



## ROBOT MÓVIL RASTREADOR DE LÍNEAS CONTRASTADAS BASADO EN UN MICROCONTROLADOR DE 8 BITS

Victor Ramos-Viterbo y Yurián Zerón-Gutiérrez  
Laboratorio del Centro de Investigación de la Universidad La Salle  
e-mail: vramos@sparcciulsa.ulsal.mx  
labciulsa@sparcciulsa.ulsal.mx

### RESUMEN

En el presente artículo se describe el diseño, la construcción y el desempeño de un robot móvil capaz de seguir líneas contrastadas, ya sea una línea negra en un fondo blanco o una línea blanca en un fondo negro. El control del robot está basado en reglas que actúan de acuerdo a las señales obtenidas de un sistema de optosensores; estas reglas están programadas en un microcontrolador COP8 de la compañía National Semiconductor.

### ABSTRACT

The current paper describes the design, construction and performance of a line tracker mobile robot; the lines must be contrasted, a white line in a black background or a black line in a white background. The robot control is based in a set of rules dependent on the optosensors inputs, these rules are programmed in a COP8 microcontroller of the National Semiconductor Corporation.

### INTRODUCCIÓN

En los últimos años la necesidad creciente de automatizar y controlar procesos específicos de una forma sencilla y a un bajo costo, ha impulsado la creación y desarrollo de dispositivos programables capaces de realizar las más diversas tareas. Estos dispositivos son denominados microcontroladores y substituyen a grandes computadoras en aplicaciones específicas. Los campos de aplicación en los que se han hecho imprescindibles los microcontroladores son principalmente en la industria automotriz, en la de electrodomésticos, en la de equipo para telecomunicaciones y en la de equipo médico. En la actualidad, una casa típica norteamericana cuenta en promedio con 35 microcontroladores distribuidos en distintos equipos, y se espera que para el año 2000 sean más de 240.

En el laboratorio del Centro de Investigación se han realizado aplicaciones con microcontroladores (5, 6). Este trabajo es una primera aproximación en el área de robótica,

con el que se demuestra la utilidad del microcontrolador, ya que permite substituir un gran número de componentes, y como consecuencia se obtiene una sustancial reducción en las dimensiones y el peso de cualquier tipo de robot. Características deseables en el diseño de éstos.

El robot que se describe en el presente artículo es de tipo móvil, ya que está diseñado para hacer seguimiento de trayectorias en un plano. En una primera etapa sólo sigue trayectorias contrastadas sobre la superficie de trabajo de manera autónoma. Este robot es escalable de tal forma, que en el futuro será posible enviarle información de control no sólo de los sensores ópticos que tiene integrados, sino de cualquier otra fuente como una PC remota que le indique qué hacer en función de algún algoritmo o programa.

### EL ROBOT

El robot que se diseñó se denomina "Prototipo Omnidireccional Móvil" (P.O.Mo.) y está

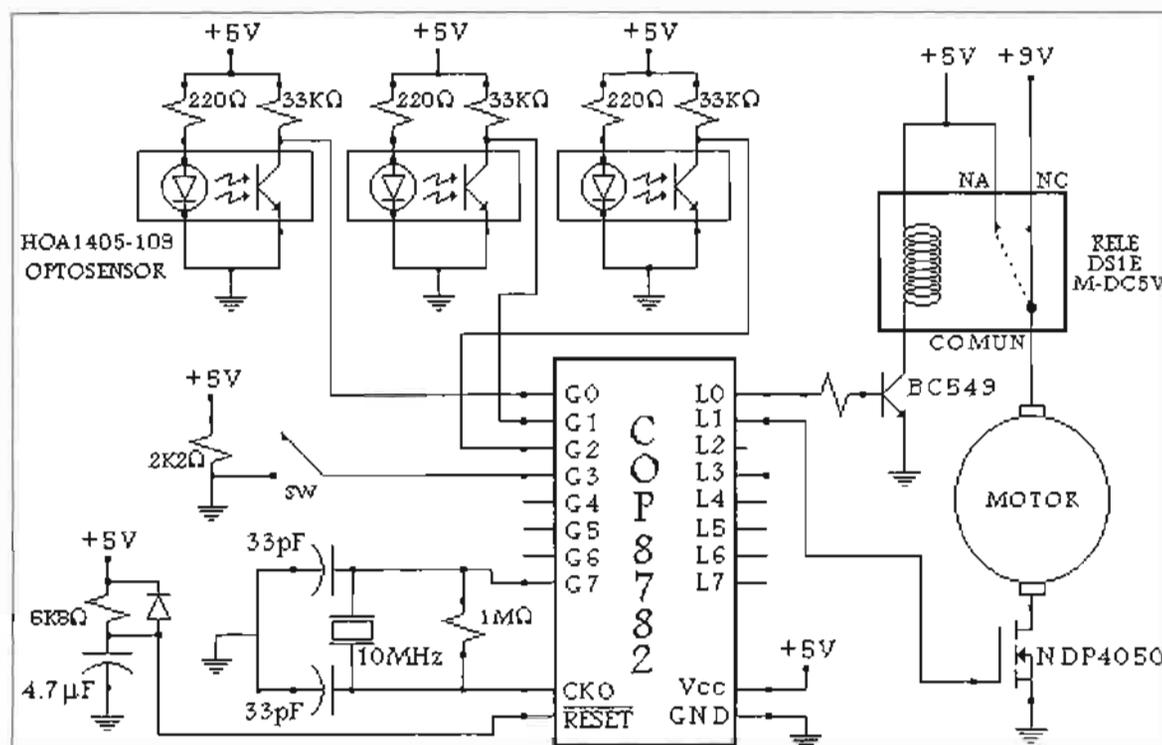


Figura 1. Diagrama esquemático del circuito de control del robot.

integrado por un mecanismo simple de robot móvil sobre tres puntos de apoyo. Los dos puntos de apoyo principales del robot son dos ruedas motorizadas independientemente que se encuentran alineadas por sus ejes de rotación en la parte delantera del robot. El tercer punto de apoyo, se encuentra en la parte

posterior del robot siendo éste una "rueda loca" que permita cualquier tipo de cambio en la dirección.

Como los motores son independientes, si ambos giran, le proporcionan tracción al robot para que éste avance en línea recta, si alguno de ellos es detenido, el robot usará la rueda asociada a este motor como eje o centro de un arco que será descrito por la rueda que se encuentra girando. La separación de 5 cm existente entre las ruedas determina el radio de giro mínimo del prototipo.

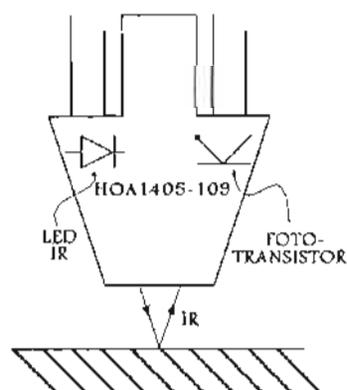


Figura 2. Optosensor Infrarrojo.

## EL CIRCUITO

El circuito (Figura 1) consta de tres distintos bloques funcionales. El primero está formado por los dispositivos sensores que permiten detectar y codificar un cambio en el contraste de la superficie como una señal de voltaje. Estas señales son enviadas al segundo bloque funcional del circuito que es el encargado de interpretar estas señales y está constituido por

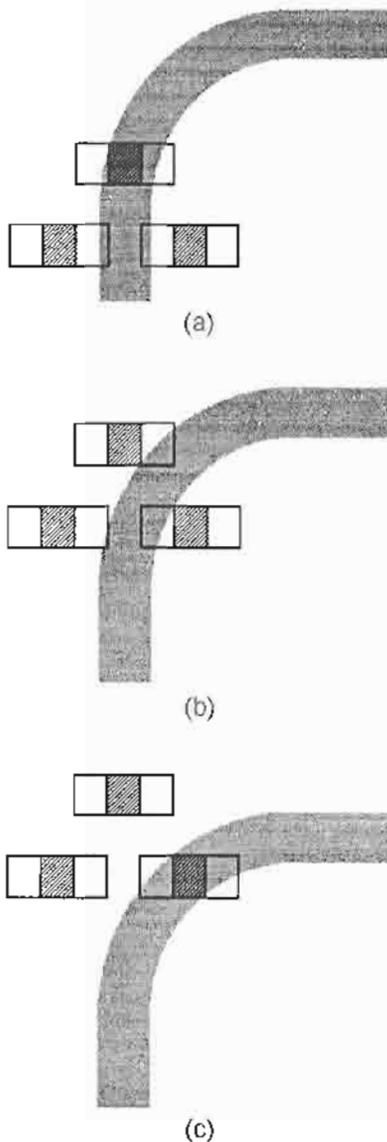


Figura 3. Diferentes situaciones que se les presentan a los sensores.

un microcontrolador que de acuerdo al programa contenido en su memoria ordena alguna acción de control para que el tercer bloque funcional del circuito actúe. Este tercer bloque lo forman dispositivos de potencia que activan los motores de acuerdo a las órdenes dadas por el microcontrolador.

El circuito de lectura está formado por tres optosensores infrarrojos del tipo HOA1405-109

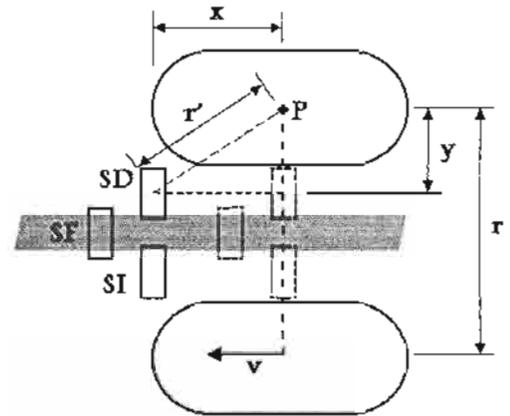


Figura 4. Ubicación de los Optosensores.

(3) que tienen su sensibilidad óptima a cinco milímetros de la superficie.

Cada sensor contiene un LED infrarrojo como emisor y un fototransistor como receptor acomodados en un ángulo convergente de tal forma que el fototransistor pueda recibir la luz infrarroja a través del rebote de ésta, en una superficie clara. (figura 2).

Los sensores están arreglados en triángulo, uno al frente y dos atrás, separados por una distancia poco mayor al ancho de la línea que describe la trayectoria (Figura 3a), donde el sensor frontal tiene la función de indicar anticipadamente la proximidad de una curva cerrada (Figura 3b), y los dos sensores traseros forman un riel óptico que le indica al robot hacia donde se realiza el cambio en la trayectoria (Figura 3c).

La velocidad de respuesta del arreglo de optosensores en relación con la velocidad de corrección del robot, está determinada por el coeficiente de precisión  $\rho$ , lo que es lo mismo, la longitud de los radios de los optosensores con la rueda eje ( $r'$ ), entre el radio de la otra rueda ( $r$ ). Experimentalmente se ha encontrado que para asegurar el seguimiento correcto de la trayectoria que marca la ruta, el coeficiente de precisión debe cumplir:

$$\rho = \frac{r'}{r} > 1$$



ya que esto asegura que la velocidad tangencial de los optosensores será mayor a la velocidad de giro de la rueda y, por lo tanto, el optosensor detectará variaciones en la ruta con sólo hacer ligeros cambios en la dirección (Figura 4).

El tercer bloque está formado por dos circuitos de potencia idénticos, uno para cada motor. Estos circuitos tienen la función de activar los motores en dos posibles velocidades. La activación de cada motor se realiza con un cero lógico en la compuerta del FET que aterriza al motor, la velocidad la define la conexión que realiza el relevador (4) a dos distintos voltajes de alimentación, para velocidad alta utiliza la conexión normalmente cerrado (NC) a 9 V y para velocidad baja utiliza la conexión normalmente abierto (NA) a 5 V.

### EL MICROCONTROLADOR

Se seleccionó un microcontrolador COP8782CJ de la serie de COP8 de 8 bits de National Semiconductor (1). Este dispositivo cuenta con dos puertos de entrada/salida de 8 bits cada uno (puerto L y puerto G), una terminal para restaurar (RESET), una de reloj (CK1), y dos de alimentación (Vcc y GND).

Para hacer la lectura de los sensores se configuraron los cuatro bits menos

G7	G6	G5	G4	G3	G2	G1	G0
NO	NO	NO	NO	TL	SD	SF	SI

**Figura 5. Configuración del puerto G.**  
TL=tipo de lógica (1-positiva y 0-negativa), SD=Sensor Derecha, SF=Sensor Frontal, SI=Sensor Izquierda.

L7	L6	L5	L4	L3	L2	L1	L0
NO	NO	NO	NO	MD	VBD	MI	VBI

**Figura 6. Configuración del puerto L.**  
VBI = Velocidad Motor Izquierdo (1-baja y 0-alta), MI = Motor Izquierda (1-encendido), VBD=Velocidad Motor Derecha (1-baja y 0-alta), MD = Motor Derecha (1-encendido).

significativos del puerto G como entradas, dejando los cuatro bits más significativos en tercer estado (figura 5).

Existen dieciseis posibles palabras de entrada en el puerto G, de las cuales las primeras ocho son para lógica positiva y las restantes ocho son para lógica negativa (Tabla 1).

**Tabla 1. Condiciones de los sensores.**

TL	SD	SF	SI	CÓDIGO	ESTADO	LÓGICA	CONDICIÓN
0	0	0	0	00h	robot al aire	Pos.	0
0	0	0	1	01h	curva suave a la izquierda	Pos.	1
0	0	1	0	02h	no opera	Pos.	2
0	0	1	1	03h	curva cerrada a la izquierda	Pos.	3
0	1	0	0	04h	curva suave a la derecha	Pos.	4
0	1	0	1	05h	sobre el riel	Pos.	5
0	1	1	0	06h	curva cerrada a la derecha	Pos.	6
0	1	1	1	07h	próximo a curva cerrada	Pos.	7
1	0	0	0	08h	próximo a curva cerrada	Neg.	8
1	0	0	1	09h	curva cerrada a la derecha	Neg.	9
1	0	1	0	0Ah	sobre el riel	Neg.	10
1	0	1	1	0Bh	curva suave a la derecha	Neg.	11
1	1	0	0	0Ch	curva cerrada a la izquierda	Neg.	12
1	1	0	1	0Dh	no opera	Neg.	13
1	1	1	0	0Eh	curva suave a la izquierda	Neg.	14
1	1	1	1	0Fh	no opera	Neg.	15



Tabla 2. Códigos y función de cada regla.

MD	VBD	MI	VBI	CÓDIGO	FUNCIÓN	REGLA
0	0	0	0	00h	motores detenidos	0
0	0	0	1	01h	no opera	x
0	0	1	0	02h	motor izquierdo velocidad alta	1
0	0	1	1	03h	motor izquierdo velocidad baja	2
0	1	0	0	04h	no opera	x
0	1	0	1	05h	no opera	x
0	1	1	0	06h	no opera	x
0	1	1	1	07h	no opera	x
1	0	0	0	08h	motor derecho velocidad alta	3
1	0	0	1	09h	no opera	x
1	0	1	0	0Ah	ambos motores velocidad alta	4
1	0	1	1	0Bh	no opera	x
1	1	0	0	0Ch	motor derecho velocidad baja	5
1	1	0	1	0Dh	no opera	x
1	1	1	0	0Eh	no opera	x
1	1	1	1	0Fh	ambos motores velocidad baja	6

Para enviar las señales de control a los motores se configuraron los cuatro bits menos significativos del puerto L como salidas, dejando los cuatro bits más significativos en tercer estado (Figura 6).

Para el control de los motores existen 16 posibles palabras de control, de las cuales sólo siete de ellas tienen una función específica (Tabla 2).

## EL PROGRAMA

El programa tiene un ciclo principal en el que se lee el puerto G y se compara continuamente con alguno de los códigos predefinidos como posibles condiciones, que simplemente indican la regla válida por medio del código de salida en el puerto L. Después de activar una regla, el programa regresa al ciclo principal para mantenerse al tanto de los cambios en las condiciones en los optosensores. En caso de no haber cambios, el ciclo se cierra siempre con la misma regla.

Se utilizó un sistema de desarrollo COP 8 Debug Module de MetaLink Corporation, para la emulación y depuración del programa, así como para la programación del circuito integrado de microcontrol.

En el apéndice A, se muestra el programa contenido en la memoria del microcontrolador, que sólo ocupa 186 bytes de los posibles 4 Kbytes de memoria ROM con que cuenta.

## CONCLUSIONES

El usar un microcontrolador para controlar el actuar del robot móvil, resultó ser una gran alternativa, ya que redujo el número de componentes que se usarían normalmente, y por lo tanto el peso y el riesgo de falla.

A pesar de que el COP8782CJ es el microcontrolador más chico de la familia de 8 bits, éste está sobrado para esta aplicación en particular, ya que sólo se utilizaron cuatro bits de cada puerto, dejando 8 posibles entradas/salidas sin usar (cuatro por puerto). Por otro lado sólo se usó el 5% de la memoria destinada para programas. Estos resultados muestran el potencial del microcontrolador para realizar funciones más complejas sujetas a un mayor número de variables.

## REFERENCIAS

1. COP8 Databook, National Semiconductor, 1994.



2. iceMaster COP8 Debug Module User's Manual v 1.2, Metalink Corporation, 1994.
3. Optoelectronics Device Data, Motorola, 1989.
4. Relay Technical Databook, Aromat Corporation, 1993.
5. Ramos Viterbo, V.; Caracas, G. y Gómez Ramírez, E. Implementación de un control difuso para un motor de corriente directa en un microcontrolador. *VI CIECE, Instituto Tecnológico de Morelia, Morelia, Michoacán, MÉXICO*, del 11 al 15 de marzo de 1996.
6. Gómez Ramírez, E.; Martínez Casas, G. y Figueroa Nazuno, J.; Diseño e implementación de una neurocomputadora en microcontroladores. *VI Congreso Nacional sobre Informática y Computación. Aguascalientes, Ags., MÉXICO*, del 23 al 25 de octubre de 1991.

0027 309B JSR X'009B  
0029 9DD6 LD A,PORTGP

Lee condición 4

002B 950F AND A,#X'0F  
002D 9204 IFEQ A,#X'04  
002F 308B JSR X'008B  
0031 9DD6 LD A,PORTGP

Lee condición 5

0033 950F AND A,#X'0F  
0035 9205 IFEQ A,#X'05  
0037 3093 JSR X'0093  
0039 9DD6 LD A,PORTGP

Lee condición 6

003B 950F AND A,#X'0F  
003D 9206 IFEQ A,#X'06  
003F 309F JSR X'009F  
0041 9DD6 LD A,PORTGP

Lee condición 7

0043 950F AND A,#X'0F  
0045 9207 IFEQ A,#X'07  
0047 30A3 JSR X'00A3  
0049 9DD6 LD A,PORTGP

Lee condición 8

004B 950F AND A,#X'0F  
004D 9208 IFEQ A,#X'08  
004F 30A3 JSR X'00A3  
0051 9DD6 LD A,PORTGP

Lee condición 9

0053 950F AND A,#X'0F  
0055 9209 IFEQ A,#X'09  
0057 309F JSR X'009F  
0059 9DD6 LD A,PORTGP

Lee condición 10

005B 950F AND A,#X'0F  
005D 920A IFEQ A,#X'0A  
005F 3093 JSR X'0093  
0061 9DD6 LD A,PORTGP

Lee condición 11

0063 950F AND A,#X'0F  
0065 920B IFEQ A,#X'0B  
0067 308B JSR X'008B  
0069 9DD6 LD A,PORTGP

Lee condición 12

006B 950F AND A,#X'0F  
006D 920C IFEQ A,#X'0C  
006F 309B JSR X'009B  
0071 9DD6 LD A,PORTGP

Lee condición 13

0073 950F AND A,#X'0F  
0075 920D IFEQ A,#X'0D  
0077 3097 JSR X'0097

**APÉNDICE A. Programa utilizado.**

Configuración del puerto L

0000 BCD1FF LD PORTLC,#X'FF

Configuración del puerto G

0003 BCD500 LD PORTGC,#X'00

0006 BCD40F LD PORTGD,#X'0F

Lee condición 0

0009 9DD0 LD A,PORTLD

000B 950F AND A,#X'0F

000D 9200 IFEQ A,#X'00

000F 308B JSR X'008B

0011 9DD6 LD A,PORTGP

Lee condición 1

0013 950F AND A,#X'0F

0015 9201 IFEQ A,#X'01

0017 308F JSR X'008F

0019 9DD6 LD A,PORTGP

Lee condición 2

001B 950F AND A,#X'0F

001D 9202 IFEQ A,#X'02

001F 3097 JSR X'0097

0021 9DD6 LD A,PORTGP

Lee condición 3

0023 950F AND A,#X'0F

0025 9203 IFEQ A,#X'03



```
0079 9DD6 LD A,PORTGP
Lee condición 14
007B 950F AND A,#X'0F
007D 920E IFEQ A,#X'0E
007F 308F JSR X'008F
0081 9DD6 LD A,PORTGP
Lee condición 15
0083 950F AND A,#X'0F
0085 920F IFEQ A,#X'0F
0087 308B JSR X'008B
0089 2009 JMP X'0009
Activa regla 0
008B BCD000 LD PORTLD,#X'00
008E 8E RET
Activa regla 1
008F BCD002 LD PORTLD,#X'02
0092 8E RET
Activa regla 2
0093 BCD003 LD PORTLD,#X'03
0096 8E RET
Activa regla 3
0097 BCD008 LD PORTLD,#X'08
009A 8E RET
Activa regla 4
009B BCD00A LD PORTLD,#X'0A
009E 8E RET
Activa regla 5
009F BCD00C LD PORTLD,#X'0C
00A2 8E RET
Activa regla 6
00A3 BCD00F LD PORTLD,#X'0F
00A6 8E RET
```