por defecto el robot asume el movimiento más brusco dentro de su velocidad, que es el accel (8) 100, 100. El mas suave es el accel (5) 1, 1
Open	Open	Apertura de la pinza, pero sin pararse a hacerlo; el robot la abre mientras se mueve hacia otro punto
Openi	Openi	Apertura de la pinza cuanto acaba el movimiento anterior, y no sigue hasta que se ha abierto
Close y closei	Close Closei	Idem, pero cierre
Delay	Delay 4	Después de esta instrucción, el robot para 4" aproximadamente. Mínimo delay = 16 ms
Set	Set A=P	Ponle al punto A las coordenadas que tiene el P. A, copia las coordenadas del P
Dest	Set A = dest	El punto A copia, (adopta), las coordenadas del punto de destino al que se dirige el robot en ese momento.
#pdest	Set #A = #pdest	Idem al anterior, si el punto de destino es de precisión.
Shift (..BY...)	Set A = shift (P BY -20, 0, k*50)	El punto A copia (adopta), las coordenadas del P, pero cambiadas -20 mm en el eje X, 0 mm en el Y, y 50 mm multiplicados por el valor instantáneo de la variable k en el Z. Haciendo set A = shift (A BY 43, 15, 0), cambian las coordenadas X e Y del propio punto A.

Drive	Drive 2,-45,50	Se gira el eje 2, -45 grados a un 50% de la velocidad de monitor.
Rx, Ry, Rz	Ry 30	Se produce un giro de 30° alrededor del Y _{world} .
Pause	Pause	Al encontrarse esta instrucción, el programa, se para. Para que siga, tecleamos proceed.
Halt	Halt	Al encontrarse esta instrucción, el programa, se para. Solo se puede seguir con Execute, y además se ejecuta desde el principio.
Call	Call coger	Si sé esta ejecutando el programa ETI, al llegar a esta instrucción, se ejecuta coger y después, sigue ejecutándose ETI. Estos subprogramas, deben terminar siempre con la instrucción "return".

SALTOS Y TESTS

<i>Comando-instrucción</i>	<i>Sintaxis - ejemplo</i>	<i>Efecto</i>
Goto <i>vete</i>	GOTO 10 (Cuidado con los saltos hacia atrás; pueden producir infinitos bucles, y que no salga de ellos el programa)	Salta a la línea de programa cuya etiqueta o comienzo es 10. (Que no es lo mismo que la décima línea de programa, que la podemos comenzar escribiendo la etiqueta 50 por ejem.)
If.....goto <i>Si.....vete</i>	IF (condición) GOTO 15 If (i==3) or (j>8) goto 15	Si la variable i igual a 3 "o" la j es mayor de 8, vete (salta) a la línea de programa con etiqueta 15. Si no se cumple ninguna de las dos condiciones, pasa a la línea siguiente.
If.....then..... else.....end <i>si.. entonces.... si no.....fin</i>	IF j >= 4 THEN move a ELSE move b END	SI la variable j, es > ó = 4, ENTONCES, muévete al punto a, SI NO lo es, muévete al punto b, FIN. Permite ejecutar unas u otras instrucciones en función de sí se cumplen o no ciertas condiciones.
If.....then.....end <i>Si..entonces.fin</i>	IF j >= 4 THEN move a END	SI j >= 4, ENTONCES muévete al punto a, si no, pasa a la línea siguiente
For...to...step... end <i>De.hasta..paso fin</i>	FOR filas = 1 TO 6 STEP 2 move d move #e END	DE ó para valor de la variable filas 1 HASTA 6, CONTANDO de DOS en DOS, repite estas instrucciones.....FIN. Permite ejecutar cierto número de veces un bloque de instrucciones. Esta instrucción es la más potente, ella sola crea la variable, y la incrementa en cada vuelta

While.....do End <i>Mientras...haz fin</i>	WHILE (co < 3) DO Moves #r Moves #s co = co+1 END	MIENTRAS co sea menor de 3 HAZ (repite) los movimientos al punto #r y al #s FIN. Se ejecuta un bucle de instrucciones mientras se cumpla la condición establecida.
Do.....until <i>Haz.....hasta</i>	DO Move a Move b k = k+1 UNTIL (k > 6)	HACE, se mueve al punto a y después al b HASTA que k sea mayor de 6. Se repite un bloque de instrucciones hasta que se cumpla una condición. Al estar la condición al final, siempre se ejecuta al menos la primera vez.
Case.....of <i>Caso.....de</i>	CASE (piezas) OF VALUE 20, 40: delay 5 VALUE 10, 30: moves c ANY call Mto END	Según los valores que tenga la variable piezas, el robot, esperara 5", se moverá a c, ó se ejecutará el subprograma mto. Se selecciona un bloque de instrucciones a ejecutar en función de los valores que tiene cada expresión. Clarifica los programas al sustituir a los if..goto o if...then...else..end múltiples.

DIALOGO CON EL OPERADOR

<i>Comando- instrucción</i>	<i>Sintaxis - ejemplo</i>	<i>Efecto</i>
Prompt	PROMPT "piezas =", p	Con esta instrucción, podemos asignar valores a las variables. El programa se para hasta que tecleemos el número de piezas p.
Type Type/B, Type/Xn,	TYPE "z vale:", z TYPE /B, "z vale:", z TYPE /X10, "z vale:", z	Con esta instrucción presentamos mensajes en la pantalla mientras se ejecuta el programa, también podemos saber el valor de las variables en ese momento. En el ejemplo, el control nos presenta el valor de z. Hay varios caracteres de control, si ponemos /B, además de lo anterior, el control emite un pitido, con /X10, escribe el mensaje dejando 10 espacios libres en la izquierda, ...
Timer 2	TIMER (2) = 5 TIMER 2 = 5 i = TIMER(2)	Ponemos el temporizador 2 a 5 seg y empieza a cronometrar al llegar a esa instrucción. También le podemos asignar a una variable el valor de un temporizador. El Delay es inexacto, con timer aseguramos cronometrajes y tiempos exactos.

ENTRADAS Y SALIDAS DIGITALES

<i>Comando- instrucción</i>	<i>Sintaxis - ejemplo</i>	<i>Efecto</i>
If sig ...goto ...	If sig(1001,-1002) goto 7	Si la entrada 1001 esta a 1 y la 1002 esta a 0 (signo -) vete a la etiqueta 7. En el robot, las entradas van direccionadas de la 1001 en adelante, y las salidas a partir de la 1
Wait	wait sig(-1002) or sig2	Espera hasta que la entrada 1002 este a 0 “o” la salida 2 este a 1. Esta instrucción interrumpe el programa hasta que la condición sea valida. El wait, también se puede emplear con otras variables, wait j > 3, wait timer(4) >6
Signal	SIGNAL 2, -5	Pone a 1 la salida 2, y a 0 la salida 5. Permite accionar o desaccionar los contactos de salidas digitales exteriores.
React ...	REACT 1006, dejar, 45	Si 1006 se pone a 1, acaba el movimiento en curso y salta al subprograma dejar, con prioridad 45 (si saltamos a dos programas a la vez, primero se haría el de prioridad mas alta, de la 1 a la 127)
Reacti	REACTI 1006, dejar, 45	Idem a la anterior pero salta inmediatamente.
Ignore	IGNORE 1006	Si la 1006 se pone a 1 después de esta instrucción, ya no se hace el salto. A partir de la línea en que se ha programado, ignore, anula al React y al Reacti
Reset	Reset	Pone a 0 todas las salidas
Runsig	RUNSIG 3	Pone a 1 la salida 3, solo mientras se esta ejecutando el programa.
Bits (instrucción)	BITS 2, 4 = 7 BITS 2, 4 = ^B0111	A partir de la salida 2 del robot y cogiendo 4 (de la 2 a la 5), ponlas en el mismo estado que los bits del 7 decimal pasado a binario; o sea, se pondrían a 1 las salidas 2, 3 y 4, y a 0 la 5. Esta instrucción sirve para activar o desactivar hasta 8 salidas con una sola instrucción.
Bits (función)	i = BITS(1004,3)	La variable i tendrá el valor decimal correspondiente al binario que forman los estados de las entradas 1004 a la 1006 (3 entradas). Con esta instrucción, podemos “leer” los estados de hasta 8 señales

CAMBIO DE REFERENCIAL

<i>Comando- instrucción</i>	<i>Sintaxis - ejemplo</i>	<i>Efecto</i>
TOOL	TOOL pinza	Desplaza el origen tool de la brida del robot a la pinza; previamente hay que meter las coordenadas de pinza con respecto al origen tool.
MOVE a:b	MOVE a:b	Muévete a las coordenadas de b pero con origen en el punto a, si a lo cogemos con here, el origen a tendrá la orientación tool Permite desplazarse con respecto a otro punto en vez de con respecto al origen world
FRAME	SET p = FRAME(p1,p2,p3,p4) SET p[i] = FRAME (p1[i],p2[i],p3[i],p4[i])	Permite definir un nuevo origen o referencial que se adapte mejor al puesto de trabajo. El origen esta en p4 (que se suele hacer igual a p1), el eje X, va en la dirección p1-p2, el eje Y va de p4 hacia la zona de p3, y el Z, queda definido por los otros dos. .Para hacer movimientos con respecto a este origen, haremos: move p:a .Si queremos tener distintos orígenes en función del valor de la variable i, emplearemos la 2ª sintaxis.
INVERSE	SET b = INVERSE(p):a	Se utiliza para obtener las coordenadas de un punto con respecto a otro origen.