

# Desarrollo didáctico de una plataforma móvil para recolección de objetos metálicos en ambientes conocidos

Royman López B. \*, Fabián Panesso L. \*\*, Carlos D. Paternina A. \*\*\*

Laboratorio de Robótica y Producción Automática, Universidad del Norte  
(robotica@uninorte.edu.co)

## Resumen

*Este artículo contiene una aplicación de robótica móvil. Específicamente se muestran los principales rasgos del robot móvil que fue desarrollado para la feria universitaria de grupos estudiantiles realizada en la Universidad del Norte. Este robot fue diseñado para operar en lazo abierto y sus componentes fueron obtenidos de módulos de ensamble FISCHERTECHNIK. En esta aplicación el entorno es totalmente conocido. También se describe el análisis cinemático del manipulador del robot y las técnicas de programación, para lo cual se utilizó el programa PIC-BASIC.*

**Palabras claves:** Denevit-Hartenberg, análisis cinemático, PIC, robot móvil.

## Abstract

*This paper presents an application of mechanics and electronics to build a mobile robot. The robot was developed for the student groups fair that took place at the Universidad del Norte. This robot was designed to work in open loop and its components were obtained out of FISCHERTECHNIK joining modules. In this application the environment is totally known. Also, the article describes the kinematics analysis of the robot manipulator and the programming techniques, with the use of the PIC-BASIC program.*

**Key words:** Denevit-Hartenberg, kinematics analysis, PIC, mobile robot.

Fecha de recepción: 14 de marzo de 2000

\* Estudiante de noveno semestre del programa de Ingeniería Mecánica de la Universidad del Norte. Miembro del Grupo de Automatización y Robótica.

\*\* Estudiante de noveno semestre del programa de Ingeniería Electrónica de la Universidad del Norte. Miembro del Grupo de Automatización y Robótica.

\*\*\* Coordinador Laboratorio de Robótica y Automatización, Universidad del Norte. Ingeniero Industrial, Universidad del Norte. Maestría en Ingeniería Industrial, University of South Florida. Ph.D. en Ingeniería Industrial de esta última universidad. Miembro de IIE (Institute of Industrial Engineers), SME (Society for Manufacturing Engineers) y de RI (Robotics International) de SME.

## 1. INTRODUCCIÓN

Los descubrimientos actuales de la robótica emergen en un entorno dotado de un sinnúmero de necesidades, las cuales motivan al hombre a desarrollar mecanismos que suplan la actividad humana en determinadas tareas que le pueden ocasionar riesgos a su integridad personal o le generen un excesivo gasto de energías. Entre algunas de las

labores que realizan dispositivos robóticos en la actualidad se encuentran: limpieza (de hogares, ventanas, plantas nucleares, etc.); automatización de procesos industriales (soldadura, maquinado, ensamble, etc.); labores de inteligencia militar (localización y desactivación de minas) y hasta entretenimiento familiar (tales como el robot AIBO desarrollado por la *SONY Entertainment Company* [5]).

Esta incesante búsqueda de tecnología también se produce en las aulas universitarias, donde se gestan los inicios de la investigación científica. En el Laboratorio de Robótica y Automatización de la Universidad del Norte se están desarrollando varias líneas de investigación, entre las cuales se han generado las de robótica móvil y la de inteligencia artificial, con objeto de realizar autómatas multitareas para aplicar en algunas actividades de uso práctico.

En este artículo se mostrarán los principales rasgos de un robot móvil con un brazo manipulador, construido a partir de módulos de *FISCHERTECHNIK* y programado en *PIC-BASIC*, el cual fue presentado durante la feria de grupos universitarios en representación del grupo de investigación en robótica y automatización.

## 2. DESCRIPCIÓN DE LA TAREA

El robot fue construido con el fin de recoger monedas situadas en un entorno conocido, específicamente a distancias ya establecidas y para operar en

carencia de algún tipo de obstáculos diferentes a las monedas.

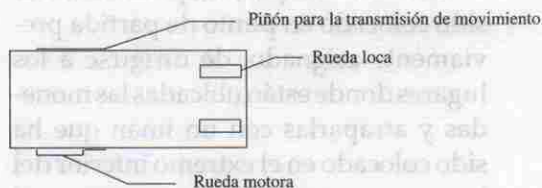
El robot es capaz, después de haber sido colocado en punto de partida previamente asignado, de dirigirse a los lugares donde están ubicadas las monedas y atraparlas con un imán que ha sido colocado en el extremo inferior del brazo, y posteriormente se regresa al punto de inicio. Para diferenciar las labores de locomoción y las de manipulación de las monedas, se colocaron 3 led's, dos de los cuales se encuentran encima de los dos motores de las ruedas y sólo se encienden cuando la plataforma se está desplazando, y otro encima del brazo, que se enciende cuando el brazo está en operación.

## 3. CARACTERÍSTICAS MECÁNICAS DEL ROBOT

### 3.1. Plataforma de desplazamiento

Esta plataforma consta de dos ruedas motoras ubicadas en la parte trasera y dos ruedas locas ubicadas en la parte delantera del dispositivo. Las ruedas traseras poseen una rueda dentada que recibe el suministro de potencia de un piñón, que a su vez recibe la potencia del motor por medio de una reducción a través de una pequeña caja de engrajes. Los motores están posicionados verticalmente, con la ayuda de módulos de fijación, por lo que la transmisión de movimiento es realizada a 90°. Dichos motores requieren alimentación de 6-9 V [2] y suministran una potencia nominal de 225 mWatts. En la figura 1

se puede observar una vista genérica de la parte inferior de la plataforma puesta en cuestión.



**Figura 1**

Vista inferior de la plataforma del mecanismo

La dirección del dispositivo es lograda mediante el cambio de sentido de rotación en los motores, lo que es dado por el cambio de polaridad de éstos. Esto genera movimientos rectilíneos de avance y retroceso cuando ambos motores se ponen a girar en el mismo sentido de rotación, y movimientos de giros hacia la izquierda y hacia la derecha

cuando alguno de los dos motores gira en sentido inverso al otro.

El chasis está constituido por un módulo de ensamble de FISCHERTECHNIK, hecho de plástico, que fue adaptado para recibir las dos ruedas locas en su parte delantera.

### 3.2. Manipulador

#### 3.2.1. Descripción

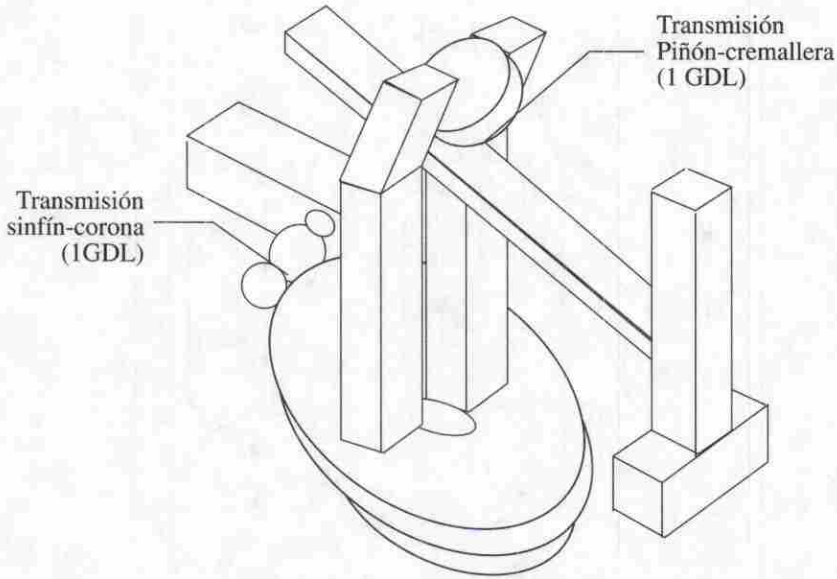
El sistema de manipulación del robot está constituido por un brazo al cual fue acoplado un imán en su extremo inferior con el fin de magnetizar monedas, labor para la cual fue preparado. En la figura 2 se visualizan de forma genérica los dispositivos del robot.

Este manipulador posee dos grados de libertad<sup>1</sup>, fruto de las dos entradas de movimiento controlado que posee: dos

**Tabla 1**  
Especificaciones del robot

CARACTERÍSTICA	ESPECIFICACIÓN
Altura máxima	20 cm
Longitud	25 cm
Ancho	21 cm
Número de motores	4
Grados de libertad del manipulador	2
Velocidad máxima de desplazamiento	0.1 m/s
Peso	0.75 Kg

<sup>1</sup> Grados de libertad: Mínimo de coordenadas independientes que se necesitan para definir la posición y orientación de un cuerpo en el espacio.



**Figura 2**  
Esquema genérico del manipulador del robot

motores de 6-9 V que a través de una transmisión sinfín-corona suministran la rotación al robot a lo largo del eje Z y mediante una transmisión de tipo cremallera permiten el avance y el retroceso lineal a lo largo del eje de la cremallera para atrapar la moneda con el imán. En la figura 2 se pueden observar detallados los dos grados de libertad del robot.

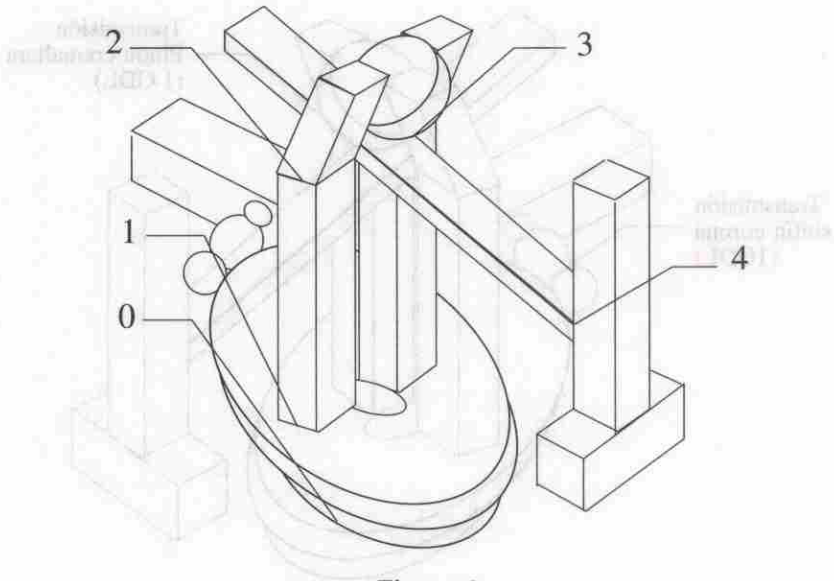
### 3.2.2. Análisis cinemático

Para realizar el análisis cinemático del manipulador se seguirá el método del algoritmo de Denavit-Hartenberg para la obtención del modelo cinemático directo [1]. Para aplicar este método, primero se numeran cada uno de los eslabones y se localizan los sistemas de

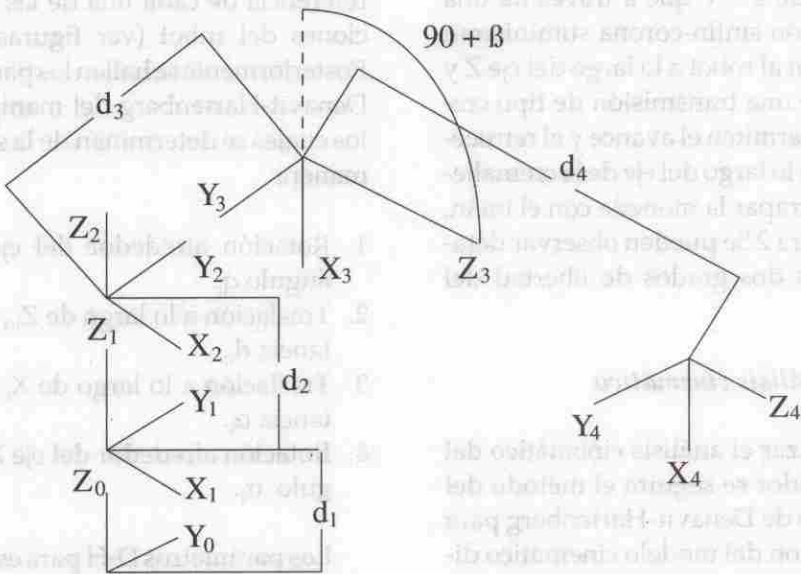
referencia de cada una de las articulaciones del robot (ver figuras 3 y 4). Posteriormente se hallan los parámetros Denavit-Hartenberg del manipulador, los cuales se determinan de la siguiente manera:

1. Rotación alrededor del eje  $Z_{i-1}$  un ángulo  $q_i$ .
2. Traslación a lo largo de  $Z_{i-1}$  una distancia  $d_i$ .
3. Traslación a lo largo de  $X_i$  una distancia  $a_i$ .
4. Rotación alrededor del eje  $X_i$  un ángulo  $\alpha_i$ .

Los parámetros D-H para este manipulador se describen en la tabla 2.



**Figura 3**  
Designación de las articulaciones del manipulador



**Figura 4**  
Sistemas de ejes de coordenadas para el manipulador en estudio

**Tabla 2**  
Parámetros D-H para el manipulador

Articulación	$\theta$	$d$	$A$	$\theta$
1	$q_1$	$D_1$	0	0
2	0	$D_2$	0	0
3	0	$D_3$	0	$90 + \beta$
4	0	$D_4$	0	0

Una vez calculados los parámetros se procede a encontrar la matriz de transformación homogénea utilizando las siguientes ecuaciones:

$$T = {}^0A_4 = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4$$

donde las matrices A se definen mediante la expresión general:

$${}^{i-1}A_i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & -\cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Vale la pena aclarar que a pesar de que este manipulador no maneja un volumen de trabajo exactamente cilíndrico, se puede aplicar la expresión general de la matriz A, debido a que el manipulador queda siempre contenido en un plano que se determina a partir del ángulo  $q_1$ , dado por la articulación rotacional 1.

Luego la matriz ( que representa la localización del sistema final con respecto al sistema de referencia de la base del robot) es la siguiente:

$$T = {}^0A_4 = \begin{bmatrix} \cos q_1 & -\cos(90 + \beta) \sin q_1 & \sin(90 + \beta) \sin q_1 & d_1 \sin(90 + \beta) \sin q_1 \\ \sin q_1 & \cos(90 + \beta) \cos q_1 & -\sin(90 + \beta) \cos q_1 & -d_1 \sin(90 + \beta) \cos q_1 \\ 0 & \sin(90 + \beta) & \cos(90 + \beta) & d_1 + d_2 + d_3 + d_4 \cos(90 + \beta) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Para calcular la matriz de transformación de este robot se tienen los siguientes datos:  $d_1=2.5$  cm,  $d_2=3.5$  cm,  $d_3=2.0$  cm,  $d_4=7$  cm,  $\beta=30^\circ$ . Se tomó, para darle valor a  $q_1$ , una rotación de  $30^\circ$  en la corona. Por lo tanto, reemplazando en la matriz anterior se obtiene:

$$T = {}^0A_4 = \begin{bmatrix} 0.866 & -0.250 & 0.433 & 3.031 \\ 0.5 & 0.433 & -0.749 & 5.243 \\ 0 & 0.866 & -0.5 & 4.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

#### 4. CARACTERÍSTICAS ELÉCTRICAS Y DE MANEJO DE POTENCIA

El mecanismo que se ensambló está compuesto por tres motores, tres led's y un buzzer con respecto a la parte que requiere manejo de corriente. Los motores trabajan a un voltaje de 6 a 9 voltios DC y tienen una demanda de corriente de 25 mA. Los led's y el buzzer trabajan con una corriente de 16 mA.

Estas cantidades de corriente requeridas por los dispositivos son todas manejadas por el microcontrolador PIC PS3, que se encarga del control principal del mecanismo debido a la programación en *software* que posee en su memoria. La entrega de estas cantidades de corriente las realiza a través de sus pines y gracias a una unidad de soporte

e interfaz de potencia que regula el voltaje que viene de la fuente y lo entrega bajo las especificaciones que le suministra el dispositivo de control. Dicha interfaz recibe el nombre de ZUMNY y ya trae configurado los pines de salida para las distintas funciones, ya sean digitales o análogas, con su equivalente respectivo de demanda de corriente [3].

La alimentación general del dispositivo de interfaz ZUMNY se realiza a través de un adaptador de corriente, que transforma el voltaje de una red normal de electricidad (110V AC) a un voltaje nominal de 9V DC, que sin carga puede llegar a medir 13.52V DC (valor práctico). Ver figura 5.

## 5. PROGRAMACIÓN

### 5.1. Microcontrolador PIC PS3

El microcontrolador PIC PS3, conocido para el manejo del lego FISERTCHENICK como PICSTIC, es un microcontrolador de la familia 16C84, que posee 16 pines. Este dispositivo permite guardar en memoria una serie de instrucciones que componen un programa que será ejecutado a través de la interfaz ZUMNY con los debidos niveles de corriente, que alimentan a través de un bus serial un bloque de uniones (pistas) en forma de circuito impreso, donde son conectados los distintos dispositivos (motores, led's, sensores, luces incandescentes, etc.) para recibir la energía para su funcionamiento.

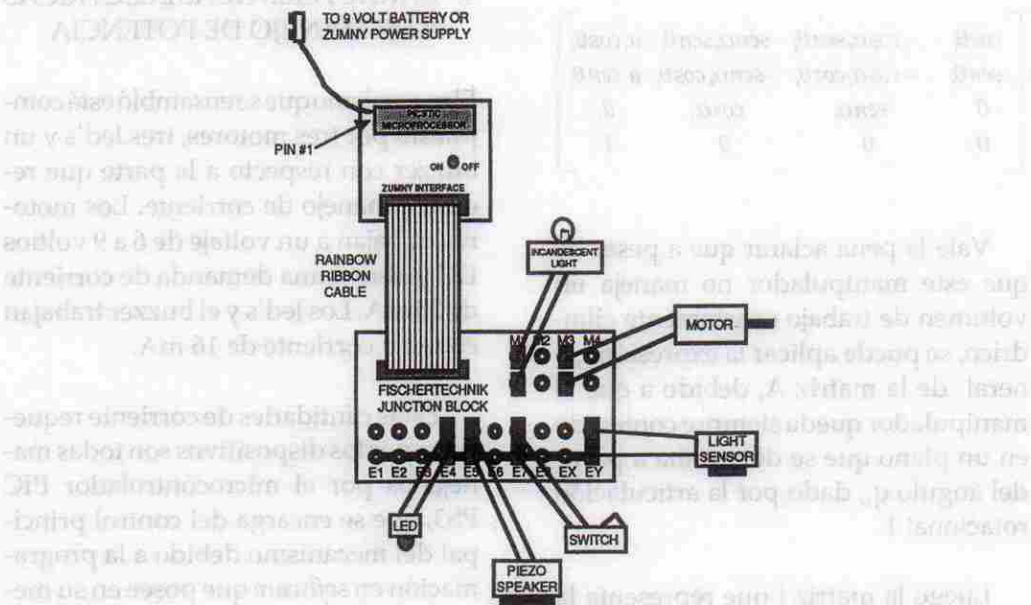


Figura 5  
Diagrama de conexiones e interfase

La descripción de las funciones de los pines es la siguiente, según su identificación en el dibujo:

PIN	NOMBRE	FUNCIÓN
1	V+	Voltaje positivo de entrada para el PICSTIC (7-18VDC)
2	GND	Tierra común para PICSTIC y entradas I/O.
3	PA3	I/O digital adicional que puede ser usada mediante rutinas cortas en Assembler o por uso de los comandos PEEK y POKE.
4	PA4	I/O digital adicional que puede ser usada mediante rutinas cortas en Assembler o por uso de los comandos PEEK y POKE.
5	5VDC	Salida positiva de 5VDC del circuito de regulador de potencia del PICSTIC para alimentación de circuitos externos. La corriente de salida puede ser hasta de 100 mA.
6	RESET	Permite reiniciar el programa que se va a ejecutar colocando el pin en cero lógico (0 V).
7	PB0	Pin I/O digital direccionable por el PICSTIC como N° 0.
8	PB1	Pin I/O digital direccionable por el PICSTIC como N° 1.
9	PB2	Pin I/O digital direccionable por el PICSTIC como N° 2.
10	PB3	Pin I/O digital direccionable por el PICSTIC como N° 3.
11	PB4	Pin I/O digital direccionable por el PICSTIC como N° 4.
12	PB5	Pin I/O digital direccionable por el PICSTIC como N° 5.
13	PB6	Pin I/O digital direccionable por el PICSTIC como N° 6.
14	PB7	Pin I/O digital direccionable por el PICSTIC como N° 7.
13-2	ADC1	Convertidor de entrada análogo-digital de 12 bit medido positivo(+) con respecto a tierra. En medida diferencial de dos canales, éste es el canal negativo.
14-2	ADC0	Convertidor de entrada análogo-digital de 12 bit medido positivo(+) con respecto a tierra. En medida diferencial de dos canales, éste es el canal positivo.

## 5.2. Lenguaje de Programación

El lenguaje de programación del microcontrolador PIC PS3 es el PicBasic vs. 2.0 6/97. Para poder trabajar con este *software*, es necesario contar con las siguientes especificaciones de *hardware*:

- IBM PC o computador compatible
- MS-DOS 3.3 o superior
- 512K de RAM
- 1 MB de espacio en el disco duro
- Drive de 3 1/2



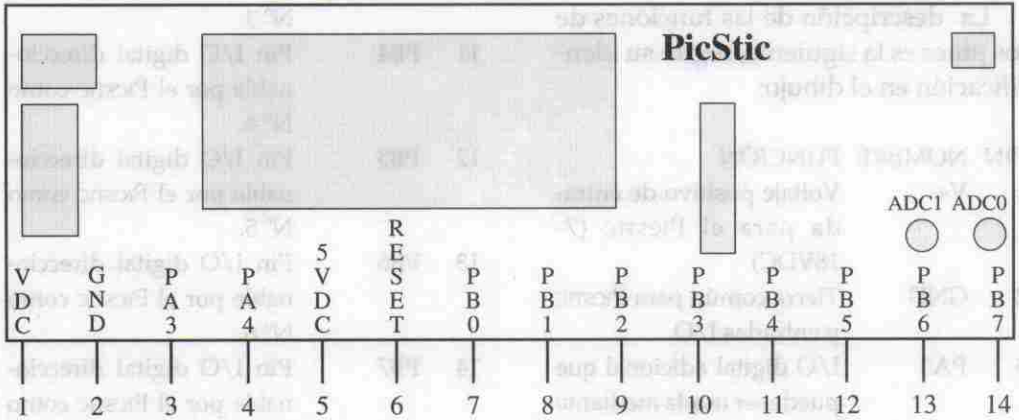


Figura 6  
Microcontrolador PIC PS3

- Puerto paralelo de impresora con conector DB-25

Para realizar la programación se cuenta con una unidad de circuitería externa que permite bajar la línea de comandos del programa a través del puerto paralelo de la impresora por medio de una interfaz de entrada con conector DB-25 a una salida del tipo telefónica RJ-11. La tarjeta de programación cuenta con su respectiva fuente adaptadora de 110V AC a voltaje de DC (aprox. 12V).

El lenguaje de programación se baja al computador bajo el DOS o sobre cualquier versión con las características antes mencionadas de Windows 3.1 o Windows 95.

Para cargar y ejecutar un programa se procede de la siguiente manera:

1. En el momento de crear un programa y para almacenarlo, se necesita contar con un procesador de texto o block de notas, en el cual se realiza la programación con los comandos respectivos que aparecen en el manual del PICBASIC [6] y son posteriormente guardados bajo las extensiones (\*.txt o \*.bas); con esto se asegura que el archivo sea de texto en ASCII.
2. El *software* también cuenta con un archivo ejecutable (PBC.EXE), que es un compilador de PICBASIC. Cuando se tiene el archivo en formato de texto y dentro de la carpeta donde se encuentran los archivos del PICSTIC, si se envía el archivo de texto hacia el compilador (PBC.EXE); éste convierte el código ASCII en lenguaje de ensamblador.
3. También existe un archivo (PM.EXE),

que se encarga de transformar el programa en assembler, si la compilación fue correcta, en un archivo del tipo lenguaje de máquina con extensión (\*.hex). Ambos archivos, éste y el obtenido en el numeral anterior, son creados dentro de Windows en la carpeta de escritorio (para el caso analizado).

4. El siguiente paso es dirigirse a la carpeta de escritorio y seleccionar el archivo (\*.hex) y enviarlo a la carpeta de PICSTIC, a la subcarpeta FISCHER, que es donde se encuentran grabados todos los programas de muestra y ejecutables por el microcontrolador. Si no se encuentra el archivo de extensión (\*.hex), quiere decir que hubo un error en la compilación, porque la secuencia de comandos o el programa realizado en el procesador de texto presenta inconsistencias de variables o fallas de programación.
5. Una vez se tiene el archivo dentro de la subcarpeta FISCHER se puede bajar el programa al PICSTIC a través de un archivo ejecutable (EPIC.EXE), que tiene ejecución sobre la tarjeta de programación.
6. La pantalla de EPIC.EXE es totalmente amigable e indica todo lo que hay que hacer y si las condiciones para bajar el programa son buenas. Una vez se empieza a bajar el programa al PICSTIC se encienden 2 led's que presenta la tarjeta de programación, y cuando éstos se apagan queda todo

listo y el programa ya está en el dispositivo, y puede separarse éste de la interfaz y colocarse en la ZUMNY.

### 5.3. Manejo de solución y programación del mecanismo

Habiendo definido la tarea que debía cumplir el mecanismo, el primer paso en su programación fue definir una secuencia de los movimientos que tenía que realizar el robot para trabajar cada uno, como una subrutina del programa principal. Al final de esta etapa se obtuvo la secuencia de rutinas que se muestra en la figura 7.

Además de las subrutinas definidas, se decidió incluir 2 subrutinas que se repiten según la ejecución de las diferentes etapas de la tarea: subrutina de encendido de led's y subrutina de sonido.

#### 5.3.1. Explicación de las subrutinas

##### • Temporización

La subrutina de temporización es la encargada de ocasionar retardos en la ejecución de una función de la tarea, es decir, ejecuta una pausa en la ejecución de los comandos del programa. Esta subrutina es implementada de la siguiente manera:

pause 2000 'Ocasiona una pausa de 2 minutos en la ejecución.

Esta función de pausa se utiliza en varias ocasiones en la programación,



**Figura 7**

Subrutinas de programación

como por ejemplo: tiempo de duración de encendido o apagado de motores, cambio de dirección de motores, encendido o apagado de led's y buzzer, control de movimiento del brazo, y en general, para el control total del tiempo de ejecución de la tarea o de sus etapas parciales.

- *Desplazamiento de motores (adelante / atrás)*

Los motores utilizados para desplazar el mecanismo se encuentran sincroni-

zados y actuando bajo la misma entrada de corriente. Debido a esto, cada uno está trabajando con un 50% de energía total de la salida del microcontrolador (M1 Fischertechnik Unión), ya que presentan las mismas características de funcionamiento. Esto quiere decir que no se tiene la posibilidad de controlar un giro en el robot, y que sólo se puede desplazar hacia adelante o hacia atrás, porque no es posible apagar un motor y dejar el otro encendido.

Bajo estas limitaciones, la programación se hace fácil por medio de los siguientes comandos:

- low 1 'Apaga un motor
- high 1 'Enciende un motor
- low 0 'Cambia de dirección motor
- high 0 'Cambia de dirección motor

Hay que tener presente que para cambiar de dirección los motores y para prenderlos o apagarlos es necesario una temporización, para darle una pausa a la ejecución del programa y poder proteger los dispositivos en sus diferentes estados [3].

- *Rotación del brazo*

La rotación del brazo es tan sólo el mismo control que se realizó en el numeral anterior sobre los motores, pero con la diferencia de que, debido a que se necesita ubicar el brazo en una posición específica para recoger la moneda y como no se cuenta con sensores que controlen el desplazamiento automáticamente por *software*, y además no se

tiene memoria de la posición en que queda el mecanismo cada vez que realiza la tarea, se tuvo la necesidad de colocar 2 controles o interruptores de 2 estados (*on/off*): uno para controlar la dirección de giro y otro para detener el movimiento del brazo y devolver el control al programa principal.

Estos interruptores son controlados por el programa a través del comando IF, es decir, se censa si fue presionado o no y se ejecuta la función que halla en cada caso. Se censa a través de un lazo como el siguiente:

```
loop: if pin(#) =  
1 then run      'Si el switch es presionado ejecuta run  
                'Sigue censando  
    goto loop  
pin(#): Es el Número del pin donde se encuentra conectado el switch en el FISCHERTECHNIK.
```

- *Activación de bajada y subida del brazo*

Para implementar esta aplicación, debido a que con el PISCTIC sólo se pueden controlar 2 salidas para motores, es necesario incluir un circuito adicional que convierta I/O digitales en análogas que puedan ser controladas desde el procesador. Puede ser mediante un puente H de transistores o una conversión de las salidas PS3 y PS4 del PISCTIC a manejo de potencia a través de un buffer 74S04 para cada pin; y mediante los comandos PEEK y POKE se puede realizar control sobre el dispositivo conectado.

Por facilidad de programación se

utilizó un puente H de transistores de potencia de referencia 2N2222A y diodos 4004 conectados a una fuente de 9V, que puede ser la misma que alimenta las ZUMNY [4].

Las subrutinas de led's y de sonido se encuentran en los programas de demostración del PICBASIC.

Al definir todas las subrutinas y probarlas se encadenan para tener el programa general y se realizan los respectivos ajustes, si son necesarios.

## CONCLUSIONES

- El mecanismo ha sido desarrollado para trabajar en lazo abierto sin recibir ningún tipo de retroalimentación de su entorno, debido a que no posee sensores de posicionamiento externo. Estos tipos de robots serían útiles para labores en las cuales sea óptimo un tipo de programación textual a nivel del robot, en la cual el robot sigue una rutina previamente establecida e invariable durante la operación. El paso siguiente es implementar una estructura de control bajo esquemas de lazo cerrado, con conocimiento de las variables del entorno que se relacionan con el mecanismo.
- La magnitud de los componentes utilizados para el desarrollo del robot sólo permite la realización de tareas con fines didácticos y limita su utilidad a la implementación de modelos de estudio. Sin embargo, las experiencias aprendidas durante

el desarrollo de este proyecto pueden ser extendidas hacia sistemas móviles de mayor alcance y complejidad.

- Debido a las limitaciones de manejo de corriente para motores que posee el PISCTIC, si se quiere implementar un diseño que tenga más de 3 motores es necesario una circuitería externa adicional para convertir las señales de las salidas digitales en análogos que puedan manejar los dispositivos.

### Referencias

BARRIENTOS, Antonio; PEÑÍN, Luis Felipe; BALAGUER, Carlos y ARACIL, Rafael. *Fundamentos de robótica*. Madrid, McGraw-Hill, 1997.

FISCHERTECHNIK. *Experimenta Computing: Assembly manual*. Berlín, Cornelsen experimenta, 1994.

TECHNOLOGY SERVICES INSTITUTE. *Zummy Interface Manual Vs. 2.0*, 1997.

TECHNOLOGY SERVICES INSTITUTE. *PicBasic for the PicStic Vs. 2.0 6/97*.

ROBOT: *Science & Technology*. September-October 1999.

### BIBLIOGRAFÍA

FU, K.S, R.C. GONZÁLEZ, C.S.G. LEE. *Robótica: control, detección, visión e inteligencia*. México, McGraw-Hill, 1994.

NORTON, Robert, L. *Machine design an integrated approach*. New Jersey, Prentice-Hall, 1996.

### Agradecimientos

Al revisor de este artículo, quien hizo posible su corrección y contribuyó eficazmente con su rectificación.