

Robot Rastreador: Dispositivos Lógicos Programables

M. Rizzo[†] E. Rojas Fredini[†] S. Sartorelli[†] P. Bocchio[†] C. Vizzarri[†]
F. G. Alva[†] A. Estepa[†] G. T. Bertoncello[†] A. Martinez[†]
Supervisores: S. Roatta[†] E. J. Padula[†]

[†]Facultad de Ingeniería y Ciencias Hídricas, Universidad Nacional del Litoral, Santa Fe, Argentina

Abstract— El presente trabajo consiste en el diseño e implementación de un robot rastreador de línea utilizando sensores infrarrojos de corto alcance, motores de corriente continua y un microcontrolador PIC16F84A.

Keywords— Robot rastreador, Microcontrolador, PIC16F84A, CNY70, Motor CC, PWM

1. INTRODUCCIÓN

El objetivo consiste que al situar el robot sobre una superficie de fondo blanco, en la cual hay trazado un camino curvo demarcado por una cinta negra, éste sea capaz de recorrer el camino, el cual puede estar formado por secciones tanto rectas como curvas. Para comenzar el recorrido, el robot debe tener sus dos sensores situados a los costados de la línea negra, de forma que estos reflejen contra la superficie blanca; al darle alimentación al robot éste pondrá sus dos motores en funcionamiento y comenzara a moverse siguiendo la línea. Su movimiento será recto mientras sus dos sensores detecten estar sobre un fondo blanco, al llegar a una curva donde alguno de los sensores, izquierdo o derecho, detecten estar sobre un fondo negro el microcontrolador, PIC16F84A, hará que el motor correspondiente al lado del sensor que detecto el fondo negro rote en sentido contrario y a la vez que sus dos motores disminuyan sus revoluciones por minuto, lo cual hará que el robot gire siguiendo la curva trazada por la cinta negra.

2. Materiales

Los componentes electrónicos utilizados para armar el robot rastreador se pueden ver en la tabla 1 junto con el presupuesto de cada uno. En la tabla 2 podemos ver algunas aclaraciones de los componentes.

3. Módulos

3.1. Sensores infrarrojos

Los sensores infrarrojos usados son CNY70. Éstos producen una salida analógica determinada por la magnitud de luz infrarroja reflejada. Por esto es que es posible sensorar cuando los costados del robot estan sobre una superficie blanca, i.e. la salida del sensor será un voltaje determinado, o sobre una superficie

Cuadro 1: Presupuesto de componentes

Cantidad	Componente	Precio (ARG)
1	16F84A	19,60
1	L293B	15,00
2	CNY70	13,50
2	Resistencia 4,7K	0,60
2	Resistencia 220	0,60
1	Resistencia 1k	0,60
2	Capacitor 0,1uF	0,50
2	Capacitor 33pF	0,50
1	Cristal	2,75
1	Push button	3,25
1	NTE40106B	3,25
8	1N4007	0,25
1	MC7805CT	2,00
1	Bateria 9V	14,75
1	Conector 9V	3,10
1	Socalo para PIC	1,00
1	Socalo para L293B	1,00
2	Potenciometro estéreo	4,00
1	Placa 10x10cm	3,50
2	Motor CC de 9V	26,00
1	Protoboard	23,00

negra, i.e. la salida del sensor será un voltaje contrario al anterior. Los voltajes correspondiente a una superficie blanca o negra sean superiores o inferiores dependen de si se realizo la conexion de los sensores en la forma A o B, las cuales se especifican la Fig. 2. La configuración utilizada en el proyecto fue la A. Los sensores se dispusieron en paralelo y apuntando en la misma dirección, hacia abajo del robot como se puede apreciar en la Fig. 8.

Al obtener las señales analógicas provenientes de los dos sensores infrarrojos, las digitalizamos mediante un *Schmitt Trigger*. La razón de esto es que es preferible el funcionamiento de éste, a dejar que el PIC16F84A lo haga usando sus niveles. El trigger es un comparador que evita que pequeños ruidos produzcan cambios bruscos en el valor detectado por el comparador. Esto lo realiza haciendo uso de umbrales diferentes para detectar el cambio de un valor positivo a uno negativo y vice-versa. Podemos

Cuadro 2: Detalle de componentes

Componente	Detalle
16F84A	Microcontrolador PIC
L293B	Driver, usado para motor CC
CNY70	Sensor infrarrojo
NTE40106B	Schmitt Trigger inversor
1N4007	Diodo
MC7805CT	Regulador de voltaje
Potenciometro estéreo	Usado para pruebas

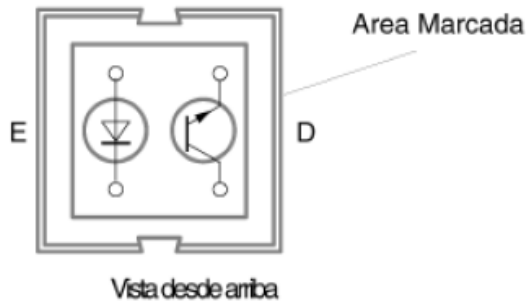


Figura 1: CNY70

ver el proceso en la Fig. 3.

Como dijimos anteriormente debido a la conexión en la que se disponen los sensores, en la señal de salida de éste un voltaje alto representa alta reflexión y uno bajo representa baja reflexión. Sin embargo el Schmitt Trigger invierte su señal de salida, lo cual implica que una entrada de voltaje bajo al microcontrolador represente alta reflexión y un voltaje alto represente una baja reflexión.

Las salidas del Schmitt Trigger fueron conectadas al PIC16F84A de forma que:

- El *sensor derecho* se conecte al *puerto RA0*.
- El *sensor izquierdo* se conecte al *puerto RA1*.

3.2. Control de los motores

El robot rastreador hace uso de dos motores de corriente continua. El control de estos está basado en el driver L293B de cuatro canales capaz de proporcionar en cada una de sus salidas hasta 1A. Cada canal es controlado por una señal de control TTL. En la Fig. 7 podemos ver el diagrama de pines del componente y su descripción. Las conexiones entre el puerto B, usado como salida, del PIC16F84A y el driver L293B son de la siguiente manera:

- *Puerto RB0* con Pines 2 y 7.
- *Puerto RB1* con puerto Enable 1.

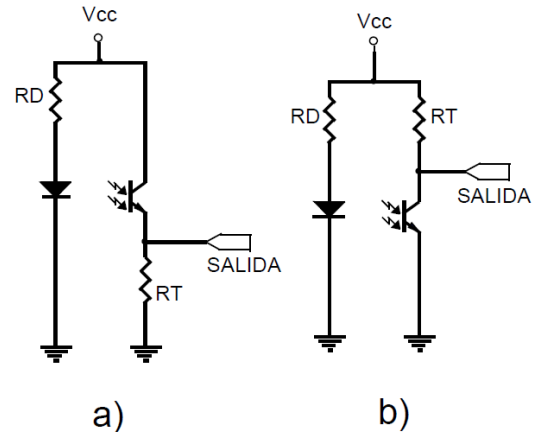


Figura 2: Las dos posibles tipo de conexión del CNY70: A voltaje alto cuando refleja, B voltaje bajo cuando refleja

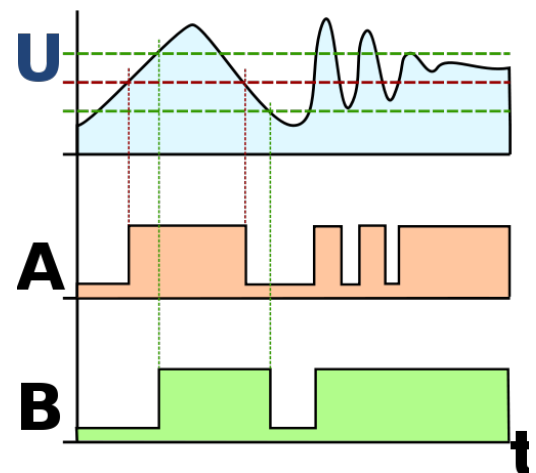


Figura 3: (U)Una señal analógica, (A)efecto de un comparador normal, (B)efecto de un Schmitt Trigger

- *Puerto RB2* con Pines 10 y 15.
- *Puerto RB3* con puerto Enable 2.

La conexión entre el driver L293B y los dos motores se puede ver en la Fig. 4.

3.3. Lógica del robot

El microcontrolador PIC16F84A será el encargado de controlar la lógica del robot, controlando según sus entradas de los sensores las salidas necesarias a los motores para reproducir las acciones que lo mantengan siguiendo la línea negra. En la Fig. 5 podemos ver la tabla de verdad del PIC. En esta es posible relacionar cada acción del robot con sus entradas desencadenantes y sus salidas necesarias.

3.3.1. Control de la lógica de las ruedas

Según lo dicho anteriormente, el proceso de control de las ruedas sigue el pseudo-código siguiente:

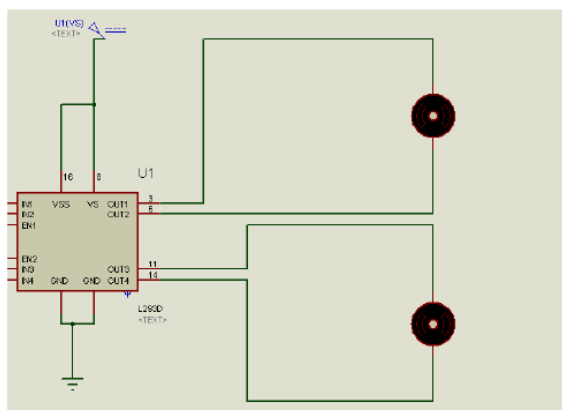


Figura 4: Conexion entre L293B y Motores CC

Entradas		Salidas				
RA0	RA1	RB3	RB2	RB1	RB0	Salida
0	0	1	0	1	1	Recto
0	1	1	0	1	0	Izquierda
1	0	1	1	1	1	Derecha
1	1	0	0	0	0	Parado

Figura 5: Tabla de verdad

1. If sensor derecha en blanco?
2. If sensor izquierda en blanco?
3. Ejecutar: derecho
4. Else
5. Ejecutar: izquierda
6. Else
7. If sensor izquierda en blanco?
8. Ejecutar: derecha
9. Else
10. Ejecutar: parado

El ensamblador correspondiente a la configuración inicial del PIC y este pseudo-código es:

```

;Configuracion inicial
start
    BSF STATUS,5 ;Set Banco 1
    MOVWF b'00011111'
    MOVWF TRISA ;Set PortA/input
    MOVWF b'00000000'
    MOVWF TRISB ;Set PortB/output
    BCF STATUS,5 ;Set Banco 0
    BSF STATUS,0
    CLRF PORTB
    CLRF PORTA
    MOVWF d'254' ;Definimos el ciclo
    MOVWF CICLO ;de trabajo
    GOTO INIZIO ;Seccion de PWM

next0
    BTFSS PORTA,0 ;If derecha=blanco

```

```

GOTO next1
GOTO next2

```

```

next1
    BTFSS PORTA,1 ;If izquierda=blanco
    GOTO derecho
    GOTO izquierda

```

```

next2
    BTFSS PORTA,1 ;If izquierda==1
    GOTO derecha
    GOTO parado

```

```

derecho
    MOVWF b'00001011'
    MOVWF PORTB ;Envia senal avance
    GOTO next0

```

```

izquierda
    MOVWF d'127' ;Definimos el ciclo
    MOVWF CICLO ;de trabajo(50%)
    MOVWF b'00001010'
    MOVWF PORTB ;Rueda izquierda gira en
    RETURN ;sentido contrario

```

```

derecha
    MOVWF d'127' ;Definimos el ciclo
    MOVWF CICLO ;trabajo(50%)
    MOVWF b'00001111'
    MOVWF PORTB ;Rueda derecha gira en
    RETURN ;sentido contrario

```

```

parado
    MOVWF b'00000000'
    MOVWF PORTB ;Envia senal de parado
    GOTO next0

```

3.3.2. Control de la potencia de las ruedas

Como dijimos al comienzo, al girar en ambas direcciones se hace que las dos ruedas tengan menor velocidad, haciendo que los motores trabajen a una fracción de su capacidad. Logramos esto usando *PWM(Pulse Width Modulation)*, la cual nos permite lograr que el voltaje medio aplicado a los motores sea una fracción del que aporta la alimentación, sin la necesidad de utilizar un potenciómetro. Podemos ver ejemplos de señales usando PWM en Fig. 6. Esta modulación posee dos niveles y la información codificada en el ancho de pulso determinara el voltaje con el cual trabajaran los motores en cada tiempo de simbolo. La implementación consiste en alimentar al motor con el voltaje total de la fuente de alimentación un tiempo T_{on} y no brindarle ninguna alimentación durante un tiempo T_{off} ; la suma de estos dos tiempos nos da el período del ciclo: $T_{on} + T_{off}$, que será constante. Luego para graduar el voltaje con el trabajan los motores, y por ende la velocidad que estos alcanzarán, asignaremos más o menos tiempo de T_{on} . Debemos notar que al ser constante el período del ciclo:

- Al aumentar T_{on} disminuira igualmente T_{off}
- Al disminuir T_{on} aumentara igualmente T_{off}

En nuestro código correspondiente a la PWM la variable CICLO contiene el ciclo de trabajo, estando este

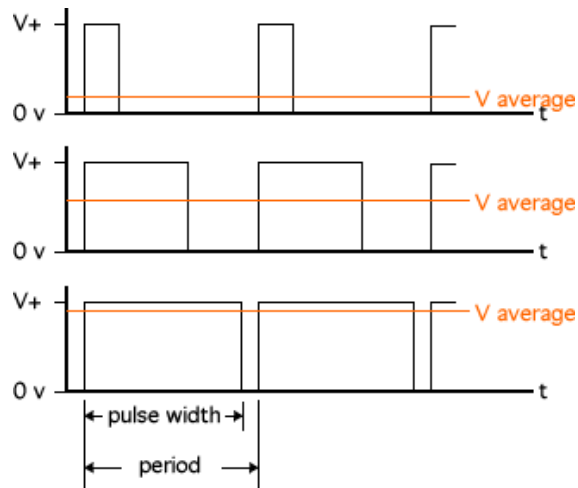


Figura 6: Ejemplo de señales moduladas con PWM, y como es la tensión de salida

entre 0 y 255, donde 0 corresponde a el ancho más corto y 255 al más largo, incrementando linealmente el ancho de pulso. El código es:

INIZIO

```
CALL    next0
MOVWF   CICLO,0 ;CONTATORE1=CICLO
MOVWF   CONTATORE1
```

T.ON

```
BSF     PORTB,1 ;RB1=1
BSF     PORTB,3 ;RB3=1
CALL    DELAY_15s
DECFSZ  CONTATORE1,1 ; While
CONTATORE1>0
GOTO    T.ON ;CONTATORE1 es 127
MOVWF   d'255' ;CONTATORE1=255
MOVWF   CONTATORE1
MOVF    CICLO,0
SUBWF   CONTATORE1,1 ;255-CICLO
```

T.OFF

```
BCF     PORTB,1 ;RB1=0
BCF     PORTB,3 ;RB3=0
CALL    DELAY_15s
DECFSZ  CONTATORE1,1 ; While
CONTATORE1>0
GOTO    T.OFF ;CONTATORE1 es 128
GOTO    INIZIO
```

DELAY_15s

```
MOVWF   d'2'
MOVWF   CONTATORE2 ;CONTATORE2=2
```

TEMPO

```
NOP
DECFSZ  CONTATORE2,1 ; While
CONTATORE2>0
GOTO    TEMPO
GOTO    FINE
```

FINE

```
RETURN
```

3.4. Código completo

```
INT_VAR UDATA 0x0C
w_saved RES 1
```

```
status_saved RES 1
```

```
ESTATUS EQU 0x03
TRISA EQU 0x85
TRISB EQU 0x86
PORTA EQU 0x05
PORTB EQU 0x06
CICLO EQU 0x0C
CONTATORE1 EQU 0x0D
CONTATORE2 EQU 0x0E
```

STARTUP CODE 0x000

```
NOP
MOVLW   high start
MOVWF   PCLATH
GOTO    start
```

INT_VECTOR CODE 0x004

```
GOTO    interrupt
```

PROG CODE

interrupt

```
MOVWF   w_saved
SWAPF   STATUS,w
MOVWF   status_saved
SWAPF   status_saved,w
MOVWF   STATUS
SWAPF   w_saved,f
SWAPF   w_saved,w
RETFIE
```

start

```
BSF     STATUS,5
MOVLW   b'00011111'
MOVWF   TRISA
MOVLW   b'00000000'
MOVWF   TRISB
BCF     STATUS,5
BSF     STATUS,0
CLRF    PORTB
CLRF    PORTA
MOVLW   d'254'
MOVWF   CICLO
GOTO    INIZIO
```

next0

```
BTFSS   PORTA,0
GOTO    next1
GOTO    next2
```

next1

```
BTFSS   PORTA,1
GOTO    derecha
GOTO    izquierda
```

next2

```
BTFSS   PORTA,1
GOTO    derecha
GOTO    parado
```

derecho

```
MOVLW   b'00001011'
MOVWF   PORTB
GOTO    next0
```

izquierda

```
MOVLW   d'127'
MOVWF   CICLO
MOVLW   b'00001010'
MOVWF   PORTB
RETURN
```

```

derecha
    MOVW    d'127'
    MOVWF   CICLO
    MOVW    b'00001111'
    MOVWF   PORTB
    RETURN

parado
    MOVW    b'00000000'
    MOVWF   PORTB
    GOTO    next0

INIZIO
    CALL    next0
    MOVF    CICLO,0
    MOVWF   CONTATORE1

T_ON
    BSF     PORTB,1
    BSF     PORTB,3
    CALL    DELAY_15s
    DECFSZ  CONTATORE1,1
    GOTO    T_ON
    MOVW    d'255'
    MOVWF   CONTATORE1
    MOVF    CICLO,0
    SUBWF   CONTATORE1,1

T_OFF
    BCF     PORTB,1
    BCF     PORTB,3
    CALL    DELAY_15s
    DECFSZ  CONTATORE1,1
    GOTO    T_OFF
    GOTO    INIZIO

DELAY_15s
    MOVW    d'2'
    MOVWF   CONTATORE2

TEMPO
    NOP
    DECFSZ  CONTATORE2,1
    GOTO    TEMPO
    GOTO    FINE

FINE
    RETURN

END

```

4. Trabajos futuros

Durante las pruebas realizadas al robot surgió un inconveniente; como el robot está programado para funcionar al máximo de velocidad en rectas, tiene problemas para corregir la marcha cuando se encuentra con una curva cerrada después de una recta larga. Esto se debe a la inercia que trae el robot, las posibles soluciones a este problema son las siguientes:

1. Reducir ancho de pulso en líneas rectas a un 90 % y en las curvas subirlo a 70 %.
2. Utilizar motores con caja reductora.

Pin	Nombre	Descripción	Patillaje
1	Chip Enable 1	Habilitación de los canales 1 y 2	
2	Input 1	Entrada del Canal 1	
3	Output 1	Salida del Canal 1	
4	GND	Tierra de Alimentación	
5	GND	Tierra de Alimentación	
6	Output 2	Salida del Canal 2	
7	Input 2	Entrada del Canal 1	
8	Vs	Alimentación de las cargas	
9	Chip Enable 2	Habilitación de los canales 3 y 4	
10	Input 3	Entrada del Canal 3	
11	Output 3	Salida del Canal 3	
12	GND	Tierra de Alimentación	
13	GND	Tierra de Alimentación	
14	Output 4	Salida del Canal 4	
15	Input 4	Entrada del Canal 4	

Figura 7: Diagrama de pines de L293B y su descripción

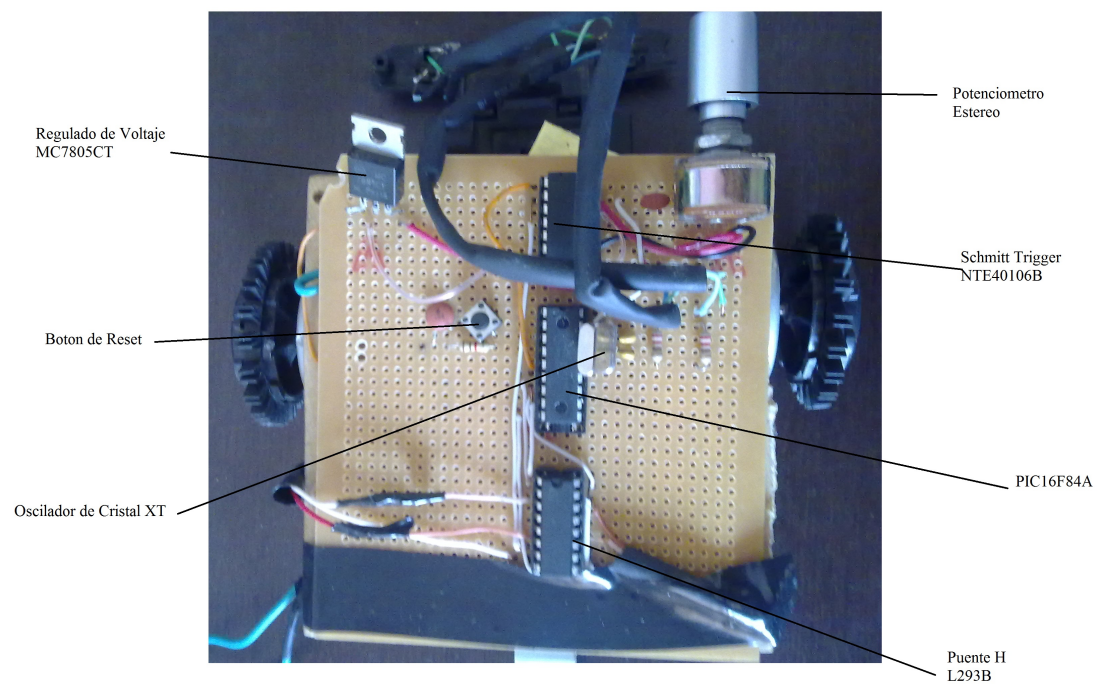
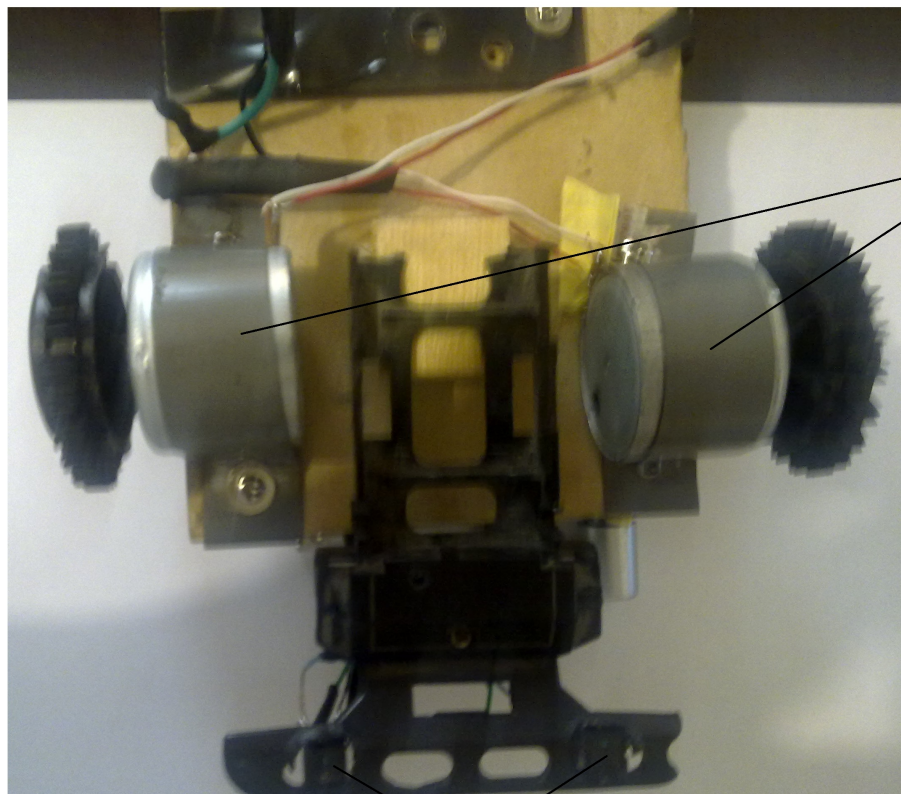


Figura 8: Vista superior del robot y sus componentes



Motores de 9V

Sensores Opticos Reflexivos
CNY70

Figura 9: Vista inferior del robot y sus componentes