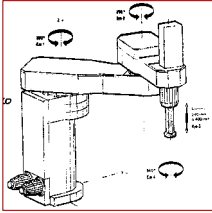


# Introducción a la Robótica



UNIVERSIDAD DE SEVILLA



escuela superior de  
INGENIEROS DE SEVILLA

**UNIVERSIDAD DE SEVILLA**  
**Dpto. De Ingeniería de Sistemas y Automática**  
**Escuela Superior de Ingenieros**

- ◆ **CONCEPTOS GENERALES**
- ◆ **PROGRAMACIÓN Y OPERACIÓN**
- ◆ **PRACTICAS CON SCORBOT-V**



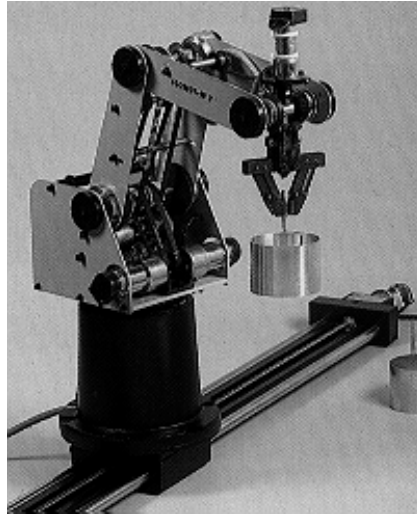
**Francisco R. Rubio**

## Índice

- ◆ **Introducción**
- ◆ **El robot SCORBOT-V**
- ◆ **Lenguaje ACL de programación del robot SCORBOT-V**

# Introducción

## **Robot SCORBOT-V Utilizado para prácticas**



# Introducción

- ◆ Robot de enseñanza
  - Menor carga útil
  - Menor repetitividad y precisión
  - Peor software
  - Suficiente para prácticas
- ◆ Lenguaje de programación: ACL

# Índice

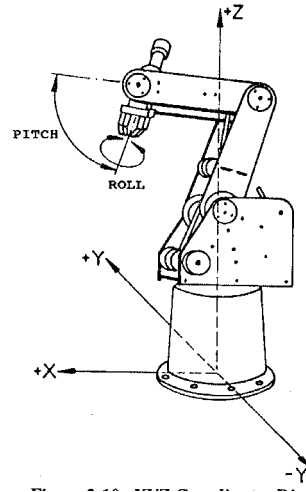
- ◆ Introducción
- ◆ El robot SCORBOT-V
- ◆ Programación del robot SCORBOT-V

## El robot SCORBOT-V

- ◆ Componentes básicos
  - El brazo del robot
  - El controlador
  - La botonera de enseñanza
  - El terminal de programación

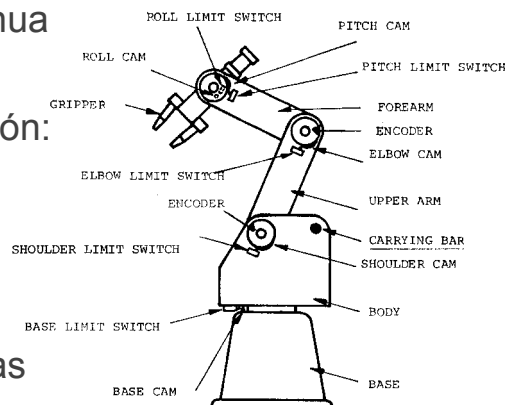
# El brazo del robot

- ◆ Tipo RRR
- ◆ Grados de libertad:
  - Base
  - Hombro
  - Codo
  - Elevación
  - Giro
- ◆ Garra con movimiento casi paralelo



# El brazo del robot

- ◆ Servomotores de continua (24V, 2A) con reductora
- ◆ Codificadores de posición:
  - Ópticos
  - Incrementales
  - Posición HOME por microinterruptores
- ◆ Transmisiones indirectas
  - Engranajes
  - Correas



## El brazo del robot

- ◆ Repetitividad: 0.5 mm
- ◆ Velocidad máxima: 600 mm/s
- ◆ Carga máxima: 1 Kg
- ◆ Peso: 11 Kg aproximadamente

## El robot SCORBOT-V

- ◆ Componentes básicos
  - El brazo del robot
  - El controlador
  - La botonera de enseñanza
  - El terminal de programación

## El controlador

### ◆ Sistema microprocesador:

- Motorola 68010
- 384K EPROM
- 64 K RAM de usuario
- Multitarea en tiempo real
- Sistema operativo propio  
(ordenador sólo como terminal via RS-232)



## El controlador

### ◆ Servocontroladores:

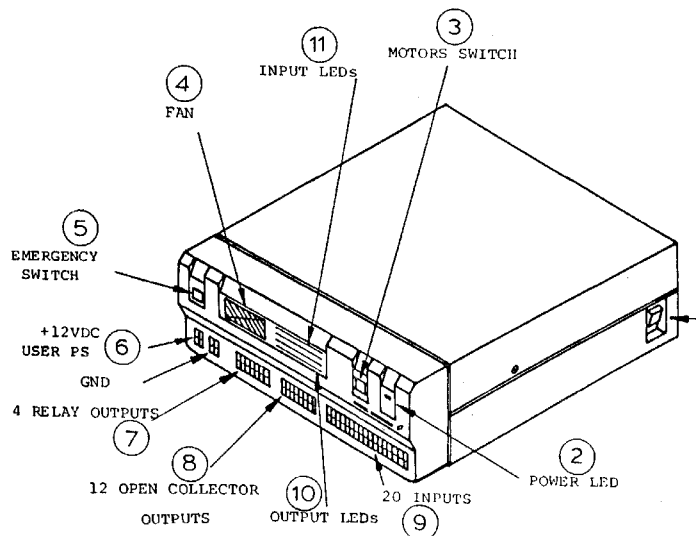
- 4 tarjetas para 8 motores
- Fuente de alimentación interna (24V, 18A) para potencia de motores
- Con microinterruptores como protecciones térmicas



# El controlador

- ◆ Señales de entrada/salida:
  - 16 entradas
  - 4 salidas de tipo relé (1-4)
  - 12 salidas a colector abierto (5-16)
- ◆ Otros:
  - Pulsador de emergencia
  - Interruptor de motores
  - Fuente de alimentación adicional (12 V, 2 A)
  - Indicadores LED para entradas/salidas

# El controlador



# El robot SCORBOT-V

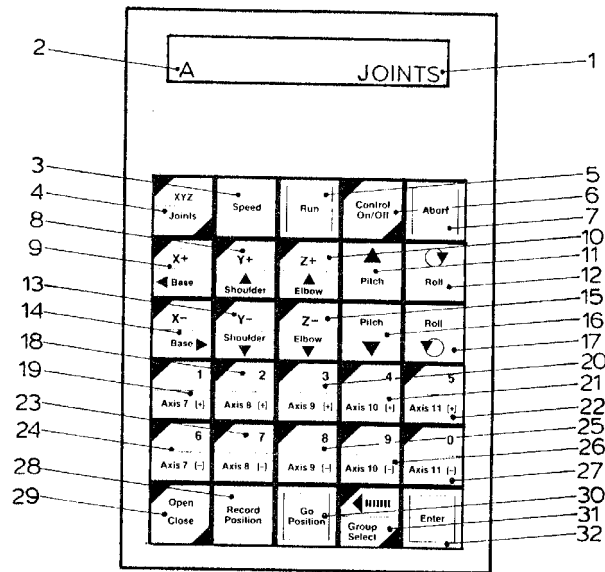
- ◆ Componentes básicos
  - El brazo del robot
  - El controlador
  - La botonera de enseñanza
  - El terminal de programación

## La botonera de enseñanza

- ◆ Mover brazo hasta posición deseada:
  - Coordenadas articulares
  - Coordenadas cartesianas
- ◆ Almacenar posiciones
- ◆ Fijar la velocidad del brazo
- ◆ Mover el brazo a una posición previamente almacenada
- ◆ Ejecutar/abortar programas en ACL
- ◆ Activar/desactivar control de ejes



# La botonera de enseñanza



## El robot SCORBOT-V

- ◆ Componentes básicos
  - El brazo del robot
  - El controlador
  - La botonera de enseñanza
  - El terminal de programación

## El terminal de programación

- ◆ Permite programar al controlador desde una computadora
- ◆ ATS: Advanced Terminal Software

## Índice

- ◆ Introducción
- ◆ El robot SCORBOT-V
- ◆ Programación del robot SCORBOT-V

## **Programación del robot SCORBOT-V**

- ◆ Lenguaje de programación: ACL (Advanced Control Language)
  - Ejecución directa de comandos
  - Control de datos de Entrada/Salida
  - La botonera de enseñanza
  - Programación por parte del usuario
  - Sincronización en la ejecución de programas
  - Manejo simple de ficheros

## **Programación del robot SCORBOT-V**

- ◆ Conexión por medio del puerto serie (RS-232)
- ◆ Modo de ejecución de los comandos:
  - Directos: de ejecución inmediata
  - Indirectos o de edición: escritos en programas

# **Programación del robot SCORBOT-V**

- ◆ Tipos de comandos:
  - Definición y almacenamiento de posiciones
  - Manipulación de posiciones
  - Definición y manipulación de variables numéricas
  - Control de movimientos del robot
  - Tiempo real y control de programas
  - Control del flujo de un programa
  - Control de entradas y salidas
  - Visualización e interfaz con el usuario
  - Otros (interfaz con usuario, ...)

## **Definición y almacenamiento de posiciones**

- ◆ Posiciones conocidas por todos los programas
- ◆ En zona especial de memoria
- ◆ Formato del nombre:
  - ATS: cinco caracteres como máximo
  - Botonera de enseñanza: números
  - Posición '0': HOME

## Definición y almacenamiento de posiciones

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
DEFP <pos>	D, E	Reserva memoria para una posición
DIMP <vpos[n]>	D, E	Reserva memoria para un vector de n posiciones
HERE <pos>	D, E	Graba una posición absoluta en coordenadas articulares
HERE <vpos[2]>		
HERER <pos>	D	Graba una posición relativa a la posición actual del robot
HERER <pos2> <pos1>	D, E	Graba una posición <pos2> relativa a la posición <pos1>

## Definición y almacenamiento de posiciones

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
TEACH <pos>	D	Graba una posición absoluta en coordenadas cartesianas
TEACHR <pos>	D	Graba una posición relativa a la posición actual del robot
TEACHR <pos2> <pos1>	D	Graba una posición <pos2> relativa a la posición <pos1>
UNDEF <pos>	D	Borra las coordenadas de la posición pero sigue estando definida
DELP <pos>	D, E	Borra posición o vector de posiciones de la memoria RAM de usuario

## Manipulación de posiciones

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
SETP <pos2> = <pos1>	D, E	Copia la posición <pos1> en la posición <pos2>
SHIFT <pos> BY <eje> <valor>	D, E	Cambia el valor de un eje de una posición <pos> asignando el nuevo valor al eje especificado
SHIFTC <pos> BY <coord> <valor>	D, E	Igual, pero en en coord. cartesianas <coord> : X, Y, Z, P, R <valor> : en décimas de mm

## Manipulación de posiciones

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
SET <var>=PVAL <pos> <eje>	D, E	Asigna a la variable <var> el valor del <eje> de la posición <pos>
SET <var>=PVALC <pos> <coord>	D, E	Igual, pero con la coordenada <coord>
SETPV <pos> <eje> <val>	D, E	Graba <val> como coordenada para el <eje> de la posición <pos>
SETPVC <pos> <coord> <val>	D, E	Graba <val> como coordenada cartesiana <coord> de la posición <pos>

## Definición de variables numéricas

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
DEFINE <var1> ... <var8>	E	Crea variables “locales”
GLOBAL <var1> ... <var8>	D, E	Crea variables “globales”
DIM <var[n]>	E	Crea un vector de n variables “locales”
DIMG <var[n]>	D, E	Crea un vector de n variables “globales”
DELVAR <var>	D, E	Borra una variable de la RAM de usuario

## Manipulación de variables numéricas

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
SET <var1> = <var2>	D, E	Asigna el valor de <var2> a <var1>
SET <var1> NOT <var2>		Asigna el valor negativo lógico de <var2> a <var1>
SET <var1> COMPLEMENT <var2>		Asigna el valor complementario de <var2> a <var1>
SET <var1> ABS <var2>		Asigna el valor absoluto de <var2> a <var1>
SET <var1> = <var2> <oper> <var3>		<oper> : +, -, *, /, SIN, COS, TAN, ATAN, EXP, LOG, MOD, OR, AND

## Control de movimientos del robot

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
MOVE <pos>	D, E	Mueve al robot a la posición <pos> a la velocidad especificada
MOVE <pos> {<tiempo>}	D, E	Mueve al robot a la posición <pos> en un tiempo <tiempo> en 10 ms/unidad
MOVED <pos> {<tiempo>}	E	Igual, salvo que el programa no continúa hasta que no se haya alcanzado la posición <pos>
MOVEL <pos> {<tiempo>}	D, E	
MOVELD <pos> {<tiempo>}	E	Igual con movimiento lineal
MOVEC <pos1> <pos2>	D, E	Movimiento circular del robot a <pos1>
MOVECD <pos1> <pos2>	E	pasando por <pos2>

## Control de movimientos del robot

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
MOVES <vpos> <inicio> <fin> {<tiempo>}	D, E	Mueve al robot por todas las posiciones del vector <vpos>, desde inicio hasta fin
MOVESD <vpos> <inicio> <fin> {<tiempo>}	E	
SPEED <valor>	D, E	Especifica la velocidad de movimiento (entre 1 y 100)
MPROFILE PARABOLE TRAPEZE	D, E	Ajusta el perfil de movimiento
INT_ON <eje1> ...	D, E	Activa/desactiva el servocontrol integral
INT_OFF <eje1> ...		



## Control de movimientos del robot

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
HOME {<n>}	D, E	Busca la posición de referencia
CON	D	Activa el servocontrol de los ejes
COFF	D	Desactiva el servocontrol de los ejes
TON	D	Activa (desactiva) las protecciones
TOFF	D	térmicas de los motores
OPEN	D, E	Abre la garra
CLOSE	D, E	Cierra la garra

## Tiempo real y control de programas

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
RUN <prog> {<prioridad>}	D, E	Ejecuta un programa
STOP {<prog>}	E	Aborta la ejecución de un programa
ABORT {<prog>}	D	Igual, y detiene el movimiento de los ejes
SUSPEND <prog>	D, E	Detiene la ejecución de un programa
CONTINUE <prog>	D, E	Continúa con la ejecución de un programa
SET <var>=TIME	D, E	Asigna a la variable var el valor de la variable de sistema TIME en ese instante
DELAY <tiempo>	E	Suspende el programa durante el tiempo especificado (en 10 ms)

## Tiempo real y control de programas

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
WAIT <var1> <cond> <var2>	E	Suspende la ejecución de un programa hasta que se cumpla la condición especificada.
PRIORITY <prog> <valor>	E	Especifica la prioridad de un programa (entre 1 y 10)
TRIGGER <prog> BY IN/OUT <n> <estado>	E	Ejecuta un programa condicionado al estado de la entrada/salida n (0: OFF 1:ON)

## Control del flujo de programas

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
IF <var1> <cond> <var2>	E	Ejecución condicional <cond>: <, >, =, <=, >=, <>
ANDIF <var1> <cond> <var2>	E	Condición lógica producto
ORIF <var1> <cond> <var2>	E	Condición lógica suma
ELSE	E	Sigue al IF y precede al ENDIF
ENDIF	E	Final de un bloque IF
FOR <var1> = <valor1> TO <valor2>	E	Bucle FOR
ENDFOR	E	Final de bucle FOR
LABEL <n>	E	Etiqueta ( n entre 0 y 9999 )
GOTO <n>	E	Salto incondicional
GOSUB <prog>	E	Salto a subrutina

## Control de entradas y salidas

<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
SET OUT[<n>] = <estado>	D, E	Fija el estado de la salida n a 0 o a 1 <estado> : 0 -> OFF    1-> ON
IF IN[<n>] = <estado>	D, E	Ejecución condicionada al valor de la entrada n.

## Visualización e interfaz con el usuario

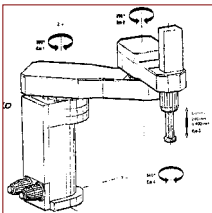
<u>Comando</u>	<u>Modo</u>	<u>Descripción</u>
PRINT "cadena" <arg>	E	Presenta por pantalla un texto y el valor de una variable
PRINTLN "cadena" <arg>	E	Igual, pero introduce inicialmente un retorno de página
READ "cadena" <var>	D	Presenta por pantalla una cadena y espera la introducción del valor de la variable var por teclado
GET <var>	E	Espera que se introduzca un carácter por teclado y asigna a <var> el código ASCII del carácter

## Otros

- ◆ Manipulación de programas
- ◆ Funciones de edición
- ◆ ...

En el laboratorio

## Introducción a la Robótica



UNIVERSIDAD DE SEVILLA



escuela superior de  
INGENIEROS DE SEVILLA

**UNIVERSIDAD DE SEVILLA**  
**Dpto. De Ingeniería de Sistemas y Automática**  
**Escuela Superior de Ingenieros**

- ◆ **CONCEPTOS GENERALES**
- ◆ **PROGRAMACIÓN Y OPERACIÓN**
- ◆ **PRACTICAS CON SCORBOT-V**



**Francisco R. Rubio**