

## CARRITO SEGUIDOR DE LÍNEA NEGRA

**RESUMEN:** En el presente informe se describe el procedimiento que se llevó a cabo para el diseño y montaje de un carrito seguidor de línea negra, con todas sus partes, la elección de sus componentes, el lenguaje a utilizar para la programación del PIC y, además, el código fuente y los datasheet de los mismos dispositivos utilizados.

**Palabras clave:** PIC 16F877A, sensor CNY70, Lm358, assembler.

### INTRODUCCIÓN

En la electrónica una de las más importantes e interesantes áreas es la de la robótica sólo porque hoy en día tiene una aplicación en casi todos los campos sino porque permite crear una cantidad ilimitada de modelos al igual que una ilimitada de aplicaciones prácticas. Se puede utilizar en el hogar, en el colegio, en la industria, en el comercio, en la investigación, en fin, en todos los campos del conocimiento y del quehacer humano puede haber una aplicación de la robótica.

Además, en nuestro caso, nos permite una aproximación a la programación y a la aplicación práctica de los conocimientos obtenidos teóricamente en clase.

### MÉTODO DE DISEÑO

Al comenzar a trabajar en el proyecto lo primordial era, primero (obvio), definir el proyecto, luego los materiales a utilizar y después el lenguaje a utilizar para programar el PIC.

Después de tener definido el proyecto, nuestros siguientes pasos fueron los siguientes:

- a. Definir si queríamos seguidor de línea blanca o negra. Para saber qué sensores utilizaríamos.
- b. Escoger los motores a utilizar. Teniendo en cuenta el peso de los motores y la potencia de los mismos.
- c. Escoger entonces el driver para controlar el motor.
- d. Seleccionar los amplificadores más adecuados para la aplicación.
- e. Definir la ubicación de los sensores.
- f. Elegir las llantas, el número de ellas y la ubicación en el seguidor.
- g. Se eligió un pequeño acondicionador de señal a la salida de los CNY70.
- h. Escogimos las baterías a utilizar para alimentar el carrito.
- i. Elegimos la carcasa del carrito.
- j. Elegimos el PIC.
- k. Escogimos el lenguaje de programación del PIC.

### JUSTIFICACIÓN DE ELECCIÓN

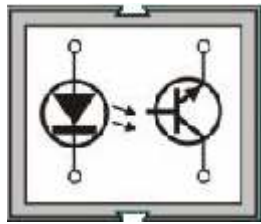
- a. Como ya se mencionó, se debió elegir entre seguidor de línea negra o blanca para poder elegir entre los diferentes tipos de sensores disponibles y adaptarse a las necesidades de la aplicación. Se eligieron entonces los sensores CNY70, que tienen gran precisión y son de fácil uso.



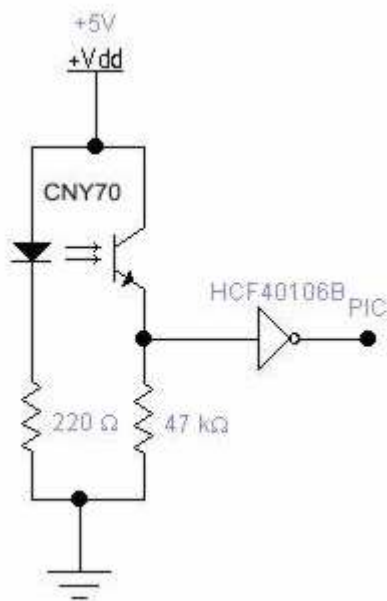
**Figura 1.** Encapsulado del sensor.

Se utilizó la siguiente configuración para los sensores CNY70 teniendo en cuenta

su configuración interna, como se muestra a continuación.



**Figura 2.** Diagrama interno del CNY70.



**Figura 3.** Conexión de los CNY70.

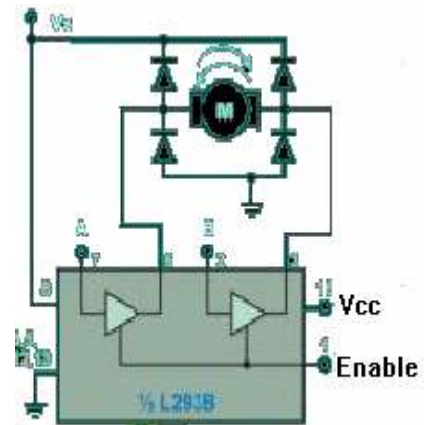
- b. Los motores eran un factor decisivo en el montaje puesto que son los que mueven toda la estructura deben tener una buena potencia, pero también deben cumplir con condiciones de consumo y peso que no los hagan inoperantes a la hora de su uso.



**Figura 4.** Motor DC.

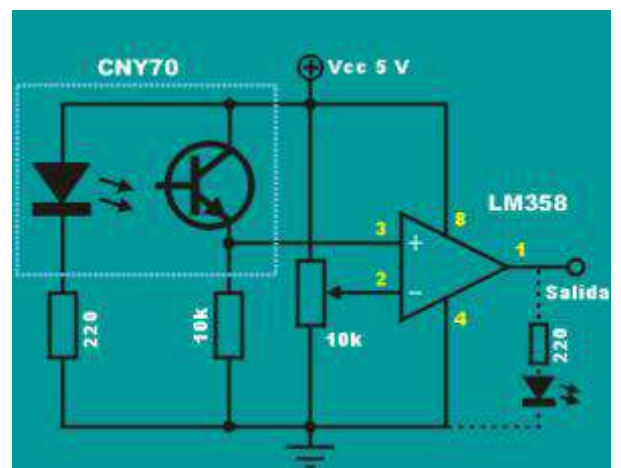
Por eso se eligió este tipo de motor que es muy común, de bajo costo y que se adapta a nuestras necesidades. 20518-890961 a 5V.

- c. Para el control de dirección y funcionamiento del motor se escogió el L293D, que lleva incorporados diodos para contrarrestar los picos que pudieran presentarse a causa de las bobinas de los motores. Además por la potencia que se obtiene por los puentes H que contiene.



**Figura 5.** L293D

- d. Se eligió este operacional por su buen funcionamiento y, además, porque puede ser alimentado por una sola fuente.  
e.



**Figura 6.** Amplificador.

- f. Se decidió ubicar los sensores al frente con una separación igual a la del ancho de la línea negra, para que detecten los bordes de la misma.

- g. Se escogió el HCF40106 para que enviara una señal limpia al PIC.
- h. Se eligieron tres llantas. Dos traseras grandes de aproximadamente 10cms. De diámetro y una central delantera de 4cms. Aproximadamente. Esta se ubicó sola adelante para permitirle mejor maniobrabilidad y giro al carrito.
- i. Como se trataba de dispositivos de bajo consumo casi todos elegimos ubicar baterías recargables tipo AA, que aunque brindan muy poca corriente, son livianas y permiten la alimentación a todos nuestros componentes. Además el tiempo de uso del carrito va a ser por instantes cortos lo que nos permite esta elección. No optamos por baterías más grandes tipo eléctricas pues si bien su potencia y amperaje son muy buenas también tienen gran peso y tamaño, más adecuadas para un **SUMO**.
- j. La carcasa a utilizar será en acrílico por su bajo peso y costo. El diseño y la forma están por definir hasta el montaje total del carrito. Ésta influye en el consumo de potencia que al final tenga el seguidor.
- k. El PIC elegido es el que hemos venido utilizando en el laboratorio, pues ya es conocido por nosotros y nos brinda los rendimientos que necesitamos para esta aplicación. De pronto el oscilador debería ser de una frecuencia un poco mayor, pero nos damos cuenta que tampoco es tan crítico este aspecto para el uso que le vamos a dar. El oscilador indicado debería ser más o menos del doble, generalmente a trabajar con un rango de frecuencias de 10MHz.

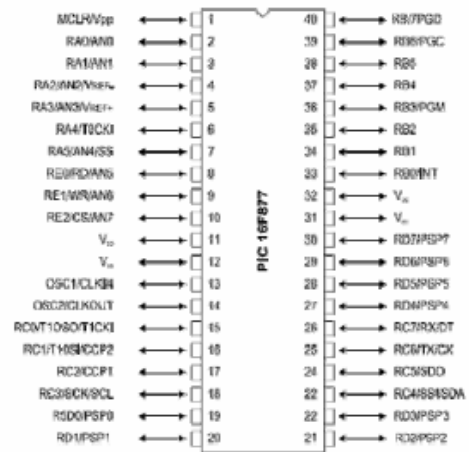
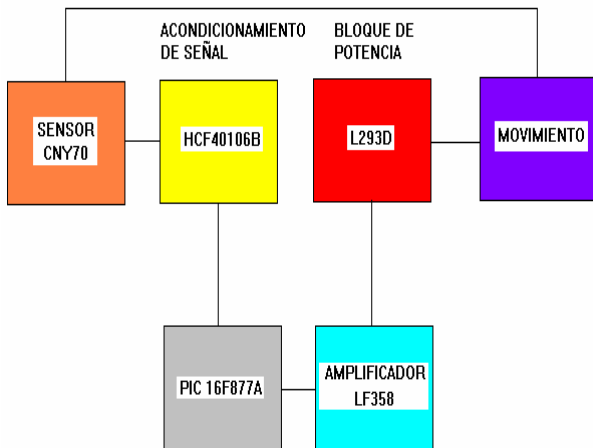


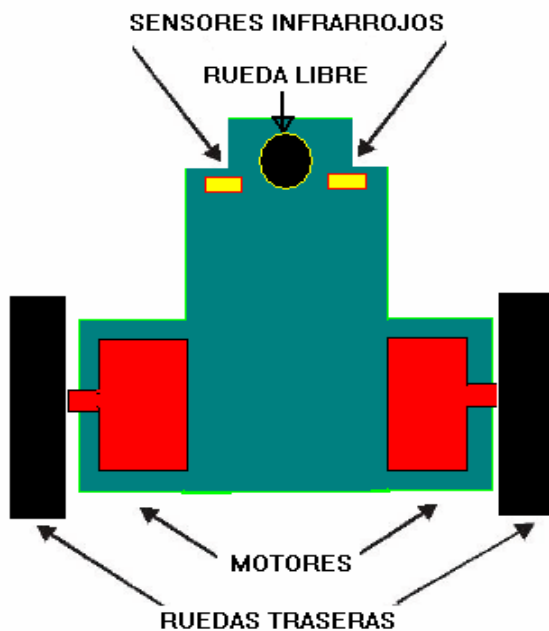
Figura 7. PIC 16f877A.

- l. Por último, la elección del lenguaje a utilizar para programar el PIC, fue crítica puesto que en la materia hemos venido trabajando en lenguaje C y es muy fácil trabajar con él por su alto nivel. Sin embargo, como habíamos comenzado a trabajar en un diseño anterior que no nos dio y a último momento nos decidimos por trabajar el proyecto actual nos vimos abocados a pedir asesoría y basarnos en un programa previo que venía diseñado para un PC16F45A. Utilizamos el compilador CCS y miramos el código y de ahí nosotros desarrollamos nuestro propio código en ASSEMBLER directamente. No nos pareció necesario, es más, nos pareció innecesario pasarlo luego a C. por esta razón presentamos en ASSEMBLER el código, ya que de todas maneras así lo compila sobre el PIC.

## DIAGRAMA DE BLOQUES



## DIAGRAMA SEGUIDOR DE LÍNEA



## CÓDIGO EN ASSEMBLER

```

INCLUDE "P16F877A.Inc"
CBLOCK .12
aux1,aux2,aux3,aux4,aux5,aux6,aux7,aux8,aux9,
aux10,aux11,aux12,CONTADOR,aux13,aux14,a
ux15
Endc

ORG 00
GOTO INICIO
ORG 05
    
```

```

INICIO bsf STATUS,RP0
        clrf TRISB
        movlw b'11111111'
        movwf TRISA
        bcf STATUS,RP0
        CLRF PORTB
        CLRF PORTA
        movlw .4
        movwf CONTADOR
        movlw b'00001111'
        movwf aux4
        movlw b'00000110'
        movwf aux5
        movlw b'00000011'
        movwf aux6
        movlw b'00000001'
        movwf aux7
        movlw b'00000000'
        movwf aux8
        movlw b'00001100'
        movwf aux9
        movlw b'00001000'
        movwf aux10
        movlw b'00000111'
        movwf aux11
        movlw b'00001110'
        movwf aux12
    
```

```

CICLO movf PORTA,0
        xorwf aux4,0
        btfsc STATUS,2
        GOTO ADELANTECON
    
```

```

        movf PORTA,0
        xorwf aux5,0
        btfsc STATUS,2
        GOTO ADELANTE
    
```

```

        movf PORTA,0
        xorwf aux6,0
        btfsc STATUS,2
        GOTO DERECHA
    
```

```

        movf PORTA,0
        xorwf aux7,0
        btfsc STATUS,2
        GOTO DERECHA
    
```

```

        movf PORTA,0
        xorwf aux8,0
        btfsc STATUS,2
        GOTO ATRAS
    
```

```

        movf PORTA,0
        xorwf aux9,0
        btfsc STATUS,2
        GOTO IZQUIERDA
    
```

```

        movf PORTA,0
        xorwf aux10,0
        btfsc STATUS,2
        GOTO IZQUIERDA
    
```

**movf** PORTA,0  
**xorwf** aux11,0  
**btfsc** STATUS,2  
**GOTO** DERECHA

**movf** PORTA,0  
**xorwf** aux12,0  
**btfsc** STATUS,2  
**GOTO** IZQUIERDA  
**GOTO** CICLO

#### **DERECHA**

**movlw** b'00000101'  
**movwf** PORTB  
**call** pausa  
**movlw** b'00000000'  
**movwf** PORTB  
**call** pausa  
**GOTO** CICLO

#### **IZQUIERDA**

**movlw** b'00000110'  
**movwf** PORTB  
**call** PAUSA  
**movlw** b'00000000'  
**movwf** PORTB  
**call** PAUSA  
**GOTO** CICLO

#### **ADELANTECON**

**movlw** b'00000100'  
**movwf** PORTB  
**call** PAUSA2  
**DECFSZ** CONTADOR,1  
**GOTO** CICLO  
**GOTO** PARAR

#### **ADELANTE**

**movlw** b'00000100'  
**movwf** PORTB  
**GOTO** CICLO

**atras movlw** b'00001000'  
**movwf** PORTB  
**GOTO** CICLO

#### **PAUSA movlw .1**

**movwf** aux1  
**movlw** .40  
**movwf** aux2  
**movlw** .40

**movwf** aux3  
**decfsz** aux3,F  
**GOTO** \$1  
**decfsz** aux2,F  
**GOTO** \$5  
**decfsz** aux1,F  
**GOTO** \$9  
**Return**

#### **PAUSA2 movlw .1**

**movwf** aux13  
**movlw** .250  
**movwf** aux14  
**movlw** .120  
**movwf** aux15  
**decfsz** aux15,F  
**GOTO** \$1  
**decfsz** aux14,F  
**GOTO** \$5  
**decfsz** aux13,F  
**GOTO** \$9  
**Return**

#### **PARAR**

**CLRF** PORTB  
**goto** PARAR

#### **END**

## CONCLUSIONES

- Nos pudimos dar cuenta que es muy fácil realizar casi cualquier programa o montaje utilizando tanto lenguaje C como el ASSEMBLER, sabiendo de dónde partir y cuáles serán los objetivos propuestos.
- La compilación de todos los códigos deben pasar invariablemente a ASSEMBLER pero es mucho más fácil programar en lenguaje C, aunque a veces para afinar detalles se puede ir al código básico y realizarlo mejor.
- A la hora del montaje encontramos dificultades especialmente a la hora de calibrar nuestros motores aún utilizando los componentes más idóneos para el caso. Esta dificultad nos impidió presentar el proyecto en el tiempo señalado.
- Los sensores también fueron de alguna dificultad, aunque al final se lograron calibrar teniendo en cuenta los parámetros exactamente y con la mayor precisión posible.
- Nos queda con marcado interés la conclusión casi obvia de tener en cuenta los imprevistos a la hora de realizar cualquier montaje pues, aunque la electrónica digital es en sí bastante precisa al contrario de la analógica, siempre hay inconvenientes y retrasos, generalmente debidos al factor humano.
- También como en estos proyectos se combinan factores

analógicos, mecánicos y digitales, esta confluencia hace aún más crítico el tiempo de preparación de los proyectos.

- Por último, al ir desarrollando el proyecto nos dimos cuenta de que hay muchísimas aproximaciones a un mismo problema y que es necesario tener en cuenta sobre todos los aspectos: mínimo número de componentes y costos. Teniendo en cuenta estos factores se puede llevar a cabo un trabajo eficiente, sin tener en cuenta estos ítems se puede llevar a cabo un trabajo sólo eficaz.



