



UNIVERSIDAD DE ANTOFAGASTA  
DEPARTAMENTO DE INGENIERIA ELECTRICA Y ELECTRONICA



## DISEÑO Y CONTRUCCION DE ROBOT MANTA-RAYA 2

Autor: Fernando Guerra H.



## Emisor IR:

En el Robot fueron probados dos modelos diferentes de emisores IR

1.- Emisor IR bajado desde Internet.

Este modelo fue bajado desde una página de la U. de Chile, pero con ciertas modificaciones echas por nosotros, y con esto es posible ajustar la frecuencia de cada timer NE555 que trabajan como osciladores de onda cuadrada.

Circuito esquemático del emisor:

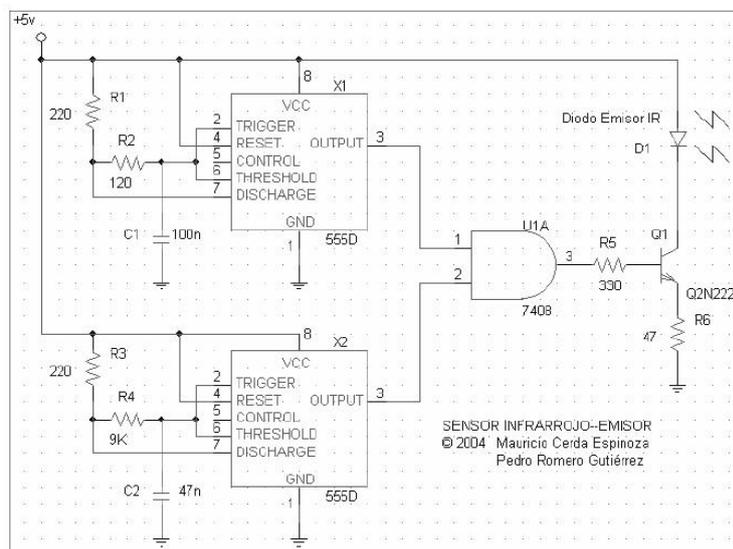


Figura 1. Circuito emisor IR con timer 555.

Es bastante sencillo y de bajo costo, el gran problema que posee es en la calibración de los componentes, mas bien dicho en los potenciómetros para obtener la frecuencia exacta de trabajo del receptor IR IRM8601S ([www.victronics.cl](http://www.victronics.cl)), por lo cual constantemente se esta descalibrando y provocando errores en la señales que salen de cada oscilador NE555.

De la salida de cada 555 son ingresadas a una compuerta lógica del tipo NAND (7400) la cual multiplica estas señales y genera un pulso que puede ser detectado por el receptor IR. El oscilador 1 entrega en su pin 3 una señal cuadrada aproximadamente a 38 [Khz] que es la frecuencia que decepciona el receptor IR, modulada por el otro oscilador 555 que entrega una frecuencia de 1,3 [Khz]. Y por medio de un amplificador del tipo de transistores usando el 2N2222 o utilizando un operacional LM386 se aumenta la corriente de la compuerta lógica para luego alimentar los diodos infrarrojos que transmitirán la señal.



La pérdida de calibración de la frecuencia de 38[Khz] implica en la distancia y sensibilidad del dispositivo de recepción infrarrojos, es decir, a medida que se aleje de los 38[Khz], ya sea mayor o menor que este valor, la detección se dificulta, produciendo que el receptor lo capte a una menor distancia. Así como lo muestra la siguiente figura.

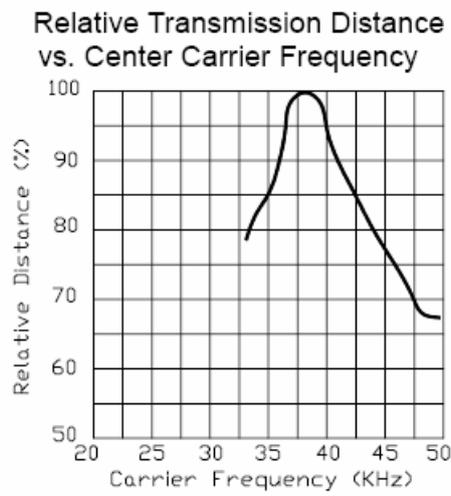


Figura 2. Relación frecuencia y distancia de recepción de IRM8601S

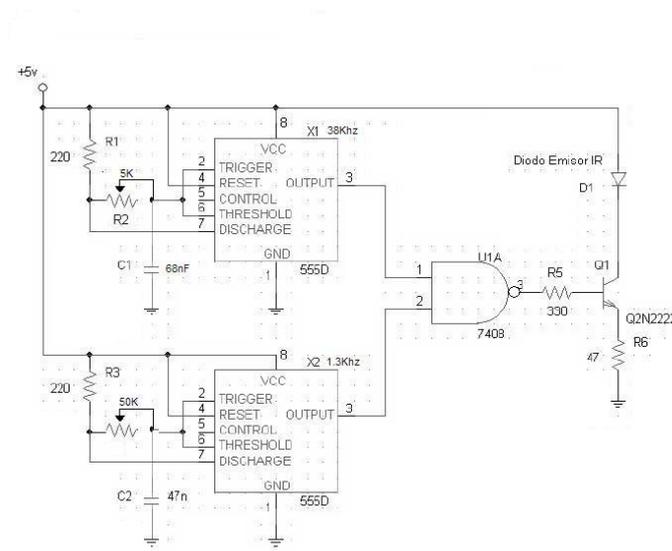


Figura 3. Circuito de Emisor IR.

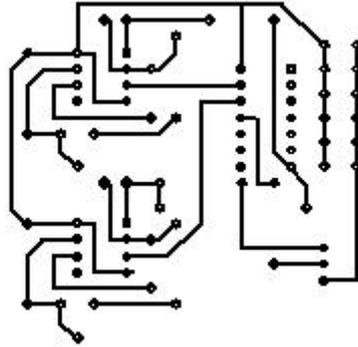


Figura 4. PCB de Emisor IR.

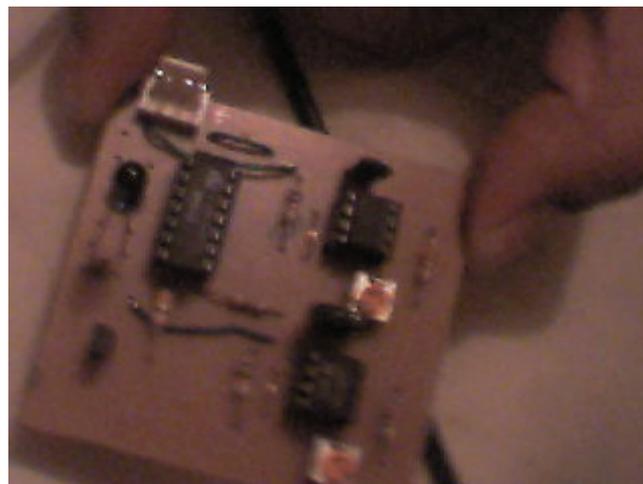


Figura 5. Circuito ya armado.

## 2.- Emisor IR por medio de PIC.

Para solucionar este problema de la calibración surgió una idea: “porque no usar un PIC para que entregue la oscilación exacta”. Y es por eso que se desarrollo un programa en Ensamblador (ASM) para el PIC 16F84A que funciona a 4 [Mhz], en el cual los pines asignados como salida RA0 y RA1 entregan cada uno la oscilación exacta que se requiere; es decir; en RA0 entrega 38 [Khz], mientras que RA1 entrega 1,3 [Khz] exactos, sin distorsión, sin variación y sin perder la calibración pues la oscilación cuadrada la entrega un programa que realiza las interrupciones en los puertos mencionados y multiplicadas después por una compuerta lógica NAND de un7400.

Existe un inconveniente, pues el programa es bastante largo para obtener la frecuencia requerida. El programa fue echo de acuerdo a la frecuencia de trabajo, es decir por cada línea de comando en Ensamblador corresponde a 1



[uS], para la frecuencia de 38 [Khz] son necesario hacer 26 líneas de comando con las instrucciones NOP, BSF, BCF y GOTO, mientras que para 1.3 [Khz] son necesaria 770 líneas con las instrucciones antes mencionadas, es claro que el programa tiene alrededor de 780 líneas en total, un poco largo pero funciona muy bien.

Fue agregado en su etapa final, un amplificador de transistores del tipo 2N2222, la razón de utilizar este amplificador es porque en la salida del 74LS00 es muy baja con una corriente máxima de salida de 25[mA], la cual solo puede alimentar un par de diodos infrarrojos, y al aplicarle un amplificador a transistores a su salida, reincrementa notablemente la cantidad de diodos infrarrojos, en el circuito armado fueron utilizados 8 Leds Infrarrojos y por cada uno de ellos circula una corriente de 7[mA], la cual es bastante para que entregue una señal de buena calidad y amplitud.

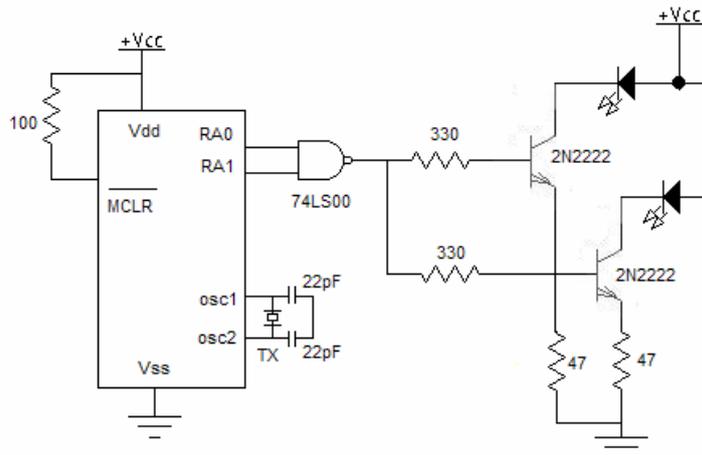


Figura 6. Circuito Emisor IR con PIC.

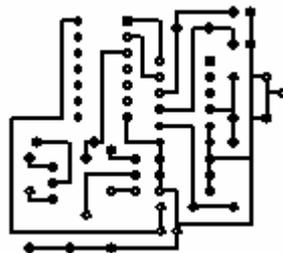


Figura 7. PCB para Emisor IR con PIC



Figura 8. Circuito PIC armado.

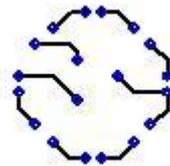


Figura 9. PCB para Leds Infrarrojos.



Figura 10. Circuito Leds ya armado.

La conexión de los led's IR es en serie, divididas en dos grupos de cuatro led's para aprovechar toda la corriente que entregan los transistores y como es usada una batería de 9[V] para alimentar todo el sistema no habría problema con el voltaje. Hay que considerar que el PIC y el 7400 solo trabajan con 5[V] y los transistores 2N222 se pueden alimentar con los 9[V], es por eso que se incluyo un limitador de voltaje que se muestra mas adelante, para estabilizar el voltaje a 5[V].

Para el diseño de las placas de montaje y la distribución de los componentes y la batería se tuvo que considerar las dimensiones de las mismas. Esto es clave, pues entre las placas con sus componentes y la alimentación se deben



contener dentro de un espacio muy reducido y cumplir con las bases impuestas sobre el tamaño y peso que exigen los organizadores al momento de diseñar y armar la baliza emisora de IR. Las cuales indican que su altura no debe ser mayor a 10[cm] y a 5X5[cm] en su base, y su peso no debe superar los 300[g.]. es por esto que se debe insistir en el buen diseño de la ubicación físicas de todos los componentes involucrados en ella.



## Receptor IR:

Este circuito esta basado en el diseño entregado por el fabricante del emisor IR IRS8601S.



Figura 11. Modulo de recepción Infrarroja IRS8601S.

De acuerdo a este esquema, se diseño el receptor para que reciba la señal del emisor IR que esta apostado sobre una plataforma fija de 5X5[cm] en el robot enemigo, de esta forma se es posible su detección, y entregando el receptor IR en su salida una interrupción de "0" y "1" hacia las entradas del PIC ya programado para que este ultimo realice ciertas rutinas de combate.

Se puede apreciar que es muy sencillo de fabricar, pues a la salida del receptor IR se puso un inversor del tipo 74LS04, esto es debido a que la salida de receptor IR (pin 3) entrega 5 [Vcc] y cuando recibe una señal infrarroja este valor tiende a subir, alrededor de 2.5 [V] y es por esto que se utiliza el inversor, pues en la salida de este entrega 0 [V] y 2.5[V], que para la entrada del PIC se tomarían estos valores como 0 y 1 lógicos.

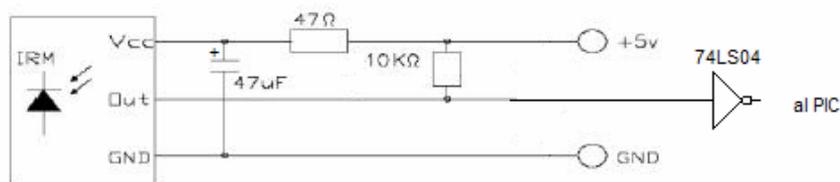


Figura 12. Circuito de Receptor IR.

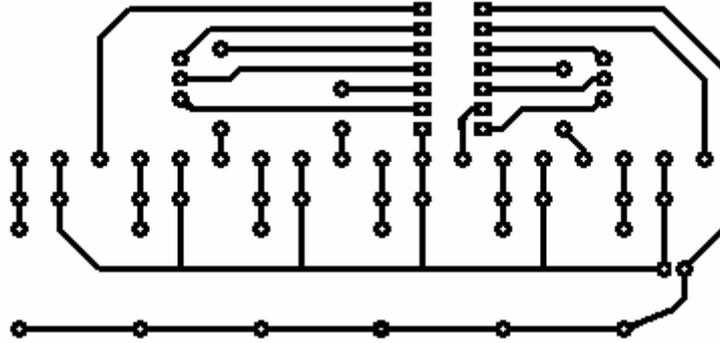


Figura 13. PCB de Receptor IR.



## Puente H:

Este tipo de circuito en particular es necesario para controlar la dirección y potencia de los motores de continua que tienen la labor de mover al robot por la pista. Hay dos tipos diferentes de puentes H que se utilizaron, con transistores y un circuito integrado que posee dos medios puentes H.

Para poder manejar los motores de corriente continua se utiliza el puente H, como muestran las figuras siguientes. Es básicamente constituido por cuatro interruptores abiertos y a medida que se ingresa un valor "1" estos se cierran de a pares logrando que le motor gire en un sentido o en otro.

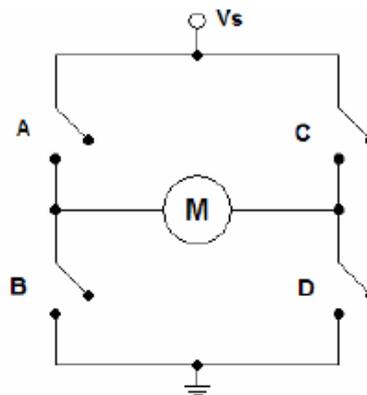


Figura 14. Puente H sin activar.

De esta forma, si A, D están cerrados (en 1), y B, C abiertos (en 0) la corriente circula en un sentido en el motor, lo que hace que este gire hacia un lado.

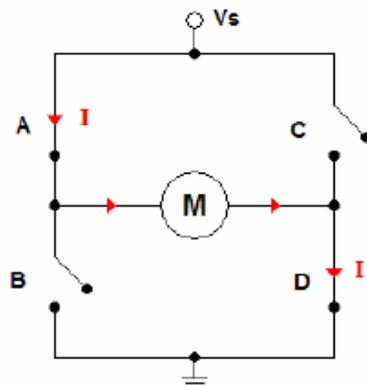


Figura 15. Puente H haciendo el giro horario.

Ahora, si B, C están cerrados (en 1), y A, D abiertos (en 0), entonces se produce un flujo de corriente en sentido contrario, lo que hace que el motor gire en la otra dirección.

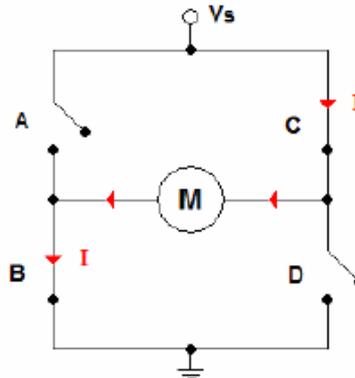


Figura 16. Puente H haciendo el giro anti-horario.

Así se hace circular la corriente en sentidos diferentes desde el punto de vista del motor haciendo que este gire en sentidos contrarios dependiendo de los interruptores cerrados. Estos interruptores se construyen con transistores trabajando en corte-saturación y se les cambia de estado mediante compuertas lógicas, y esto se puede realizar con transistores del tipo NPN y PNP o por medio de circuitos integrados que ya poseen esta configuración.

#### 1.-Puente H con transistores.

A pesar de su bajo costo por la cantidad de componentes, su gran problema radica en el control de las corrientes que circulan por el mismo, ya que en las pruebas con este tipo de circuito se quemaron varios transistores 2N2222 y 2N2907 a pesar que pueden manejar una corriente de 1[A] y al funcionar por bastante tiempo estos se queman inevitablemente ya que el consumo de los motores en conjunto (2) es muy alto cuando se quiere que funcionen a una mayor velocidad. Es por ello que se tiene que limitar las corrientes y diseñarlo para la aplicación que se requiera con las resistencias adecuadas. Para las entradas que controlan el giro de los motores, las corrientes deben ser muy pequeñas y así como la que circula por los transistores, que deberían de ser de un máximo de 50[mA] sin carga (sin los motores), pero en fin es muy complicado calcular los valores exactos de las resistencias, además que el circuito en si se pone engorroso de tanto componentes y por ende la fabricación del PCB.

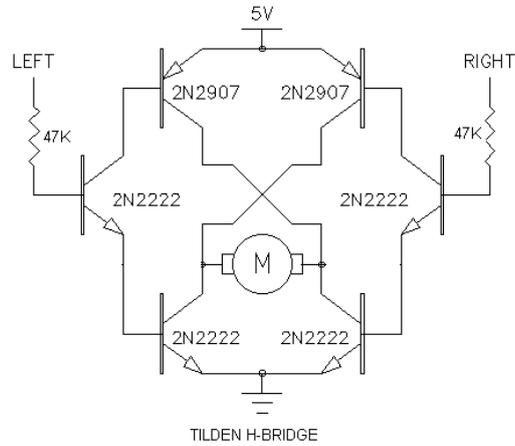


Figura 17. Circuito de Puente H.

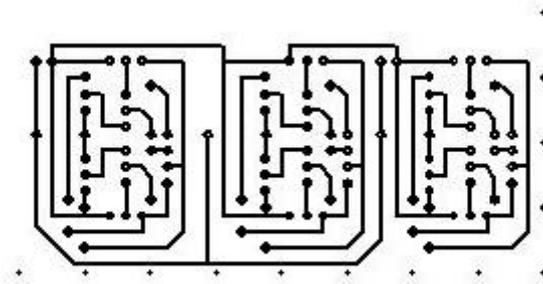


Figura 18. PCB Puente H.



Figura 19. Puente H ya armado.



## 2.-Puente H con Circuito Integrado.

Este es un mejor diseño, el circuito integrados SN754410 posee dos medios puentes H, es decir que puede controlar tanto dos motores de continua o un motor paso a paso, aunque un poco mas caro que el modelo con transistores, en este se pueden manejar corrientes hasta de 1[A] con su respectivo disipador de calor puesto en los pines centrales. Su gran característica es la utilización de una mínima cantidad de componentes, basta con solo unas cuantas resistencias para la entrada de control de dirección y ya estaría funcionando en perfectas condiciones. Hay que tener en cuenta que la alimentación del dispositivo, ya que para que este funcione de debe alimentar por el pin 16 con solo 5[V], mientras que para la alimentación de los motores se puede hacer por medio del pin 8 que puede llegar hasta lo 36[V], se tiene que considerar que la alimentación del CI y de los motores debe ser de forma independiente, es decir, que una batería alimenta el CI y otra a los motores, de lo contrario por experiencia al trabajar en conjunto con los PIC, puede generar errores en la realización de las rutinas de este ultimo.

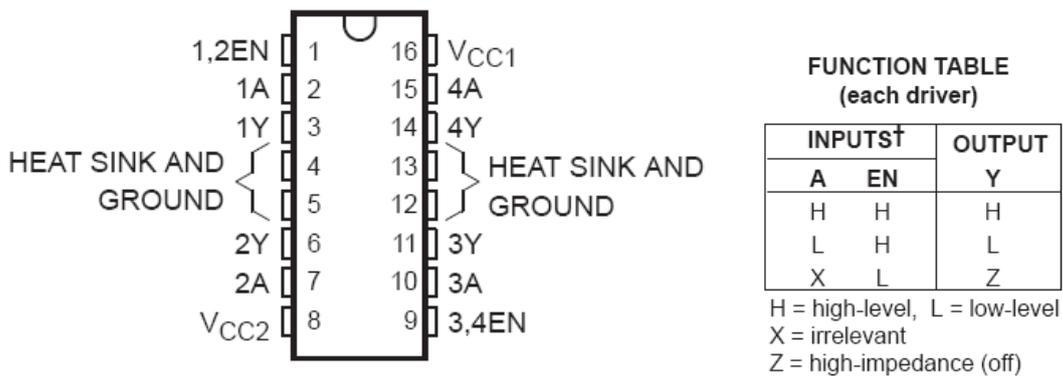


Figura 20. Puente H SN754410.

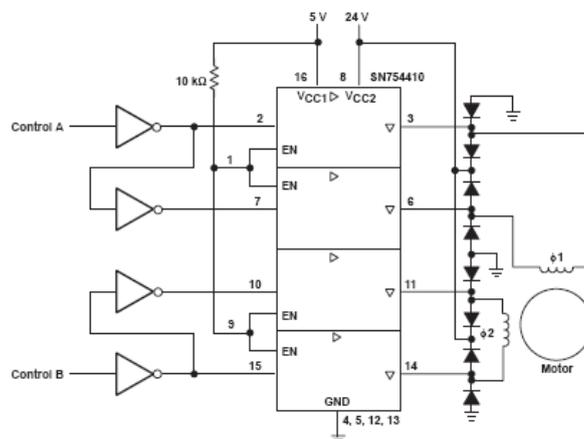


Figura 21. Circuito con SN754410

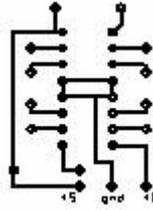


Figura 22. PCB de Puente H con SN754410.



### Sensor Táctil:

La propiedad de este sensor táctil es de evitar golpear y quedarse atascado el robot en los bordes de la pista de combate y también evitar chocar con el cilindro que esta en el centro del la pista.

Es diseño es muy sencillo al utilizar un interruptor de un polo y dos posiciones en donde uno de los pines es el común que va hacia una de las entradas de interrupción el PIC, los otros dos proporcionan los valores lógicos a trabajar, es decir, uno de ellos entrega el "0" lógico (0[V]) y el otro el "1" lógico (5[V]). Como ejemplo, si no se choca contra ningún obstáculo en el pin común habrá "0" por tanto el PIC no hará nada en particular a esta entrada, pero si choca contra un objeto en el pin común habrá "1" y en PIC programado hará una rutina cuando en su entrada haya este valor.

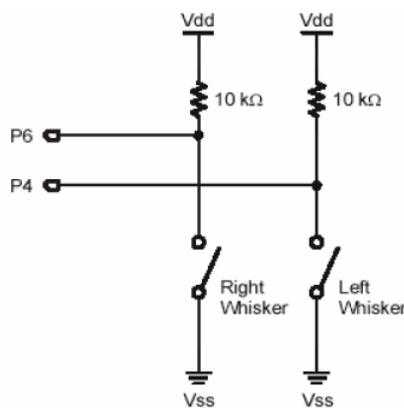


Figura 23. Alimentación del interruptor.

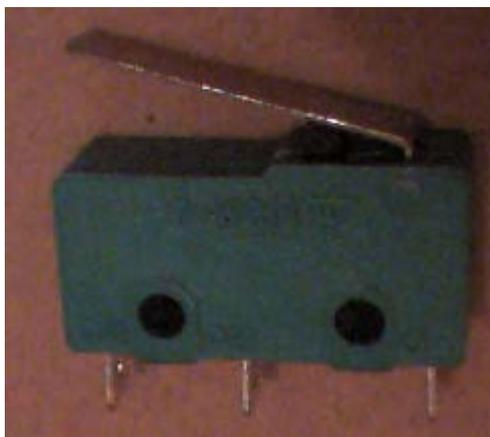


Figura 24. Interruptor – Sensor Táctil.



## **Motores:**

1.-Opción 1(utilizada en la competencia):

Para que el robot se mueva fueron utilizados los servos Hi-Tech modelo HS311 ([www.mirax.cl](http://www.mirax.cl)) y los Futaba. Estos servos fueron modificados para que funcionen como simples motores de continua. Este cambio fue necesario para poder utilizar los puentes H y también porque la programación de los tiempos de oscilación del tren de pulso para el control de la dirección de los servos implica que se tenga un conocimiento mas avanzado en la programación en ASM, en PicBasic es mas fácil pues se declaran directamente la duración de la onda cuadrada en milisegundos.



Figura 25. Hi-Tech HS-311.

Se procedió a eliminar el sistema de control, desoldando de los motores y reemplazarlos por un par de cables desde la alimentación de los mismos motores que luego van hacia los puentes H que controlaran su dirección.



Figura 26. Eliminación de sistema Servo.

Hay que tener cuidado en la alimentación de los motores ya que pueden soportar un máximo de 6[V], así que cuando se trabajen con los puentes H, ya sean estos de transistores o CI, se debe tener en consideración que la alimentación de los motores no debe superar este valor. Cuando se trabajen con los puentes H de CI (SN754410) debe tenerse cuidado en la ubicación del dispositivo, verificando la ubicación del pin 1, y también de las polarizaciones



que este posee, pues para alimentar el CI se necesita solo 5[V] mientras que para el voltaje que alimenta los motores puede llegar hasta lo 36[V].

También es necesario cortar el limitador de giro en los engranajes, que hace que el servo funcione solo en 180°, y de esta forma el eje del motor gire de forma continua o 360°, entonces el robot pueda avanzar, retroceder y hacer giros.



Figura 27. Corte de limitador de engranaje.

## 2.-Opción 2 (pruebas posteriores):

Para esta opción la cual implica saber un poco mas de programación en ASM es el de utilizarlos de forma directa como servos, es decir que por medio de las salidas del PIC 16F84A se puede controlar la dirección de servo. Todos los servos tienen tres cables de entrada, uno generalmente de color negro que se conecta al negativo de la alimentación, otro normalmente de color rojo que se conecta al positivo de la alimentación (4,8 a 6[V]) y por último uno que suele ser amarillo o blanco mediante el cual el servo recibe la señal de control.

La señal de control es un pulso de duración variable que se repite cada 20 milisegundos. La duración del pulso va desde 1[ms] hasta 2[ms] y determina la posición del servo, por ejemplo: si el pulso dura 1[ms], el cuerno del servo se posiciona en un extremo, si el pulso dura 1,5[ms] el cuerno toma la posición central y si el pulso dura 2[ms] entonces se posicionará en el otro extremo y así se producirá una relación biunívoca entre la duración del pulso y la posición que tome el servo.

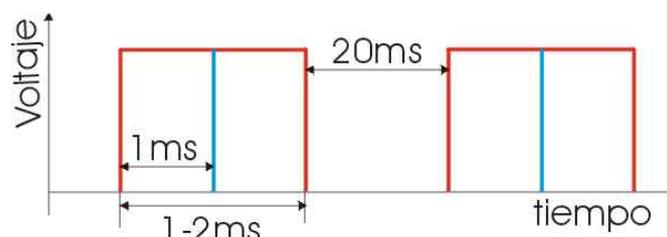


Figura 28. Señal de control.

Así, por medio de la programación en ASM, PicBasic o C, se pueden obtener estos valores y controlar un servos con un solo pin de salida del PIC, a



diferencia de utilizar un puente H en el cual se necesitan 2 pines de salidas del PIC, esto concluye en la optimización de las escasas salidas del PIC, en donde con el primer sistema solo es posible controlar 2 motores con 4 salidas, mientras que con este sistema se pueden controlar 4 servos de forma independientes y ahorrando espacio en el circuito del puente H y sus componentes.



### **Sistemas de control de Robot:**

Para los dos modelos de robot que se desarrollaron fueron utilizados dos tipos diferentes de control. Uno es por medio de un sistema ya armado y configurado llamado Basic Stamp 2 de Parallax ([www.parallax.com](http://www.parallax.com)) y el otro por medio de la programación de PIC's 16F84A de Microchip usando el grabador PIC-PG2.

#### 1.- Control con Basic Stamp 2:

Este sistema ya viene ensamblado, se trabaja bajo el lenguaje Basic por medio del programa PICBASIC que se puede bajar de la página oficial del fabricante.

Los BASIC Stamps son pequeñas computadoras que ejecutan programas en PBASIC. Tienen pines de I/O (entrada/salida) programables que pueden conectarse a diferentes dispositivos tales como pulsadores, LEDs, parlantes, motores, potenciómetros, etc. con la interfaz apropiada. Los pines de I/O del BASIC Stamp pueden ser conectados a reles, redes RS-232 y GPS (BSp24). Los BASIC Stamps poseen su propia EEPROM (donde almacenan su código), microcontrolador (ejecuta e interpreta el PBASIC "tokenizado"), regulador de tensión, y resistencias para comunicación serie.

Los Stamps ejecutan un BASIC "simbolizado". Esto significa que puede programar un BASIC Stamp usando un PC y un cable serie. No necesita "herramientas de desarrollo" para ser programado. El código es escrito en su PC usando editores de DOS o Windows gratuitos disponibles en Internet.



Figura 29. Basic Stamp 2.

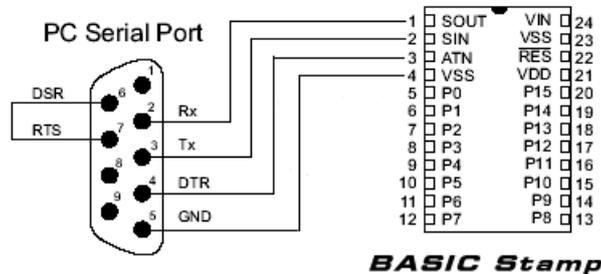


Figura 30. Conexión por Puerto serial de Basic Stamp 2.

2.- Control con PIC 16F84A (4Mhz):

El 16F84A es un microcontrolador PIC, el cual se puede programar por medio del lenguaje Ensamblador, y para ello se utilizo el programa PIC Simulator IDE, con el cual se puede compilar y simular a la vez los programas realizados para este PIC.

Las características del PIC 16F84A son las siguientes:

El primer paso importante el ver el diagrama de pines del PIC16F84, en el cual se observa como están distribuidos sus pines. Este circuito integrado cuenta con 2 puertos configurables como entradas o salidas según sea el caso y consta de 18 pines las cuales se encuentran asignadas de la siguiente manera:

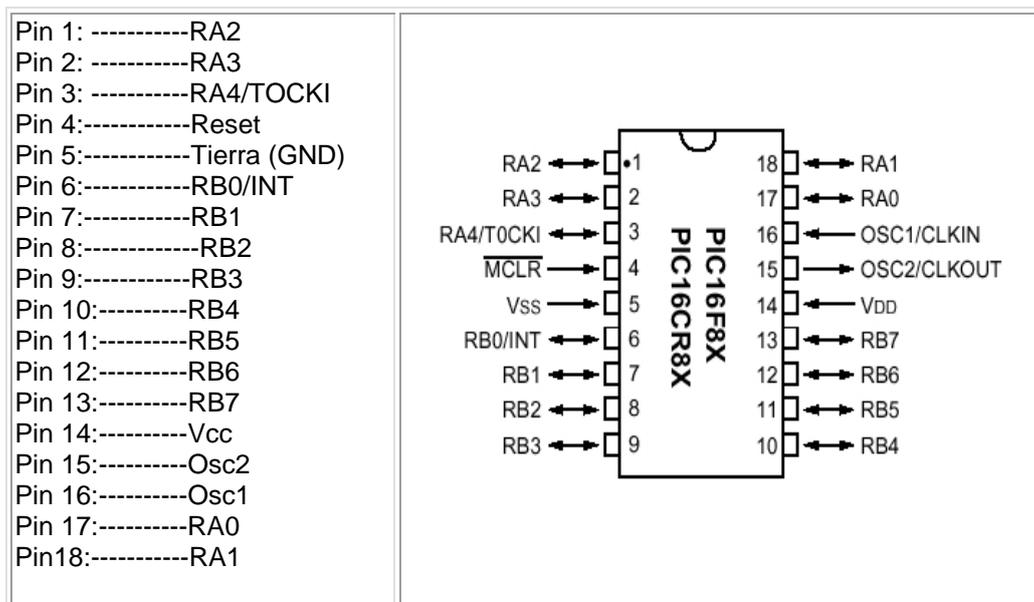


Figura 31. Disposición de pines en PIC 16F84A



El puerto A, el cual tiene sólo cinco pines que puedes configurar como entrada o salida. La pata 3, o sea, RA4/TOCKI puede ser configurado a su vez como entrada/salida o como temporizador/contador. Cuando es salida se comporta como colector abierto, por lo tanto debemos poner una resistencia Pull-up a Vcc de 1 Kohm. Cuando es configurada como entrada, funciona como disparador Schmitt Trigger por lo que puede reconocer señales con un poco de distorsión.

El puerto B, tiene ocho pines que igualmente se pueden configurar como entrada o salida. Los pines 15 y 16 son únicamente para el oscilador externo. El pin 4, o sea, el Reset se debe conectar con una resistencia de 10 Kohm a Vcc para que el PIC funcione, si se desea resetear entonces se puede poner un micropulsador con una resistencia de 100 Ohm a tierra.

El oscilador externo es necesario para que el PIC pueda funcionar, puede ser conectado de cuatro maneras diferentes: XT (oscilador compuesto por un cristal y dos condensadores), RC (oscilador compuesto por una resistencia y un condensador), HS (oscilador compuesto por un cristal de alta velocidad), LP (oscilador compuesto por un cristal de baja frecuencia y bajo consumo de potencia).

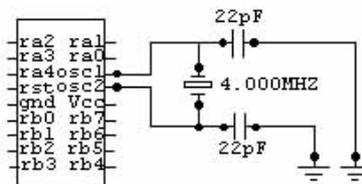


Figura 32. Oscilador XT.

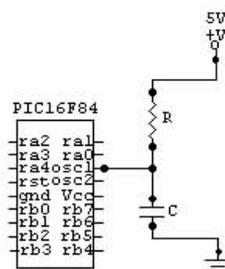


Figura 33. Oscilador RC.

Es necesario conocer donde se alojan los datos dentro del PIC, es decir en su banco de memoria y por ende conocer la tabla de registros. Esta tabla está dividida en dos partes llamadas BANCO 0 y Banco 1. Interesa en estos momentos las opciones de: STATUS, PORTA, PORTB, TRISA y TRISB.



Para que el PIC pueda trabajar se debe configurar sus puertos como entrada o como salida según sea el caso, si se asigna un cero lógico (0) a un pin éste será salida y si se asigna un uno lógico (1) éste será entrada. Esta asignación de pines se hace programando los registros TRISA y TRISB.

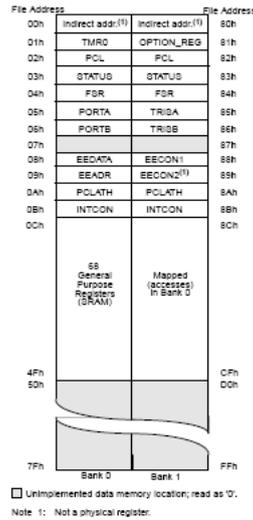


Figura 34. Ubicación de los Bancos, puertos y memoria.

El PIC 16F84A, se programa para que las RAX sea salidas y las RBx sean las entradas de datos (0,1). El reloj de 4[Mhz] la entrega un cristal que oscila a esa frecuencia y con la ayuda de dos condensadores como muestra el circuito.

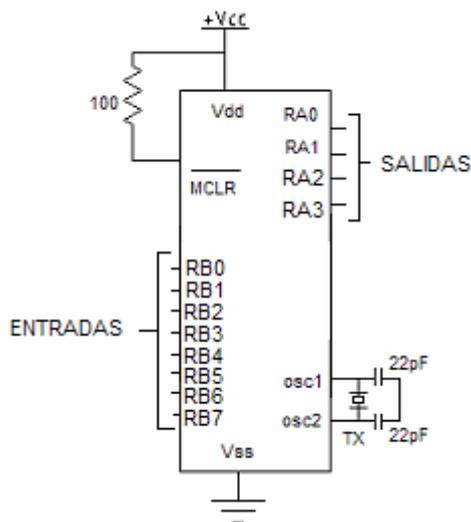


Figura 35. Circuito del PIC.



Este diseño de circuito es muy general ya que al estar libres los puertos, cada uno de ellos se puede configurar como uno desee que trabajen, dependiendo de la aplicación que uno desee obtener.

#### A.- PIC de Desplazamiento:

Las entradas del PIC RBx son utilizadas para tomar datos de control y poder realizar los movimientos necesarios para la defensa y el ataque del robot. Las salidas RAx, están configuradas para alimentar con un valor de activación para un puente H que determina el giro de un motor de c.c., es decir, RA0-RA1 para un motor y RA2-RA3 para el otro motor.

Por medio de los sensores de contacto (interruptores) se evita que choque y quede estancado en un borde o esquina de la pista de combate, es decir, que el robot avanza en línea recta, al chocar este retrocede gira en sentido opuesto a las choques y vuelve a moverse en línea recta.

#### B.- PIC de Torreta:

Para este PIC se tomaron como entrada los sensores IR, los cuales funcionan como un seguidor de luz, en este caso luz IR, y la programación esta basada en seguir la luz IR que viene del Robot enemigo y apuntar constantemente con el cañón que contiene las Balabot, semejante a una torreta de tanque. De esta forma en robot va a estar siempre apuntando al enemigo sin importar en que posición este el chasis de este o en el caso de haberse trabado en alguna esquina o por ultimo que algún motor de movimiento falle y se desplace en círculos.

#### Programa en ASM:

```
estado          equ    0x03
entrada         equ    0x06          ; las entradas son RBx
salida         equ    0x05          ; las salidas son RAx
ret_1          equ    0x0C          ; contador de FF hasta 00
ret_2          equ    0x0D
ret_3          equ    0x0E

                org    0x00
                bsf    estado, 5          ; pasa al banco 1
                movl  w 0xff          ; configura "entrada" como entradas
                movwf entrada
                movl  w 0x00          ; configura "salida" como salidas
                movwf salida
                bcf   estado, 5          ; vuelve al banco 0

ir_in          clr   salida          ; borra salidas
                btfsc entrada, 5
                call  rotacion_der
                btfsc entrada, 6
                call  rotacion_der
                btfsc entrada, 1
```



```
call rotacion_i zq
btfsc entrada, 2
call rotacion_i zq
btfsc entrada, 3
call retardo_1
goto ir_in

rotacion_der    bsf  salida, 0
                bcf  salida, 1
                bcf  salida, 2
                return

rotacion_i zq    bsf  salida, 1
                bcf  salida, 0
                bcf  salida, 2
                return

retardo_1       bsf  salida, 2
                bcf  salida, 0
                bcf  salida, 1
                decfsz    ret_1, 1
                goto  retardo_1
                decfsz    ret_2, 1
                goto  retardo_1
                call  retardo_2
                return

retardo_2       bsf  salida, 2
                bcf  salida, 0
                bcf  salida, 1
                decfsz    ret_1, 1
                goto  retardo_2
                decfsz    ret_2, 1
                goto  retardo_2
                call  retardo_3
                return

retardo_3       bsf  salida, 2
                bcf  salida, 0
                bcf  salida, 1
                decfsz    ret_1, 1
                goto  retardo_3
                decfsz    ret_2, 1
                goto  retardo_3
                return

end
```

Para los 16F84A, se utilizó el programa PIC Simulator IDE para realizar la programación y respectiva simulación del mismo en el PIC seleccionado. Este programa es capaz de mostrar una imagen del PIC y sus respectivos pines y asignación de pines, tiene un compilador en ASM, también posee un osciloscopio para ver la señal digital con la cual se esta trabajando y por ultimo que es muy importante un visualizador de los pasos por los cuales corre el programa sobre las líneas de código.

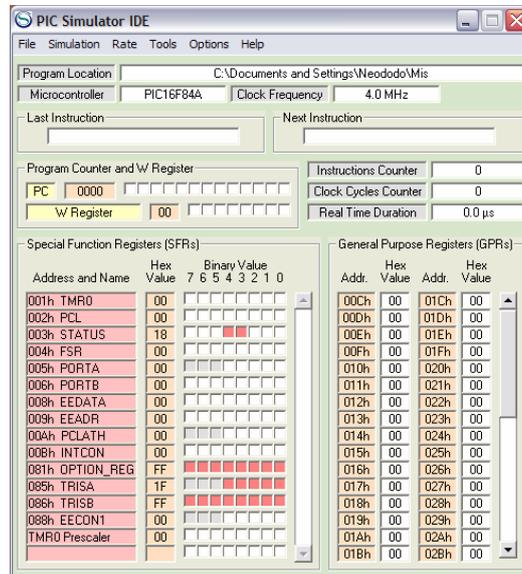


Figura 36. Programa PIC Simulator IDE.

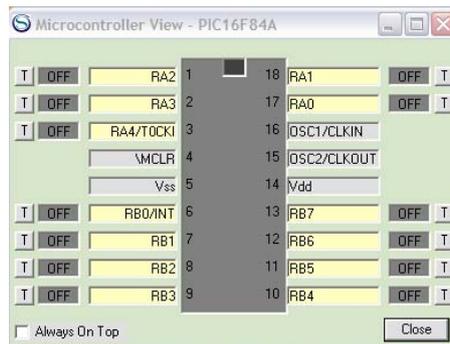


Figura 37. Imagen del Microcontrolador a trabajar.

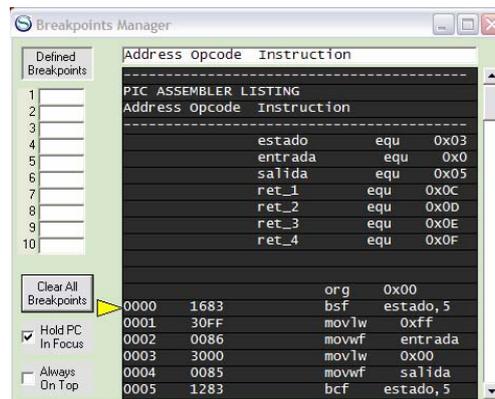


Figura 38. Breakpoint Manager. Para ver los pasos.



```
Assembler - emisor_IR.asm
File Edit Tools Options
estado      equ    0x03
entrada     equ    0x06      ;las ent
salida      equ    0x05      ;las sal
ret_1       equ    0x0C      ;contadc
ret_2       equ    0x0D
ret_3       equ    0x0E
ret_4       equ    0x0F

          org    0x00
          bsf    estado,5      ;pasa al
          movlw 0xff      ;configu
          movwf entrada
          movlw 0x00      ;configu
          movwf salida

-----
0158 0080      end
Number of errors = 0
-----
Lin 165, Col 0
```

Figura 39. Editor de Ensamblador.

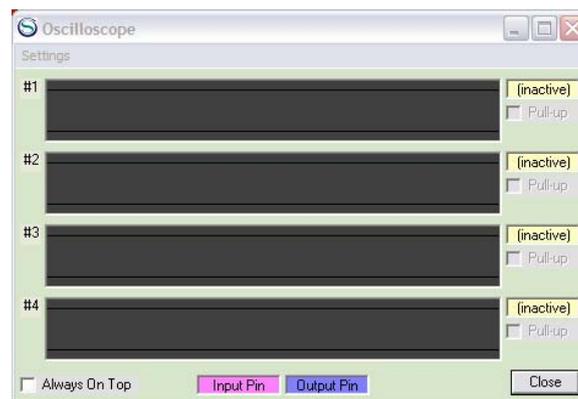


Figura 40. Osciloscopio.

Para el 16F84A la circuitería que lo acompaña es muy sencilla, con solo unos pocos componentes se puede armar el circuito de la figura 35.

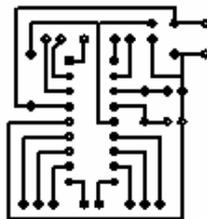


Figura 41. PCB de PIC



Figura 42. PCB del PIC ya armado.



### Polarizacion de los circuitos:

Para obtener una correcta alimentación de todos los circuitos involucrados, y en especial atención a los PIC, se utilizo un pack de cuatro pilas AA en un porta pilas, al que se le agrego un regulador de voltaje del tipo L7805CV el cual regulariza a 5 [V] por el lado positivo de la fuente. Esto es necesario porque sin carga las pilas entregan alrededor de 7.8 [V] y con carga un poco mas de 6[V] y con el regulador se puede estabilizar en los 5[V] que se necesitan para el correcto trabajo del los dispositivos como por ejemplo los PIC, la alimentación es de 5 [V], así como también las compuertas lógicas. Además este dispositivo puede manejar voltajes de entrada  $V_i$  de hasta 35 [V] y una corriente de salida lo de 1.5 [A], pero como el consumo de corriente es bajo, el dispositivo escogido es bueno.

A este esquema básico se le agrego un interruptor para encendido, haciendo más fácil la activación del robot.

También se le proveyó varias salidas de alimentación, de esta forma puede suministrar energía a varios circuitos a la vez. Se eligió el modo de placas independientes, es decir, una placa para el PIC, otra para el puente H, etc. en vez de hace una sola que contenga a todos los circuitos necesarios para que funcione el robot ya que si una de ellas falla, se puede reemplazar con facilidad

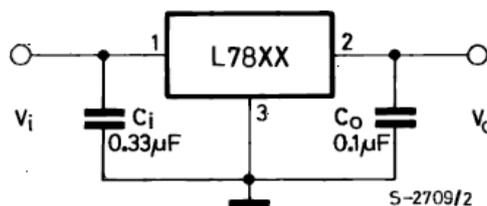


Figura 43. Circuito de regulador de Voltaje.

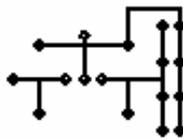


Figura 44. PCB del regulador

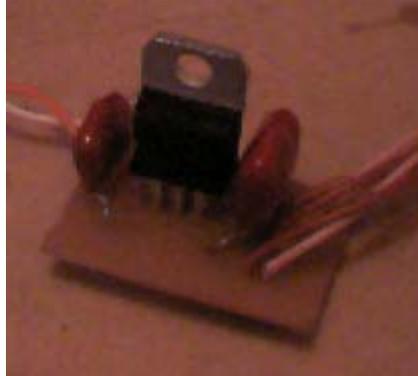


Figura 45. Regulador ya armado.



Figura 46. Regulador con interruptor, salidas y porta-pilas.



### Grabado de los PIC:

Para grabar los PIC 16F84A se utilizo el programador PIC-PG2 ([www.olimex.cl](http://www.olimex.cl)), el cual funciona por medio del puerto serial y se transfiere el programa compilado por medio el programa IC-Prog, el cual toma el programa hecho por el usuario, utilizando la compilación en Hexadecimal (\*.hex) y configurado par que tome al PIC-PG2 como un programador del tipo JDM. Se tienen que seguir las siguientes instrucciones para poder grabar correctamente el PIC:

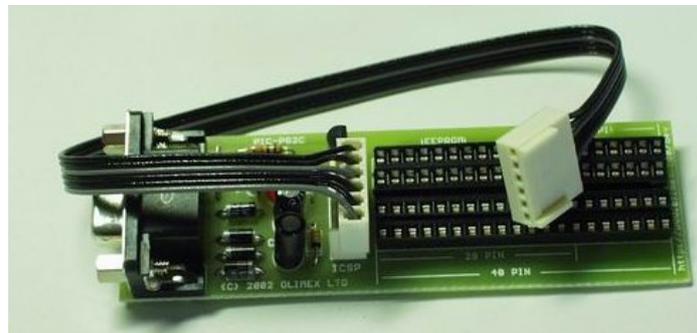


Figura 47. Programador PIC-PG2

#### 1.-Usuarios de Windows XP y NT:

XP y NT no permite acceder a los puertos seriales ni paralelos directamente, se debe instalar un driver especial que permita acceder a los puertos.

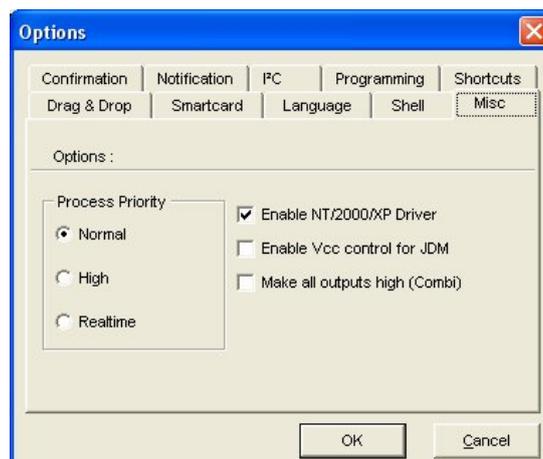


Figura 48. Configuración NT/2000/XP.



2.-Se tiene que descargar el driver de la página de IC-Prog. Poner en el mismo directorio de ICProg.exe. En IC-Prog hacer click en "Settings" y luego en "Options". Seleccionar "MISC" y hacer click en "Enable NT/2000/XP Driver". ICProg se va a reiniciar con el nuevo driver instalado.

3.-Ahora que esta el software instalado, se tiene que empezar a comunicar con el PG2C. Para configurar el ICProg - hacer click en "Setting" y luego seleccionar Hardware.

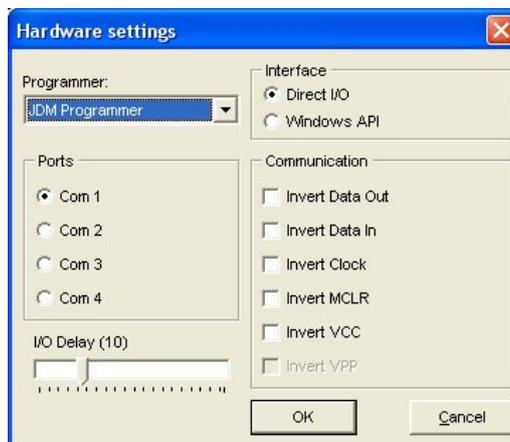


Figura 49. Selección de unidad.

4.-Seleccionar "JDM Programmer" desde el menú. Asegurarse de que no este ninguna de las opciones de invertir seleccionadas. Luego seleccionar "OK" y cerrar la ventana.

5.- El lenguaje hexadecimal.

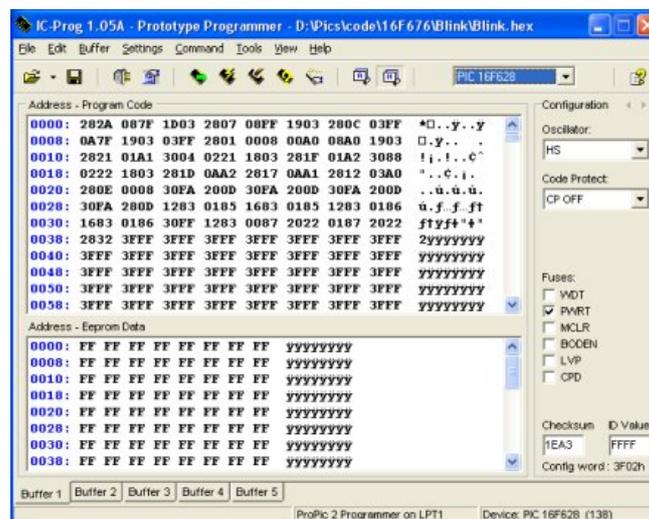


Figura 50. Lenguaje Hexadecimal.



Ahora se puede abrir un archivo HEX para hacer la prueba. El archivo HEX debe tener la configuración del PIC. ICProg reconocerá esa configuración y seleccionara las opciones para el oscilador en forma automática.

6.-Se debe seleccionar el procesador a programar, también se puede hacer de forma automática cuando en HEX se declara a que tipo de procesador se esta realizando el programa. Se debe tener en cuenta que programar el 16f873 es muy diferente de programar el 16F873A.

7.-La ubicación de los diferentes PIC's esta indicada en la misma placa

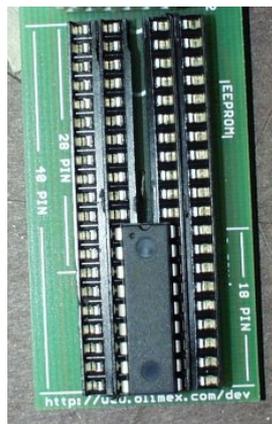


Figura 51. Programador PG2C.

El pin1 para los pic de 28 y 40 pin es la línea blanca entre los 2 sockets en la parte superior de la imagen. El pin 1 de los PIC's de 14 y 8 pin es la segunda línea blanca (ver figura 50). El pin 1 para los PIC's de 18 pines es la línea que viene mas abajo que no se alcanza a ver por que esta cubierta por el pic. Verificar que la ranura que señala el pin 1 en el PIC debe calzar con el indicador entre los sockets. Para PIC's pequeños (16F873, 16F84A, 16F628, etc.) el PIC utiliza los sockets interiores como se muestra en la figura, mientras que un PIC grande como el 16F877 utiliza los socket exteriores.

8.-Instalar correctamente el PIC en el socket correspondiente. Ahora se debe volver al ICProg. Hacer click en el botón que tiene un rayo. El PG2C empezará a programar. El led del PG2C empezará a parpadear. Cuando termine deberá aparece el siguiente mensaje "Successfully verified!".



9.-Si aparece el mensaje "Programming failed at code address 0000h" se debe realizar ciertas comprobaciones del dispositivo de grabación, este es un error genérico que significa algo anda mal. Se debe comprobar lo siguiente:

- El cable serial debe esta correctamente conectado en el comX
- Asegurarse de que esta seleccionado el comX en la opción hardware setup.
- El PIC debe estar orientado correctamente en el socket o el cable ICSP esta sujeto firmemente en la tarjeta de desarrollo que se esté utilizando.
- Asegurarse de seleccionar el PIC correcto. Recuerda que el 16F873 es diferente al 16F873A.



Figura 52. Mensaje de error.



### Chasis:

Se pensó bastante en como seria el chasis del robot. Este paso es muy importante pues sobre él se tiene que ensamblar todos los dispositivos, sensores, motores etc. Y debe de cumplir con las reglas establecidas en las bases en donde solo se limita a la altura que el Cowbot debe tener. La cual es de 40[cm] como máximo, esto se debe a que sobre los 40[cm] se debe colocar la baliza de detección sobre una base firme, y teniendo esto como regencia se procedió a la búsqueda del mejor modelo optimo para que realice el trabajo que se le asigne (los diseños que aparecen los obra de uno de los integrantes del equipo).

#### 1.-Diseño en 3D.

Hubieron varios modelos diseñado por uno del equipo que fueron modelados en 3D, y a partir de ahí se eligió el mas adecuado y el mas factible de hacer con los materiales que se eligieron con posterioridad. El modelo mostrado en la figura 53 se desecho pues a pesar de ser de bajo perfil y un centro de gravedad bajo, ocupaba mucho volumen haciéndolo un blanco fácil, mientras que la figura 54 muestra el modelo mas adecuado con un centro de gravedad elevado y con un volumen menor y por ende el mas indicado para la fabricación.

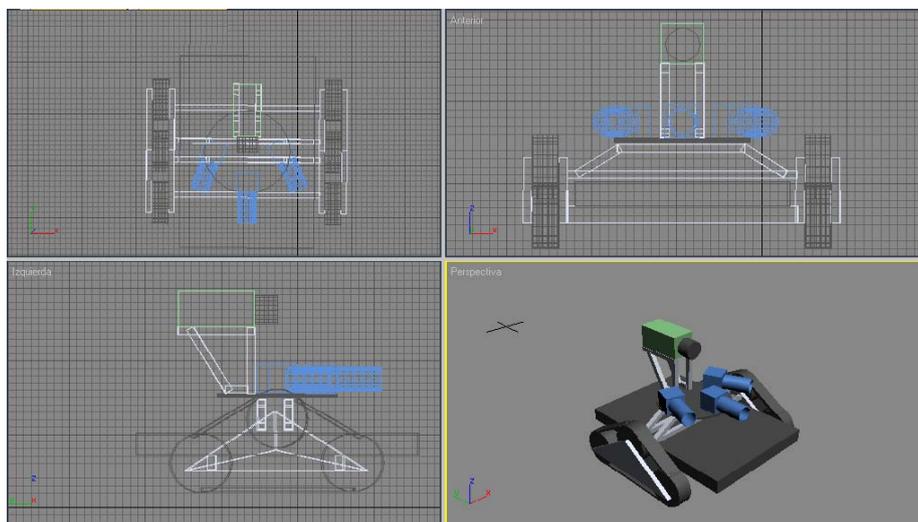


Figura 53. Primer diseño.

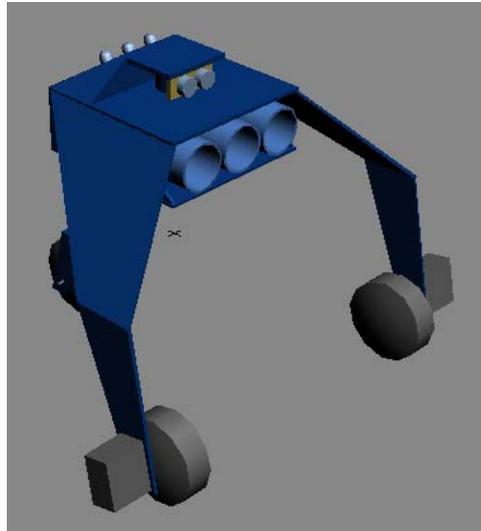


Figura 54. Segundo y final diseño.

## 2.-Chasis final.

El chasis el robot fue diseñado para que ocupe el menor espacio posible en la cancha, es decir que al tener extremidades largas hace que los sensores de los demás robots no reboten en las extremidades al ser éstas estilizadas y delgadas, además que cuando es atacado con las balabots estas pasan por debajo del robot sin tocar el mismo. La única debilidad presente son las ruedas utilizadas que son de gran tamaño y por el sensor táctil colocado en la parte trasera del mismo.

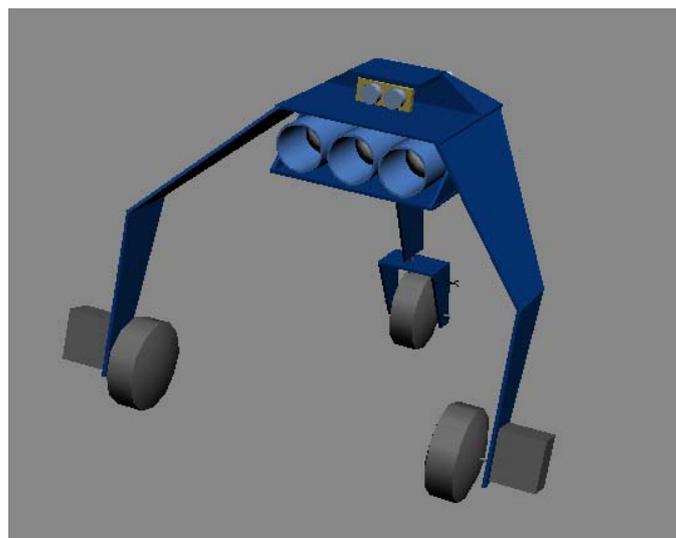


Figura 55. Chasis modelado en 3D.



Figura 56. Chasis ya ensamblado.

### 3.-Materiales usados.

Se pensó bastante en que tipo de material se usaría para la fabricación del chasis, y se llegó a la conclusión que el aluminio era el material mas adecuado, es liviano, maleable, fácil de hacer cambios y resistente. Pero como debía tener una cierta firmeza en las extremidades se opto por hacer un armazón de metal mas duro y pesado, y se uso la lamina de fierro que fue soldada tomando consideración las dimensiones interiores del robot.

### 4.-Pintura.

Otro punto importante a mencionar era la pintura, pues como se utilizaba en su mayoría sensores IR, se opto por pintar el robot con pintura negra opaca (pintura de pizarrón), y de esta forma el color negro absorbería toda luz emitida hacia el, sin reflejar casi nada de luz (incluyendo la IR) y la que es reflejada se desviaba por los ángulos que poseía el robot (a similitud de los aviones Stealth).



Figura 57. Pintado del chasis y vista de polin.



Un punto importante fue que se agregó un polin en la parte superior donde se monta la torreta, esto es debido al peso que esta posee, este polin ayuda a su desplazamiento mas suave y continuo, produciendo una mayor precisión en el blanco (ver figura anterior).



Figura 58. Vista completa de robot con torreta.

#### 5.-Rueda Pivotante.

Un elemento del chasis que acarrió bastantes problemas fue la rueda pivotante trasera, pues esta provocaba la inestabilidad del robot, es decir, que hacia de sus movimientos previamente programados los realizara de forma errática, así también daba dificultad para hacer un giro completo sin que esta rueda se trabase, también cuando el robot se desplazaba en forma recta y dependiendo de la posición de la rueda trasera, este no iba en la dirección deseada, en cambio realizaba una amplia curva ya sea a la derecha o izquierda, dependiendo de cómo estaba ubicada la rueda trasera. La solución a este problema fue el colocarle una rueda mas robusta con el eje pivotante puesto sobre rodamientos que facilitaban el giro rápido de la rueda y por ende el cambio de dirección del robot con bastante precisión, pero hacia que ocupase un volumen mayor al deseado y ser así un blanco mas fácil para los sensores de los demás robots y de las balabots.



Figura 59. Rueda pivotante usada.



### Cañón:

El cañón es esencial para la batalla del los Cowbot, con el se puede atacar al robot contrincante y derrotarlo con una balabot. Existen ciertas reglas respecto a la utilización del cañón y la fuerza con que lanza la balabots. En general se puede utilizar los cañones que uno desee, pero solo pueden ser lanzadas solo un máximo de tres pelotas de ping-pong. La fuerza con esta son lanzadas tiene un máximo de 3[m] en vertical, así mismo no es posible la utilización de energía nuclear, química o explosivos para el lanzamiento de las balabots.

En el robot ensamblado fueron probados varios métodos de lanzamiento de las pelotas, como por ejemplo usando dos motores encontrados y girando en sentido opuesto, así como los lanzadores de pelotas de baseball, y en realidad uno de los competidores uso este método, otro fue el de utilizar aire comprimido y un sistema semejante a las armas de este tipo, pero se necesitaba dedicación y mucho tiempo para realizar este tipo de lanzador y como ese tiempo no se poseía fue desechado, también se pensó en utilizar resorte, solenoides, etc. pero todos tenían cierta dificultad y complejidad en su fabricación, y estaba distando que la filosofía de que "mientras mas sencillo su sistemas mas fácil de armar y reparar".

Es por esto que se opto por la utilización de elásticos como medio de propulsión de las pelotas y un motor servo para la liberación de la traba y así provocar el lanzamiento de la pelota. Este sistema es muy sencillo, pero difícil y engorroso la recarga de las balabots. Pero por el poco tiempo que se poseía fue la mejor y mas optima idea.

Se usaron dos elásticos cruzados amarrados en sus extremos en la boca del cañón y en su centro puestas sobre una cruz de metal donde se extrae un gancho que se fija a una lamina de aluminio que se libera desde el servo y así se lanza la balabot. Dependiendo del largo del elástico es la fuerza que entrega este a la pelota de ping-pong.



Figura 60. Cañones con los elásticos montados.



Figura 61. Cañones cargados.

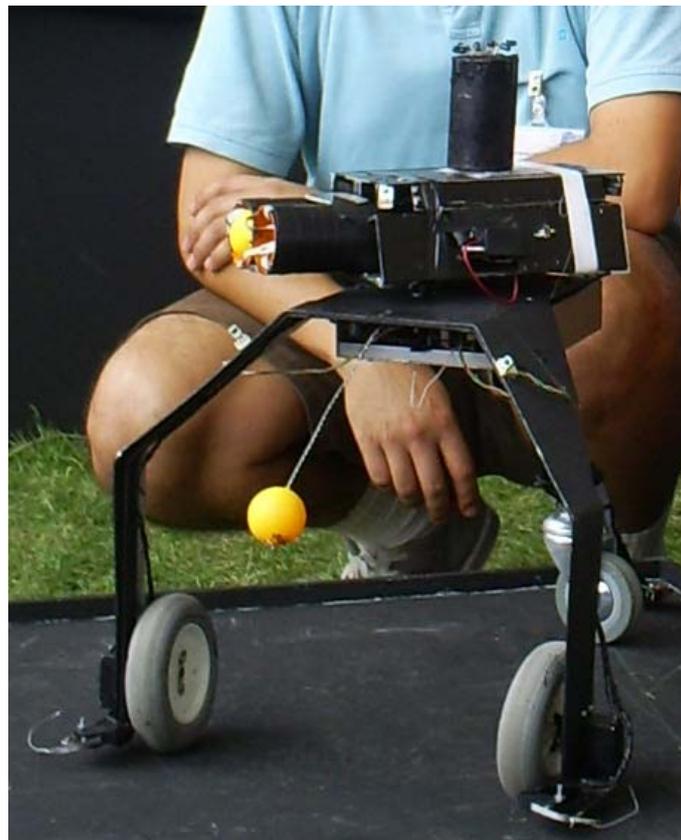


Figura 62. Robot ensamblado final.



Figura 63. Manta-Raya 2 en combate.

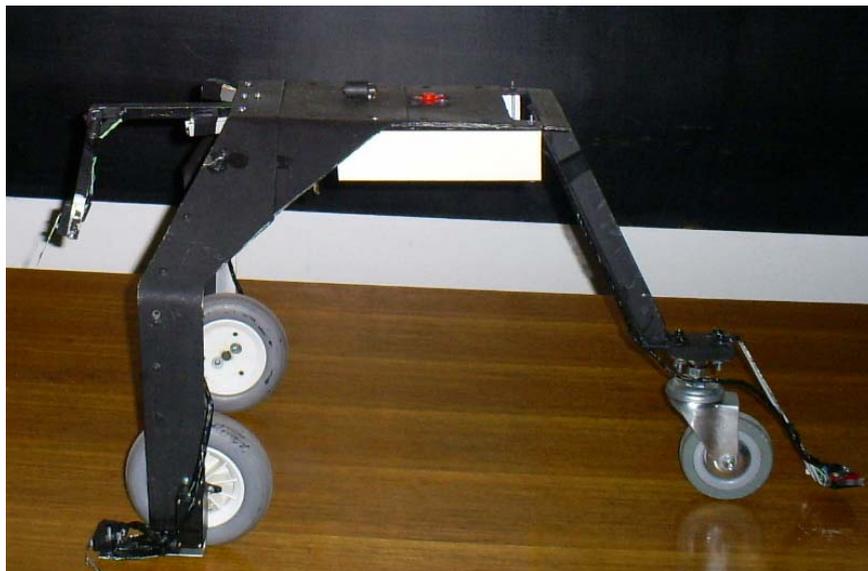


Figura 64. Manta-Raya 2 sin torreta.

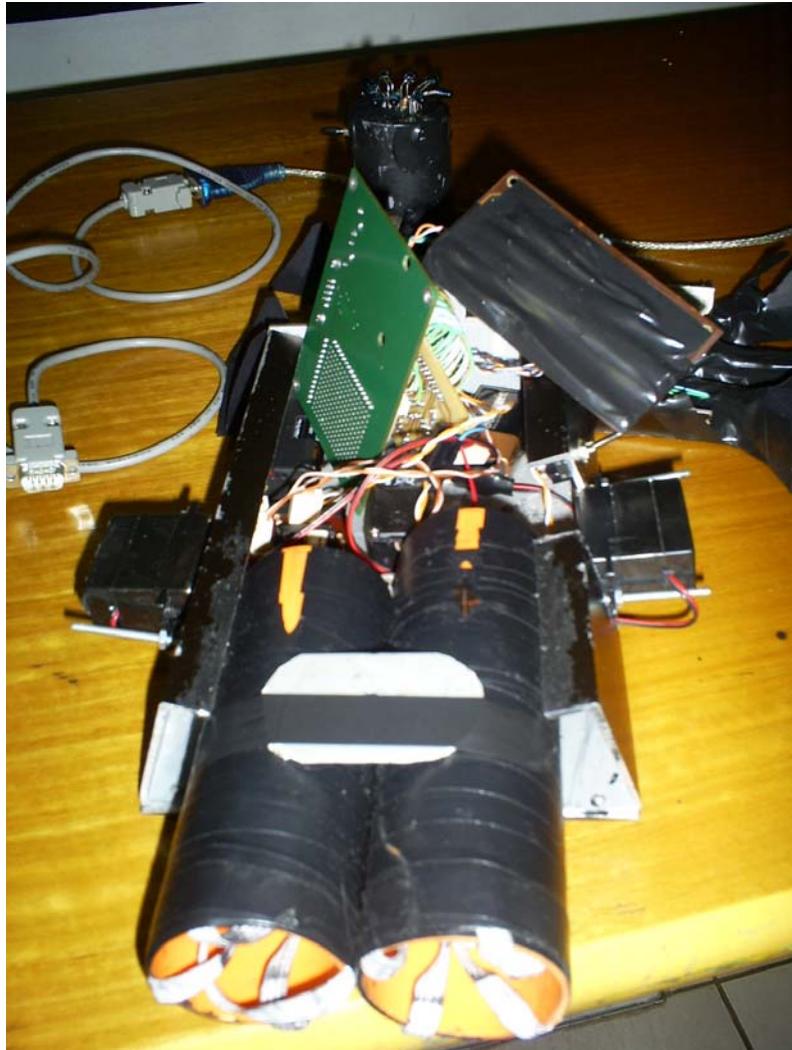


Figura 65. Torreta de Manta-Raya 2.