

UNIVERSIDAD DEL VALLE DE GUTEMALA

FACULTAD DE INGENIERÍA



**“Micro Robot de una pulgada cúbica controlado por
Radio Frecuencia.”**

**Trabajo de graduación presentado por Antonio Rodil Almenara
para optar al grado académico de Licenciado en Ingeniería
Electrónica.**

Guatemala
2009

**“Micro Robot de una pulgada cúbica
controlado por Radio Frecuencia.”**

UNIVERSIDAD DEL VALLE DE GUTEMALA
FACULTAD DE INGENIERÍA

**“Micro Robot de una pulgada cúbica
controlado por Radio Frecuencia.”**

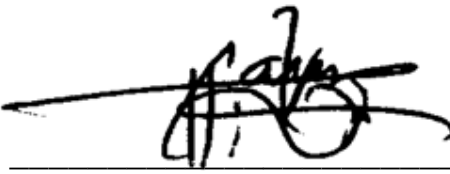
**Trabajo de Graduación presentado por Antonio Rodil Almenara para optar
al grado académico de Licenciado en Ingeniería Electrónica.**

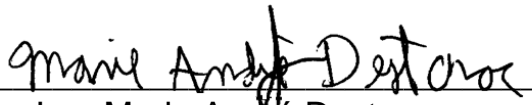
Guatemala
2009

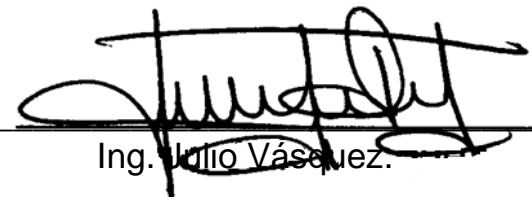
Vo.Bo. :

(f) 
Ing. Carlos Esquit.
Asesor

Tribunal Examinador:

(f) 
Ing. Carlos Esquit.

(f) 
Ing. Maria Andre Destarac.

(f) 
Ing. Julio Vasquez.

20 Noviembre 2009

ÍNDICE

LISTA DE FIGURAS, TABLAS y GRÁFICAS...	viii
RESUMEN	x
I. INTRODUCCIÓN...	1
II. OBJETIVOS	3
III. MARCO TEÓRICO...	4
A. Robots Industriales...	4
1. Robots de paletizado	4
2. Robots de soldadura...	5
3. Robots de carga y descarga	5
4. Robots de Scara...	5
5. Robots de ensamblaje...	6
6. Robots de aplicación de pintura	6
7. Robots de verificación de productos...	6
B. Robots recreativos...	6
C. Robots científicos...	7
1. ALICE2002...	8
a. Locomoción...	8
b. Microcontrolador	9
c. Fuente de alimentación	10
d. Sensores...	10
e. Comunicación...	10
f. Modularidad	11
1. Cámara lineal...	12
2. CMOS Cámara de color...	12
3. Paneles solares...	13
4. Herramientas adicionales externos al robot...	13
D. Comunicación serial en radiofrecuencia...	14
1. Radiofrecuencia...	15
2. Señales digitales..	17
3. Comunicación serial inalámbrica...	18
4. Velocidad de transmisión...	19
a. Bits de datos...	19
b. Bits de parada...	19
c. Bits de paridad...	19
E. Microrrobótica y sus aplicaciones...	20
1. Aplicaciones...	21

VII.	CONCLUSIONES Y RECOMENDACIONES...	58
VIII.	BIBLIOGRAFÍA...	60
IX.	APÉNDICE	61
	A. Diagramas de flujo de los softwares...	61
	1. Diagrama del transmisor...	61
	2. Diagrama del receptor...	66
	3. Diagrama del diseño experimental...	67
	B. Programas...	68

LISTA DE FIGURAS Y TABLAS

Figuras:

1. Robot de Paletizado...	5
2. Robot de Scara...	5
3. Robot de mediana estatura...	6
4. Micro Robot en forma de ratón...	7
5. Micro Robot elaborado por los Laboratorios de Scandia en Alburquerque...	7
6. Motores SWATCH utilizados como minutos y horas...	9
7. Distribución de sensores en ALICE2002. b) Funcionamiento del sensor IR de proximidad...	10
8. Dispositivos RF...	11
9. ALICE2002 con cámara lineal de 102 pixeles...	12
10. ALICE2002 con cámara CMOS de color...	13
11. Estación de carga para 5 ALICE.....	13
12. Alice, micro robot desarrollado por laboratorio de Suiza EPFL...	14
13. Modulador de onda.....	16
14. Frecuencia demodulada y centrada...	17
15. Señales de modulación digital...	17
16. Modulación OOK...	18
17. Diagrama de un sistema de retroalimentación...	22
18. Esquema de un sistema de control remoto...	23
19. Estator y rotor de un motor...	28
20. Diagrama del circuito del control remoto...	34
21. Diseño de PCB del control remoto con componentes...	34
22. Diseño del PCB del control remoto, sólo las trayectorias...	35
23. Diagrama del módulo de transmisión LINX...	35
24. Memoria EEPROM del PIC 16F88...	38
25. Pantalla de PIC Wizard...	40
26. Diagrama del circuito receptor, con Led's para prueba...	43
27. Diagrama del módulo de recepción LINX...	43
28. Diagrama del receptor con el juego de compuertas OR para el diseño experimental...	46
29. Diagrama del cronómetro...	46
30. Diagrama del circuito de interfaz con la computadora...	47
31. PCB del circuito de interfaz con la computadora...	47
32. Pantalla del programa de Visual Basic para diseño experimental...	48
33. Diagrama de bloques de todo el sistema. a) Receptor sin diseño experimental. b) Receptor con el diseño experimental...	49
34. Circuito receptor...	50
35. Diseño del PCB del receptor sin componentes...	50
36. Diseño del PCB del receptor con componentes...	51
37. Adaptador para programación ICSP...	51
38. Programador USB MELABS...	51
39. Adaptador y programador conectados...	52

40.	Esquema del integrado TC4468...	53
41.	Batería Energizer A23...	54
42.	Dimensiones CR2032...	54
43.	Motor GH712 Sin caja de engrane...	55
44.	Motor TPM...	55
45.	Chasis elaborado con un polímero...	56
46.	Otras opciones para chasis...	56

Tablas:

1.	Rangos y clasificación de frecuencias...	15
2.	Especificaciones entre baterías...	25
3.	Especificaciones eléctricas del control remoto...	36
4.	Datos eléctricos de consumo PIC y dispositivo RF...	44
5.	Datos eléctricos de consumo del Driver...	53
6.	Datos eléctricos de batería Energizer A23.....	54
7.	Datos eléctricos CR2032...	54
8.	Datos eléctricos de consumo PIC y dispositivos RF...	56

Gráficas:

1.	Batería A23 Energizer. Descarga vrs. Tiempo...	26
2.	Batería GP27A. Descarga vrs. Tiempo...	26

RESUMEN

Al final del proyecto tendremos un robot de una pulgada cúbica con habilidades para moverse hacia adelante, atrás, izquierda y derecha. Estos movimientos serán transmitidos al robot a través de una comunicación serial inalámbrica, es decir un control remoto.

Este trabajo se desarrolló con la motivación de ir a Nagoya, Japón y participar en el concurso anual de micro robótica. El objetivo del certamen, es pasar un laberinto en menor tiempo, con el robot.

Para realizar esta comunicación, se utilizaron microcontroladores y programación en PIC C Compiler. Se desarrolló un protocolo de comunicación con los parámetros del RS232 para transmitir las señales. También se compraron unos módulos de transmisión y recepción para mandar las instrucciones.

Los pasos que se siguieron son el de investigar cuáles son los microcontroladores apropiados para dicha comunicación. También se tiene que diseñar el circuito transmisor y receptor. En el receptor se tiene que hacer énfasis en los componentes a elegir, pues puede ser más grande que pulgada cuadrada.

Al final se verá que no fue posible la construcción completa de dicho micro robot, debido a las dificultades que se presentaron con la adquisición de partes. Sin embargo, se deja un historial bastante sólido para que futuras generaciones puedan tomar este trabajo de tesis y guiarse para construir algo mejor.

En diversas partes del mundo hay laboratorios desarrollando tecnología en micro robótica. Instituto Federal de Tecnología en Lausana, Suiza, con ALICE 2002. Laboratorio de Sandia en Albuquerque, Estados Unidos. Aquí se desarrolló un micro robot de 1.6cm cúbicos. MIT también tiene su aporte en esta rama con su granja de micro robots. Y varios más. Actualmente en Centro América no hay información de proyectos en micro robótica.

I. INTRODUCCIÓN

El objetivo principal de este trabajo, es la construcción de un micro robot de una pulgada cúbica y su sistema de control por medio de radio frecuencia. La idea fue planteada por uno de los asesores que estudió una maestría en Nagoya, Japón. Todos los años en esta ciudad se hace un concurso de micro robótica, que consiste en atravesar un laberinto con dicho robot.

Parte de la motivación de este trabajo, fue ir a Japón a participar en dicho concurso y aprender un poco sobre la cultura científica de dicho país. No fue uno de los objetivos, pero sirvió para tomar una idea y que otros estudiantes tomen la iniciativa, participen y puedan desempeñarse a nivel internacional en este campo de los microrobots.

El trabajo comenzó con una investigación en internet del concepto de un micro robot. También se buscaron las aplicaciones que esta rama de la robótica tiene, así como los componentes que se utilizarían para dicha construcción.

Para la elaboración de este proyecto se plantearon ciertos parámetros de construcción del sistema de control y la estructura del micro robot. Parámetros como: qué componentes usar, cómo diseñarlo, funcionalidad, tipo de control remoto y otros.

Uno de los objetivos y retos fue el de integrar un sistema con pocos componentes para la transmisión de datos hacia el micro robot. Éste consiste en un control remoto que solamente transmite y que es capaz de grabar el recorrido en sentido inverso, de tal forma que el micro robot, al comando de una instrucción, pueda regresar a su punto de partida.

Lo más complejo de este proyecto, fue escoger los componentes adecuados para que sea un robot de una pulgada cúbica. También tiene relevancia la estructura que éste debe tener para albergar todos los elementos que lo conformarán: micro controlador, resistencias, módulo de recepción, y sobre todo el tipo de motores que se utilizarán de tal manera que tenga el torque y potencia suficiente para moverse de manera libre y correcta. Además es de suma importancia la fuente de alimentación que éste tiene que llevar. Toda esta integración es lo que hizo de este trabajo, un verdadero reto.

Al final se verá que no fue posible la construcción completa de dicho micro robot. Sin embargo, se deja un historial bastante sólido para que futuras

generaciones puedan tomar este trabajo de tesis y guiarse para construir algo mejor, sin tropezar por los mismos obstáculos que se expondrán en el cuerpo de la investigación.

En diversas partes del mundo hay laboratorios desarrollando tecnología en micro robótica. Un ejemplo es el Instituto Federal de Tecnología en Lausana, Suiza, con ALICE 2002. Éste es un micro robot modular con varias aplicaciones según configuración. Puede tomar fotos, comunicarse bidireccionalmente, entre otras. Tiene 8 pulgadas cúbicas y utiliza unos motores swatch que ahorra energía y proveen del torque necesario.

Otro laboratorio es Sandia en Albuquerque, Estados Unidos. Aquí se desarrolló un micro robot de 1.6cm cúbicos. Posee sensores de temperatura, dispositivos de comunicación, sensor químico y otras funciones. Éste planea ser un micro robot de reconocimiento.

MIT también tiene su aporte en esta rama con su granja de micro robots. Estos están equipados con 17 sensores de diferentes tipos. Actualmente los robot pueden comunicarse entre ellos con sensores IR. Se hacen equipos y juegan captura a la bandera. Pueden ser utilizados para la exploración a Marte ya que contiene sensores para mapeo de terreno.

Actualmente en Centro América, no se encontró información de proyectos en micro robótica. Pero en México en la universidad de Anáhuac Mayab, Mérida, un estudiante diseñó un micro robot controlado por imágenes. Es básicamente un imán de unos 400micrómetros de espesor cuyo movimiento es controlado por campos magnéticos. Al momento sólo han logrado que dicho robot se mueva en un eje. Algunas de las aplicaciones que éste tendría sería el procesamiento de señales bioeléctricas con aplicaciones médicas. Sin embargo, en Guatemala es un campo poco conocido.

Hay muchos más proyectos de desarrollo por diversas partes del mundo. Todos con el propósito de optimizar tiempo y recursos. La tendencia de esto es a la nanotecnología. Se pretende desarrollar nano robots que puedan servir en el campo quirúrgico de tal manera que estos procedimientos disminuyan en riesgo. Tal es el caso del micro robot de 250 micrómetros de ancho de la universidad de Monash en Australia.

II. OBJETIVOS

A. Generales:

- Elaborar un diseño para fabricar un micro-robot.
- Implementar un sistema de control inalámbrico para un micro robot, con pocos componentes.

B. Específicos:

- Construir un micro robot de una pulgada cúbica.
- Diseñar y desarrollar un control remoto con una comunicación serial por medio de módulos de radio frecuencia.
- Lograr que el control remoto tenga capacidad para grabar un recorrido corto y lo ejecute en sentido contrario.

III. MARCO TEÓRICO

El área de la robótica es muy amplia. Hay diferentes usos para dichas máquinas según el campo de aplicación y complejidad.

Se define entonces qué es un robot: es una máquina equipada con actuadores, sensores, microcontroladores y una armazón. Éste tiene como finalidad desarrollar actividades que normalmente hacen los humanos, pero con tiempo muy prolongado, o tareas que no puede realizar una persona por la dificultad que representa.

Con la definición anterior se pueden separar los robots según su finalidad:

- Industrial.
- Recreativa.
- Científica.

A. Robots industriales.

Los robots industriales son máquinas empleadas por el humano para realizar tareas de producción y construcción. Generalmente éstas tienen forma de brazo y hay varias clasificaciones:

- Robots de paletizado.
- Robots de soldadura.
- Robots de carga y descarga.
- Robots de Scara.
- Robots de ensamblaje.
- Robots de aplicación de pintura.
- Robots de verificación de productos.

1. Robots de paletizado. Estos robots sirven para ubicar productos, materiales y objetos en general, sobre una tarima.

Algunos de estos tienen forma de cintas de transporte y numerosas carretillas que permiten ubicar las tarimas en diferentes alturas. Se encuentran más que todo en lugares de almacenaje (Figura 1).

Figura 1: Robot de paletizado^[6].



2. Robots de soldadura. Los robots de soldadura, se emplean masivamente en la industria de ensamblajes de autos, aviones y todo lo que requiera una unión entre dos metales. Su diseño es en forma de brazo, parecido a los de paletizado pero su “mano” tiene unos electrodos especiales que proporcionan la corriente necesaria para soldar un metal.

3. Robots de carga y descarga. Estos robots se componen normalmente de dos ejes (X e Y), y como el nombre lo indica, sirve para desplazar un objeto de un punto A hacia un punto B. Sirve también para trasladar una pieza de una parte del proceso a otra etapa del mismo.

4. Robots de scara. Robots de Scara son aquellos que tienen toda la libertad de movimiento en el eje X y Y, pero muy limitados en el eje Z. El movimiento en Y es muy simple, hacen pequeñas presiones siempre sobre un mismo plano.

Estos robots son utilizados en la industria para manufactura de componentes electrónicos de uso masivo (Figura 2).

Figura 2: Robot de Scara^[7].



5. Robots de ensamblaje. El mismo nombre lo indica. Sirven para armar carros, electrodomésticos, etc. Cualquier objeto que se produzca en masa, y se componga de varias piezas.

6. Robots de aplicación de pintura. Básicamente tiene forma de brazo humano y en la parte final del brazo, se encuentra una herramienta para pintar, ya sea paredes, carros, tela, etc.

7. Robots de verificación de productos. Estos robots pueden tener en la parte que simula la mano, una cámara o un sensor infrarrojo.

La cámara sirve para tomar la imagen y por medio de un histograma analizar el objeto o dibujo que verifica el producto. Los sensores infrarrojos son para leer un código de barras, que debe de ser único para los productos.

B. Robots recreativos.

Los robots recreativos también pueden llegar a ser de tipo científico, ya que se empieza como un robot juguete cuando en realidad es la plataforma para un experimento.

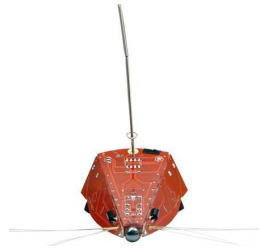
De estos hay de diversos tamaños: grandes, medianos, pequeños y micros; y todos tienen la finalidad de recrear (Figuras 3 y 4).

Los robots grandes, normalmente son andróides o con forma de animales.

Figura 3: Robot de mediana estatura^[12].



Figura 4: Micro Robot en forma de Ratón^[13].



C. Robots científicos.

Dentro de esta categoría se podrían incluir los robots médicos, de exploración, de medición y los androides.

Al igual que en la categoría anterior hay de diferentes tamaños. Se delimitará el tema a micro robots. Varios laboratorios en el mundo se dedican a la investigación sobre micro robótica con el fin de buscar nuevas formas de realizar tareas científicas, como la recolección de muestras en un lugar estrecho y contaminado.

El laboratorio de “Sandia” en Albuquerque, Estados Unidos (Figura 5); realizó en septiembre de 2002, lo que podría ser el robot más pequeño del mundo. Las dimensiones son de un cuarto de pulgada (1.6cm cúbicos) con un peso de 28 gramos. Contiene un procesador de 8Kbytes de ROM, sensor de temperatura y dos motores. Se planea en un futuro colocarle una cámara miniatura para tomar imágenes, micrófono, dispositivos de comunicación y un sensor químico. La aplicación que éste tendría es el de poder penetrar en tuberías para buscar químicos peligrosos o movimiento humano. Algunos de los desarrolladores de este proyecto son Ed Heller y Doug Adkins entre otros muchos.

Figura 5: Micro Robot elaborado por los laboratorios Sandia en Albuquerque^[1].



Otro caso es el de un micro robot desarrollado por Siegwart, Pilles Caprari del Instituto Federal de Tecnología en Lausana, Suiza. Ellos construyeron una flotilla de micro robots que los llamaron ALICE.

1. **ALICE2002.** La idea de ALICE comenzó en 1998 y cada año han ido perfeccionando algún detalle haciendo diferentes versiones.

Las características de este robot son:

- 8 centímetros cúbicos.
- 2 motores SWATCH.
- Dos ruedas.
- 3 sensores infrarrojos, dos al frente y uno atrás.
- Modular.
- Reprogramable.
- Capacidad para durar entre 15 y 20 horas.

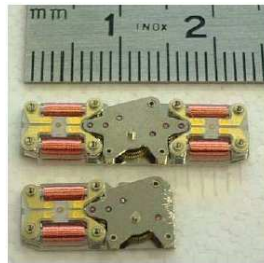
A continuación se hablará de la locomoción, microcontroladores, sensores, fuente de alimentación y dispositivos de comunicación así como su modularidad de ALICE2002 por separado.

a. Locomoción. La movilidad del micro robot es de mucha importancia para poder diseñar una estructura adecuada para ensamblar todos los componentes. Para esto es imprescindible escoger cuidadosamente los motores.

En el caso de ALICE2002, se escogieron 2 motores SWATCH, como actuadores. Un actuador es cualquier elemento que pueda provocar un efecto determinado a un proceso.

Los motores SWATCH, a diferencia de otros, son de muy bajo consumo, 2mA. Proporcionan el torque suficiente y son de un tamaño aproximado de 2.3cm de largo (Figura 6).

Figura 6: Motores SWATCH utilizados como minutos y horas^[2].



Normalmente se usan para relojes y se mueven en dirección a las agujas del reloj. Consta de 6 pasos por cada revolución, controlado por 3 pines del microcontrolador. Sin embargo existen también bidireccionales, que son utilizados normalmente en los tacómetros de los carros y velocímetros.

Estos motores tienen la desventaja que no se encuentran comercialmente disponibles sino que hay que comprar todo el paquete en donde se encuentran. Los SWATCH de ALICE son del reloj “T-touch” vendido por TISSOT.

b. Microcontrolador. El microcontrolador utilizado por ALICE2002, es el PIC 16F877 con el empaquetado especial PT44 de 12x12x1 mm.

Este microcontrolador tiene varias funciones y periféricos que son adecuados para este tipo de trabajo (ADC, Timers, USART, PWM, etc).

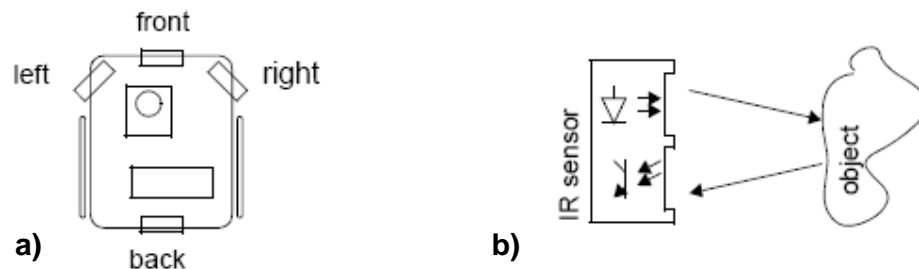
Para una mayor información de este PIC es necesario bajar el manual de especificaciones que se encuentra en Internet. Una página adecuada sería www.microchip.com.

c. Fuente de Alimentación. Al inicio se utilizaron las baterías tipo moneda, pero estos tienen problemas con la corriente máxima que pueden proporcionar. Por lo que se optó por utilizar acumuladores. También se utilizó un regulador para estabilizar el flujo de alimentación. Sin embargo hay que tener cuidado pues cuando el sistema requiera de más corriente, el voltaje puede bajar produciendo malfuncionamiento en los componentes.

d. Sensores. Para que un robot tenga una utilidad de trabajo, es necesario que tenga sensores de cualquier tipo: temperatura, tacto, proximidad, etc.

ALICE2002 está equipado con 4 sensores infrarrojos de proximidad. Tres al frente y uno atrás.

Figura 7: a) Distribución de sensores en ALICE2002. b) Funcionamiento del sensor IR de proximidad^[2].



Estos sensores son de fácil uso y de bajo costo. La marca utilizada por ALICE2002 es SFH9201 de Infineon, por su disponibilidad y tamaño compacto (Figura 7a). Lo que hace que este robot tenga la capacidad de localizar obstáculos a 4cm y comunicación local entre robots a una distancia de 6cm.

e. Comunicación. ALICE2002 tiene tres tipos de comunicación principalmente dependiendo del módulo que se le adjunte.

Para comunicación local entre robots se utilizan los sensores IR mencionados anteriormente. Incluye también un módulo de comunicación IR para distancias de 10m de tal manera que pueda recibir instrucciones a través

de un control remoto IR. El componente utilizado es el TSOP18xxSS3V de Vishay (Figura 8).

Tiene la opción de una comunicación bidireccional por medio de dispositivos de radiofrecuencia. Los integrados utilizados son los HX1000 como transmisor y el RX1020 como receptor, ambos a 433.92 MHz, elaborados por RF Monolithics, Inc.

Esta comunicación RF da la facilidad del robot de comunicarse con otro robot e intercambiar datos, o con un computador. La distancia a la que fue probada fue de 10 metros a 1 kBaud de velocidad.

Estos chips tienen una dimensión de 5.99 x 3.99 x 2.29 mm.

Figura 8: Dispositivos RF.



f. Modularidad. Un importante aspecto a tomar en cuenta para la construcción de un micro robot, es la versatilidad que este puede tener. De aquí la importancia de la modularidad, pues por el tamaño, es más fácil equipar un micro robot con ciertos sensores y otro con otro tipo de sensores.

ALICE2002 tiene varios módulos:

- Comunicación IR unidireccional.
- Comunicación bidireccional RF.
- Sensores de tacto.
- Cámara lineal.
- CMOS cámara de color.
- Paneles solares.
- Herramientas adicionales externos al robot.

Se especificará a partir de la cámara lineal, pues los tres primeros fueron vistos en las secciones anteriores.

1. Cámara lineal. ALICE2002 puede llevar montado en su parte superior, una cámara lineal. Para esto se utiliza el sensor TSL2301 manufacturado por TAOS. Esta cámara es un poco más pequeña que todo el robot en sí, y se puede leer directamente por medio de un procesador a través de un protocolo serial (Figura 9).

El chip tiene 102x1 píxeles y un ADC de 8 bits. Se alimenta con 3V y un refrescamiento de la imagen a 50Hz, con un consumo de 2.2mA.

Esto lo hace capaz de detectar objetos a más distancia. Sin embargo estos se limitan a puntos brillantes, líneas verticales blancas y negras, así como el ancho de dichas franjas.

Figura 9: ALICE2002 con cámara lineal de 102 píxeles^[2].



2. CMOS cámara de color. Este tipo de cámaras maneja un número mayor de información que las anteriores, por que le hace difícil el trabajo al microcontrolador que ALICE2002 tiene incorporado. Una solución podría ser leer ciertas líneas de la imagen, mandarlas a un computador, y luego leer las otras líneas.

En la Figura 10 se puede observar que ya está instalada pero todavía no está del todo operable por la complejidad.

Cuando se logre dar todo el funcionamiento la cámara, hará que ALICE2002 mejore su rendimiento.

Figura 10: ALICE2002 con cámara CMOS de color^[2].



3. Paneles solares. Los paneles solares fueron utilizados más que todo por robots ALICE anteriores al ALICE2002. El panel utilizado fue un Panasonic BP-2422 con dimensiones de 24 x 22mm. Tiene un voltaje nominal de 2.1V y una corriente máxima de 6.6mA. La batería recargable es una Panasonic VL-2020.

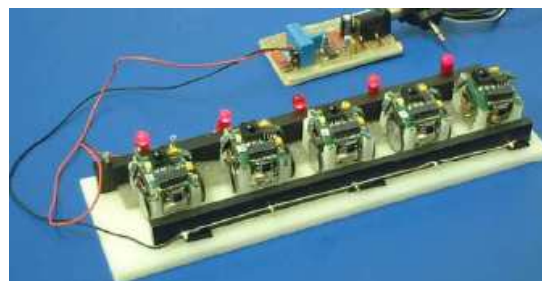
Este módulo funciona bien, pero tiene un inconveniente. Al haber mucha luz, el panel provee suficiente energía para mover al micro robot, pero los sensores IR se quedan sin poder transmitir porque al haber mucha luz, se crea una interferencia. Esto es de fácil solución si se controla la manera de cargar la batería y de usar los sensores, o simplemente prescindir de ellos.

4. Herramientas adicionales externos al robot. Estas herramientas son necesarias para complementar el funcionamiento de ALICE2002. Estos son: programador, recargador, “debugger” y control remoto.

El programador es útil para poder re-programar el PIC del micro robot. ALICE también cuenta con un cargador para recargar a 5 robots al mismo tiempo (Figura 11).

Luego el control remoto maneja una comunicación serial RS232 con sensores IR y un microcontrolador 16F84.

Figura 11: Estación de carga para 5 ALICE^[2]

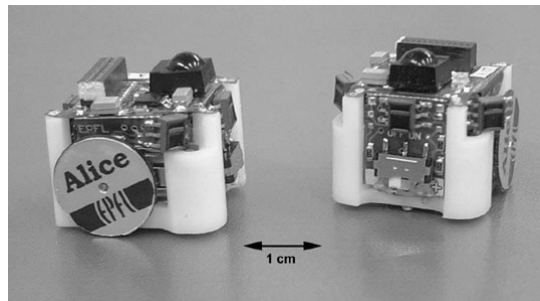


ALICE2002 se ha desenvuelto en:

- La exposición de Suiza en el 2002.
- RobOnWeb.
- Navegación un construcción de mapas.
- Mini soccer game.
- Proyectos de estudios sobre hormigas.

Éstas y otras muchas más aplicaciones pueden tener estos increíbles robots. Este laboratorio ha desarrollado muchos más micro robots.

Figura 12: Alice, micro robot desarrollado por laboratorio de Suiza EPFL^[2].



ALICE2002 es un buen ejemplo para desarrollar un micro robot, y tiene todo lo necesario para empezar a investigar las aplicaciones que pueda tener en Guatemala.

D. Comunicación serial en radiofrecuencia.

Para entender un poco cómo se comunican el transmisor y el receptor del micro robot, se debe entender lo que sucede cuando estamos enviando una señal a través del aire.

Se hablará sobre la radiofrecuencia y los tipos de modulaciones que hay. Luego, se explicará la modulación de señales digitales y análogas. Finalmente, comunicación serial inalámbrica.

1. Radiofrecuencia. La radio frecuencia, es el espectro electromagnético con el que se pueden generar ondas aplicando corriente alterna, para luego transformarlo en una señal útil que puede ser usado para transmitir información.¹

Este espectro se divide en campos de frecuencias según la longitud de onda y frecuencia, según se puede observar en Tabla 1.

Tabla 1: Rangos y clasificación de frecuencias.

Extra Low Frequency	ELF	3Hz - 30Hz
Super Low Frequency	SLF	30Hz - 300Hz
Ultra Low Frequency	ULF	300Hz - 3kHz
Very Low Frequency	VLF	3kHz - 30kHz
Low Frequency	LF	30kHz - 300kHz
Middle Frequency	MF	300kHz - 3MHz
High Frequency	HF	3MHz - 30MHz
Very High Frequency	VHF	30MHz - 300MHz
Ultra High Frequency	UHF	300MHz - 3GHz
Super High Frequency	SHF	3GHz - 30GHz
Extra High Frequency	EHF	30GHz - 300GHz

El espectro UHF tiene un rango de 300MHz a 3GHZ y tiene una longitud de onda de 1 metro a 100 milímetros.

Hay una entidad internacional que se encarga de asignar los usos de estos rangos de frecuencia para la comunicación de información que es la “Federal Communications Commission” (FCC).

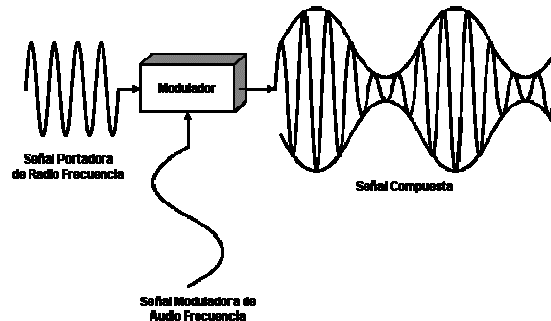
Dentro de las radiofrecuencias UHF se encuentra un pequeño rango asignado al uso público, sin necesidad de pagar alguna cuota o comprar el espectro, ni de preocuparse por hacerle algún tipo de interferencia. Estas frecuencias son las utilizadas para los controles remotos de portones, carros a control, etc. Estos dispositivos son, por lo general, de baja potencia. Como se mencionó al principio, las ondas de radio se pueden transformar para enviar mensajes a través del aire. La manera de hacer esto es modulando la señal. La definición de modulación que aparece en el libro de Stremler del año 2005 es:

<< El proceso por el cual una propiedad o un parámetro de una señal, se varía proporcionalmente a una segunda señal.>>²

¹ Marcus, Abraham; Marcus, Williams. 1965. *Elements of Radio*. Pág. 49.

Es decir, el mensaje que se está enviando se convertirá en una frecuencia que será insertada en una onda portadora para la transmisión.

Figura 13: Modulador de onda^[10].



Entra el mensaje en la señal portadora y luego se modula con otra señal para transmitirla por un canal determinado. La Figura 13 muestra la modulación AM.

Hay varios tipos de modulación:

- AM (modulación en amplitud).
- FM (modulación en frecuencia).
- PM (modulación de pulso).

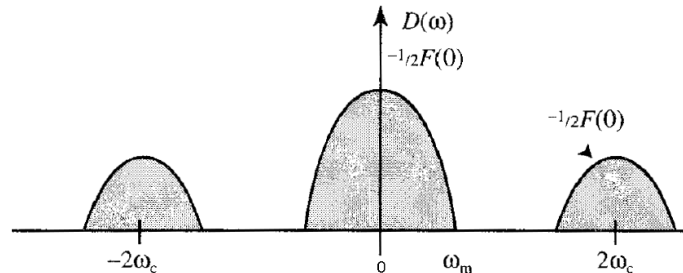
Modulación en amplitud: es la forma de transmitir una onda portadora por medio de la variación de la amplitud. Para trabajar con este tipo de modulación hay varias ecuaciones matemáticas para calcular que frecuencia se debe utilizar para determinada portadora y cuál sería la frecuencia intermedia necesaria para demodular la señal. Pero no es nuestro objetivo dar a conocer esas fórmulas, sólo la base teórica – práctica. Si se desea mayor detalle se puede acudir a las referencias.

Después de transmitida la señal por medio de un transmisor, la señal debe de ser recibida por un receptor la cual tendrá que interpretarla y obtener la información que esta trae.

Ese proceso de interpretación es la demodulación de la señal. Para dicho proceso es necesario trasladar el espectro de la señal a su posición original. Dicho de otro modo, la señal que sale del modulador, está centrada en la frecuencia de la portadora y se utiliza un filtro pasa banda para eliminar

las colas no deseadas. Al momento de entrar en el demodulador, se desea que el espectro se centre en $\omega = 0$ y se utiliza un filtro pasa bajo para obtener el mensaje (Figura 14).

Figura 14: Frecuencia demodulada y centrada^[10].

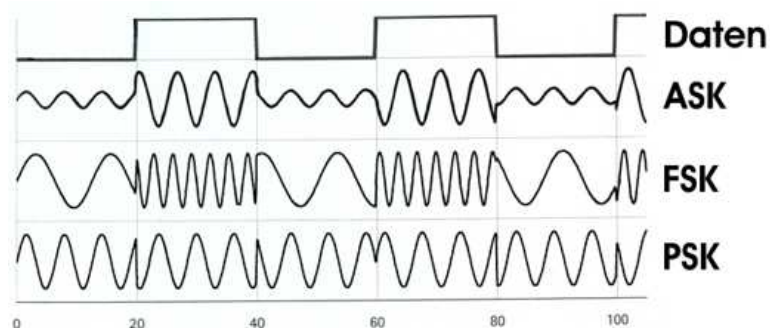


Las curvas representan la señal portadora ya demodulada. El símbolo ω_m es la frecuencia que delimita el filtro, o ancho de banda. La curva que está en el centro, es la que se desea obtener después del filtro.

Luego de hacer esto todavía hay que pasarla por un amplificador para poder interpretar bien el mensaje, además por las características del demodulador, éste siempre va reduciendo la señal a la mitad.

2. Señales digitales. Una señal digital se compone de valores discretos y no de variables continuas como las analógicas. Solamente asume dos estados en su lógica: 1 ó 0, claro que en la práctica estos dos valores deben asumir un valor en voltaje que puede variar dependiendo de la aplicación (Figura 15).

Figura 15: Señales de modulación digital^[10].



² Streamler, Ferrel. 1989. *Sistemas de Comunicación*. 1ª ed. Español. México. Fondo Educativo Interamericano, S.A. Pág. 205.

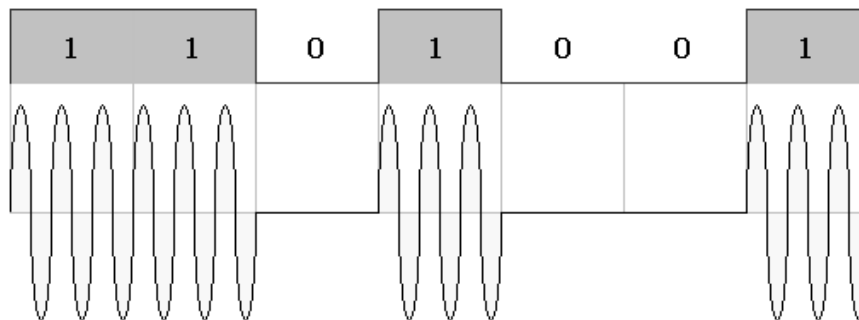
Esta figura muestra cómo es que se transmiten señales digitales, por medio de modulaciones digitales. Estas modulaciones son:

- FSK (Frequency-Shift Keying).
- ASK (Amplitude-Shift Keying).
- PSK (Phase-Shift Keying).
- OOK (On-Off Keying).

Y hay otras más, pero éstas son las más comunes.

En la lista mencionada, hay dos tipos de modulación que son muy parecidos: ASK y OOK. ASK es una modulación en amplitud que al momento de mandar un estado lógico de 1, se modula a una amplitud mayor que cuando se transmite un cero. Mientras que en OOK se transmite un cero, para un nivel lógico de cero; no hay amplitud (Figura 16).

Figura 16: Modulación OOK^[10].



Para la demodulación de estas señales, es necesario tener un tiempo de muestreo que sea dos veces mayor al tiempo de transmisión, según el teorema de Nyquist.

3. Comunicación serial inalámbrica. Una comunicación serial se refiere a la transmisión de datos digitales utilizando únicamente tres líneas: receptor, transmisor y tierra. Los datos se reciben bit por bit a diferencia de la comunicación paralela que se recibe byte por byte.

Este tipo de comunicación es un poco más lenta que la paralela pero su alcance es mayor. Si se utiliza un cable, el largo del paralelo no puede ser mayor a 20 metros, mientras que en el serial puede llegar a los 1,200 metros.

Hay ciertas características que hay que tener en cuenta al momento de configurar la comunicación serial. Estas características son:

- Velocidad de transmisión.
- Los bits de datos.
- Bits de parada.
- Bits de paridad.

Es necesario que, tanto el transmisor como el receptor tengan la misma configuración.

Ahora describiremos un poco sobre que es cada característica.

4. Velocidad de transmisión. Velocidad de transmisión también es conocida como el *baud rate* o bits por segundo. Por ejemplo, si se dice que se transmite a 400 baudios significa que se está transmitiendo a 400 bits por segundo. Esto también depende del protocolo utilizado, ya que hay algunos que se manejan por flancos y no por cambios de estado. También influye la frecuencia que se utiliza.

Por lo que una mejor definición para el *baud rate* sería: número de bits que se transmiten en un tiempo T.

a. Bits de datos. Es la cantidad de bits en la transmisión. Puede ser de 5, 7 y 8 bits, dependiendo del protocolo utilizado y del tipo de información a mandar.

b. Bits de parada. Estos bits se utilizan para indicar el fin de la transmisión de una trama de datos. Los valores comunes son: 1, 1.5 y 2 bits. Entre más bits de parada mayor será la seguridad de que se transmitieron bien los datos, sin embargo será más lenta la transmisión.

c. Bits de paridad. Los bits de paridad sirven para hacer verificación de errores. Hay diferentes tipos de paridad: par, impar, marcada y espaciada. Por poner un ejemplo: si se desea transmitir 0101 con paridad impar, el bit de paridad sería uno lógico pues así se mantendría la paridad impar, es decir tres estados lógicos en alto.

La comunicación inalámbrica, como el nombre lo indica, es la transmisión de datos a través del aire. Y serial se refiere al tipo de envío de datos. Al hacer este tipo de transmisiones, hay que tener cuidado con la interferencia y la seguridad en la transmisión.

E. Microrrobótica y sus aplicaciones.

Es un poco difícil decir cuál es el tamaño inicial en el que se comienza la microrrobótica. Sin embargo, haciendo un pequeño estudio podemos establecer que comienza con un volumen de 8 pulgadas cúbicas y terminaría hasta donde ya no es fácilmente manipulable por el ser humano a través de medios convencionales; aproximadamente 1 centímetro cúbico.

La microrrobótica es un campo en exploración pero con avances ya bien marcados en el ámbito internacional. En Guatemala, es un área no explorada.

La microrrobótica deja de serlo cuando empieza la nanorrobótica. Es una diferencia de tres posiciones de corrimiento del punto decimal.

Para la construcción de un micro robot hay que tener en cuenta varios aspectos que no son del todo fáciles de resolver, sobre todo si se quiere una aplicación robusta.

Los aspectos a tomar en cuenta son:

- Motores.
- Fuente de alimentación.
- Sensores.
- Comunicación.
- CPU.
- Módulos adicionales.
- Montaje.
- Movilidad.

1. Aplicaciones. Hay numerosos proyectos en Internet referentes a microrobótica. Dependiendo de cómo sea construido y los sensores que se incorporen, así será su uso. Normalmente tiene aplicaciones como:

- Recreación.
- Estudios de movimientos de insectos.
- Espionaje.
- Exploración y mapeo de terrenos.
- Mediciones de toxicidad en tuberías o en lugares donde no entran máquinas grandes.
- Toma de imágenes de lugares poco usuales.

F. Sistemas de control

Todas las máquinas, incluso un micro robot, dependen de un sistema de control para así poder operarlos eficientemente.

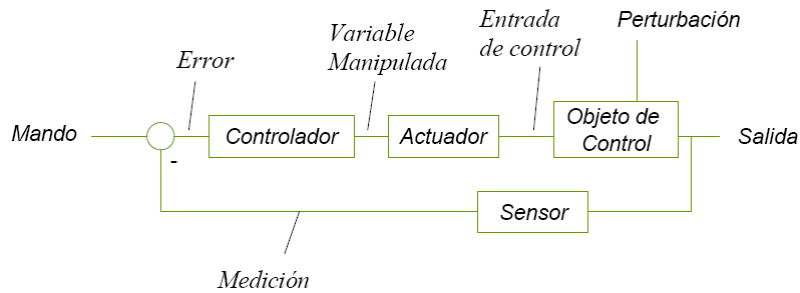
Los sistemas de control son formas de poder manipular máquinas o procesos para así minimizar los fallos. Hay dos clases de sistemas: abierto y cerrado.

1. Sistemas abiertos. Dentro de estos sistemas hay sistemas estáticos y dinámicos. Los sistemas de control estáticos son aquellos que la salida sólo depende de la entrada actual del sistema. Un ejemplo de estos es la de un tanque –válvula.

Los sistemas de control dinámicos son los que la salida dependen de la entrada actual y anteriores (poseen cierta memoria). Ejemplo de esto es la medición del nivel de un tanque de agua.

2. Sistemas cerrados. Es el sistema en donde la salida es medida para modificar la entrada de tal forma que la nueva salida se mantenga cerca del valor deseado. A esto también se les llama sistemas con retroalimentación. Ejemplo de esto es mantener el nivel de un tanque de agua. El esquema siguiente muestra mejor el sistema.

Figura 17: Diagrama de un sistema de retroalimentación.



Se va a definir cada una de las palabras que se mencionan en la Figura 17 para entender mejor el funcionamiento.

a. Mando. Es la señal de entrada de todo sistema. Es la variable que se va modificando en el transcurso del sistema hasta producir una salida.

b. Error. Es la diferencia entre la señal medida y la de entrada. Esta diferencia sirve para calibrar la entrada y mantener la salida en el valor deseado.

c. Controlador. Es una interfaz que toma la señal de entrada (error) y la procesa para producir la variable manipulada que es la que va a determinar el funcionamiento del actuador.

d. Actuador. Es elemento que es capaz de generar una fuerza a partir de líquidos, energía eléctrica y aire. Ésta recibe la orden del controlador y da la salida necesaria para activar un elemento final como una válvula. Hay tres tipos: hidráulicos, neumáticos y eléctricos.

e. Objeto de control: Es lo que se pretende mover o modificar para generar un tipo de salida. Por ejemplo una válvula, un motor, una compuerta, etc.

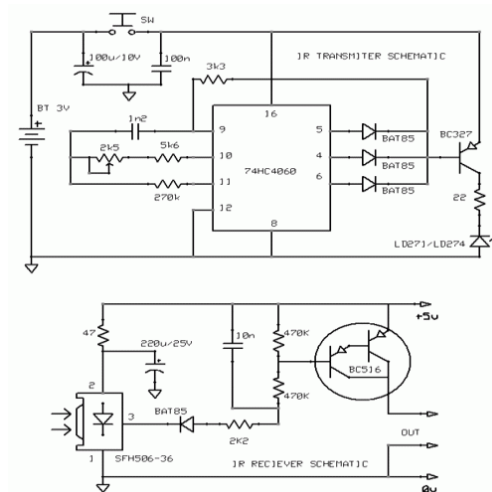
f. Sensor: Es el dispositivo que indica el estado de la salida, es decir la medición del sistema. Depende de este instrumento el funcionamiento de todo el sistema. Es por eso que debe de ser el más caro y mejor calibrado.

3. Sistema de control remoto. Básicamente es todo lo que se describió anteriormente solamente que se parte en dos. La señal de mando y el controlador están en un dispositivo llamado transmisor y el actuador en un receptor. El sensor puede estar en cualquiera de estos dependiendo de la construcción del sistema.

Si se toma un ejemplo de un control remoto por radio frecuencia para controlar un robot. El transmisor debe tener ciertos botones que generará una señal de mando que viajará a través del aire por medio de un modulador RF. El receptor capta la señal y la procesa en otro controlador para luego mandar una señal a un actuador.

Si el sensor está en el receptor, éste tendría que tener la capacidad de convertirse en transmisor, de lo contrario sería un sistema de control abierto.

Figura 18: Esquema de un sistema de control remoto.



La Figura 18 muestra un diagrama eléctrico de un receptor (abajo) y un transmisor (arriba), de un control infrarrojo. La diferencia del RF es nada más la forma en que envía la señal por el aire.

G. Potencia y motores

Una parte muy esencial de cualquier proyecto, tiene que ver con la potencia que se debe generar para poder tener en funcionamiento todos los componentes adecuadamente, sobre todo si se trata de motores.

1. Dimensional mAh. Un factor importante a considerar cuando se va a utilizar una batería en determinado proyecto, es examinar la capacidad de corriente que esta pueda proporcionar.

La dimensional mAh es una unidad de capacidad de generación de corriente eléctrica que pasa por las terminales de una batería. Esto se entiende mejor explicando la relación que se tiene con la medida de carga Culombio.

$$1 \text{ A} = 1 \frac{\text{C}}{\text{s}} \Rightarrow 1 \text{ A} \cdot \text{s} = 1 \text{ C}$$

Si lo operamos para compararlo con segundos y horas:

$$1 \text{ A} \cdot \text{h} = 1 \text{ A} \cdot \text{h} \cdot \frac{3600\text{s}}{1\text{h}} = 3600 \text{ A} \cdot \text{s} = 3600 \text{ C}$$

Por transitividad de la igualdad:

$$1 \text{ A} \cdot \text{h} = 3600 \text{ C}$$

Miliamperio hora, es la milésima parte de Ah, es decir 3.6C. Dependiendo de cuanta carga almacene una batería, así es el tiempo que se va a descargar. El tiempo de descarga viene dado por la siguiente relación:

$$\text{tiempo de descarga} = \frac{\text{carga electrica bateria}}{\text{consumo electrico dispositivo}}$$

Al hacer un sencillo despeje podemos encontrar la carga eléctrica del dispositivo:

$$\text{consumo electrico dispositivo} = \frac{\text{carga electrica bateria}}{\text{tiempo de descarga}}$$

Para visualizar mejor esto lo presentar dos ejemplos sencillos:

- Una batería tiene una carga de 1000mAh y un dispositivo consume 20mA. Al aplicar la fórmula de tiempo de descarga da como resultado 50 horas de consumo.
- Si otro dispositivo utiliza una batería de 800mAh y tarda 400 horas en descargarse, significa que tiene un consumo de 2mA.

2. Baterías. Encajando lo anterior; esta relación no es del todo lineal. Cuando una batería llega a su mínimo voltaje, la función empieza a tomar una leve curva en donde la descarga empieza a acelerar para finalmente quedar sin energía.

Hay otras características que afectan el comportamiento de descarga de una batería, como por ejemplo la temperatura y la resistencia. Esto determina la cantidad de corriente que se esté suministrando y el tiempo de utilidad. El material con que se construya la batería también es de importancia pues cada elemento tiene diferentes propiedades eléctricas.

Para ejemplificar, se hará una comparación entre dos baterías la A23 Energizer y GP27A Super Alkaline.

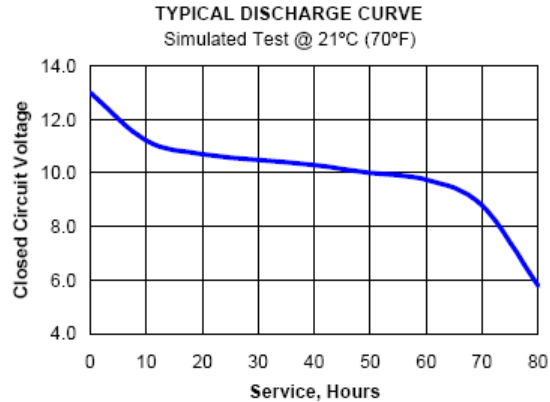
Tabla 2: Especificaciones entre baterías.

	A23	GP27A	Dim.
Qímica	Dióxido de Magnesio	Zinc/Dióxido de Magnesio	
Voltaje	12	12	V
Peso promedio	7.5	4.4	g
Volumen	0.14	0.08	cubic inch
Capacidad promedio	40	20	mAh

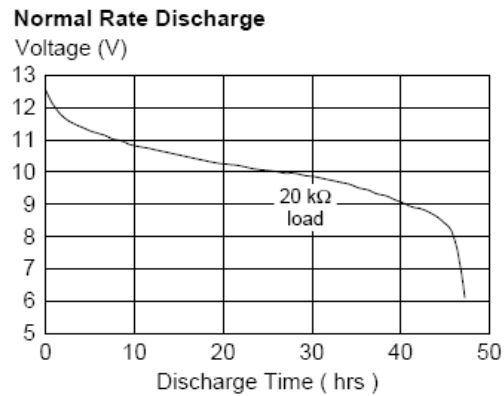
La capacidad promedio fue probada hasta los 6V con una resistencia de 20kΩ a una temperatura de 20°C, para las dos baterías.

Se puede ver que ambas baterías están casi hechas del mismo material, tienen el mismo voltaje, sin embargo la capacidad de dar corriente varía.

Gráfica 1: Batería A23 Energizer. Descarga vrs. Tiempo.



Gráfica 2: Batería GP27A. Descarga vrs. Tiempo.



Se puede ver en las Gráficas 1 y 2 que el comportamiento de descarga de las baterías es el mismo, sin embargo la duración en horas es diferente, casi 30 horas de diferencia. Esto ocurre en condiciones normales de uso.

También se observa en la Tabla 2, que la capacidad de carga es mayor en la batería A23 es mayor que la GP27A. Ahora bien, dependiendo cuál es la exigencia que se le pida a la batería, es decir si el dispositivo que se le conecte demanda más corriente / hora, la batería lo va a proporcionar, pero no durará la misma cantidad de tiempo sino menos, y la gráfica cambiaría drásticamente de horas a minutos.

3. Potencia. La potencia es la cantidad de trabajo en relación al tiempo. En términos de electricidad, es una multiplicación entre la diferencia de potencial y la corriente que circula por todo el circuito, dicho de otra forma, es el cambio de energía en el tiempo:

$$P = \frac{dE}{dt}$$

La medición se hace en watts o caballos de fuerza (HP). 1HP = 736watts. Dependiendo de los componentes que tenga el circuito a alimentar y la cantidad de corriente que estos demanden, va a ser la potencia que requiere para trabajar. Esto es de vital importancia al momento de considerar la fuente de energía de un proyecto.

En el caso de un proyecto pequeño como el de un micro robot, se tiene que tener especial cuidado pues no hay espacio para cualquier fuente de corriente.

4. Motores. Básicamente hay dos tipos de motores: los de corriente continua o DC, y los de corriente alterna o AC.

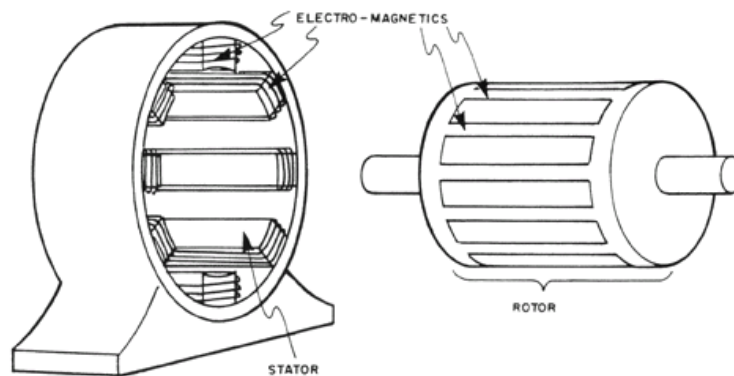
En términos generales, los motores son máquinas que convierten la energía eléctrica en energía mecánica. Se compone de dos partes principales: el estator y el rotor.

a. Estator. Es la parte fija del motor. Dentro del mismo, se encuentra dividido en lo que se llaman devanados, que son secciones de motor bobinados y separados por un desfase de 60°. La forma de este y su funcionamiento, va a depender de si es de AC o DC.

b. Rotor. Esta es la parte que se mueve de un motor, que por lo general se va a mover rotativamente. Se encuentra en el centro del estator ya que por medio del campo magnético que se genera, ésta empezará a girar y a producir energía mecánica.

Depende del tamaño del motor y de cuanta fricción tenga que vencer, así va a ser la corriente que demande.

Figura 19: Estator y Rotor de un motor^[8].



c. Torque. El torque, es un vector que mide la tendencia de una fuerza a rotar un objeto sobre un eje determinado. Así como la fuerza se usa para describir un jalón o un empuje, el torque se puede pensar como un retorcimiento. Las dimensionales son Newton Metro (N.m).

El torque es descrito matemáticamente con la siguiente expresión:

$$\boldsymbol{\tau} = \mathbf{r} \times \mathbf{F}$$

Donde \mathbf{r} es la posición del objeto y \mathbf{F} es la fuerza aplicada al objeto, que es perpendicular al vector \mathbf{r} . También se puede expresar como:

$$\boldsymbol{\tau} = \frac{d\mathbf{L}}{dt}$$

Donde \mathbf{L} es el vector del momento angular y t es el tiempo. Esto nos lleva a hablar de un torque rotacional, que es para entender la fuerza y potencia que un motor debe ejecutar.

1. Potencia y torque. Ya definidos los términos potencia y torque, ahora se debe relacionar estos dos conceptos. Para ello se usará el ejemplo de un motor de combustión.

En un motor de gasolina, se genera una combustión en el área de los pistones, haciendo que el gas haga presión sobre estos y genere un torque al cigüeñal haciéndolo girar.

Ahora bien, la potencia en un motor se debe medir con las revoluciones por minuto (rpm) y el torque que se genere. Al multiplicar las revoluciones por el torque ($F \times d / \text{min}$), obtenemos la potencia del motor. Esto va a depender de la velocidad en que este el vehículo.

En un motor eléctrico, la potencia dependerá de la corriente que éste necesite la cual se proporcionará dependiendo de cuánto torque deba vencer. Si se tiene que vencer un torque alto, entonces demandará mucha corriente.

IV. DELIMITACIÓN E IMPACTO DEL TEMA

Este trabajo de graduación comprende la investigación, análisis, diseño y elaboración de un micro robot con dimensiones de una pulgada cúbica.

El mayor reto de este trabajo, es el diseño del micro robot y la elección de componentes.

La micro-robótica es un campo científico muy grande, por lo que el trabajo se enfocará en lo más pequeño que se puede hacer sin recurrir a maquinaria sofisticada o fuera de alcance.

El trabajo de graduación se centra en clasificar la información encontrada en los medios y escoger únicamente lo que tenga relación con robots de una pulgada cúbica o similar. Es un proyecto, que al momento en Guatemala, no se había escuchado y puede abrir campo a un área no explorada en este país.

La problemática principal es el tamaño y la compaginación de motores versus fuente de alimentación. Esto es lo que el estudiante debe de poner más atención al momento de diseñar un micro robot.

V. METODOLOGÍA

La presente metodología se dividirá en dos partes. Una que mostrará la parte global del proyecto, y dos que enumerarán los pasos para la elaboración del transmisor y receptor de manera general.

A. Transmisor.

1. Programación del PIC 16F88.
2. Diseño del circuito PCB.
3. Armar el circuito.
4. Probar el funcionamiento de los componentes.
5. Desarrollar la parte estética del transmisor.

B. Receptor

1. Programación del PIC 12F683.
2. Diseño del circuito PCB.
3. Probar el funcionamiento de los componentes.

Armado todos los circuitos, se procede a probar todo el sistema ya unificado para establecer mejoras y verificar funcionamiento.

C. Herramientas de trabajo.

Para la elaboración del software se utilizó el compilador CCS C PIC Compiler por su facilidad de programación. Además, el microcontrolador utilizado en el receptor no dispone de un módulo USART para la comunicación serial, por lo que se implementó por software. También se utilizó el MPLAB para grabar el PIC 16F84A que se utilizó como cronómetro.

Las mediciones de resistencia, corriente, voltaje, etc; se utilizó un multímetro METEX M-3640D.

En la elaboración de los PCB's, se utilizó el programa "Eagle", que puede ser descargado libremente en Internet. Se optó por este software de diseño, por tener una amplia variedad de componentes y por su facilidad para diseñar el diagrama de impresión.

Para el receptor experimental se utilizó el PIC 12F683, LED's, resistencias y un driver.

En la elaboración del diseño experimental se utilizó 2 PIC's 16F84A. Uno de ellos sirve como temporizador: marca el tiempo que se tarda en estar una dirección del control remoto encendido. El segundo es una interfaz entre el receptor y la computadora. Un MAX 232 para la interfase serial, displays de siete segmentos y cable DB9.

Este segundo PIC 16F84A es la interfaz entre el receptor y un programa elaborado en Visual Basic.

VI. DESARROLLO

A. Diseño experimental para el transmisor.

1. Hardware. Se hará una lista de los componentes utilizados y luego se explicará la interacción entre ellos. El funcionamiento del mismo se comprenderá, en su totalidad cuando, se explique el software.

- 1 PIC16F88 (PDIP)
- 1 módulo RF LINX TXM – 433 - LR de 433MHz.
- 1 capacitor de 10 μ F para 35V electrolítico, radial de 0.5mm de diámetro.
- 1 capacitor de 100nF cerámico.
- 1 capacitor de 10 μ F cerámico.
- 6 resistencias de 4k7 Ω de ¼ de Watt.
- 1 resistencia de 10k Ω de ¼ de Watt.
- 1 Joystick de cuatro direcciones con “Push Botton” en el centro.
- Placa de cobre de 4.6 x 4.6 cm.

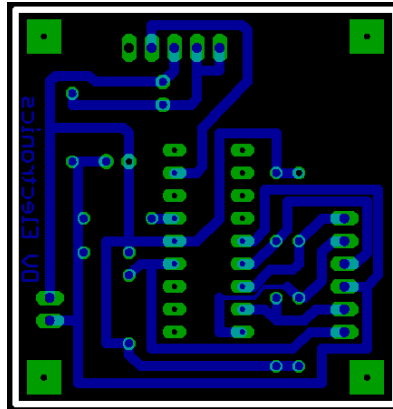
Los pines utilizados fueron (el número indica el pin):

- VSS (5)
- VDD (14)
- RA3 (2), como Tx.
- MCRL (4), \longrightarrow Entrada.
- RB0 (6), \longrightarrow Interrupción.
- RB4 al RB7 (10-13), \longrightarrow Interrupción..

Se empezará describiendo brevemente cómo es que van interconectados los componentes, y los detalles se podrán ver en el esquema del circuito (Figura 19).

Las seis resistencias de 4k7 Ω , están conectados en los pines de interrupción del RB “Port Change” y del RB0 a tierra, haciendo la función de “pull down”. Cada uno de los cuatro contactos del “joystick”, que dan las direcciones, se conectan a los pines RB4 al RB7 del PIC, para activar un

Figura 22: Diseño del PCB del control remoto, sólo las trayectorias.



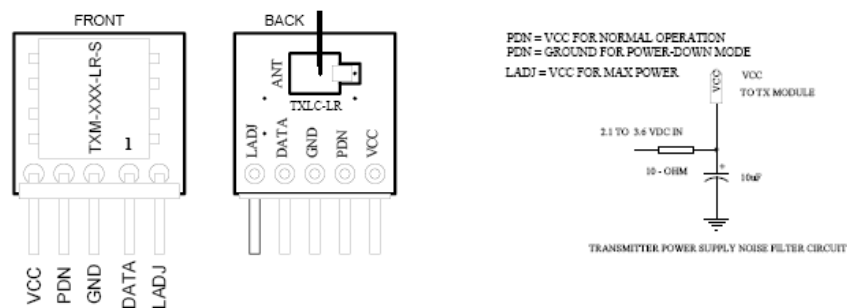
El módulo de transmisión contiene cinco pines: LADJ, DATA, GND, PDN y VCC. De los cuales, el único no utilizado es el LADJ (Level Adjust) que sirve para proporcionar intensidad a la señal de salida.

La forma en que opera este módulo es de la siguiente manera: la trama de datos sale del pin Tx del PIC, al pin DATA. El integrado convierte la señal binaria en una señal análoga por medio de la modulación digital OOK (On-Off Keying).

Los pines PDN y VCC van conectados entre si, debido a que el PDN (Power Down) sirve para habilitar o deshabilitar la transmisión. Si esta a tierra o al aire, no habrá transmisión.

Las resistencias y capacitor que se conectan en el módulo son para filtrar ruido que pueda ser causado por la alimentación de las baterías. Lo mismo aplica para el conjunto conectado al VCC y al MASTER del microcontrolador.

Figura 23: Diagrama del módulo de transmisión LINX.



a. Datos eléctricos. Estos datos fueron medidos con un multímetro METEX M-3640D. El voltaje de alimentación esta en un rango donde el microcontrolador y el módulo Tx puedan operar simultáneamente.

Tabla 3: Especificaciones eléctricas del control remoto.

	Min.	Typ.	Max.	Dim.
Voltaje de alimentación	2.1	2.9	3.6	V
Corriente de consumo	2.7	3.4	4	mA
*Potencia consumida	5.7	9.9	14.4	mW

* Con $P=VI$ asumiendo caja negra.

2. Software y transmisión. El programa contiene tres interrupciones: RB Port Change, External RB0 Interrupt y Timer0 interrupt.

La ejecución del programa se mantiene cíclica en un “loop” principal, que tiene como función la transmisión de la instrucción al receptor. Al momento de detectarse una interrupción, el puntero salta a la función específica para atenderla.

El “loop” principal consta de un ciclo *while*. Adentro se encuentra un condicionamiento *if*, que sirve para guardar el tiempo que demoró un movimiento, controlado por una bandera *j*. Y una rutina *switch* para comparar la variable que envía la rutina de interrupción RB Port Change y transmitir el movimiento.

Los códigos para los movimientos son:

47: Adelante.

57: Atrás.

67: Derecha.

77: Izquierda.

Antes de enviar el movimiento, se transmite un código único para que solamente el receptor decodifique la información. Éste es cualquier carácter que el programador desee. En este caso es el signo \$, guardado en una variable llamada “code”.

En la interrupción RB Port Change, es cuando se empiezan a transmitir los datos de manera: código - movimiento. Si no hay movimiento, lo que se está transmitiendo es un código 33, que significa “no hacer nada”. Pero todas las transmisiones se mandan junto con un “stream”, que es una serie de caracteres que ayudan a eliminar el ruido. El “stream” utilizado es la cadena QVF.

Cuando se detecta la interrupción "RB Port Change", hay dos posibilidades: cambio de estado de bajo a alto o, alto a bajo. Si fue la primera, la rutina de servicio guarda en una variable el estado del puerto b, lo compara y lo condiciona con otra variable de control para detectar cuál fue el pin que generó la interrupción. Los cuatro pines que activan esta interrupción son para cada uno de los movimientos: adelante, atrás, derecha e izquierda.

Detectado el pin, se manda en una variable global con el código que se debe de transmitir y se cambia el valor de dos banderas.

```
current=input_b();

#if LOWTOHIGH
if ((!bit_test(last,4))&&(bit_test(current,4)))
{
    data = 47;
    i=1;
    j=0;
}
```

La bandera *i* es para habilitar la condición que guarda el movimiento en la memoria EEPROM y para comenzar el Timer0.

```
case 47: {
    if(i==1)
    {
        write_eeprom(address,57);
        address += 0x01;
        cont2++;
        i=0;
        enable_interrupts(INT_TIMER0);
    }
    fputc(code,QVF);
    fputc(data,QVF);
}
break;
```

Si el cambio fue de alto a bajo, la rutina lo detecta y apaga el Timer0, cambia las banderas y manda a transmitir el código 33.

```
#if HIGHTOLOW
if (!(bit_test(current,4))&&(bit_test(last,4)))
{
    disable_interrupts(INT_TIMER0);
    data = 33;
    j=1;
}
}
```

El External RB0 interrupt, se activa cuando se presiona el botón del joystick. Cuando esto sucede, el puntero salta a la rutina que atiende esta interrupción. Las variables que manejan esta rutina de interrupción son *cont*, *cont2*, *tim* y *time*.

La variable *cont* va aumentando de uno en uno comparándolo con *cont2*, que es la que almacena la cantidad de veces que se transmite un movimiento. Esta comparación sirve como límite en un ciclo *while* que controla la lectura de la EEPROM.

Al momento de entrar en el ciclo *while* se lee las primeras dos posiciones de la memoria EEPROM, comenzando por el último espacio grabado. La primera lectura es el tiempo (círculo rojo, Figura 23) y se guarda en la variable *time*; la segunda es el movimiento (círculo verde, Figura 23) que se trasladada a la variable *data*.

Figura 24: Memoria EEPROM del PIC 16F88

Code	EEPROM Data	User ID	Configuration	Device ID
0000-	067 001 077 003 047 007	255 255	255 255	255 255
0008-	255 255 255 255 255 255	255 255	255 255	255 255
0010-	255 255 255 255 255 255	255 255	255 255	255 255
0018-	255 255 255 255 255 255	255 255	255 255	255 255
0020-	255 255 255 255 255 255	255 255	255 255	255 255
0028-	255 255 255 255 255 255	255 255	255 255	255 255
0030-	255 255 255 255 255 255	255 255	255 255	255 255
0038-	255 255 255 255 255 255	255 255	255 255	255 255
0040-	255 255 255 255 255 255	255 255	255 255	255 255
0048-	255 255 255 255 255 255	255 255	255 255	255 255
0050-	255 255 255 255 255 255	255 255	255 255	255 255
0058-	255 255 255 255 255 255	255 255	255 255	255 255
0060-	255 255 255 255 255 255	255 255	255 255	255 255
0068-	255 255 255 255 255 255	255 255	255 255	255 255
0070-	255 255 255 255 255 255	255 255	255 255	255 255
0078-	255 255 255 255 255 255	255 255	255 255	255 255
0080-	255 255 255 255 255 255	255 255	255 255	255 255
0088-	255 255 255 255 255 255	255 255	255 255	255 255
0090-	255 255 255 255 255 255	255 255	255 255	255 255
0098-	255 255 255 255 255 255	255 255	255 255	255 255
00a0-	255 255 255 255 255 255	255 255	255 255	255 255
00a8-	255 255 255 255 255 255	255 255	255 255	255 255

La variable *time* se utiliza dentro de un ciclo *for* como control de cuánto tiempo se debe mantener la transmisión del movimiento, pero este movimiento es el contrario del presionado originalmente, es decir, si el micro-

robot fue para adelante (código 47), en la memoria se graba para atrás (código 57) de tal forma que luego pueda reproducir su camino de regreso al punto de partida (Referirse al código de programa en los anexos).

Pero debido a que el tiempo es grabado como una aproximación del tiempo real, se debe tener otro ciclo *for* anidado de ajuste, tomando en cuenta la configuración del reloj interno del microcontrolador. Esta variable es *r* que va aumentando mientras siga siendo menor a 224. El último número es lo que se denominó como *índice de resolución*, que retarda el tiempo grabado acorde al ciclo de ejecución de las instrucciones y así dar una transmisión casi exacta del movimiento en tiempo más o menos igual al original. Este *índice* se calculó de manera empírica basado en un cálculo que se explicará en el Timer0.

Terminado el proceso anterior, se procede a inicializar nuevamente las variables y a limpiar la memoria EEMPROM.

La interrupción del TIMER0 se para el cálculo del tiempo que tarda la transmisión de un movimiento y se activa sólo al momento de transmitir el movimiento, y se desactiva al dejar de presionar el "joystick".

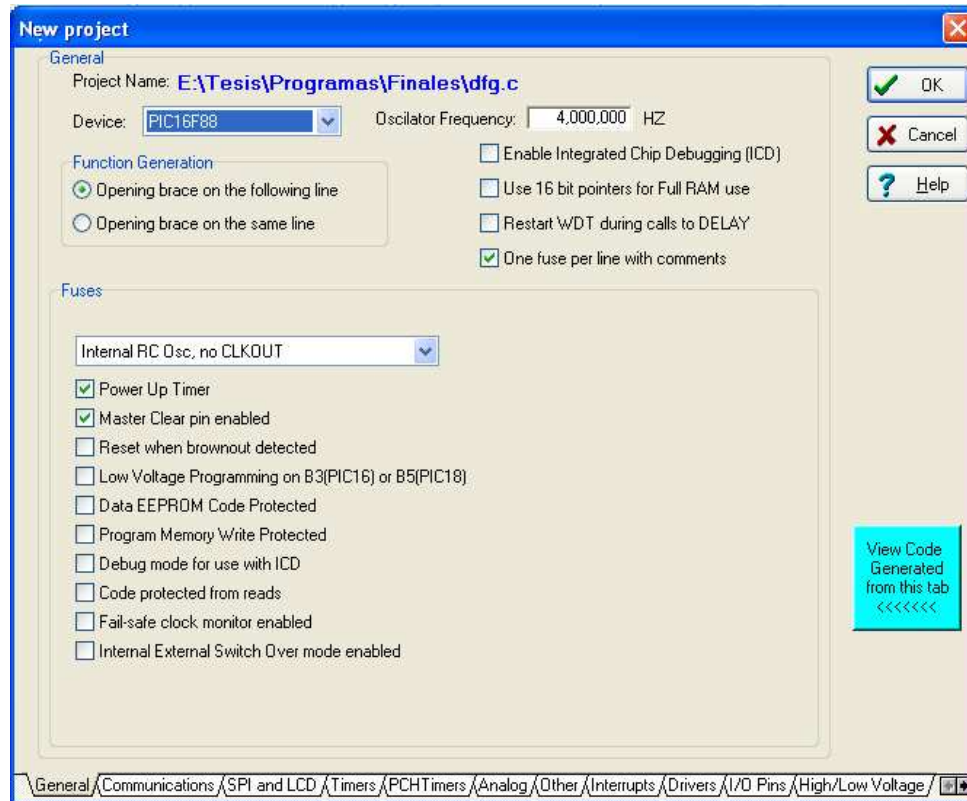
En la configuración del reloj interno, PIC C tiene la opción de elegir la resolución para determinar cada cuanto se efectuará un "overflow". Al escoger una resolución de 256µs se obtiene que cada 65.5ms ocurrirá un "overflow" y por lo tanto una interrupción. Ahora bien, para determinar cuántas interrupciones se llevan a cabo en 1 segundo se hizo una simple división: $1 / 0.0655 = 15.26$. Significa que pasan 15 llamadas a la rutina de interrupción para llegar al segundo, por lo que se utilizó una variable llamada *t2* para controlar esto a través de una condición *if*. Cumplida la condición, se aumenta la variable *t*, que es la que guarda los segundos y es la que posteriormente se graba en la memoria EEPROM.

Partiendo del cálculo anterior, se resto $255 - 15 = 240$, y en base a este cálculo intuitivo se empezó a calibrar el *índice de resolución* mencionado anteriormente.

Este programa ocupa, en la RAM un %16 y en la ROM, un 17% de espacio, es decir, un total de 33% utilizado.

a. **Configuración del PIC 16F88.** La configuración del PIC se hace desde el PIC Wizard de PIC C Compiler. Ésta es una ventana donde se seleccionan los “FUSES” del PIC, que van a determinar el comportamiento del PIC dependiendo de la programación.

Figura 25: Pantalla de PIC Wizard.



En la Figura 25 se observa la primera pantalla de configuración para un PIC. Se selecciona el PIC, la frecuencia y los FUSES. Luego en la parte de abajo de la figura, hay varias pestañas que son las diversas funciones que tiene el PIC.

Seleccionado todo lo que se va a utilizar, la configuración quedó de la siguiente manera:

```
#include <16F88.h>
#device adc=8

#FUSES NOWDT           //No Watch Dog Timer
#FUSES INTRC_IO       //Internal RC Osc, no CLKOUT
#FUSES PUT             //Power Up Timer
#FUSES MCLR           //Master Clear pin enabled
#FUSES NOBROWNOUT     //No brownout reset
#FUSES NOLVP          //No low voltage progming, B3(PIC16) or //B5(PIC18) used for I/O
#FUSES NOCPD          //No EE protection
#FUSES NOWRT          //Program memory not write protected
```

```
#FUSES NODEBUG      //No Debug mode for ICD
#FUSES NOPROTECT    //Code not protected from reading
#FUSES NOFCMEN      //Fail-safe clock monitor disabled
#FUSES NOIESO       //Internal External Switch Over mode disabled

#use delay(clock=4000000)
#use rs232(baud=4800, parity=N,xmit=PIN_A3,rcv=PIN_A2,bits=8,stream=QVF)
```

Se observa que la tasa “*baud rate*” es de 4800, que es la velocidad de transmisión de signos. Esto tiene que ser igual en el receptor. Para comprender cada FUSE, consultar el manual del PIC.

Se debe tener precaución en la pestaña donde se configura cada pin como entrada y salida. El pin A3, por ejemplo, se configura como salida, pues es el de transmisión. Luego el pin A5 se configura como entrada, ya que se habilitó el MASTER CLEAR. También los pines que generan interrupción son entradas, el resto pueden ir como salidas por seguridad.

3. Discusión y resultados – transmisor. El transmisor no tiene restricción de tamaño, por lo que puede variar el micro controlador utilizado y los componentes. Sin embargo, es recomendable cambiar el módulo transmisor. Se explicará más a detalle en el receptor.

En el software, la calibración para tomar el tiempo de los movimientos esta sujeto a dos parámetros: el *índice de resolución* y la *constante para el segundo*. El primero fue comentado anteriormente y su valor es 224. El segundo es el número 14, que fue el número de veces que tiene que ocurrir un “overflow” en el programa, para aumentar en un segundo la variable *t*.

Algo similar se hizo para la *constante del segundo*. Se experimentó con el número 15, y luego al comparar con los resultados del *índice de resolución*, se optó por bajarlo al número 14.

Se realizaron pruebas con la función *get_timer0*, con el propósito de ver si se podía calibrar aun más, por medio de una comparación al momento de aproximar la cantidad de segundos que se grabaría en la memoria EEPROM; pero no dio resultado. El manejo adecuado de esta función, puede mejorar la resolución del tiempo.

El software desarrollado tiene una resolución de un segundo. Esto quiere decir que si el tiempo real de movimiento del robot es de 7.56 segundos, en la memoria es probable que se grabe 8 segundos y en casos esporádicos 7

segundos; haciendo que el micro robot no regrese a su punto exacto de partida.

El circuito de este transmisor tiene un detalle que es de considerar. Las resistencias que están en el puerto B, para generar la interrupción de *Port Change*; están configuradas en *pull down*. Realmente hay que ponerlas en *Pull Up*, es decir con las resistencias a +5V no a GND. Esto por una razón: al estar las resistencias a tierra, estas pueden servir de antena y genera ruido. Aunque el software tiene un protocolo definido, este ruido no es por datos sino por falsas interrupciones, que pueden generar molestias. Esto es algo que se debe de mejorar al circuito de la figura 19.

B. Diseño experimental para el receptor.

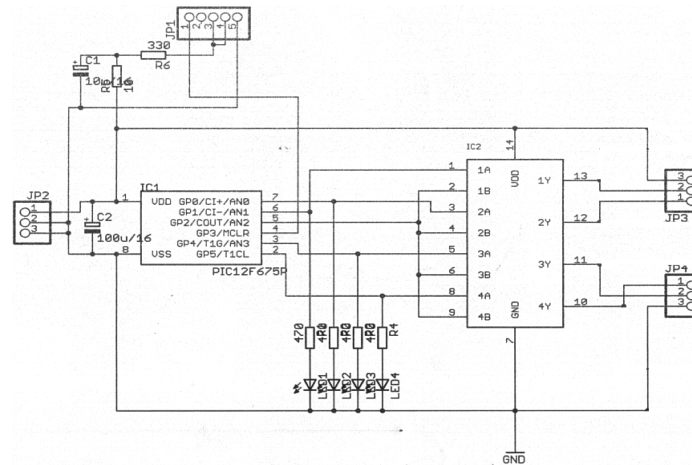
1. Hardware. De la misma manera que en el transmisor, se hará una lista de los componentes utilizados y explicaremos la interacción entre ellos.

Los componentes utilizados son:

- 1 PIC 12F683 (DIP)
- 1 TC4468 (DIP)
- 1 módulo RF LINX RXM – 433- LR de 433 MHz.
- 1 resistencia de 10Ω
- 1 resistencia de 330Ω
- 1 capacitor de $10\mu\text{F}$
- 14 conectores individuales divididos en cuatro grupos.

La mejor forma de explicar la interacción entre estos componentes, es mostrando el diagrama del circuito:

Figura 26: Diagrama del circuito receptor, con Led's para prueba.

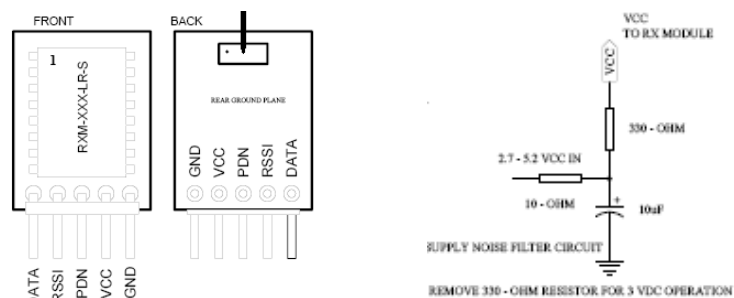


Se describirá el funcionamiento de los conectores, para entender el funcionamiento total del circuito. Se observa que hay cuatro grupos: J1, J2, J3 y J4, cada uno son su razón de ser.

J1 como se observa en la figura 35, es el conector para el módulo de recepción LINX. Las entradas 3 y 4 van unidos entre sí, pues es la conexión de alimentación del receptor que van a un arreglo de resistencia y capacitor para filtrar ruido. La entrada 1 es donde ingresa la señal de datos al receptor, que funciona al contrario que el transmisor, es decir, recibe la señal OOK y la transforma en una trama binaria de datos el cual puede interpretar el PIC. Hay una conexión que va a tierra, la número 5. El pin que queda libre en el módulo, llamada RSSI, proporciona una señal analógica dependiendo de la intensidad de la señal de recepción.

El largo de la antena, tanto para el receptor como para el transmisor; deben ser del mismo tamaño y su altura depende de la frecuencia que se desea utilizar. En este caso mide 6.5 pulgadas para la frecuencia de 433MHz, según especificaciones de fabricante.

Figura 27: Diagrama del módulo de recepción LINX.



J2 es la alimentación del circuito. J3 y J4 simula las salidas de los posibles motores para el micro robot. Hay dos salidas adicionales, los pines 3, que son extensiones de VCC y GND para el circuito del diseño experimental.

Los pines utilizados en el PIC 12F683 son (el número indica el pin):

- VCC (1)
- VSS (8)
- GP0 (7) } Como posibles salidas a un motor.
- GP1 (6) }
- GP2 (5) → Salida del PWM (no utilizado).
- GP3 (4) → Como entrada de datos del módulo receptor.
- GP4 (3) } Como posibles salidas a un motor.
- GP5 (2) }

Las especificaciones eléctricas del receptor son:

Tabla 4: Datos eléctricos de consumo PIC y dispositivo RF.

	Min.	Typ.	Max.	Dim.
Voltaje de alimentación	5.8	5.8	6.12	V
Corriente de consumo	6.8	7.2	7.2	mA
Potencia consumida	39	42	44	mW

1. Software y recepción. El software consiste en un *while* “infinito” con una serie de tres *if* anidados entre sí y un *switch* dentro del último *if*.

La función *kbhit()* es predeterminada en PIC C Compiler que sirve para que el microcontrolador detecte cuando hay una transmisión entrando en el pin de recepción.

Cuando el protocolo de comunicación RS232, controlado por software, detecta el primer bit que entra en el pin GP3, la función sabe que hay datos ingresando. Este regresa un valor de TRUE al programa principal, y eso sirve como condicional en la sentencia *if*; Dando la facilidad de que se pueda “polear” constantemente la entrada en el ciclo “infinito” *while*.

```

do{
    if(kbhit())
    {
        code = fgetc(QVF);
        .
        .
        .
    }
    While(1);
}

```

El primer *if* que aparece controla la entrada del código \$, y para esto tiene que detectar que sea precedido por el “stream”, que son las dos formas de seguridad en la transmisión. Si la primera trama que entra es el código, se cumple la condición del segundo *if* y este entra a otro ciclo *while* condicionado.

Dentro de este ciclo está el último *if* que, al igual que el primero, tiene como condicional la función *kbhit()*. Este detecta la entrada de la segunda trama que es ya el código de movimiento. Al ingresar, guarda el movimiento en una variable y evalúa por casos en la función *switch*. Si, por ejemplo, el movimiento es para atrás (código 57), se da un *output_a(0x2E)* que significa que se van a poner en alto las salidas GP0, GP4 y con la GP2, que siempre se mantiene encendida, va hacia el driver TC4468 provocando una salida lógica de 1 y otra de 0 hacia los motores.

Si la transmisión es el código 33, la salida sera 0x00 que mantiene en bajo los pines provocando movimiento nulo.

El espacio utilizado en la memoria ROM es un 9% y en la RAM un 7%, haciendo un total de 16% en espacio utilizado para este software.

Se realizó un diseño experimental que demostrara la capacidad del control remoto para guardar un recorrido y ejecutarlo en sentido inverso. Para ello se debe simular las salidas de los motores y contar el tiempo que demore una instrucción de movimiento enviado por el transmisor. Se armaron dos circuitos y un programa en Visual Basic para dicha comprobación.

El primer circuito consta de un PIC 16F84A conectado a dos displays de siete segmentos. Éste tiene como objetivo activarse inmediatamente después de que se ejecute un movimiento para empezar un cronómetro de segundos. Esto se logra con la interrupción RB “Port Change” en el pin RB5. Se puede llegar a tomar hasta un minuto y medio con nueve segundos para un movimiento, ya que son dos displays y puede contar hasta 99.

Para activar la interrupción, se toman las salidas del Driver TC4468 y se introducen a una serie de compuertas OR. La salida de la última compuerta genera la señal hacia el pin RB5 activando la interrupción, y comienzo del cronómetro. Al momento de soltar el “joystick”, éste vuelve a cero y detiene el tiempo, deshabilitando la interrupción TMR0 del PIC.

Figura 28: Diagrama del receptor con el juego de compuertas OR para el diseño experimental.

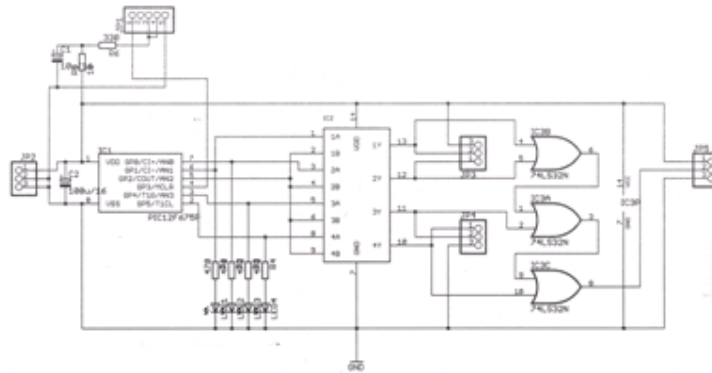
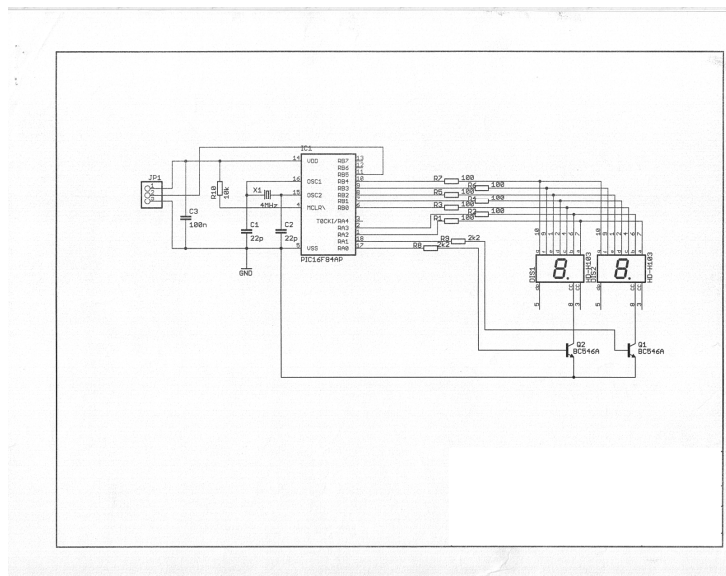


Figura 29: Diagrama del cronómetro.



El otro circuito es otro PIC 16F84A. Su software es muy similar al del 12F683 con la única diferencia que tiene un código más para detectar el “push botton” del transmisor. El código es 87. En el transmisor el código está en la interrupción RB0.

La función de este segundo circuito es transmitir las instrucciones a la computadora por medio del puerto serial. Para ello es necesario el uso de un MAX232.

Figura 30: Diagrama del circuito de interfaz con la computadora.

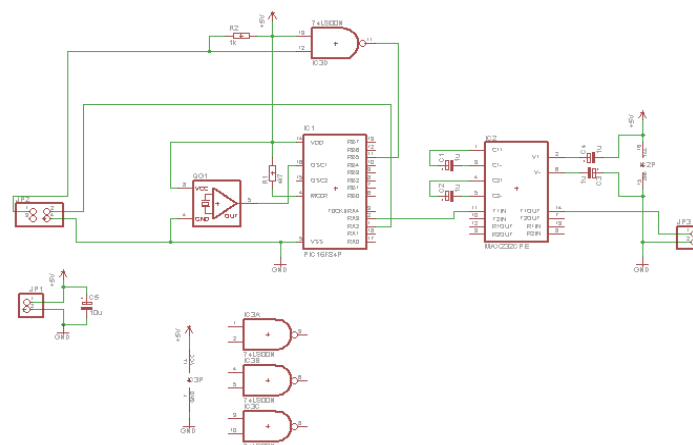
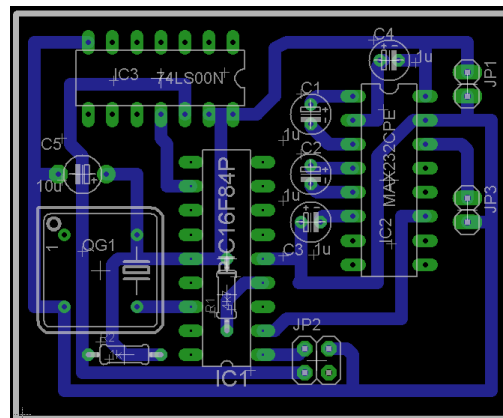
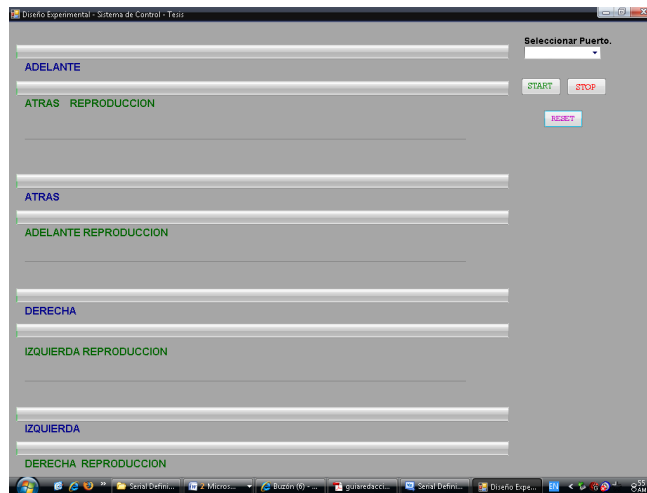


Figura 31: PCB del circuito de interfaz con la computadora.



Las instrucciones son decodificadas por un programa hecho en Visual Basic, que tiene como misión mostrar de una manera visual el progreso de los movimientos del micro robot.

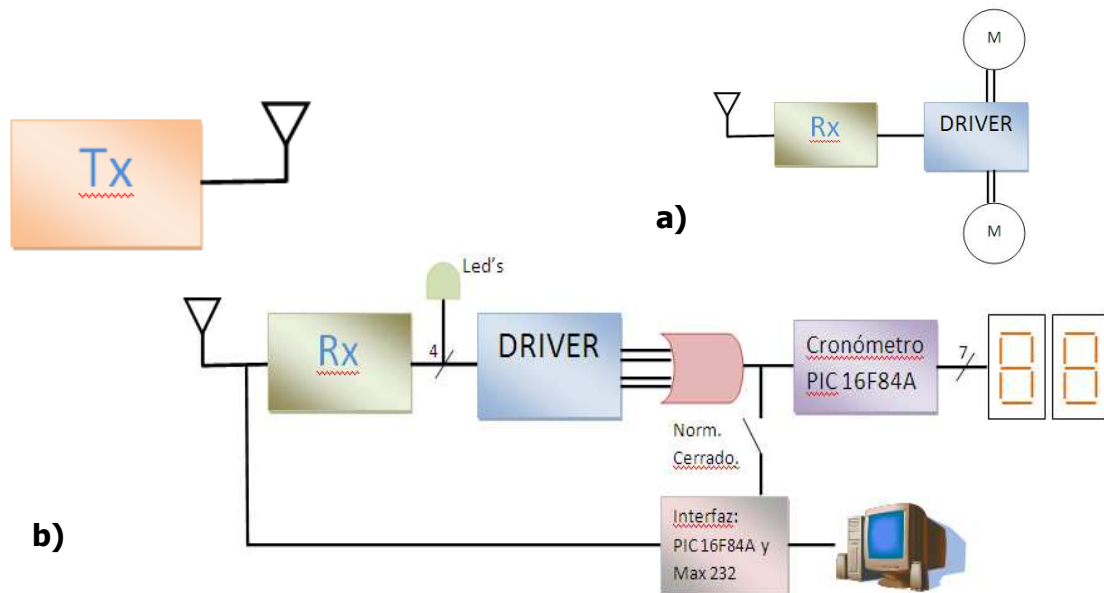
Figura 32: Pantalla del programa de Visual Basic para Diseño Experimental.



En la Figura 32 se puede apreciar la forma en que se simula el avance de cada movimiento y la comparación con la trayectoria inversa, por medio de barras de progreso. Cuando detecta el código 87, es cuando el programa sabe que vienen los códigos del trayecto inverso. Si detecta un código 97, sabe que hubo un error de algún tipo y despliega un mensaje indicando que el microrobot esta dañado. Esto se simula con el switch que conecta la salida de la compuerta OR con el PIC de interface (ver Figura 33 B).

El programa tiene un botón de RESET para poder hacer varias medidas. Un botón de STOP que detiene la entrada serial, y START que abre el puerto.

Figura 33: Diagrama de bloques de todo el sistema. a) Receptor sin diseño experimental. b) Receptor con el diseño experimental.



1. Discusión y resultados – receptor. El receptor debe instalarse en la estructura del micro robot de una pulgada cúbica. Sin embargo, este diseño no es del todo funcional. Diversos factores deben mejorarse. Uno de ellos es la utilización de otros módulos receptores, ya que los utilizados fueron muy grandes para el propósito.

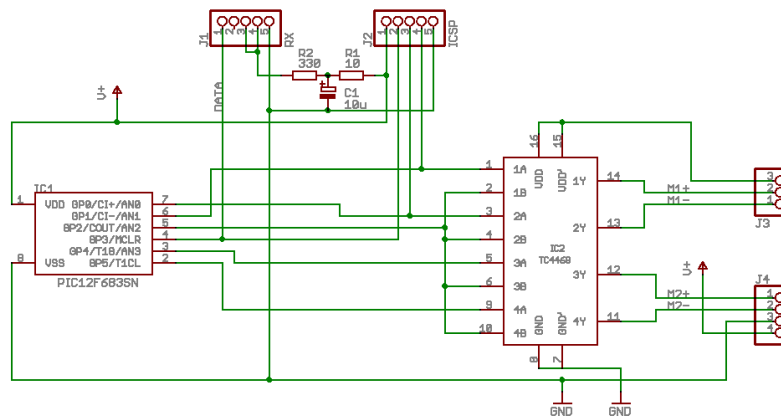
Los receptores que se recomiendan son los circuitos integrados HX1000 como transmisor y el RX1020 como receptor. Ambos trabajan a 433.92 MHz, elaborados por RF Monolithics, Inc (Figura 8).

Estos receptores, con su comunicación RF, dan la facilidad de comunicarse con otro robot e intercambiar datos entre sí o con un computador. Tienen un alcance de 10 metros a 1 kBaud de velocidad. Estos chips tienen una dimensión de 5.99 x 3.99 x 2.29 mm, ideales para un micro robot de una pulgada cúbica. Dejan el espacio necesario para la fuente de alimentación y motores.

Los componentes a utilizar como micro controladores, resistencias, capacitores, etc, deben ser de superficie. Estos se pueden comprar en la página www.mouser.com.

El diseño recomendado para el circuito que se instalará en el micro robot es el que aparece en la Figura 34, que en su mayoría ya fue explicado.

Figura 34: Circuito receptor.



El driver TC4468, puede ser opcional dependiendo de los motores a utilizar y de la fuente de alimentación.

J2 es el conector para el adaptador del programador de microcontroladores. El PIC 12F683 tiene la capacidad de ser programado en el circuito por medio del “In Circuit Serial Programming”; esto a través de dos pines. El pin 6 recibe los datos y el otro lleva la señal del reloj para la programación (pin7). También se debe alimentar normalmente con VCC y GND, que son las entradas 1 y 2 de J2. Esta es una forma para programar un circuito con integrados de superficie.

A continuación, en las Figuras 35 y 36, están los diagramas PCB:

Figura 35: Diseño del PCB del receptor sin componentes.

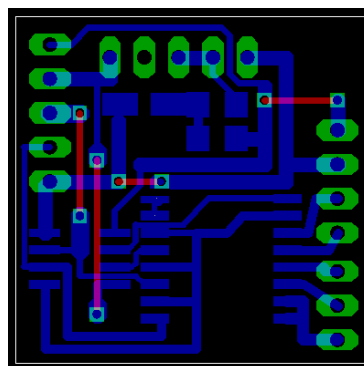
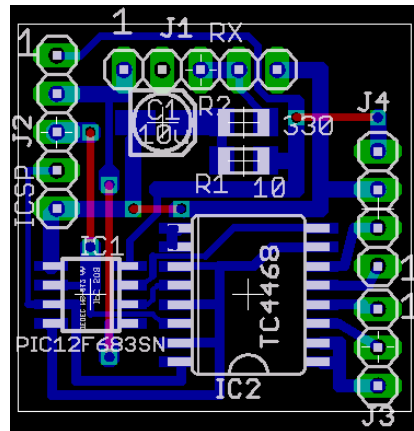


Figura 36: Diseño del PCB del receptor con componentes.



El adaptador especial, es una resistencia de 10kΩ y un diodo, con unos conectores para transmitir desde el programador USB, al circuito. Los pines y conectores van a depender del tipo de programador utilizado por el diseñador del micro robot. En este caso es un programador MELABS. Ver Figuras 37, 38 y 39.

Figura 37: Adaptador para programación ICSP.



Figura 38: Programador USB MELABS.



Figura 39: Adaptador y programador conectados.



Las entradas de J3, son la alimentación positiva del driver TC4468 y la conexión de un motor.

J4 contiene la entrada para la alimentación positiva del PIC y el módulo receptor y la tierra. También esta la conexión del segundo motor al driver de potencia.

Hasta aquí se ha hablado de todo lo relacionado con la recepción de datos, programación y componentes del circuito. En la parte C se explicará todo lo referente a motores, chasis del motor y fuente de alimentación del micro robot.

Con esto se puede dar una lista de los componentes electrónicos que, basados en lo anterior, son los que podrían servir para la construcción del micro robot:

- 1 PIC 12F683 (SOIC)
- 1 TC4468 (SOIC) →Opcional
- Integrados RF HX1000 y RX1020.
- 1 resistencia de 10Ω
- 1 resistencia de 330Ω
- 1 capacitor de $10\mu\text{F}$
- 14 conectores individuales divididos en 4 grupos.

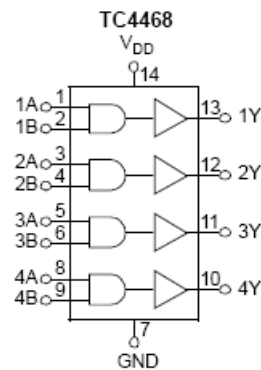
C. Parte mecánica y de diseño del micro robot.

En esta sección se discutirá sobre los componentes mecánicos del micro robot. Este es uno de los factores más importantes al cual se le debe poner especial cuidado, ya que de esto depende el éxito o no del micro robot. Por estos factores fue que el objetivo de construcción no se logró al 100%.

1. Potencia y motores. Fue en este tema en donde se hizo una investigación amplia de componentes y formas de alimentación, ya que son estos los que determinarán el espacio a utilizar en la construcción del micro robot; en este caso de una pulgada cúbica.

En este proyecto se observó la necesidad de utilizar un driver de potencia para alimentar los motores. El utilizado fue el TC4468, que es un integrado que contiene una serie de cuatro compuertas AND. Depende de la alimentación proporcionada, va a ser la corriente obtenida:

Figura 40: Esquema del integrado TC4468.



La Figura 40 explica de una forma ilustrativa el funcionamiento exacto de este driver.

El voltaje de funcionamiento va de 4.5 a 18V. En este caso se optó por 12V. Por esto, fue necesaria la separación completa de la alimentación del PIC y del módulo receptor. Sin embargo, esto no es nada recomendado para un micro robot de una pulgada cúbica.

Tabla 5: Datos eléctricos de consumo del Driver.

	Min.	Typ.	Max.	Dim.
Voltaje de alimentación	11.75	11.9	12	V
Corriente de consumo	* 0.608		64.8	mA
*Potencia consumida	7.14		778	mW

* Corriente consumida cuando no se activa los controles del transmisor.

Las baterías experimentadas para la alimentación del driver fue una Energizer A23, con las siguientes especificaciones eléctricas:

Tabla 6: Datos eléctricos de batería Energizer A23.

Classification:	"Miniature Alkaline"
Chemical System:	Manganese Dioxide (MnO_2)
Designation:	ANSI-1181A
Nominal Voltage:	12.0 volts
Typical Capacity:	65 mAh to 6.0 volts (rated at 20k ohms at 21°C)
Typical Weight:	8.0 grams (0.3 oz)
Typical Volume:	1.1 cubic centimeters (0.07 cubic inch)

Figura 41: Batería Energizer A23.



La batería A23 cumple su función por un período corto de tiempo, debido a que su capacidad es de 65mAh y el motor utiliza 64.8mA.

Para alimentar el PIC y el modulo se utilizaron dos baterías CR2032, con las siguientes especificaciones eléctricas y de dimensión:

Tabla 7: Datos eléctricos CR2032.

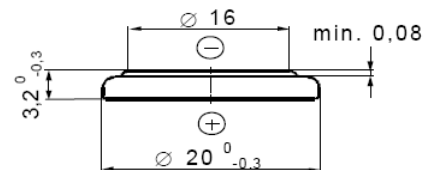
Specifications

Chemical System	Li / MnO_2
Nominal Voltage	3 V
Rated Capacity	235 mAh
Standard Discharge Current	0,4 mA
Max. Cont. Discharge Current	3,0 mA
Average Weight	2,8 g
Temperature Range	-30 - +70 °C
Self Discharge at 23°C	< 1% / year

Figura 42: Dimensiones CR2032.

Dimensions

(According to IEC 60086)



Sin embargo ninguna de estas baterías fueron adecuadas para el proyecto. Si se cambian los módulos RF y el Driver por otros que consuman menos corriente, y se consiguen los motores del ALCE (Figura 6), la CR2032 podría ser suficiente para alimentar todo.

Los motores experimentados son dos GH712 que poseen caja de un engrane. Debido a que consume mucha corriente, fue necesario quitar la caja de velocidad para reducir el consumo de corriente, bajándolo a la mitad. De 80mA con caja, consume 40mA sin caja, cada motor.

Figura 43: Motor GH712 Sin caja de engrane.



Los componentes mencionados anteriormente (baterías y motores), no fueron los primeros en utilizarse. Se hicieron varias pruebas con otros tipos de baterías y de motores.

En baterías se experimentó con FR-130 Lithium Polymer Cell de 3.7V con capacidad de 130mAh y con motores TPM Tiny Pager Motors. Sin embargo, estos motores TPM no funcionaron por falta de torque.

Figura 44: Motor TPM^[11].



En esta sección se trata de encontrar un equilibrio entre la corriente que consume el motor y la que puede proporcionar la batería.

Tabla: 8: Datos eléctricos de consumo PIC y dispositivo RF.

	Min.	Typ.	Max.	Dim.
Voltaje de alimentación	5.8	5.8	6.12	V
Corriente de consumo	6.8	7.2	7.2	mA
Potencia consumida	39	42	44	mW

Los datos de la Tabla 8, son únicamente utilizando los módulos RF y el PIC. Se le tendría que sumar la de los motores para saber el total de corriente utilizada ($7.2 + 64.8 = 72\text{mA}$).

1. Chasis. El chasis del micro robot debe de ser liviano para que el motor no utilice mucho torque para moverlo. Con peso liviano reduce el consumo de corriente. Se recomienda usar teflón (politetrafluoretileno), ya que es muy liviano y fácil de maquilar.

Figura 45: Chasis elaborado con un polímero.

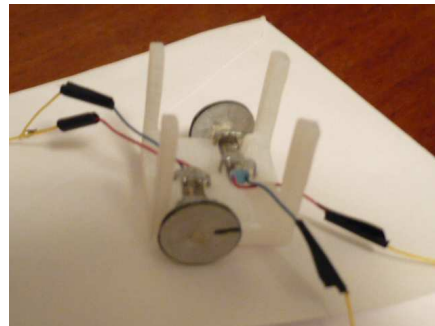


Figura 46: Otras opciones para chasis.



Es evidente que dependiendo de los componentes a utilizar, será la forma del chasis. Sin embargo, en la Figura 45 se ve una opción más abierta que el de la figura 46; se recomienda que sea así para que pese menos y para que haya más espacio y así se le puedan colocar más componentes.

Para esta sección, la lista recomendada de componentes a utilizar son:

- Motores SWATCH.
- Baterías CR2032

Para el chasis, cualquier material que sea liviano puede servir. Incluso los mismos componentes se puede buscar la manera de colocarlos con algunos sujetadores para hacerlos más liviano.

VII. CONCLUSIONES Y RECOMENDACIONES

Se lograron recopilar todos los factores que son necesarios para la construcción de un micro robot. La tesis pretendió ilustrar detalles técnicos favorables y desfavorables necesarios para poder desarrollar un micro robot de una pulgada cúbica.

Debido al hecho que no se logró el primer objetivo específico, se elaboró un diseño experimental para demostrar que el sistema de control efectivamente reproduce la trayectoria inversa.

Al presentar un listado de posibles componentes para un receptor de micro robot, se cumple con el objetivo de desarrollar la base para la elaboración de un micro robot. Se explica la interconexión y funcionamiento entre ellos, para poder construir el sistema.

Hay que tener en cuenta la parte mecánica del micro robot. Se debe tener cuidado en la integración de los componentes en algún tipo de estructura. Para ello, se recomienda tener definido los elementos más importantes como la fuente de alimentación y motores. El material a usar para la estructura del micro robot debe ser ligero ya que los motores deben poder moverlo. Una recomendación es utilizar teflón como parte de la estructura.

En este proyecto se utilizaron placas pequeñas de cobre para los circuitos, pero se recomienda buscar un PCB flexible para el circuito.

Se logró elaborar un control remoto con una comunicación serial por medio de módulos de radiofrecuencia con un protocolo sencillo. El programa utilizado fue PIC C Compiler; por su facilidad de programación. El funcionamiento fue el esperado, sin embargo se recomienda conseguir los módulos que utiliza ALICE fabricados por RF Monolithics HX1000 y RX1020.

Con la programación se logró que el control remoto tenga capacidad para grabar un recorrido corto y lo ejecute en sentido contrario. El diseño experimental ayudó a comprobarlo. Tiene una resolución de un segundo, por lo que aún se pueden hacer mejoras en el software para lograr un mejor desempeño.

El software es muy robusto en cuanto a la señal RF. Se presenta una mínima interferencia, pero se debe a que las resistencias que están conectadas al PIC 16F88 están configuradas como "pull down" cuando deben estar en

“Pull up”. Pero se recomienda utilizar un método CRC (Cyclic Redundancy Check) para aplicaciones más científicas.

Cualquier consulta sobre esta tesis, puede hacerla escribiendo al correo electrónico: tonorodil83@yahoo.com.mx.

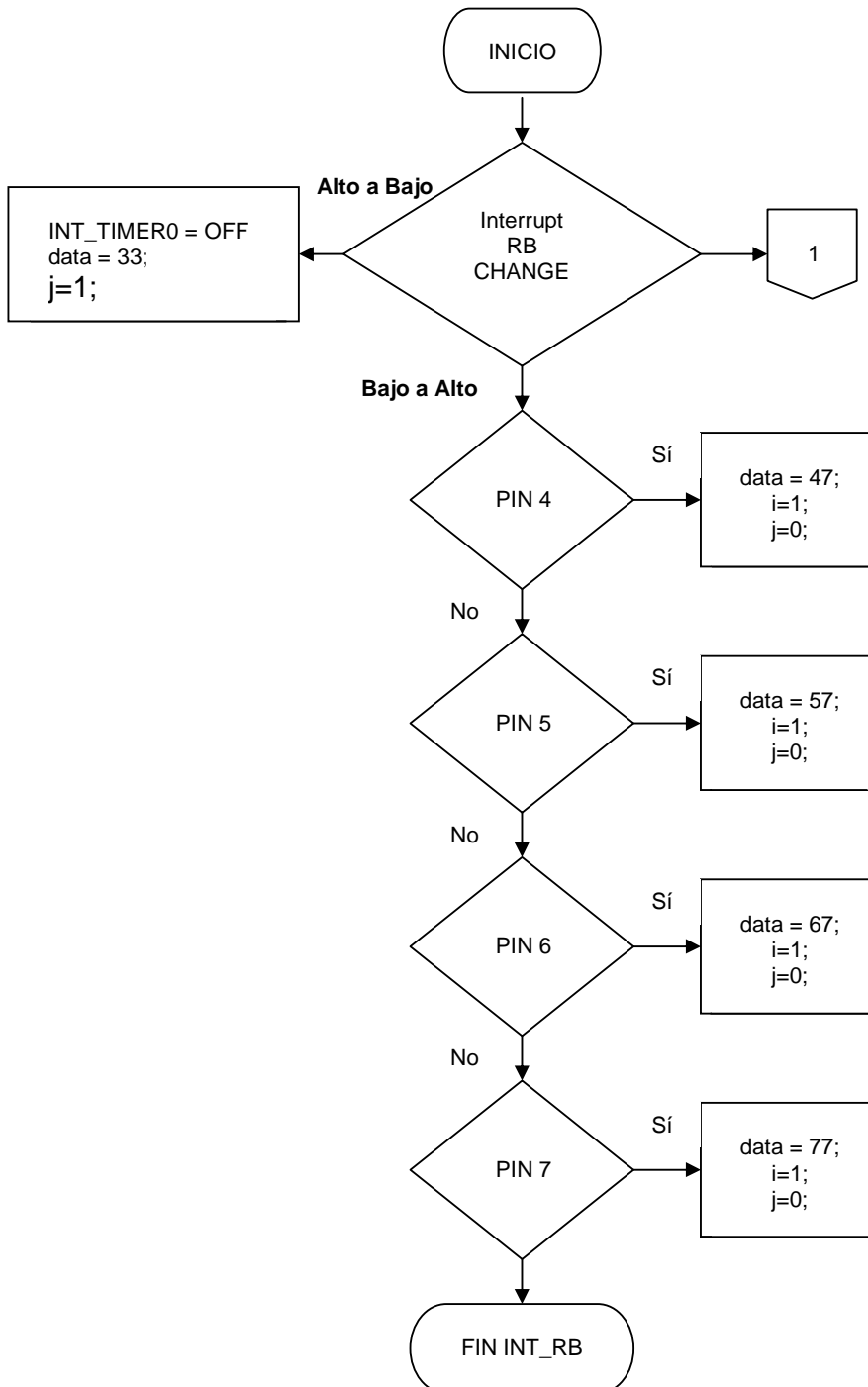
VIII. BIBLIOGRAFÍA

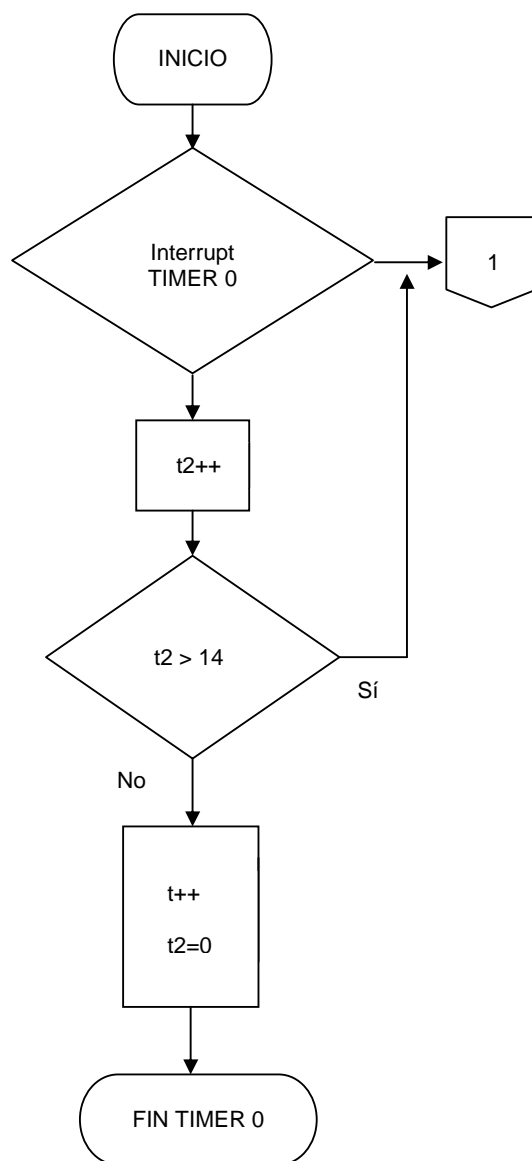
1. Caprari, Gilles. 2003. *Autonomous Micro-Robots: Applications and Limitations*. Tesis École Polytechnique Fédérale de Lausanne. Lausanne, Suiza.
2. GillesCaprari, Masoud Asadpour, Fabien Tâche, Alice oficial web site, [Citado em junio 2007],
<http://lis.epfl.ch/index.html?content=robots.php>
3. Marcus, Abraham; Marcus, William. 1965. *Elements of Radio*. Pag.49
4. *Motor CC-Motor de Corriente Continua, "Principio básico de funcionamiento de un motor cc."*,
http://www.unicrom.com/Tut_Motor.asp
5. *National Instruments, "Comunicación Serial: Conceptos Generales"*, [Citado en junio 2008]
<http://digital.ni.com/public.nsf/allkb/039001258CEF8FB686256E0F005888D1#Serial>
6. *Robots Fijos*, [Citado en septiembre 2010], <http://robotica-biri.blogspot.com/2009/03/tipos-de-robots.html>
7. *Scara, Robots*, [Citado en septiembre 2010]
<http://www.directindustry.com/prod/hirata-america/scara-robot-22536-216130.html>
8. *Sistemas de distribución de energía eléctrica*, [Citado en junio 2010],
<http://www.freewebs.com/mecatronica/electronicaindustrial.ht>
9. Stremler, Ferrel. 1989. *Sistemas de Comunicación*. Edición en español. México, Fondo Educativo Interamericano, S.A Pag.205, 206.
10. *Tecnología de Comunicación de Datos, "Modulación en Amplitud"*, [Citado en junio 2008]
<http://www.ie.itcr.ac.cr/marin/telematica/tcd/EIS/2.ModulaciondeAmplitud.pdf>
11. *Telecomunicación Digital*. 1988. SIEMENS. Berlín.
12. *Tienda de robótica*, [Citado en septiembre 2010], <http://robotica.com/iSOBOT.asp>
13. *Tiny page motor*, [Citado em octubre 2008]
http://www.solarbotics.com/robot_kits/
14. *US Frequency Allocation Chart*, [Citado en abril 2008],
<http://www.fcc.gov/>

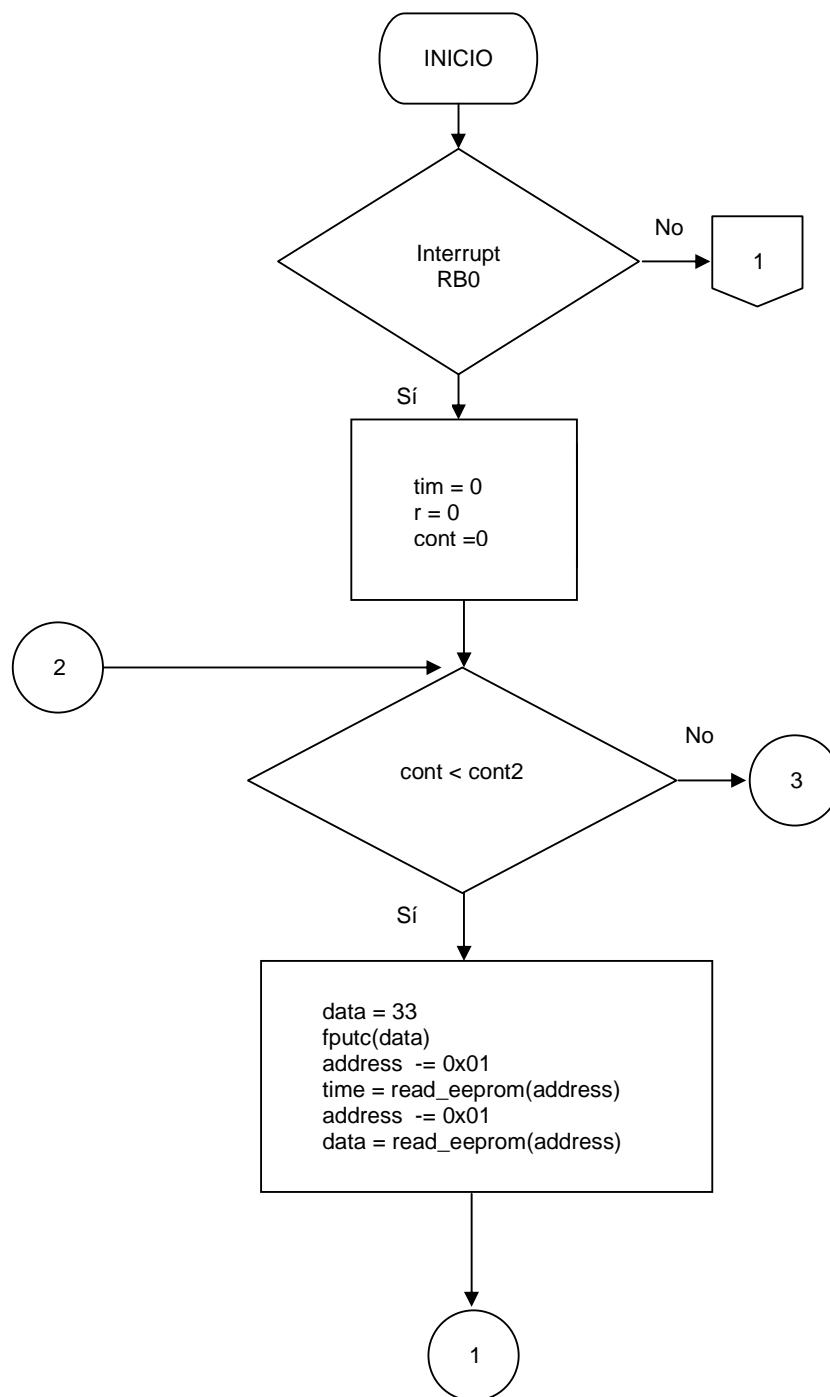
IX. APÉNDICE

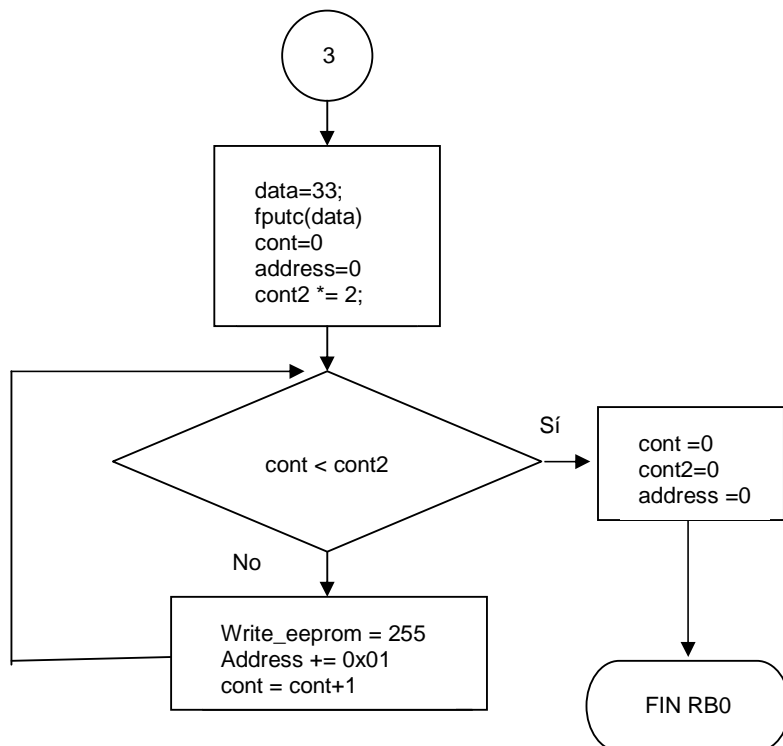
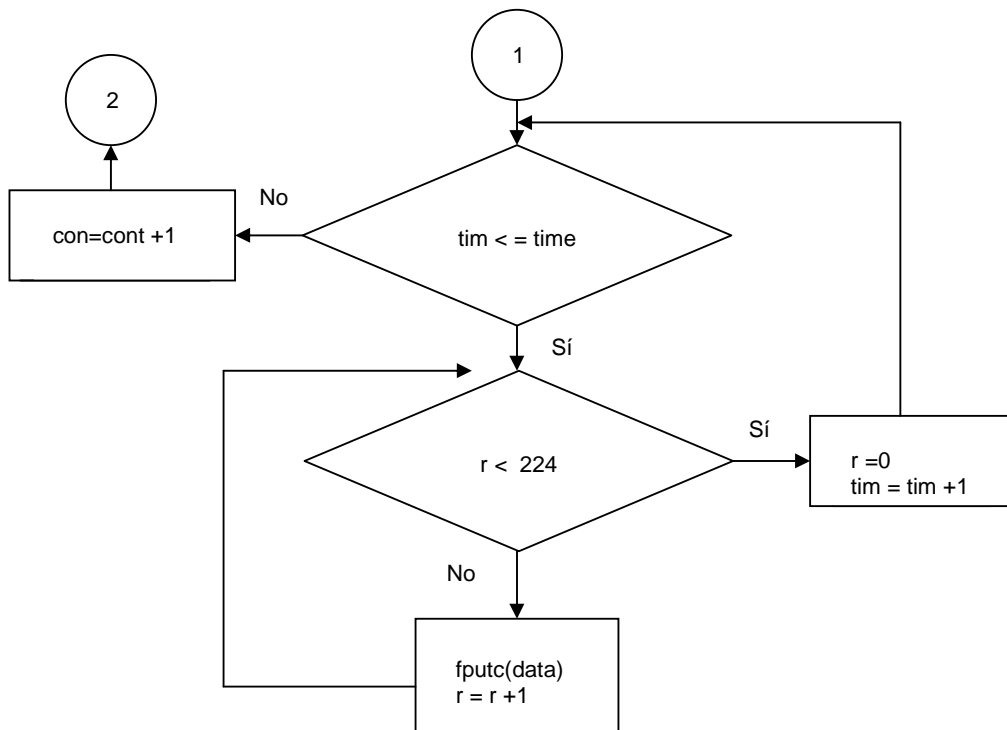
A. Diagramas de flujo de los softwares

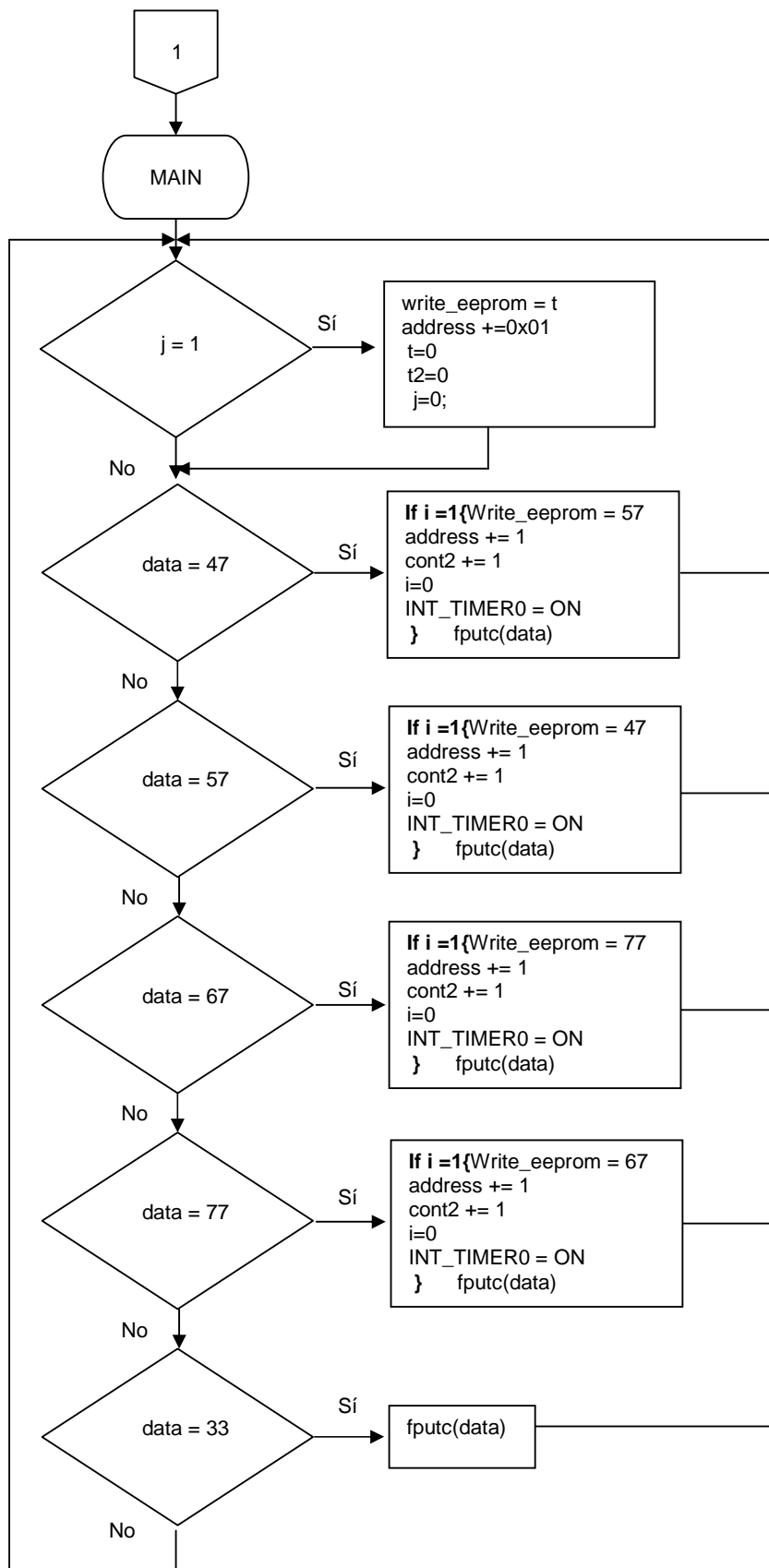
1. Diagrama del transmisor:



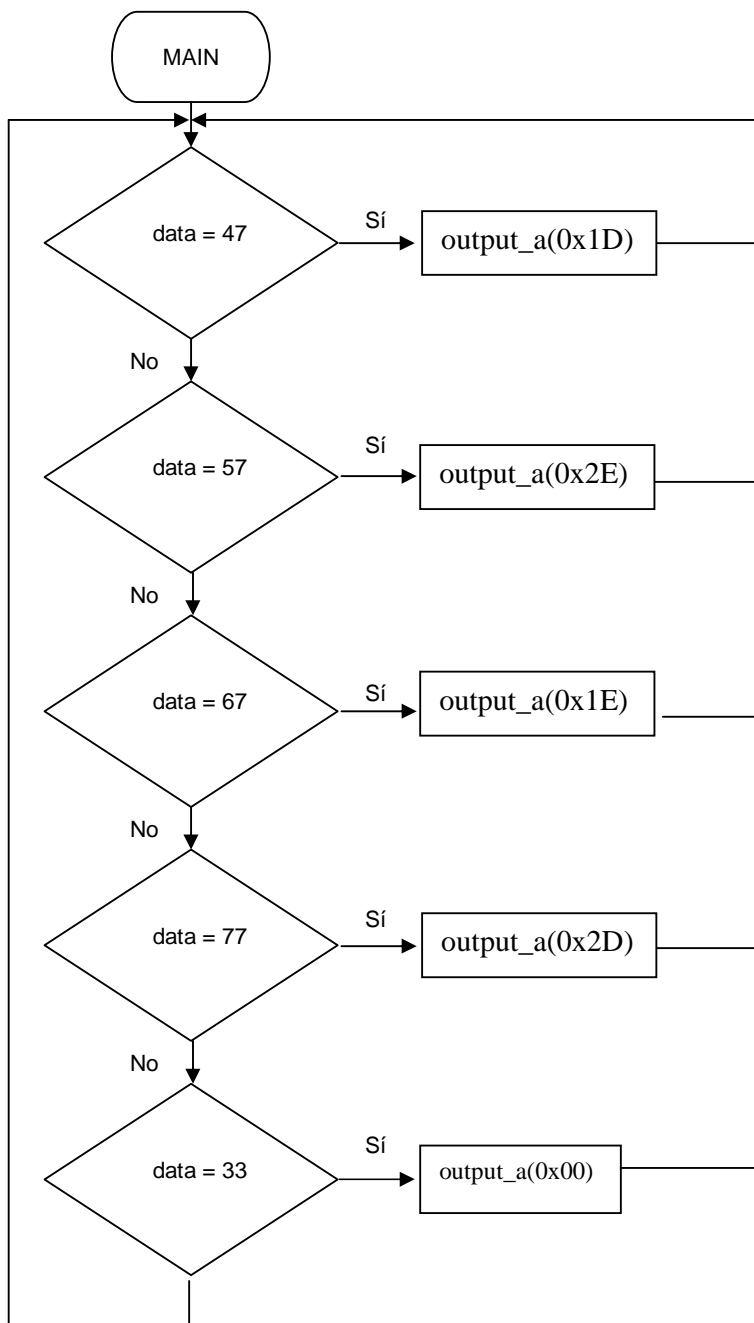




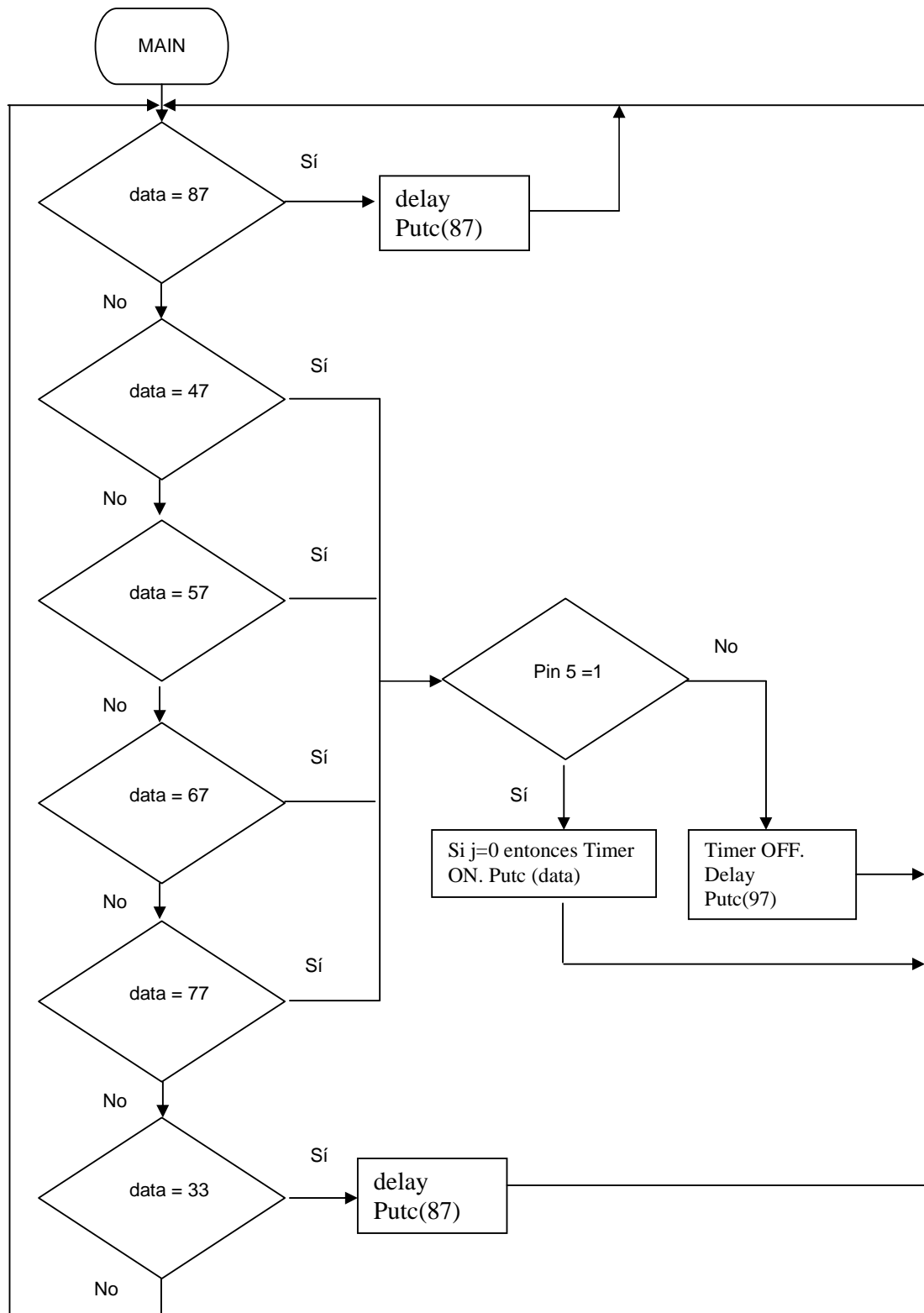




1. Diagrama del receptor:



2. Diagrama del diseño experimental:



3. Programas

```

/*****
*
*          PROGRAMA DE CONTROL REMOTO PARA
*          MICRO ROBOT
*          RECEPTOR
*
* Especificaciones:
*   - Receptor serial por RF. PIN A3 Rx
*   - Baud: 4800 con reloj de 8MHz interno. 8 bits
*   - Ejecuta los cuatro movimientos (AD, AT, DR, IZ).
*   - PIC 12F683
*   - Módulo de Rx LINX de 433MHz
*
* Autor: Antonio Rodil Almenara
* Carné: 01127
* Para: Trabajo de Tesis. Ing. Electrónica, Universidad del Valle de Guatemala.
*
* Versión: 5.0 Elaborado en PIC C compiler.
*****/

void main()          /* Rutina Principal del programa. */
{

    char data, code;
    int1 flag;

    /* Explicación de Variables:

        Variable "data" recibe el movimiento.
        Variable "code" recibe el código de seguridad.
        Variable "flag" bandera para ciclo de ejecución de movimiento.
    */

    port_a_pullups(TRUE);          // Sección de habilitación y deshabilitación de funciones.
    setup_adc_ports(NO_ANALOGS|VSS_VDD);
    setup_adc(ADC_OFF);
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_comparator(NC_NC);
    setup_vref(FALSE);
    setup_oscillator(OSC_8MHZ);
    output_a(0x00);

    do{                          // Ciclo "infinito".

    if(kbhit())                   // Condición sujeto a kbhit(), función que determina
    {                              // La recepción de datos en el PIN A3 del PIC.
        code = fgetc(QVF);        // Lo primero que debería detectar es el código de seguridad.
        if(code == '$')          // Evalúa trama para verificar código.
        {
            flag = 1;
            do{                    // Ciclo de ejecución de movimiento.
                if(kbhit())        // La segunda trama debería de ser la instrucción de movimiento.
                {
                    data= fgetc(QVF);
                    switch (data) // Evalúa el movimiento.
                    {

                        case 47:{ // ADELANTE.

```

```

        output_a(0x1D); }
        break;

    case 57:{          // ATRAS.
        output_a(0x2E); }
        break;

    case 67:{          // DERECHA.
        output_a(0x1E); }
        break;

    case 77:{          // IZQUIERDA.
        output_a(0x2D); }
        break;

    case 33:{          // Salidas a 0. No hacer nada.
        output_a(0x00); }
        break;
    }
    flag = 0;
    code = 0;

}

}while(flag==1);
}
}while(1);
}

/*****
*
*          PROGRAMA DE CONTROL REMOTO PARA
*          MICRO ROBOT
*          TRANSMISOR
*
* Especificaciones:
*   - Transmisión serial por RF. PIN A3 Tx
*   - Baud: 4800 con reloj de 4MHz interno. 8 bits
*   - Ejecuta los cuatro movimientos (AD, AT, DR, IZ).
*   - Graba el trayecto inverso para que el robot regrese a su punto de
*     partida.
*   - Modalidad de uso: "joystick" con "push botton" al centro.
*   - PIC 16F88
*   - Módulo de Tx LINX de 433MHz
*
* Autor: Antonio Rodil Almenara
* Carné: 01127
* Para: Trabajo de Tesis. Ing. Electrónica, Universidad del Valle de Guatemala.
*
* Versión: 8.0 Elaborado en PIC C compiler.
*****/

#define HIGHTOLOW TRUE          // Variable que detecta cambio en Port Change de alto a bajo.
#define LOWTOHIGH TRUE         // Variable que detecta cambio en Port Change de bajo a alto.

char data, code='$';
int1 i, j;
int cont=0, cont2=0, address=0x00;
int t=0, t2=0, r,tim, time;

```

/* Explicación de Variables:

Variable "i" bandera de grabado de movimiento en EEPROM.
 Variable "j" bandera de grabado de tiempo en EEPROM.
 Variable "data" para transmisión de movimiento.
 Variable "code" para seguridad en la transmisión de datos.
 Variable "cont" contador general.
 Variable "cont" contador general.
 Variable "cont2" contador de veces que se efectúa un movimiento.
 Variable "address" dirección de memoria EEPROM.
 Variable "t" guarda los segundos que se tarda un movimiento.
 Variable "t2" cuenta los overflows para llegar al segundo.
 Variable "r" contador del índice de resolución.
 Variable "tim" contador auxiliar de los segundos.
 Variable "time" guarda el tiempo grabado en la EEPROM, para su ejecución.

*/

```
#int_RB          /* Rutina para el interrupto RB0 INT. */
RB_isr()
{
  int current;          // Variable "current" guarda el valor actual del Puerto B.
  static int last=0;    // Variable "last" guarda el valor anterior del Puerto B.

  current=input_b();

  #if LOWTOHIGH        // Cambio estado bajo-alto en Port Change (joystick accionado).
  if (!(bit_test(last,4)&&(bit_test(current,4)))
  {
    data = 47;          // Si el movimiento es Adelante.
    i=1;
    j=0;
  }
  if (!(bit_test(last,5)&&(bit_test(current,5)))
  {
    data = 57;          // Si el movimiento es Atrás.
    i=1;
    j=0;
  }
  if (!(bit_test(last,6)&&(bit_test(current,6)))
  {
    data = 67;          // Si el movimiento es Derecha.
    i=1;
    j=0;
  }
  if (!(bit_test(last,7)&&(bit_test(current,7)))
  {
    data = 77;          // Si el movimiento es Izquierda.
    i=1;
    j=0;
  }
  #endif
  #if HIGHTOLOW        // Cambio estado alto-bajo en Port Change (joystick no accionado).

  if (!(bit_test(current,4)&&(bit_test(last,4)))
  {
    disable_interrupts(INT_TIMER0);
    data = 33;
    j=1;
  }          // Al soltar joystick se cumple cualquiera de estas condiciones.

  if (!(bit_test(current,5)&&(bit_test(last,5)))
```

```

{
    disable_interrupts(INT_TIMER0);
    data = 33;
    j=1;
}

if (!(bit_test(current,6))&&(bit_test(last,6)))
{
    disable_interrupts(INT_TIMER0);
    data = 33;
    j=1;
}

if (!(bit_test(current,7))&&(bit_test(last,7)))
{
    disable_interrupts(INT_TIMER0);
    data = 33;
    j=1;
}

#endif

    last=current;
}

#int_EXT                               /* Rutina para ejecución de trayecto inverso. */
EXT_isr()
{
    delay_ms(450);
    tim=0;
    cont=0;
    r=0;

    data = 87;                          //Código exclusivo para el diseño Experimental.
    fputc(code,QVF);                    //Código exclusivo para el diseño Experimental.
    fputc(data,QVF);                    //Código exclusivo para el diseño Experimental.
    data = 0;

    delay_ms(420);

    while(cont<cont2) // Ciclo de lectura de movimiento y tiempo
    {
        data = 33;                      //Ayuda a cambiar de movimientos, con reset en el
        fputc(code,QVF);                 //temporizador.
        fputc(data,QVF);
        delay_ms(220);

        address -=0x01;
        time = read_eeprom(address);

        address -=0x01;
        data = read_eeprom(address);

        for(tim=0;tim<=time;tim++)      // Ciclo de ejecución de movimiento.
        {
            for(r=0;r<224;r++)           // Ciclo auxiliar de ejecución de movimiento con
            {                             // índice de resolución.
                fputc(code,QVF);         // Envío de datos.
                fputc(data,QVF);
            }
        }
    }
}

```

```

    cont++;
}

data=33;           // Inicializa los valores luego de ejecución de trayecto inverso.
fputc(code,QVF);
fputc(data,QVF);
cont=0;
address=0;
cont2 *= 2;       // "cont2" por dos para tomar en cuenta la posición de grabado de
                  // tiempo en la EEPROM.

for(cont=0; cont<cont2;cont++) // Ciclo de borrado de la EEPROM luego de ejecución de trayecto.
{
    write_eeprom(address,255);
    address+=0x01;
}
cont=0;           // Inicialización de variables.
cont2=0;
address=0;
}

#endif

/* Rutina de interrupción Timer0. */
TIMER0_isr()
{
    t2++;
    if(t2>14)     // Condición para llegar al segundo.
    {
        t++;      // Incremento de segundos.
        t2=0;
    }
}

}

void main(void)  // Rutina Principal del programa. */
{
    setup_adc_ports(NO_ANALOGS|VSS_VDD); // Sección de habilitación y deshabilitación de
funciones.
    setup_adc(ADC_OFF);
    setup_spi(FALSE);
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_256);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);
    enable_interrupts(INT_RB);
    enable_interrupts(INT_EXT);
    enable_interrupts(GLOBAL);
    setup_oscillator(OSC_4MHZ|OSC_INTRC);

do // Ciclo "infinito" para transmisión de datos.
{
    if(j==1) // Si "joystick" inactivo, graba tiempo de movimiento.
    {
        write_eeprom(address,t);
        address +=0x01;
        t=0;
        t2=0;
        j=0;
    }
}
}

```

```

}
switch(data) // Examina "variable" data para transmitir movimiento.
{
    case 47: { // ADELANTE.
        if(i==1) // Graba una vez el movimiento contrario en EEPROM.
        {
            write_eeprom(address,57);
            address += 0x01;
            cont2++;
            i=0;
            enable_interrupts(INT_TIMER0);
        }
        fputc(code,QVF);
        fputc(data,QVF);
    }
    break;

    case 57: { // ATRAS.
        if(i==1)
        {
            write_eeprom(address,47);
            address += 0x01;
            cont2++;
            i=0;
            enable_interrupts(INT_TIMER0);
        }
        fputc(code,QVF);
        fputc(data,QVF);
    }
    break;

    case 67: { // DERECHA.
        if(i==1)
        {
            write_eeprom(address,77);
            address += 0x01;
            cont2++;
            i=0;
            enable_interrupts(INT_TIMER0);
        }
        fputc(code,QVF);
        fputc(data,QVF);
    }
    break;

    case 77: { // IZQUIERDA.
        if(i==1)
        {
            write_eeprom(address,67);
            address += 0x01;
            cont2++;
            i=0;
            enable_interrupts(INT_TIMER0);
        }
        fputc(code,QVF);
        fputc(data,QVF);
    }
}

```

```
    }  
    break;  
  
    case 33: { fputc(code,QVF);      // NO HACER NADA, código 33 jostick inactivo.  
              fputc(data,QVF); }  
    break;  
  
    }  
}while(1);  
}
```

```

/*****
*
*          PROGRAMA DEL DISEÑO EXPERIMENTAL
*
*
* Especificaciones:
*
*   - Receptor serial por RF. PIN A2 Rx
*   - Baud: 4800 con reloj de 20MHz. 8 bits
*   - Ejecuta los cuatro movimientos y código que
*     detecta el "push botton" del Tx (AD, AT, DR, IZ, 87).
*   - PIC 16F84A
*   - Módulo de Rx LINX de 433MHz
*
*
* Autor: Antonio Rodil Almenara
* Carné: 01127
* Para: Trabajo de Tesis. Ing. Electrónica, Universidad del Valle de Guatemala.
*
* Versión 5.0 Elaborado en PIC C compiler.
*****/

void main()
{
    char data, code='$';
    int1 flag, i=0;

    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);

do{
    // Ciclo "infinito".

if(kbhit())
    // Condición sujeto a kbhit(), función que determina
    // la recepción de datos en el PIN A3 del PIC.
    {
        code = fgetc(QVF); // Lo primero que debería detectar es el código de seguridad.
        if(code == '$') // Evalúa trama para verificar código.
            {
                flag = 1;
                do{
                    // Ciclo de ejecución de movimiento.
                    if(kbhit()) // La segunda trama debería de ser la instrucción de movimiento.
                        {
                            data= fgetc(QVF);
                            switch (data) // Evalúa el movimiento.
                                {

                                    case 87:{ //Código exclusivo para el Diseño Experimental.
                                        putc(data);
                                        i=0;

                                        }
                                        break;

                                    case 47:{ // ADELANTE.

                                        putc(data);
                                        i=0;
                                        }
                                        break;

                                    case 57:{ // ATRÁS.
                                        putc(data);
                                        i=0;
                                        }
                                        break;
                                }
                        }
            }
    }
}

```

```
        case 67:{          // DERECHA.
            putc(data);
            i=0;

            }
            break;

        case 77:{          // IZQUIERDA.
            putc(data);
            i=0;

            }
            break;
    }
    flag = 0;
    code = 0;
}
}while(flag==1);
}
}
if(i==0 && data==33)      // Al terminar cada transmisión al serial de
{                          // la computadora, manda el código 33 para que el
    delay_ms(500);        // de Visual Basic avance una posición en una matriz.
    putc(33);
    i=1;
}
}while(1);
}
```

```

;-----;
;   TEMPORIZADOR PARA DISEÑO EXPERIMENTAL.   ;
; Especificaciones:                           ;
;   - PIC16F84A                               ;
;   - Cronómetro de segundos. Toma hasta minuto y ;
;     medio con 9 segundos, es decir 99 segundios. ;
;   - Reloj de 4MHz. Resonador.               ;
;-----;
; Versión 3.4.1 Elaborado en MPLAB 8.00      ;
; Antonio Rodil Almenara 01127              ;
;-----;

```

```

list    p=16F84A      ; Definición de las directivas
#include <p16F84A.inc> ; a usar.

```

```

__CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON & _HS_OSC

```

```

;__CONFIG Es para definir el estado de ciertos registros. Para más
;detalle consultar el manual del PIC 16F84A.

```

```

;-----;
; Definición de variables a usar.
;-----;

```

```

w2    EQU    0x0C      ; Acumulador auxiliar 2.
w3    EQU    0x0D      ; Acumulador auxiliar 3.
status_temp EQU    0x0E ; Temporal de STATUS.

```

```

cblock 0x10
    seg1      ; Unidades de segundo.
    seg10     ; Decenas de segundo.
    maska     ; Máscara, selecciona bits. PORTA
    maskb     ; Máscara, selecciona bits. PORTB
    w_temp    ; Temporal de WREG.
    i         ; Acumulador para ajuste del segundo.
endc

```

```

;-----;
; Comienzo del programa.
;-----;

```

```

ORG    0x000
    goto main

```

```

ORG 0x004
    movwf w_temp
    movf  STATUS,0
    movwf status_temp
    btfsc INTCON,2      ; Chequeo de bandera TOIF.
    call  ISR
    btfsc INTCON,0      ; Chequeo de bandera RBIF.
    call  ISR2
    bcf   INTCON, 2     ; Bandera TOIF=0 en caso de que no se incremento seg1.

    movf  status_temp,0
    movwf STATUS
    movf  w_temp, 0
    bcf   INTCON,0     ; Limpia bandera RBIF.
    bsf   INTCON,3     ; Habilita RB PORT Change después de que ocurrió.

    retfie

```

```

;-----
;Inicialización de variables.
;-----

```

```

inicializa
    clrf seg1           ; Puesta a cero las variables.
    clrf seg10
    clrf w2
    clrf w3
    clrf i
    clrf maska
    clrf maskb

    movlw B'11'        ; Asignación de valor a maska.
    addwf maska
    movlw B'1111100'   ; Asignación de valor a maskb.
    addwf maskb
    movlw .0           ; Inicilización de segundos.
    movwf seg1
    movlw .0
    movwf seg10
    movlw .0
    movwf i
    clrw

    bsf STATUS,5       ; Selección de Banco 1.
    bsf OPTION_REG,7   ; Pull-ups disable, PORTB
    bcf OPTION_REG,5   ; Ciclo de instrucción de reloj interno.
    bcf OPTION_REG,3

    bsf OPTION_REG,0   ; Set del Prescaler a 1:256
    bsf OPTION_REG,1
    bsf OPTION_REG,2

    movlw .96          ; Selección puertos RB<0..4> salidas.
    movwf TRISB        ; RB<5..6> entrada. RB 7 Don't Care.
    movlw .0           ; Selección puertos RA<0..4> salidas.
    movwf TRISA

    bcf STATUS,5       ; Selección de Banco 0.
    clrf PORTA
    clrf PORTB
    bsf INTCON,3       ; Interrupto RB Port Change habilitado.
    bcf INTCON,4       ; Interrupto RB0/INT deshabilitado.
    bcf INTCON,5       ; Interrupto TMR0 habilitado.
    bsf INTCON,7       ; Interrupto Global habilitado.

    retfie

```

```

;-----
;Conversión de Tabla de 7SEG, para el Display.
;-----
TABLA7SEG  ;WREG Es Parámetro de Entrada.

    addwf PCL, F ;g f e d c b a
    retlw B'0111111';0
    retlw B'0000110';1
    retlw B'1011011';2
    retlw B'1001111';3
    retlw B'1100110';4
    retlw B'1101101';5
    retlw B'1111100';6
    retlw B'0000111';7
    retlw B'1111111';8
    retlw B'1100111';9

;-----
; Rutinas para el cambio de segundos.
;-----
; Evalua las decenas de segundo.
;-----
evalseg2

    clrf    seg1
    movlw  .1          ; Aumenta 1 las decenas de segundo y limpia las unidades de segundo.
    addwf  seg10,1
    movlw  .10
    subwf  seg10,0    ; Si seg10 = 6 evalúa las unidades de minuto y se pone en cero seg10.
    btfsc STATUS,2
    clrf   seg10

    return

;-----
; Evalua las unidades de segundo.
;-----
evalseg

    clrf    i          ; Se borra el acumulador de ajuste de segundo para el TMR0 overflow.
    movlw  .1          ; Aumenta 1 las decenas de minuto y limpia las unidades de minuto.
    addwf  seg1,1
    movlw  .10
    subwf  seg1,0    ; Si seg1 = 10 evalúa las decenas de segundo y se pone en cero seg1.
    btfsc STATUS,2
    goto  evalseg2

    return

;-----
; Rutina para perder tiempo.
;-----
delay
    movwf w2

delay0
    decfsz w2,.1
    goto  delay0
    return

```

```

;-----
; Rutina que ajusta los bits para desplegarlo en el Puerto A. RA<3..4>
;-----
ajusteA
    movwf w2
    andwf maska,0
    movwf w3
    rlf    w3,1
    rlf    w3,1
    movf  w3,0
    retfie

;-----
; Rutina que ajusta los bits para desplegarlo en el Puerto B. RB<0..4>
;-----
ajusteB

    movf  w2,0
    andwf maskb,0
    movwf w3
    rrf    w3,1
    rrf    w3,1
    movf  w3,0
    iorlw .96
    retfie

;-----
; Rutina para volver a cero el minicronómetro.
;-----
vueltacero

    clrf  seg1
    clrf  seg10

    return

;-----
; Rutina que despliega los valores en los displays 7SEG.
;-----
output

    movf seg1,w           ; Despliegue de las Unidades de Segundo.
    call TABLA7SEG
    call ajusteA

    iorlw .1             ; Termina de ajustar para display de unidades de segundo.
    movwf PORTA
    call  ajusteB
    movwf PORTB

    movlw .255
    call  delay
    call  delay

    clrf PORTA
    clrf PORTB

    movf seg10,w         ; Despliegue de las Decenas de Segundo.
    call TABLA7SEG
    call ajusteA

    iorlw .2             ; Termina de ajustar para display de decenas de segundo.
    movwf PORTA

```

```

call  ajusteB
      movwf PORTB

      movlw .255
      call  delay
      call  delay

      clrf PORTA
      clrf PORTB

      movlw .255
      call  delay
      call  delay

      goto  output

```

```

;-----
; Rutina de interrupcion para el TMR0.
;-----

```

```

ISR
      incf  i,1          ; Ajuste de 1 segundo. Incrementara después de un overflow.
      movlw .18         ; Cálculo, cuantos overflow antes de un segundo lo más exacto.
      subwf i,0
      btfsc STATUS, 2  ; Si llegó a 16 entonces empieza la evaluación de la hora y la salida
      goto  exitISR    ; del ISR.
      return

```

```

exitISR
      call evalseg
      bcf  INTCON, 2    ; Limpia la bandera del TMRO.

      return

```

```

;-----
; Rutina de interrupcion para el RB<5..7>/Port Change.
;-----

```

```

ISR2

      bsf  INTCON,5     ; Interrupto TMR0 habilitado.
      btfss PORTB,5    ; Verificación de RB Port Change.
      bcf  INTCON,5     ; de bajo a alto.
      btfss PORTB,5
      call vueltacero  ; Si fue de alto a bajo, vuelve a cero
                       ; los displays.
                       ; Para esto se pone un pull down en circuito
                       ; aparte de deshabilitar los pull up's internos.

      return

```

```

;-----
; Menu Principal
;-----
main

```

```

      call inicializa
      call output

```

```

END

```

Programa Visual Basic. Interfaz con computadora

```

Public Class Form1

    Dim b As Boolean
    Dim num As Integer
    Dim c As Integer
    Dim j As Integer

    //Objeto Forma. Plantilla de pantalla principal.

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        For i As Integer = 0 To _
            My.Computer.Ports.SerialPortNames.Count - 1
            ComboBox1.Items.Add(My.Computer.Ports.SerialPortNames(i))
        Next
        b = True
        With Label1
            .Font = New Font("Arial", 12.0!, FontStyle.Bold)
            .Text = "ADELANTE"
            .ForeColor = Color.DarkBlue
        End With
        With Label2
            .Font = New Font("Arial", 12.0!, FontStyle.Bold)
            .Text = "ATRAS"
            .ForeColor = Color.DarkBlue
        End With
        With Label3
            .Font = New Font("Arial", 12.0!, FontStyle.Bold)
            .Text = "DERECHA"
            .ForeColor = Color.DarkBlue
        End With
        With Label4
            .Font = New Font("Arial", 12.0!, FontStyle.Bold)
            .Text = "IZQUIERDA"
            .ForeColor = Color.DarkBlue
        End With
        With Label5
            .Font = New Font("Arial", 12.0!, FontStyle.Bold)
            .Text = "ADELANTE REPRODUCCION"
            .ForeColor = Color.DarkGreen
        End With
        With Label6
            .Font = New Font("Arial", 12.0!, FontStyle.Bold)
            .Text = "ATRAS REPRODUCCION"
            .ForeColor = Color.DarkGreen
        End With
        With Label7
            .Font = New Font("Arial", 12.0!, FontStyle.Bold)
            .Text = "DERECHA REPRODUCCION"
            .ForeColor = Color.DarkGreen
        End With
        With Label8
            .Font = New Font("Arial", 12.0!, FontStyle.Bold)
            .Text = "IZQUIERDA REPRODUCCION"
            .ForeColor = Color.DarkGreen
        End With
        With Label9
            .Font = New Font("Arial", 10.0!, FontStyle.Bold)
            .Text = "Seleccionar Puerto."
        End With
    End Sub
End Class

```

```
        .ForeColor = Color.Black
    End With
    With ProgressBar1
        .Visible = True
        .Minimum = 0
        .Maximum = 300
        .Step = 1
    End With
    With ProgressBar2
        .Visible = True
        .Minimum = 0
        .Maximum = 300
        .Step = 1
    End With
    With ProgressBar3
        .Visible = True
        .Minimum = 0
        .Maximum = 300
        .Step = 1
    End With
    With ProgressBar4
        .Visible = True
        .Minimum = 0
        .Maximum = 300
        .Step = 1
    End With
    With ProgressBar5
        .Visible = True
        .Minimum = 0
        .Maximum = 300
        .Step = 1
    End With
    With ProgressBar6
        .Visible = True
        .Minimum = 0
        .Maximum = 300
        .Step = 1
    End With
    With ProgressBar7
        .Visible = True
        .Minimum = 0
        .Maximum = 300
        .Step = 1
    End With
    With ProgressBar8
        .Visible = True
        .Minimum = 0
        .Maximum = 300
        .Step = 1
    End With

End Sub

Private Sub Form1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles Me.Paint
    e.Graphics.DrawLine(Pens.Gray, 25, 183, 720, 183)
    e.Graphics.DrawLine(Pens.Gray, 25, 375, 720, 375)
    e.Graphics.DrawLine(Pens.Gray, 25, 563, 720, 563)
End Sub

Private Sub Form1_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
    SerialPort1.Close()
```

```
End Sub
```

```
//Objeto Serial. Controla el puerto Serial.
```

```
Private Sub SerialPort1_DataReceived(ByVal sender As Object, ByVal e
As System.IO.Ports.SerialDataReceivedEventArgs) Handles
SerialPort1.DataReceived
```

```
    c = SerialPort1.ReadChar()
    If c = 97 Then
        SerialPort1.DiscardInBuffer()
        SerialPort1.Close()
        MsgBox("ALERTA: Miro Robot dañado!! No se detecta movimiento!!")
        MsgBox("Presione RESET y luego START para comenzar otravez.")
        c = 0
    End If
    If c = 87 Then
        j = c
        SerialPort1.DiscardInBuffer()
        c = 0
    End If
End Sub
```

```
//Objeto Timer 1. Controla el tiempo del recorrido inicial.
```

```
Private Sub Timer1_Tick(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
```

```
    If j = 87 Then
        Timer1.Stop()
        Timer2.Start()
        j = 0
    End If

    If c = 47 Then
        ProgressBar1.PerformStep()
        SerialPort1.DiscardInBuffer()
    End If

    If c = 57 Then
        ProgressBar2.PerformStep()
        SerialPort1.DiscardInBuffer()
    End If

    If c = 67 Then
        ProgressBar3.PerformStep()
        SerialPort1.DiscardInBuffer()
    End If

    If c = 77 Then
        ProgressBar4.PerformStep()
        SerialPort1.DiscardInBuffer()
    End If
    c = 0
    j = 0
End Sub
```

```

//Objeto Timer 2. Controla el recorrido del recorrido inverso.
Private Sub Timer2_Tick(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Timer2.Tick

    If c = 47 Then
        ProgressBar5.PerformStep()
        SerialPort1.DiscardInBuffer()
    End If

    If c = 57 Then
        ProgressBar6.PerformStep()
        SerialPort1.DiscardInBuffer()
    End If

    If c = 67 Then
        ProgressBar7.PerformStep()
        SerialPort1.DiscardInBuffer()
    End If

    If c = 77 Then
        ProgressBar8.PerformStep()
        SerialPort1.DiscardInBuffer()
    End If

    c = 0
End Sub

//Objeto Botón de START.

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

    ProgressBar1.Maximum = 0
    ProgressBar2.Maximum = 0
    ProgressBar3.Maximum = 0
    ProgressBar4.Maximum = 0
    ProgressBar5.Maximum = 0
    ProgressBar6.Maximum = 0
    ProgressBar7.Maximum = 0
    ProgressBar8.Maximum = 0
    ProgressBar1.Maximum = 300
    ProgressBar2.Maximum = 300
    ProgressBar3.Maximum = 300
    ProgressBar4.Maximum = 300
    ProgressBar5.Maximum = 300
    ProgressBar6.Maximum = 300
    ProgressBar7.Maximum = 300
    ProgressBar8.Maximum = 300

    Timer2.Stop()
    Timer1.Start()
End Sub

//Objeto Botón de Selección de Puerto COM1.

```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    If ComboBox1.Text <> "COM1" Then
        MsgBox("ERROR: Debe seleccionar un puerto!!")
    Else
        With SerialPort1
            .Close()
            .PortName = ComboBox1.Text
            .BaudRate = 4800
            .Parity = IO.Ports.Parity.None
            .DataBits = 8
        End With
        SerialPort1.Open()
        Timer2.Interval = 190
        Timer1.Interval = 190
        Timer2.Stop()
        Timer1.Start()
    End If
End Sub

//Objeto Botón de STOP.

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    SerialPort1.Close()
    Timer1.Stop()
    Timer2.Stop()
End Sub
End Class
```