

Automatización de juguetes

Grupo GARPE (Grupo de Automatización, Robótica y Pedagogía, ITPB)

Edgar Mario Rico, Profesor de Electrónica y circuitos digitales Mecatrónica. José Alfredo Palacio, Profesor de microcontroladores en el área Mecatrónica. Gilberto Carmona, Profesor de microcontroladores en el área Electrónica.

Víctor Manuel Hernández, Profesor de dibujo y elementos mecánicos en el área Mecánica y Mecatrónica

E-mail: automatización_juguetes@yahoo.com

Resumen

La automatización de juguetes es una idea surgida a partir de la necesidad de dar a los estudiantes del área mecatrónica, elementos mecánicos preestablecidos como lo son los vehículos de juguete electromecánicos y con base a estos desarrollar algún tipo de control. No con ello limitando al estudiante en el diseño de la parte mecánica sino, como punto de partida para que ellos conozcan su funcionamiento y que profundicen sobre la parte mecánica en semestres superiores.

Palabras Clave. Sensor infrarrojo, secuencia de control, PWM, juguete, microcontrolador, puente H, lenguaje de programación, obstáculos.

Abstract

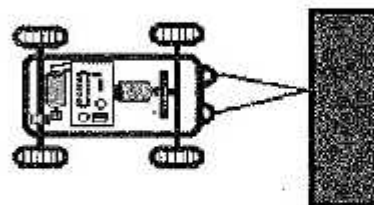
The automation of toys is an idea arisen from the need to give to the students of the area mecatrónica, preestablished mechanical elements as are it the vehicles of toy electromecánicos and based on these to develop some type of control. Not with it limiting al student in the design of the mechanical part but, like starting point so that they know their operation and that deepen on the mechanical part in upper semesters.

Key words. Infrared sensor, sequence of control, PWM, toy, microcontrolador, bridge H, programming language, obstacles

I. INTRODUCCIÓN

Al abordar un tema en las materias que dictamos en la carrera, queremos vincular al máximo elementos mecánicos y electrónicos para darle uniformidad a la razón académica del mecatrónico. Pero nos vemos con la limitante de que para los semestres iniciales son mas los conceptos electrónicos que mecánicos que los estudiantes han desarrollado. Para solventar en gran medida este problema recurrimos a los mecanismos ya implementados que se encuentran en el mercado, en lo que a juguetes se refiere y en particular los vehículos con tracción delantera o trasera y sacamos la parte eléctrica o electrónica y le damos un nuevo rumbo a su control mediante elementos electrónicos que se dan a conocer a los estudiantes como lo son los transistores, drivers de potencia, los microcontroladores, los sensores y se desarrollan una serie de talleres en diseño de circuitos impresos y en programación de los microcontroladores en lenguaje ensamblador y de alto nivel.

Figura 1.1 Vehículo detector de obstáculos



En primer lugar tratamos de nivelar a cada estudiante en los conceptos electrónicos básicos, para tal efecto dividimos a los estudiantes en dos grupos, uno mas avanzado que otro y colocamos metas definidas de acuerdo al nivel teórico-práctico adquirido en semestres anteriores.

Se desarrolló con ambos grupos el estudio del Eagle como software para el diseño de los diferentes circuitos impresos que llevarán los proyectos que definimos

como detector de obstáculos y seguimiento de líneas. Posteriormente se dió una introducción sobre el lenguaje ensamblador usado por el microcontrolador Pic16f873 de la empresa Microchip y sus respectivos pines.

Mediante software de alto nivel se programó este PIC y se obtuvo el control por parte de los estudiantes de una secuencia de detección de obstáculos simple y se sigue avanzando en el desarrollo de una segunda etapa.

II. REPASO Y ACLARACIÓN DE CONCEPTOS

Para nivelar a los estudiantes en la teoría electrónica a implementar ya sea que se haya estudiado o no en los cursos previos y particularizando en los elementos que forman parte del proyecto inicial como lo son los transistores y su implementación en el puente H para realizar los cambios de dirección en los motores que manejan la tracción y el avance adelante atrás, el concepto de sensor infrarrojo para la detección de obstáculos mediante la emisión de señales a partir del PWM generado por el microcontrolador PIC 16f873.

III. MANEJO DEL EAGLE

Luego de haber desarrollado esta primera parte comenzamos la enseñanza del Eagle con el cual diseñaremos las tarjetas explicadas en la parte introductoria obteniendo así los primeros resultados en algunos estudiantes con los que ya se llevaba mas tiempo en el proyecto y entregaron los planos de las pistas para ser llevadas a los acetatos y posteriormente a los impresos a ser soldados.

Figura 3.1 Pic 16F873 y sus salidas a las demás etapas en esquemático

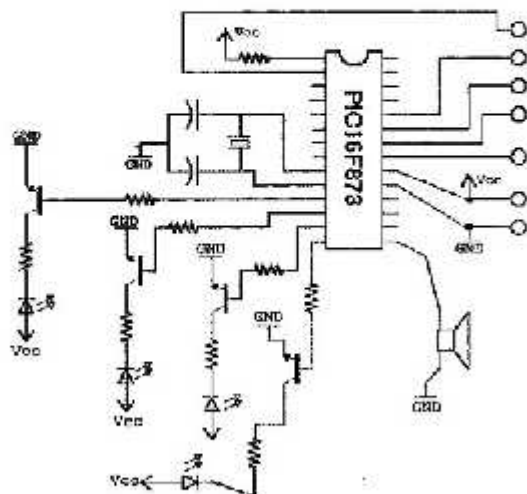


Figura 3.2 Board para el circuito de la figura 3.1

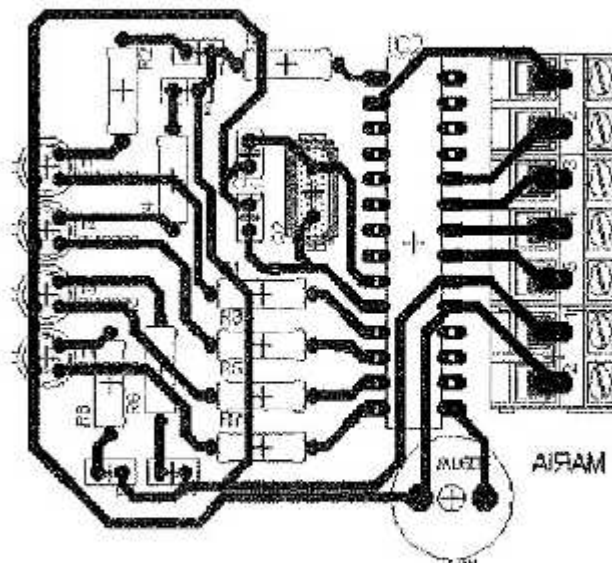


Figura 3.3 Esquema de uno de los dos Puentes H implementados

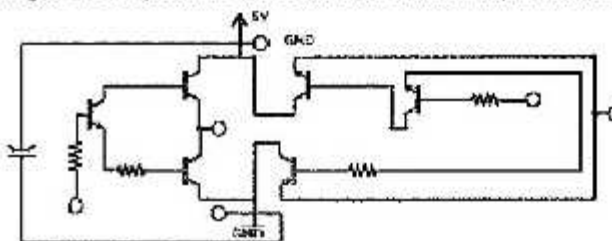


Figura 3.4 Circuito anterior en formato board con los dos puentes H

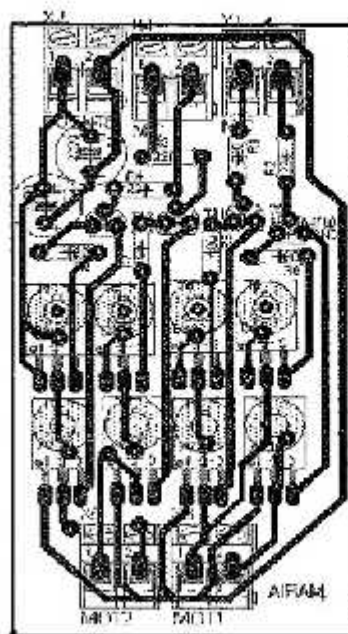
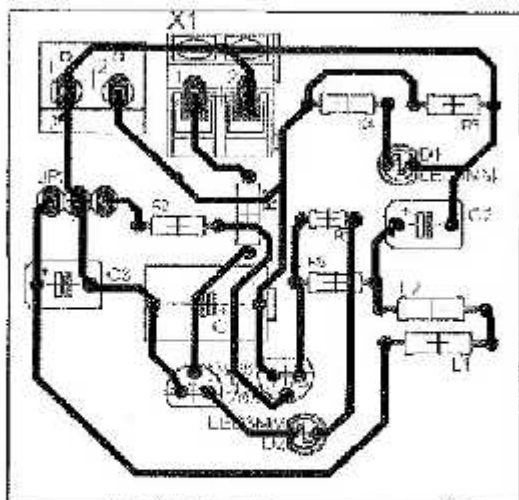


Figura 3.5 Etapas de los sensores infrarrojos



Las tres etapas tienen entradas y salidas las cuales se acoplan las diferentes tarjetas mediante borneras.

Con esta primera parte pretendemos que la totalidad de aproximadamente 20 estudiantes que asisten al taller periódicamente, estén en la capacidad de implementar estos circuitos en forma independiente y puedan en su medida, continuar con el siguiente paso que nos concierne y es el de dar a los estudiantes las bases para la programación del microcontrolador mediante lenguaje ensamblador y de alto nivel.

IV. PROGRAMACIÓN DEL MICROCONTROLADOR

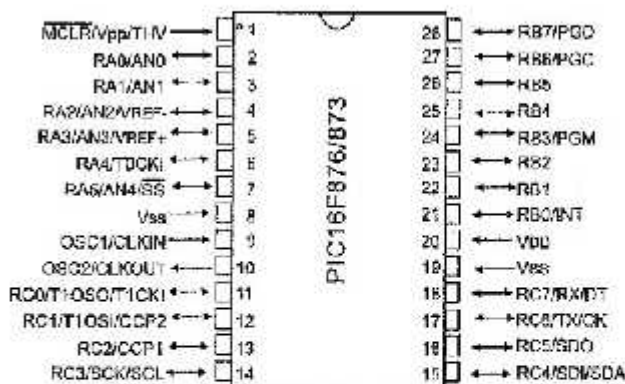
A. Lenguaje ensamblador

Para programar el microcontrolador, tenemos la posibilidad de hacerlo en lenguaje ensamblador y de alto nivel.

Daremos solo unas bases en ensamblador para conocer la forma de configurar los puertos de entrada y salida así como el conocimiento de los registros internos y la configuración de las entradas analógicas y las salidas PWM.

El PIC 16F873 es un dispositivo microcontrolador cuya programación es implementada en memoria flash sobre la base de 35 instrucciones y posee 28 pines en su estructura externa ordenados de la siguiente manera.

Figura 4.1 Esquema del microcontrolador pic16f873



Haremos uso solo del puerto B para activar los diferentes elementos como son los motores, el buzzer que indica alarma de reversa, los sensores infrarrojos y los leds indicadores de reversa o dirección. También usaremos RC2/CCP1 pues este pin está implementado como salida PWM por medio de la cual enviaremos el tren de pulsos que rebotarán sobre la pared o el obstáculo indicando la proximidad a éste y por ende la toma de decisiones en el algoritmo interno.

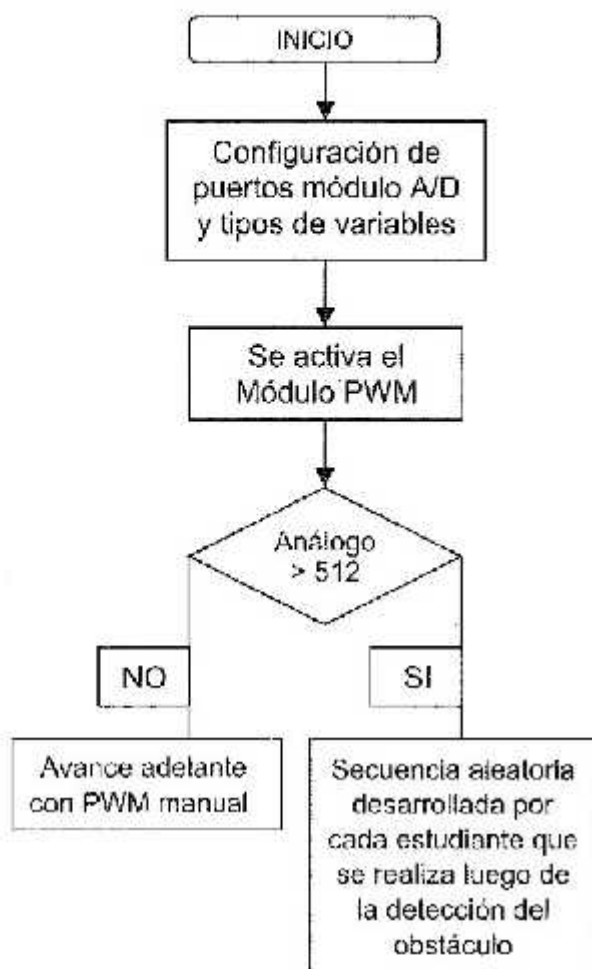
Habiendo adentrado a los estudiantes en el lenguaje ensamblador mediante programas simples, nos enfocamos ahora a la programación en lenguaje de alto nivel, ya que con este agilizamos el control que los estudiantes deben realizar.

B. Lenguaje de alto nivel para programar el PIC mediante el Picbasic

La programación se simplifica considerablemente al usar este programa pues una sola línea de código en alto nivel equivale a muchas líneas de código en lenguaje ensamblador.

Se les explicó a los estudiantes las instrucciones más relevantes en formato Picbasic con las cuales podían perfectamente programar el control del vehículo.

Para iniciar la programación haremos uso de un diagrama de flujo que explique claramente las subrutinas que queremos implementar en el primer proyecto.



La secuencia es independiente entre los estudiantes. Uno de los programas desarrollado en PicBasic por una estudiante cumpliendo con el anterior diagrama de flujo es el siguiente:

Configuración del módulo A/D y el oscilador de 20 MHZ

```

define osc 20
define ADC_BITS 10
define ADC_CLOCK 3
define ADC_SAMPLEUS 50
  
```

Configuración de puertos y tipos de variables a utilizar

```

ticon = %00010000
adcon1 = %10000010
portb = 0
porte = 0
  
```

```

trish = %00001111
trise = %00000111
analogo var word
conteo var word
efecto var byte
efecto1 var word
clear
pause 2000
gosub pwm_init; Llamada a la
                ; Configuración del PWM
  
```

Ciclo infinito en el que se chequea permanentemente el estado del canal análogo cero indicando la proximidad del obstáculo

```

while (1)
  adcin 0, análogo
  if (análogo > 512) then
    portb = %00000000
    pause 2000
  
```

Efectos con los indicadores de reversa mediante el Buzzer y los Leds de Luz a chorro

```

for conteo = 0 to 4000
  portb = %01000000
  pauscus 600
  portb = %00000000
  pauseus 400
  efecto1 = conteo//500
  if (efecto1 = 0) then
    if (efecto = 0) then
      porte = %11001000
      efecto = 1
    else
      porte = %00110000
      efecto = 0
    endif
  endif
endfor
next conteo
  
```

Se detienen los Motores por un segundo

```

portb = %00000000
pause 1000
  
```

Se genera un PWM manual para el vehículo durante 4 segundos

```

for conteo = 0 to 2000
  
```

```
portb = %10000000  
pauseus 600  
portb = %00100000  
pauseus 400  
next conteo
```

```
for conteo = 0 to 2000  
portb = %10100000  
pauseus 600  
portb = %00100000  
pauseus 400  
next conteo
```

Avance normal hacia adelante si no se detecta el obstáculo pero mediante un PWM manual

```
cise  
portb = %10000000  
pauseus 600  
portb = %00000000  
pauseus 400  
endif
```

Esta parte del programa genera un PWM para los pulsos infrarrojos que chocan con el obstáculo y son retornados a la entrada análoga del canal cero.

```
pwm_init:  
PR2=128; periodo  
CCPR1L= 6 ;cielo de dureza  
T2CON=4 ; activación del  
;timer y preescalado de 1  
CCP1CON=%00001100 ;PWM  
return  
wend
```

V. CONCLUSIONES

- Este proceso de conformación de un grupo mecatrónico ha dado la oportunidad de conocer diversas habilidades y conocimientos que manejan algunos estudiantes en la implementación y adecuación de sistemas mecánicos aptos para ser automatizados.
- Se ha logrado concienciar a los estudiantes que no solo se deben preparar para que profesionalmente actúen en el campo del mantenimiento sino también en el área del diseño y desarrollo de prototipos.
- A través de los juguetes se pueden dar soluciones a problemas reales como el rescate de víctimas, asistencia médica, entrega de víveres en catástrofes generados por sismos o derrumbes.

Referencias

1. BERRIO TOBON, Hader. Lógica para programación de computadores, Medellín: Lealon, 1995. p.17-75.
2. MALVINO, Albert Paul. Principios de electrónica, Medellín: McGraw-Hill Interamericana, 1991. p.165-299
3. OVIEDO RENGIFO, Efraín Manuel. Lógica de programación, Bogotá: Ecoe Ediciones, 2002. p. 15-199
4. SEDRA Adel S., SMITH, Kenneth C. Circuitos microelectronicos, México: Universidad de Oxford, 1998. p. 1-39, 884-915
5. PIC16F87X DATA SHEET 28/40-Pin 8-Bit CMOS FLASH Microcontrollers, Microchip, 2001