

# Implementación del proceso de escritura mediante robots de articulación horizontal

Graciela Bustos Riveros\*, Iván Galera Delgado\*,  
Alfredo Jimeno Duncan\*, Miguel Angel Jimeno Paba\*,  
Karina Llaín Arenilla\*, Hugo Marchena Roca\*, Álvaro Santiago  
Arellana\*, Félix Vargas Cantillo\*, Eduardo Zurek Varela\*\*  
Laboratorio de Robótica y Automatización de la Producción  
([robotica@uninorte.edu.co](mailto:robotica@uninorte.edu.co))

---

## Resumen

*En este artículo se muestra cómo puede ser utilizado un robot tipo SCARA (el SCORA ER-14) para el desarrollo de escritura mediante los caracteres del alfabeto occidental aplicando de forma adecuada los recursos de programación ofrecidos por el controlador del robot.*

**Palabras claves:** Joints, teach, controlador, ACL.

## Abstract

*This paper shows how a SCARA – type robot (the SCORA ER-14) can be used for writing tasks by means of the characters from the western alphabet, applying in an appropriate way the programming resources offered by the robot's controller.*

**Key words:** Joints, Teach, controller, ACL.

---

Fecha de recepción: 14 de febrero de 2001

## INTRODUCCIÓN

El proceso descrito en este artículo fue realizado en el Laboratorio de Robótica y Automatización de la Producción de la

Universidad del Norte con ayuda de algunos de sus componentes, tales como el robot SCORA-ER 14 y su respectivo controlador tipo B, un PC, un marcador y un tablero borrable. Integrando estos implementos se elaboró una aplicación en la cual se solicita por pantalla una palabra, y el robot retira de una posición inicial el marcador, se posiciona en el tablero y la escribe sobre éste.

---

\* Estudiantes del programa de Ingeniería de Sistemas de la Universidad del Norte. Miembros del Grupo de Investigación en Automatización y Robótica de la misma universidad.

\*\* Ingeniero de Sistemas de la Universidad del Norte. Profesor del Departamento de Ingeniería de Sistemas y miembro del Grupo de Investigación en Automatización y Robótica. Actualmente cursa estudios de maestría en University Of South Florida, Tampa (E.U.). (e-mail: [ezurek@uninorte.edu.co](mailto:ezurek@uninorte.edu.co))

Todo este proceso es llevado a cabo por el previo análisis sintáctico por parte del controlador y su respectiva ejecución.

# 1. DESCRIPCIÓN DE LOS EQUIPOS USADOS

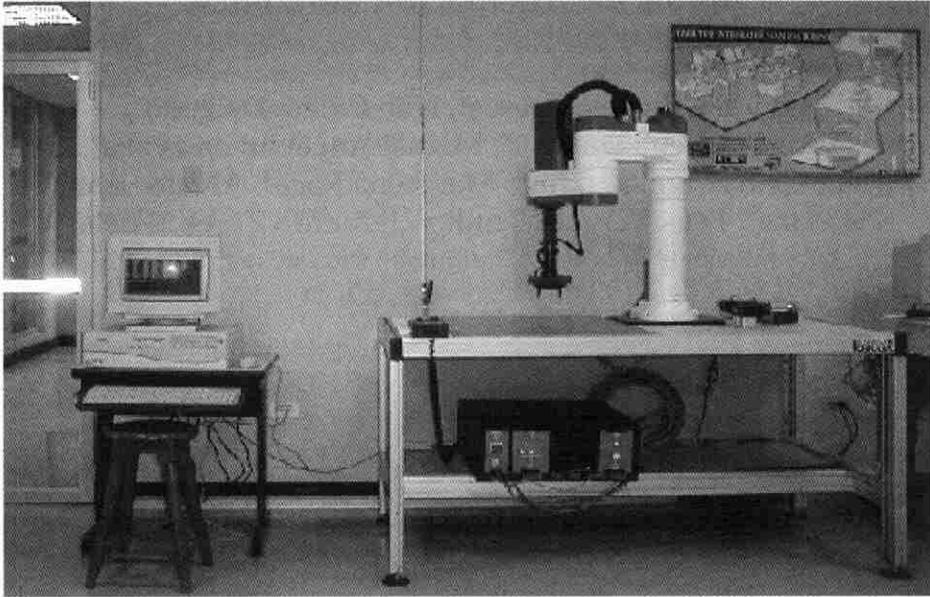


Figura 1. Robot, PC, teach pendant y controlador tipo B

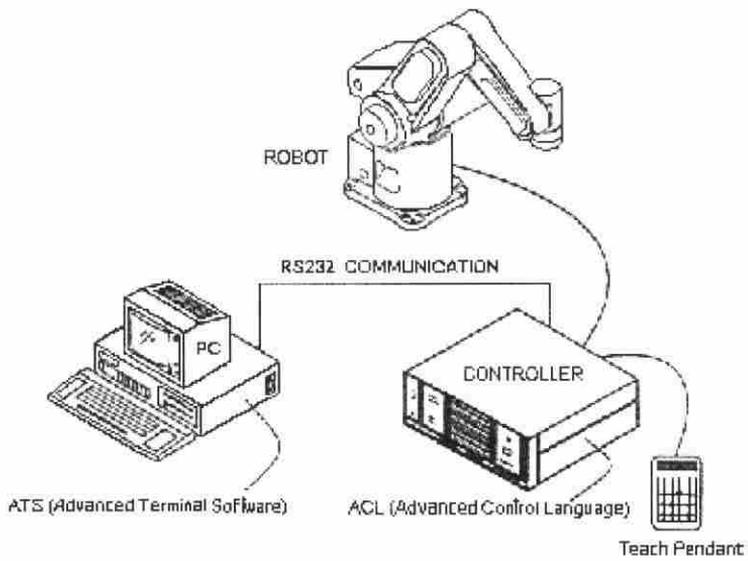


Figura 2. Gráfica de interconexión de los equipos usados[1]

En la figura 1 se muestra la forma como están colocados en el Laboratorio de Robótica los equipos que han sido utilizados; en la figura 2 se muestra cómo está la interconexión entre dichos equipos; vemos que el robot está conectado al controlador, y éste se encuentra conectado al *teach pendant* (el cual maneja el robot manualmente) y también al PC con una comunicación de RS-232.

### 1.1. Descripción física del robot SCORA-ER 14

#### Especificaciones técnicas [2]

- Estructura mecánica: Articulado horizontal
- Movimientos de los ejes (Axis):
  - Uno y dos (1 y 2): Rotación

- Tres (3): Translación
- Cuatro (4): Rotación sobre su propio eje con o sin cable de *gripper*
- Radio máximo de operación:
  - Mínimo 230 mm (9,06")
  - Máximo 500 mm (19.69")
- Peso máximo que puede soportar el brazo: 2Kg (4.4lb)
- Temperatura ambiente de operación: 2°-40°C

#### Estructura

El SCORA-ER 14 es un robot de articulación horizontal (tipo *SCARA*). Los dos primeros «*Joints* (o grados de libertad)» son de rotación y determinan la posición del *gripper* en el plano XY. El tercer *Joints* es prismático y determina la altura de la coordenada Z del *gripper*.

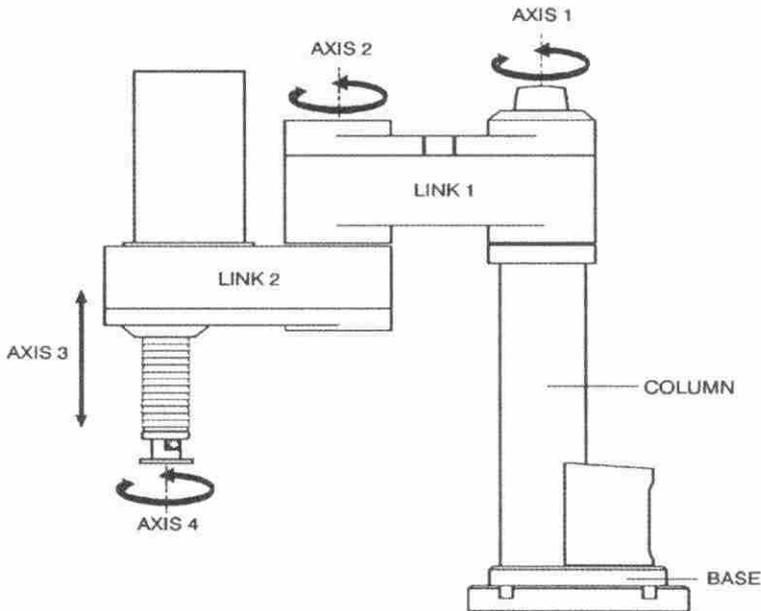


Figura 3. Partes del robot [2]

## 1.2. Descripción física del controlador

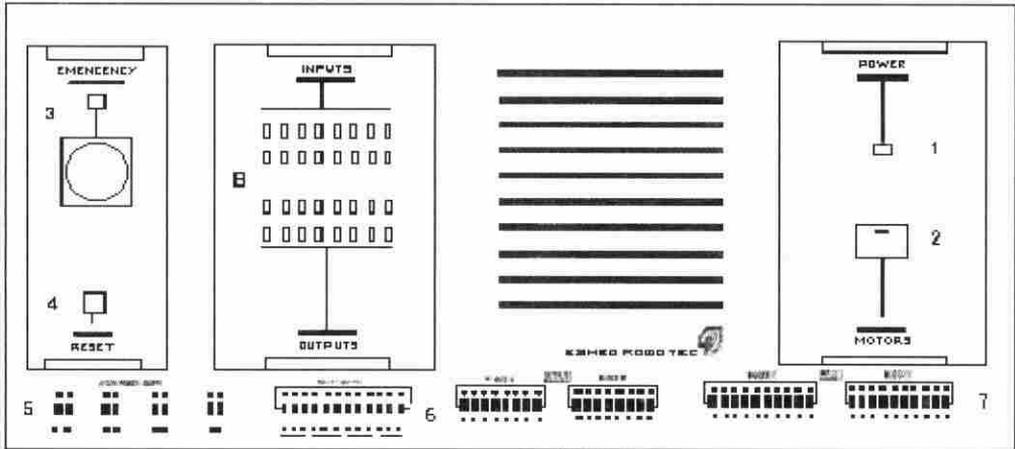


Figura 4. Esquema del controlador [3]

El controlador tiene 8 partes muy importantes, las cuales se encuentran enumeradas en la figura 4. A continuación se describe el funcionamiento de dichas partes[3]:

1. *Luz de poder:* Esta luz es de color verde, se enciende cuando el switch de energía de encuentra encendido.
2. *Switch de los motores:* Este switch conecta y desconecta el voltaje DC a todos los motores conectados.
3. *Botón y luz de emergencia:* Al usar el botón se desconecta la energía del motor, se prende una luz de emergencia y se abortan todos los programas que se estén ejecutando.
4. *Switch de reset:* Reinicia el controlador.
5. *Terminales de fuente de poder del usuario:* Permiten a los aparatos externos usados en las aplicaciones del usuario recibir energía del controlador.
6. *Terminales de salida:* Permiten al sistema robótico transmitir señales a aparatos externos en el ambiente del robot.
7. *Terminales de entrada:* Permiten al sistema robótico recibir señales en el ambiente del robot.
8. *Luces de entrada y salida:* Son 16 luces de color verde para las salidas (*output*) y 16 luces de color amarillo para las entradas (*input*).

### 1.3. Descripción física del *teach pendant*

El *teach pendant* es un dispositivo que permite controlar manualmente el robot. Mediante el uso de éste se puede ordenar al robot efectuar desplazamientos en tiempo real, y de esta forma se tiene un mejor control sobre el movimiento del mismo, con lo que resulta más práctico al momento de ubicar al robot en posiciones específicas para almacenar sus coordenadas y utilizarlas luego en el programa que se va a crear.

## 2. REQUERIMIENTOS DEL SISTEMA

El lenguaje de ACL necesita un protocolo de comunicaciones RS-232 para poder establecer la comunicación entre el controlador y el computador. Además necesita un PC que permita la interfaz con el usuario.

## 3. DESCRIPCIÓN DEL LENGUAJE DE PROGRAMACIÓN CON ACL

ACL, Lenguaje de Control Avanzado (*Advanced Control Language*), es un lenguaje de programación robótico multitarea desarrollado por *Eshed Roboted* (1982). ACL es programado dentro de un grupo de EPROMs dentro del controlador B, y puede ser accedido desde cualquier terminal standard o computadores PC a través de un canal de comunicación RS-232 [1].

## Características de ACL

- Control de usuario directo de los ejes del robot
- Programación de usuario del sistema robótico
- Control de datos de entrada / salida
- Ejecución de programas simultáneo y sincronizado
- Alto soporte multitarea
- Manejo de archivos simples

ATS, Software Terminal Avanzado, es la interfaz de usuario para el controlador de ACL. ATS es suministrado en disquete y opera en cualquier computadora. El *software* es un emulador terminal que habilita el acceso a ACL desde una computadora.

## Características de ATS

- Configuración del controlador de formato corto
- Definición de aparatos periféricos
- Claves de Acceso rápido para entradas de comando
- Manejador de *Backup*
- Manejador de impresión

## 4. DESCRIPCIÓN DEL PROGRAMA

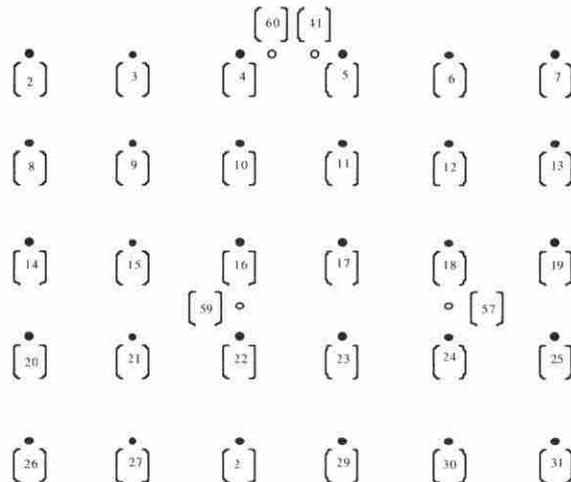
Para este proyecto se utilizó el lenguaje ACL, el cual nos permite interactuar con el robot, mediante una serie de instrucciones que éste ejecutará una vez procesadas y reconocidas.

Además de sentencias típicas de programación como condicionales IF-ELSE y bucles FOR se usaron instrucciones

únicas de ACL para establecer posiciones a partir de la posición del robot (*here*), posiciones relativas a otras (*teachr*) y establecer posiciones manualmente (*teach*). Además, dentro del programa se usaron otras como: *speed*, *speedl* (establecen velocidades en los ejes); *move*, *moved* (mueven el robot a una posición); *delay* (establece una pausa determinada entre una instrucción y otra).

El programa está basado en una matriz de puntos, en la que cada punto representa una posición en la que el robot se puede ubicar al momento de construir las letras y que almacena las coordenadas de dicha posición (todas las posiciones son relativas, es decir, dependen de la posición [2], la cual se actualiza al escribir una nueva letra).

La matriz utilizada es la siguiente:



Nombre: **cara**  
 Puntos auxiliares: cara[41], cara[49], cara[58], cara[60]

**Figura 5.** Matriz de puntos

Para la escritura de ciertas letras se limitaba un poco la matriz, por lo que fue necesario crear unos puntos auxiliares para lograr escribir todas las letras de la forma más estética posible (ver figura 5).

A partir de la matriz principal y los puntos auxiliares se empezó a diseñar las 26 letras de la manera más eficiente. Siempre se trató de buscar la ruta más corta para escribir las letras, sobre todo con aquellas que el robot no las dibujaría de un solo trazado, sino levantando el marcador. Siempre que se necesitó elevar el marcador se llevó hacia una posición inicial de la cual dependen las demás posiciones. Esa posición inicial es cara[2]; como ya se mencionó, es el mayor punto de referencia.

Al estar totalmente diseñadas las letras, se implementaron en 26 subrutinas los trazos que pertenecían a cada letra; por esto se creó un programa principal que se encarga de llamar a la subrutina correspondiente al código *ASCII* que representa la letra de la palabra digitada por pantalla. Esto se realizó con la instrucción *GET* (obtiene el código *ASCII* de un carácter leído). Para este propósito se usaron ciertos comandos especiales de *ACL* para identificar los códigos *ASCII*.

Por motivos de eficiencia, a la hora de compilar el programa éste fue dividido en dos partes: la primera consta de las 26 subrutinas y está en el archivo llamado *pra1*, y la segunda en el archivo *pra2*, que contiene el programa principal.

Los dos archivos referidos son de tipo texto que necesitan ser actualizados antes de pasar a la *compilación*. De esto se encarga el controlador, que busca línea por línea del archivo algún error de escritura; si no se encuentra ninguno, se puede proceder a la ejecución del programa a través de la línea de comandos de *ATS*.

## 5. LIMITACIONES

### 5.1. Volumen de trabajo del robot

Como se pudo ver en el esquema del robot, los ejes de éste no ofrecen una gran flexibilidad y alcance a pesar del tamaño del robot. Esto no nos favoreció, pues limitó el tamaño de las palabras que se iban a escribir a sólo nueve letras (ver figura 6).

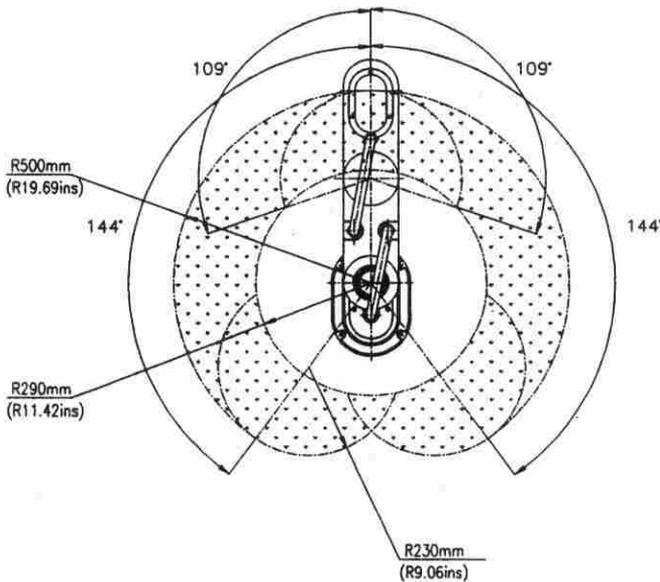


Figura 6. Volumen de trabajo del robot [2]

## 5.2. Cohesión entre el *gripper* del robot y el marcador

La presión ejercida por el *gripper* del robot fue suficiente para suministrar un agarre sobre el marcador. Sin embargo, la forma cilíndrica de éste dificultaba ciertos movimientos, sobre todo aquellos en los que se requería hacer curvas, ya que el marcador se deslizaba por la superficie y se doblaba. Para lograr una estabilidad máxima y un mejor desempeño se diseñó una cuña de madera a presión, con un orificio circular central para sujetar el marcador, con lo cual se evitó que éste se deslizara. En la figura 7 vemos cómo el marcador se encuentra con la cuña y cómo será agarrado por el *gripper*.

## 5.3. Superficie de escritura

El robot se encarga de dibujar las letras sobre un tablero borrable fijado sobre una mesa. Después de haber probado varias veces el programa, notamos una perfecta nivelación por parte del robot, pero la mesa presentaba un ligero desnivel, no detectable a simple vista, lo que provocaba una mayor presión sobre la punta del marcador a medida que éste iba escribiendo, lo cual le dañaba la punta. Esto era más notorio en las palabras que tenían mayor número de letras, ya que esto se producía en las últimas letras. Para poder solucionar este problema se tuvo que ir aumentando la altura del marcador con respecto a la mesa a medida que éste iba escribiendo.

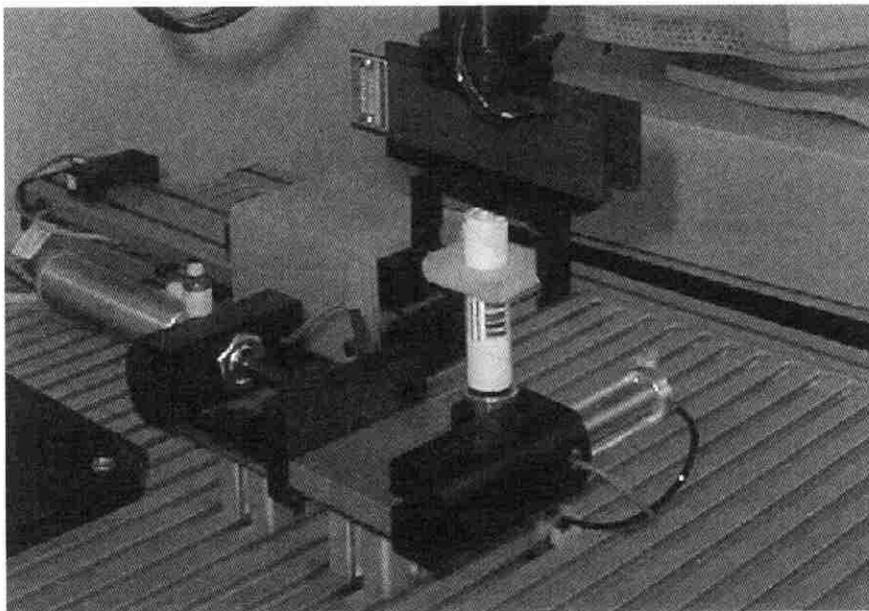


Figura 7. Muestra de las formas del *gripper* y el marcador

## 6. APLICACIÓN REALIZADA CON LA PALABRA «CASA»

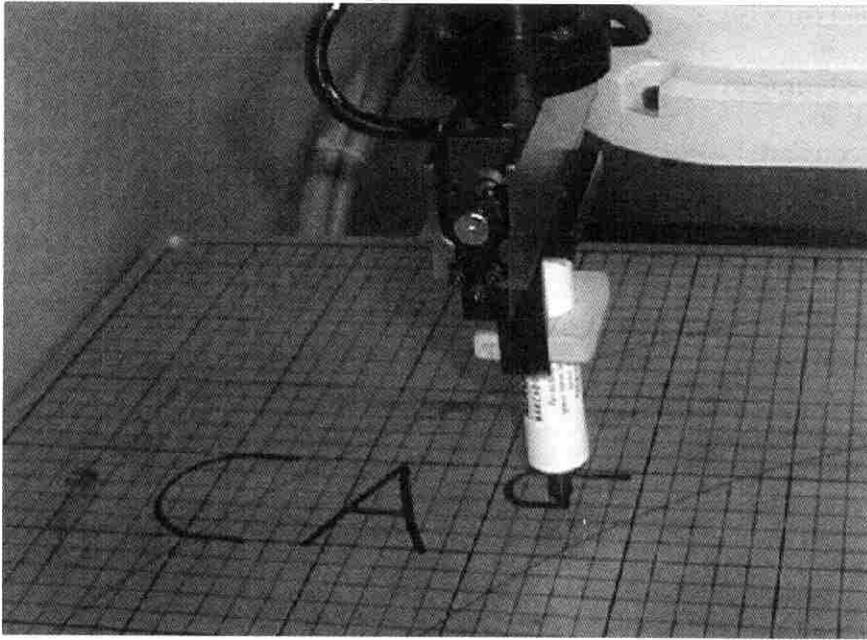


Figura 8. Palabra de ejemplo

Presentamos la aplicación mediante la palabra CASA. En la figura 8 se puede apreciar cómo es escrita esta palabra por el robot. Vamos a mostrar las subrutinas que se necesitan para escribir la palabra y también la parte del programa principal que se encarga de llamar a dichas subrutinas para escribirla.

Estas subrutinas contienen los movimientos necesarios para escribir la letra, además de una pausa entre instrucciones y el establecimiento de una velocidad para escribir.

```
PROGRAM c
  speed 10
  moved cara[1]
  delay 50
  moved cara[20]
  delay 50
  speed 10
  movec cara[25],cara[5]
  delay 50
  moved cara[1]
  delay 50
end
```

```
PROGRAM a
  speed 10
  moved cara[1]
```

```

delay 50
moved cara[2]
delay 50
moved cara[19]
delay 50
moved cara[26]
delay 50
moved cara[1]
delay 50
moved cara[10]
delay 50
moved cara[22]
delay 50
moved cara[1]
end
PROGRAM s
  speed 10
  moved cara[1]
  delay 50
  moved cara[31]
  delay 50
  moved cara[13]
  delay 50
  speedl 10
  movec cara[11],cara[6]

  delay 50
  moved cara[17]
  delay 50
  speedl 10
  movec cara[14],cara[40]
  delay 50
  moved cara[2]
  delay 50
  moved cara[1]
  delay 50
end

```

Dentro del programa principal se creó un bloque *for*, cuyo índice es KL, para controlar el número de letras

digitadas. Esto, como ya se dijo, llama a la subrutina correspondiente al código ASCII digitado. Además, se traslada hacia la derecha y un poco hacia arriba para contrarrestar el desnivel de la mesa.

```

IF LETR[KL]=65
  print "A"
  gosub a
  shiftc cara[2] by x 60
  shiftc cara[2] by z .3
ENDIF

```

```

IF LETR[KL]=67
  print "C"
  gosub c
  shiftc cara[2] by x 60
  shiftc cara[2] by z .3
ENDIF

```

```

IF LETR[KL]=83
  print "S"
  gosub s
  shiftc cara[2] by x 60
  shiftc cara[2] by z .3
ENDIF

```

Además de esta parte del programa hay que hacer la respectiva declaración de las variables. Después de haber escrito las subrutinas se devuelve el robot a su posición original.

## CONCLUSIONES

El programa realizado puede ser mejorado, si se tiene en cuenta que, por ejemplo, el reconocimiento de cadena de caracteres puede simplificarse si se logra implementar un método de ordenamiento que esté en capacidad de

aumentar la velocidad de procesamiento de la información al reducirse el número de comparaciones dentro del ciclo.

A pesar de que las técnicas de programación de robots es un área muy poco explorada, ésta se puede relacionar con aplicaciones industriales y tecnológicas para su utilización en múltiples tareas, que simplifique y maximice procesos industriales como, por ejemplo, el tallado de objetos como el vidrio.

## Referencias

- [1] ESHED ROBOTEC. *ACL, Advanced Control Language, for controller B, Reference Guide.*
- [2] ESHED ROBOTEC. *SCORA-ÉR 14, user's manual. Israel, ESHED ROBOTEC Inc.*
- [3] ESHED ROBOTEC. *CONTROLLER-B, user's manual. Israel, ESHED ROBOTEC Inc.*