



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

TESIS DE GRADO

**“DISEÑO E IMPLEMENTACIÓN DE UN CONTROLADOR REMOTO VÍA
RADIOFRECUENCIA PARA UN BRAZO ROBÓTICO DE USO DIDÁCTICO
CON TRES GRADOS DE LIBERTAD”**

Previa a la obtención del título de:

**INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN INDUSTRIAL
INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

PRESENTADA POR:

**JORGE LUIS MATAMOROS TINGO
DIEGO PATRICIO CORDERO ZAPATA**

GUAYAQUIL – ECUADOR

2007

Dedicatoria

*Al aire que mueve mis alas
Al espíritu que me da la fuerza para caminar
A su incansable esfuerzo y dedicación
A su inagotable fe en la creación divina
Aun en los momentos adversos, presentes en todos los tiempos
A ella que inspira mi mente, mi fuerza y el surgir de mi fe
Al cielo que siempre ve la luz de sus ojos
A su mano amiga siempre extendida que no duda en dar
Al agua en el desierto que es su alma para mi
Y sin fin... a mi madre
Un ángel en mi vida
Una amiga un amor
Un eterno corazón
Mi madre Ana Maria Tingo*

Agradecimientos

*A todos los que
contribuyeron de manera directa o indirecta en el
desarrollo de este trabajo y de manera especial al Ing.
Efrén Herrera nuestro director
por su valiosa guía en todas las etapas del proyecto
y a todos mis maestros que no dudan en
preparar a las generaciones futuras sin esperar nada
a todos los mencionados y los que se quedan
en mis lagunas mentales.*

Gracias Totales

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesis de Grado, nos corresponde Exclusivamente; y el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral”

(Reglamento de exámenes y títulos profesionales de la ESPOL)

Jorge Luis Matamoros Tingo

Diego Cordero Zapata

TRIBUNAL DE GRADUACIÓN

PRESIDENTE

Msc. Holger Cevallos Ulloa

DIRECTOR DE TESIS

Msc. Efrén Herrera

MIEMBROS PRINCIPALES

Msc. Cesar Martín

Dr. Boris Vintimilla

RESUMEN

Desarrollo de un sistema de control inalámbrico para el control del brazo robot Lynx 5 Arm de la marca Lynxmotion. Los movimientos del brazo robot son controlados por un usuario desde una PC y a su vez enviados mediante el puerto serie (com1) a través de un cable con conectores DB9 hembra / macho que se conectan a la tarjeta que envía los datos al transmisor de radio frecuencia RXM-433-LCS. Los datos son recibidos por el receptor inalámbrico TXM-433-LCS el cual los recibe para ejecutar los movimientos deseados por el usuario. Este puede controlar el brazo robot mediante tres pantallas que se encuentran en el menú principal y salir de ellas en cualquier momento volviendo al menú principal y pudiendo escoger cualquiera de las tres aplicaciones mostradas allí. En las tarjetas que controlan al brazo robot se muestra el estado del puerto serie mediante leds que indican si se permite la transmisión o recepción de datos desde la PC y además el estado de ejecución del programa del microcontrolador, es decir, si se están enviando datos desde la PC o si se están enviando datos vía RF.

Temario

1. La evolución de la Robótica
 - 1.1 Que es la Robótica y sus inicios
 - 1.2 Torrebot: Etapas en la construcción de un robot
 - 1.3 Características del brazo robot

2. Estructura mecánica y servomecanismos
 - 2.1 morfología del robot con respecto al brazo humano
 - 2.2 Prototipo final: Planos y montaje
 - 2.3 Control de servomecanismos
 - 2.3.1 Generación PWM sin interrupciones
 - 2.4 Servomotores a utilizarse
 - 2.4.1 Características del servo Hitec HS-422
 - 2.4.2 Características del servo Hitec Hs-81

3. Medio de comunicación para la transmisión de datos
 - 3.1 Características del Transmisor TXM-433-LC
 - 3.1.1 Características del conector serial DB-9
 - 3.1.2 Conexión del transmisor al conector serial DB-9
 - 3.2 Características del Receptor RXM-433-LC-S
 - 3.2.1 Conexión del receptor al PIC 16F877

4. Programación a usarse.
 - 4.1 Características de las tipos de programación usadas
 - 4.1.1 Software y hardware
 - 4.2 Descripción básica de los programas usados

5. Manual del Usuario
 - 5.1 Descripción de la tarjeta Controladora para Servomotores
 - 5.2 Conexiones de los diferentes servomotores del brazo robót

6. Resultados

Conclusiones

Apéndice

Glosario

Bibliografía

INDICE GENERAL

Dedicatoria.....	ii
Agradecimiento.....	iii
Declaración expresa.....	iv
Tribunal de graduación.....	v
Resumen.....	vi
Temario.....	vii
Indice general.....	viii
Indice de figuras.....	x
Indice de tablas.....	xiv
Introducción.....	1
1. La evolución de la Robótica.....	4
1.1. Que es la Robótica y sus inicios.....	4
1.2. Torrebot: Etapas en la construcción de un robot.....	9
1.3. Características del brazo robot.....	13
2. Estructura mecánica y servomecanismos.....	16
2.1. Morfología del robot con respecto al brazo humano.....	16
2.2. Prototipo final: Planos y montaje.....	17
2.3. Control de servomecanismos.....	29
2.3.1. Generación PWM sin interrupciones.....	33
2.4. Servomotores a utilizarse.....	36
2.4.1. Características del servo Hitec HS-422.....	41
2.4.2. Características del servo Hitec Hs-81.....	42
3. Medio de comunicación para la transmisión de datos.....	44

3.1. Características del Transmisor TXM-433-LC.....	46
3.1.1. Características del conector serial DB-9.....	50
3.1.2. Conexión del transmisor al conector serial DB-9.....	53
3.2. Características del Receptor RXM-433-LC-S.....	55
3.2.1. Conexión del receptor al PIC 16F877.....	60
4. Programación a usarse.....	62
4.1. Características de las tipos de programación usadas.....	62
4.1.1. Software y hardware.....	65
4.2. Descripción básica de los programas usados.....	82
5. Manual del Usuario.....	91
5.1. Descripción de la tarjeta Controladora para Servomotores.....	93
5.2. Conexiones de los diferentes servomotores del brazo robót.....	103
6. RESULTADOS.....	106
CONCLUSIONES Y RECOMENDACIONES.....	112
APÉNDICE.....	120
GLOSARIO.....	121
BIBLIOGRAFÍA.....	122

INDICE DE FIGURAS

Figura 1.1 - Esquema de microbot de reacción	8
Figura 1.2 - Esquema de un microbot con el nivel de control.....	8
Figura 1.3 - Especificación de los nombres asignados a los servomotores del brazo robot.....	10
Figura 1.4 – Brazo Robótico con la nueva pinza.....	11
Figura 1.5 – Otra posición del Brazo Robótico.....	11
Figura 2.1 – Morfología entre un brazo humano y un brazo robótico.....	14
Figura 2.2 – Paso 1 para armar el brazo robótico.....	15
Figura 2.3 – Paso 2 para armar el brazo robótico.....	15
Figura 2.4 – Paso 3 para armar el brazo robótico.....	16
Figura 2.5 - Paso 4 para armar el brazo robótico.....	16
Figura 2.6 - Paso 5 para armar el brazo robótico.....	17
Figura 2.7 - Paso 6 para armar el brazo robótico.....	17
Figura 2.8 - Paso 7 para armar el brazo robótico.....	18
Figura 2.9 - Paso 8 para armar el brazo robótico.....	18
Figura 2.10 - Paso 10 para armar el brazo robótico.....	19
Figura 2.11 - Paso 11 para armar el brazo robótico.....	20
Figura 2.12 - Paso 12 para armar el brazo robótico.....	20
Figura 2.13 - Paso 13 para armar el brazo robótico.....	21
Figura 2.14 - Paso 14 para armar el brazo robótico.....	21
Figura 2.15 - Paso 15 para armar el brazo robótico.....	22
Figura 2.16 - Paso 16 para armar el brazo robótico.....	22
Figura 2.17 - Paso 17 para armar el brazo robótico.....	23

Figura 2.18 - Paso 18 para armar el brazo robótico.....	24
Figura 2.19 - Paso 19 para armar el brazo robótico.....	24
Figura 2.20 - Paso 20 para armar el brazo robótico.....	25
Figura 2.21 - Paso 21 para armar el brazo robótico.....	25
Figura 2.22 - Paso 22 para armar el brazo robótico.....	26
Figura 2.23 - Paso 23 para armar el brazo robótico.....	26
Figura 2.24 – Diferentes pulsos cuadrados (PWM).....	29
Figura 2.25 – PWM para el movimiento de un servomotor.....	29
Figura 2.26 - Pulsos generados mediante PWM.....	30
Figura 2.27 - Gráfica de relación lineal.....	31
Figura 2.28 - Servomotor.....	34
Figura 2.29 - Diferentes componentes de un servo.....	35
Figura 2.30 - Servomotor internamente.....	35
Figura 2.31 - Otro modelo interno de un servomotor.....	35
Figura 2.32 - Cables de conexión de un servomotor.....	37
Figura 2.33 - Algunos modelos y marcas de servomotores.....	37
Figura 2.34 - Servomotor HITEC HS-422.....	38
Figura 2.35 - Servomotor HITEC HS-81.....	39
Figura 2.36 - Dimensiones del servomotor HITEC HS-81.....	40
Figura 2.37 - Otra perspectiva del servomotor HITEC HS-81.....	40
Figura 3.1 – Trama de datos.....	43
Figura 3.2 – Formato de cada dato.....	43
Figura 3.3 - Dimensión del transmisor.....	43

Figura 3.4: Conexión del transmisor TXM – 433 – LC.....	46
Figura 3.5 - Nombres de cada pin del transmisor.....	46
Figura 3.6 - Conector DB – 09.....	48
Figura 3.7 – Conector Serie.....	50
Figura 3.8 – Transmisor TXM – 433 – LC.....	51
Figura 3.9 – Puerto Serial, Fuente 5 V (7805), PIC MODEM y Transmisor...51	
Figura 3.10 - Dimensión del receptor.....	52
Figura 3.11 - Modo de conexión del receptor.....	54
Figura 3.12 - Otra forma de ver el receptor.....	55
Figura 3.13 - Nombre de los pines del receptor.....	56
Figura 3.14 – Receptor, controlador (PIC maestro y esclavo) y pines para la conexión de los servomotores	58
Figura 4.1- Pantalla principal y los controles vinculados a la misma.....	64
Figura 4.2 - Pantalla de MOVIMIENTOS LIBRES y los controles vinculados a la misma.....	65
Figura 4.3 - Pantalla de MOVIMIENTOS 2D y los controles vinculados a la misma.....	66
Figura 4.4 - Muestra la pantalla SECUENCIA PROGRAMADA y los controles vinculados a la misma.....	67
Figura 4.5 – Diagrama de flujo de la pantalla principal.....	68

Figura 4.6 – Diagrama de flujo de la pantalla de movimientos libres.....	69
Figura 4.7a – Diagrama de flujo de los MOVIMIENTOS 2D.....	73
Figura 4.7b – Diagrama de flujo de los MOVIMIENTOS 2D.....	74
Figura 4.8a – Diagrama de flujo de SECUENCIA PROGRAMADA.....	77
Figura 4.8b – Diagrama de flujo de Secuencia programada.....	78
Figura 4.9a – Diagrama de flujo del PicModem.....	81
Figura 4.9b – Diagrama de flujo de PicModem.....	82
Figura 4.10a – Diagrama de flujo de Maestro i2c.....	84
Figura 4.10b – Diagrama de flujo de Maestro i2c.....	85
Figura 4.11a – Diagrama de flujo de Esclavo i2c.....	86
Figura 4.11b – Diagrama de flujo de Esclavo i2c.....	87
Figura 5.1 – Módulo transmisor, receptor y el brazo robot.....	88
Figura 5.2 – Otra vista de los módulos transmisor, receptor, brazo robot, cable serie y una fuente de alimentación.....	89
Figura 5.3 - Descripción de tarjeta PicModem.....	91
Figura 5.4 - Diagrama eléctrico de tarjeta PicModem.....	94
Figura 5.5 - Descripción de TARJETA RECEPTORA.....	95
Figura 5.6 - Diagrama eléctrico de TARJETA RECEPTORA.....	98
Figura 5.7 – Descripción de cada articulación del brazo robot con una parte del brazo humano.....	102
Figura 5.8 - Conexión de los servomotores.....	102
Figura 6.1. Grafico de Error vs. Datos.....	104

ÍNDICE DE TABLAS

Tabla 1.1 – Torrebot.....	6
Tabla 1.2 – Características del Brazo Robot (Fabricante Linx).....	12
Tabla 2.1 – Características del Servomotor HITEC HS-422.....	38
Tabla 2.2 – Características del Servomotor HITEC HS-81.....	39
Tabla 3.1 – Grados máximos en el uso del transmisor.....	45
Tabla 3.2 – Descripción de los pines del transmisor.....	47
Tabla 3.3 – Descripción de los pines del puerto serial (DB-09).....	49
Tabla 3.4 – Grados máximos en el uso del receptor.....	55
Tabla 3.5 – Descripción de los pines del receptor.....	56

INTRODUCCIÓN

En el desarrollo del sistema inalámbrico para el control del brazo robot cumple con los siguientes objetivos:

OBJETIVOS GENERALES

- Desarrollar un prototipo de hardware y software que permita el control de un brazo robot desde una PC mediante módulos de radio frecuencia y mediante la programación del microcontroladores PIC16F877A y módulos de inalámbricos RXM433LCS y TXM433LCS.

OBJETIVOS ESPECÍFICOS

- Comunicación de interfases graficas desarrolladas en Visual Basic con hardware controlado con microcontroladores.
- Envío de datos vía RF entre módulos inalámbricos para la interpretación de datos.
- Generación de múltiples salidas moduladas para el control de servomotores que simulen el movimiento de articulaciones y eslabones de nuestro brazo robot

- Establecer la comunicación maestro / esclavo con el bus i2c, para futuras expansiones de dispositivos que se conecten al bus mencionado.

Para que el lector tenga una idea rápida acerca de lo que encontrará en cada

capítulo, a continuación brinda una descripción general de cada uno de ellos:

- **Capítulo 1:** se describe los niveles de diseño y etapas de construcción de un robot además de una reseña histórica de la evolución de los primeros robots en el mundo.
- **Capítulo 2:** en este capítulo se describe la estructura mecánica del brazo robot su ensamblaje, la descripción de los servomecanismos, además de cómo se genera el control PWM que controla a los servomotores usando los recursos del microcontrolador.
- **Capítulo 3:** se describe, los módulos RF como medio de comunicación inalámbrica, además de la explicación concerniente a la trama de datos y las características de los dispositivos mencionados.

- **Capítulo 4:** se describe la programación de control tanto a nivel hardware como a nivel software y además se explican los algoritmos de programación mediante diagramas de flujo y la descripción y uso de las pantallas que controlan al brazo robot.

- **Capítulo 5:** se describe el manual de usuario donde se describen las partes principales de las tarjetas de control, además de recomendaciones y consejos para su correcto funcionamiento

- **Capítulo 6:** se describen los resultados del presente trabajo.

- **Conclusiones:** se describen aquí las conclusiones y recomendaciones del presente trabajo.

- **Apéndice:** se describe el apéndice como medio de ayuda para profundizar características de los elementos usados.

CAPÍTULO 1

LA EVOLUCIÓN DE LA ROBÓTICA

La robótica es un concepto de dominio público. La mayoría de las personas tienen una idea de lo que es la robótica, saben algunas de sus aplicaciones y el potencial que tiene; sin embargo, no conocen el origen de la palabra robot, ni tienen idea del origen de las aplicaciones útiles de la robótica como ciencia.

1.1 QUÉ ES LA ROBÓTICA Y SUS INICIOS.

La Robótica es una tecnología que surgió como tal aproximadamente por el año 1960, desde entonces, el interés que ha despertado es superior a cualquier previsión que en su nacimiento se pudiera formular, siguiendo un proceso paralelo a la introducción de los ordenadores en las actividades cotidianas del hombre. En la actualidad se encuentran robots que han encontrado la manera de formar parte en los hogares tales como robots de limpieza, de compañía de personas, y más, y también son un elemento imprescindible en la mayoría de las industrias.

Podemos contemplar a la robótica como una ciencia en la cual se han conseguido grandes avances tecnológicos; pero todavía ofrece un

amplio campo para el desarrollo y la innovación; y es precisamente este aspecto el que motiva a muchos investigadores y aficionados a los robots a seguir adelante planteando cada vez robots más evolucionados.

EVOLUCIÓN DE LA ROBÓTICA

Desde siempre la naturaleza ha proporcionado inspiración para los desarrollos tecnológicos y el campo de la robótica no ha sido una excepción. En los últimos años, muchos investigadores enfrentados a la complejidad de los sistemas a desarrollar han comenzado a buscar en la naturaleza, y en particular en la evolución, un mecanismo que permita la obtención de sus sistemas de manera automática a partir de su interacción con los entornos y las tareas a desarrollar. La disciplina resultante de esta aproximación se denomina robótica evolutiva. La evolución artificial simula en un ordenador el procedimiento por el cual la selección natural prioriza la conservación del material genético de los individuos mejor adaptados al medio y tarea.

Desde hace cientos de años antes de Cristo, se comenzaron a crear autómatas, antecesores de los actuales. Solamente por citar algunos de estos inventos, se mencionan los siguientes:

- ✓ En 1500 A.C., Amenhotep, hermano de Hapu, construye una estatua de Memon, el rey de Etiopía, que emite sonidos cuando la iluminan los rayos del sol al amanecer.
- ✓ Salomón de Caus (1576 - 1626) construye fuentes ornamentales y jardines placenteros, pájaros cantarines e imitaciones de los efectos de la naturaleza.
- ✓ Jacques de Vaucanson, construye el pato, el autómata más conocido. Un pato hecho de cobre, que bebe, come, grazna, chapotea en el agua y digiere su comida como un pato real. Previamente construye un flautista y un tamborilero en 1738, el primero de los cuales consistía en un complejo mecanismo de aire que causaba el movimiento de dedos y labios, como el funcionamiento normal de una flauta.
- ✓ Robert Houdini construye una muñeca que escribe. También realiza un pastelero, un acróbata, una bailarina en la cuerda floja, un hombre que apunta con una escopeta y una trapecista.
- ✓ Tomás Edison construyó en el año 1891 una muñeca que hablaba **[6]**.

Como se puede observar, los autómatas contruídos hasta ese entonces, solamente servían para entretener a propios y extraños, no tenían una aplicación práctica en alguna área específica. Estas máquinas funcionaban generalmente por medio de movimientos ascendentes de aire o agua caliente. El vertido progresivo de un líquido provocaba rupturas de equilibrio (o bien la caída de un peso) en diversos recipientes provistos de válvulas. Otros mecanismos se basaban en palancas o contrapesos. Mediante sistemas de éste tipo se construían pájaros artificiales que podían “cantar” o “volar”, o puertas que se abrían solas. Las construcciones de la escuela de Alejandría se extendieron por todo el Imperio Romano y posteriormente por el mundo árabe. En el siglo XIII, Al-Djazari apareció como el heredero de todas ellas con la publicación de su libro del “Conocimiento de los Procedimientos Mecánicos”.

En el año de 1920, el escritor de origen checoslovaco Karel Capek, publicó su novela RUR (Russum’s Universal Robots), la cual fue representada en el Teatro Nacional de Praga el 25 de Enero de 1921.

Esta obra trata de dos pequeños seres artificiales de forma humana que responden perfectamente a las órdenes de su creador, aunque al final acaban rebelándose contra él. Para referirse a estos seres, el autor los llamaba **robots**, derivación del vocablo checo **robot**, que

significa “trabajo obligatorio”. Y es así como surge la palabra robot para referirse a los autómatas mecánicos de aquellas épocas. Y a partir de esta novela, se les llamaría robots a los autómatas.

Existe un miedo a los robots debido a la evolución tan acelerada que se ha proyectado en muchas de las novelas de ciencia-ficción. Y aunque muchas de estas novelas no están tan fuera de la realidad, no hay porque tener miedo al desarrollo de robots, sino todo lo contrario; ya que estos existen para poder facilitar las tareas de los humanos.

En la obra de Isaac Asimov, "Yo robot" publicada en 1940, éste postula tres leyes que los robots deberán seguir:

- Un robot no debe dañar a un ser humano o, por su inacción, dejar que un ser humano sufra daño.
- Un robot debe obedecer las órdenes que le son dadas por un ser humano, excepto cuando estas órdenes están en contradicción con la primera ley.
- Un robot debe proteger su propia existencia, hasta donde esta protección no entre en conflicto con la primera o la segunda ley.

Algunas historias más recientes de la evolución de la robótica se muestran a continuación:

- ✓ A mediados del siglo XVIII, J. de Vaucanson construyó varias muñecas mecánicas de tamaño humano que ejecutaban piezas de música.
- ✓ En 1946, el inventor americano G.C Devol desarrolló un dispositivo controlador que podía registrar señales eléctricas por medios magnéticos y reproducirlas para accionar una máquina mecánica. La patente estadounidense se emitió en 1952.
- ✓ En 1951, se realizó un trabajo de desarrollo con teleoperadores (manipuladores de control remoto) para manejar materiales radiactivos. Patente de Estados Unidos emitidas para Goertz (1954) y Bergsland (1958).
- ✓ En 1959, se introdujo el primer robot comercial por Planet Corporation. Estaba controlado por interruptores de fin de carrera.
- ✓ En 1961, un robot Unimate se instaló en la Ford Motors Company para atender una máquina de fundición de troquel.
- ✓ En 1966, Trallfa, una firma noruega, construyó e instaló un robot de pintura por pulverización.
- ✓ En 1968, un robot móvil llamado 'Shakey' se desarrollo en SRI (Standford Research Institute), estaba provisto de una diversidad de

sensores así como una cámara de visión y sensores táctiles y podía desplazarse por el suelo.

- ✓ En 1974, Kawasaki, bajo licencia de Unimation, instaló un robot para soldadura por arco para estructuras de motocicletas.
- ✓ En 1984, se creó Robots 8. La operación típica de éstos sistemas permitía que se desarrollaran programas de robots utilizando gráficos interactivos en una computadora personal y luego se cargaban en el robot. [5].

1.2 TORREBOT: ETAPAS A SEGUIR EN LA CONSTRUCCIÓN DE CUALQUIER ROBOT.

En el desarrollo de ensamblaje de cualquier tipo de robot siempre hay que tener un orden de actuación determinado, es decir una buena planificación. En este caso, se ha recurrido a una herramienta útil que divide en diferentes niveles o etapas la fabricación de robots. Los niveles proporcionados por la Torrebot, como se muestra en la tabla 1.1, son:

NIVEL DE COOPERACIÓN
NIVEL DE COMUNIDAD
NIVEL DE INTELIGENCIA
NIVEL DE CONTROL
NIVEL DE REACCIÓN
NIVEL FÍSICO

Tabla 1.1 – Torrebot

El nivel inferior *Nivel Físico* representa el primer paso y el superior *Nivel de Cooperación* el último (Ver tabla 1.1). Realmente no es necesario llegar hasta el último nivel, dependerá de la aplicación en concreto. Lo que sí es importante es empezar por el primer nivel para evaluar las necesidades y tener un cierto orden de prioridades.

También es evidente que, a medida que se avanza, se puede retroceder para ajustar detalles que hayan quedado pospuestos en niveles anteriores. El significado de cada nivel es el siguiente:

- **Nivel Físico:** Comprende la estructura física, las unidades motoras y las etapas de potencia. Es posible encontrar desde sistemas muy sencillos basados en un único motor, hasta estructuras sumamente

complejas que buscan emular las capacidades mecánicas de algunos insectos, humanos o cualquier ser vivo de la naturaleza.

- **Nivel de Reacción:** Está formado por el conjunto de sensores y sus circuitos de polarización. Estos transductores cubren un amplio margen de posibilidades, tal que podemos encontrar desde simples topes de fin de carrera, hasta micro cámaras digitales con sistemas de reconocimiento de imágenes. Un microbot que haya superado en cuanto a su construcción tanto el nivel físico como el de reacción, se denomina microbot reactivo, tal como se muestra en la figura 1.1. Este tipo de unidades trabajan cumpliendo la premisa, acción o reacción. En estos casos, los sensores son los propios controladores de las unidades motoras, sin ningún tipo de control intermedio.

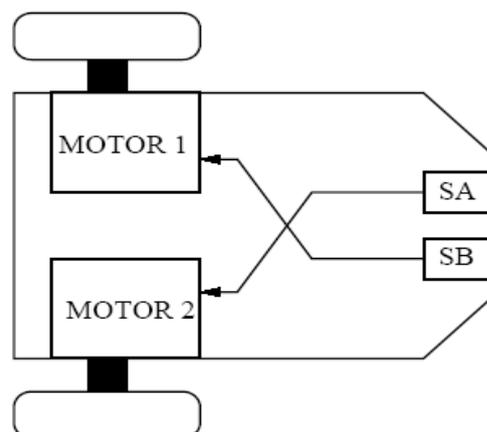


Figura 1.1 - Esquema de microbot de reacción.

SA y SB son sensores para facilitar el movimiento

- **Nivel de control:** Incluye los circuitos más básicos que relacionan las salidas de los sensores con el control de las restantes unidades, tal como se muestra en la figura 1.2. Partiendo de una simple lógica digital hasta potentes microcontroladores, donde se busca dotar al microbot de la capacidad para procesar la información obtenida por los sensores, así como actuar de una manera controlada sobre las unidades motoras.

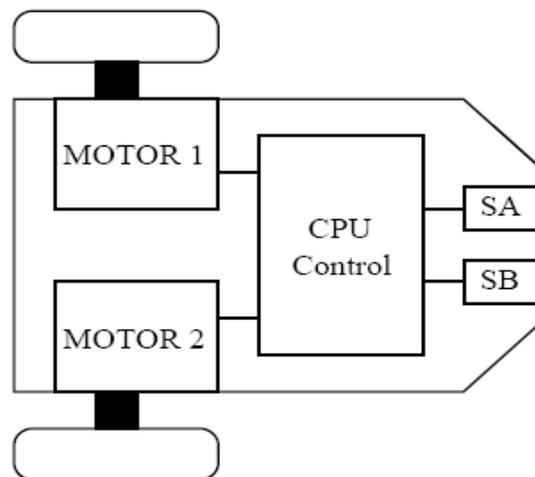


Figura 1.2 - Esquema de un microbot con el nivel de control.

SA y SB son sensores para facilitar el movimiento.

CPU Control, utiliza el valor de los sensores A y B para mejorar el movimiento de giro que le debe dar a los motores 1 y 2.

- **Nivel de Inteligencia:** En éste nivel se introducen los objetivos del microbot que tienen relativa independencia de los sensores. Éste es

el más alto nivel de inteligencia que puede alcanzar un microbot como una unidad individual.

- **Nivel de Comunidad:** Se trata de la puesta en funcionamiento de más de un microbot, dentro de un mismo entorno, de forma simultánea y sin que ninguno de ellos tenga conocimientos explícitos de la existencia de otros en su mismo entorno. A estos recintos se les denomina granjas.
- **Nivel de Cooperación:** Comprende los sistemas donde a partir de un nivel de comunidad, se planifican o programan los microbots para que tengan conocimiento de la existencia de otros, tal que posean la capacidad de cooperar para el buen desarrollo de una tarea.

1.3 CARACTERÍSTICAS DEL BRAZO ROBÓTICO.

El brazo robot de 5 ejes desarrollado por la empresa Lynxmotion, es capaz de realizar movimientos rápidos y precisos de forma repetida. Los ejes incluyen: rotación de la base, hombro, codo, muñeca y pinza, tal como se muestra en la figura 1.3; posibilitando de esta manera 5 movimientos independientes. El armado del brazo robot se lo realiza sin necesidad de herramientas especiales. Entre sus usos

más frecuentes caben destacar que son para: aficionados a la robótica, enseñanza en escuelas y universidades, demostraciones y presentaciones, experimentos de inteligencia artificial, etc.

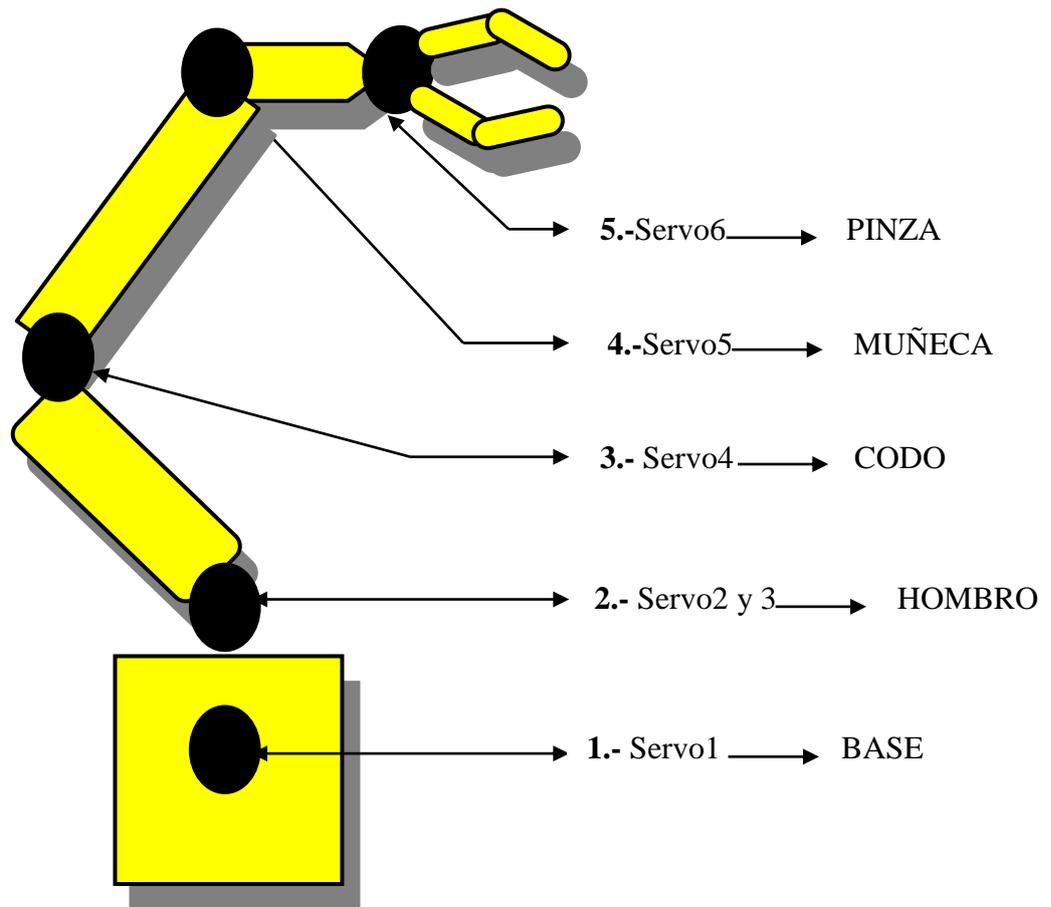


Figura 1.3 – Especificación de los nombres asignados a los servomotores del brazo robot.



Figura 1.4 – Brazo Robot con la nueva pinza



Figura 1.5 – Otra posición del Brazo Robot

Número de Ejes:	5 Incluyendo la pinza.
Altura máxima:	368 mm.
Alcance (adelante):	279 mm.
Apertura de la pinza:	50 mm.
Carga máxima extendida:	85 g / 3 oz. Aproximadamente.
Rango de giro por eje:	90° o 180 ° seleccionable.
Precisión en cada eje:	0.36° o 0.72° seleccionable
Tensión de los servos:	Nominal 6V. Puede usarse con 7.2V
Consumo en reposo:	10 mA cada servo.
Consumo en movimiento:	130 mA cada servo
Fuerza de los servos:	3500 g-cm. / 49 oz.-in
Velocidad de los servos:	90° en 0.24 segundos

Tabla 1.2 – Características del Brazo Robot (Fabricante Linx).

Los 6 servos que se utilizan son de la marca Hitec, que son de alta calidad modelo Hitec Hs-422. Todas las piezas del chasis están hechas de poli-carbonato de 3 mm. El diseño especialmente estudiado del robot, permite que éste sea armado fácilmente sin

necesidad de conocimientos especiales. Incluso los niños (con supervisión de adultos) pueden armar el brazo robot, ayudándolos a comprender su funcionamiento.

CAPÍTULO 2

ESTRUCTURA MECÁNICA Y SERVOMEKANISMOS

Todos los inventores, creadores o aficionados siempre han buscado en la naturaleza algo para poder plasmarlo en un robot, claro que no con la misma precisión pero siempre se hacen muchos intentos.

En éste caso en particular, se escogió hacer los movimientos de un brazo, el cual sería controlar la posición desde una computadora.

También se indicará las formas de armado del brazo robótico y los servomotores que se utilizan, sin olvidar que el controlador decodificará los datos para poder usar la modulación de ancho de pulsos (PWM), que va a permitir que un servomotor se mueva de una a otra posición.

2.1 MORFOLOGÍA DEL ROBOT CON RESPECTO AL BRAZO HUMANO.

Es conveniente tomar en cuenta la forma de un brazo humano de una manera morfológica, es decir, como está formado un brazo de la manera más simple, lo cual ayudaría a armar nuestro brazo robot.

Como se muestra en la figura 2.1, se trata de ensamblar el brazo robot con la misma movilidad y características que tiene un brazo humano en sus movimientos [6].

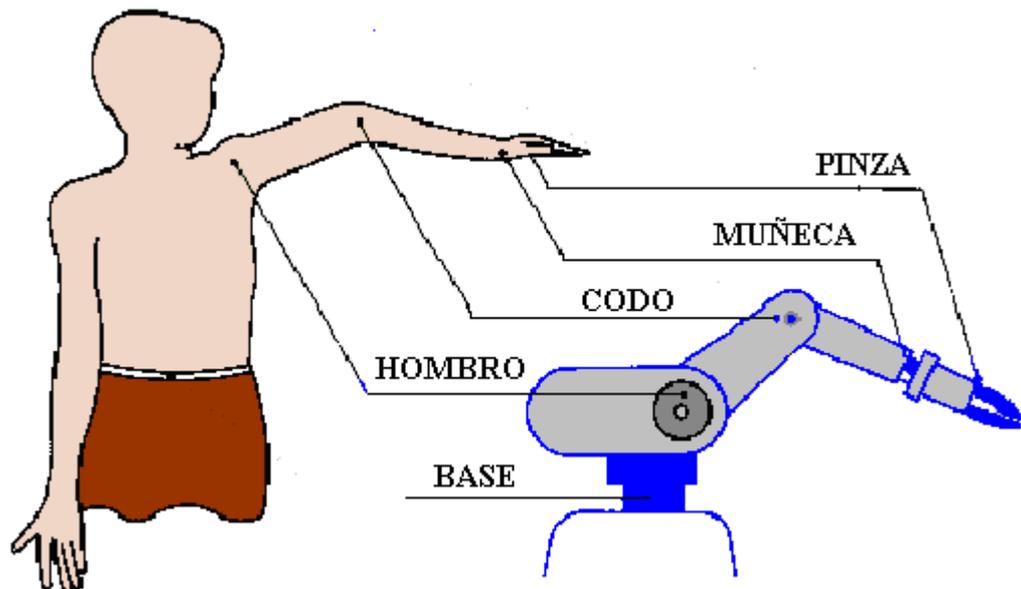


Figura 2.1 – Morfología entre un brazo humano y un brazo robótico

2.2 PROTOTIPO FINAL: PLANOS Y MONTAJE.

A continuación se explica la manera de ensamblar el brazo robot [7]:

Paso 1: Perforar los agujeros que se encuentran en el eje exterior que rota de los servomotores. Vienen 2 o 4 orificios y se les

debe perforar un ancho de 1/16" de pulgadas. Los agujeros son para centrar la pieza que se va a utilizar para que el servomotor la mueva, tal como se muestra en la figura 2.2.



Figura 2.2 – Paso 1 para armar el brazo robótico

Paso 2: Montar la estructura del antebrazo como lo indica la figura 2.3. Use 8 de los sujetadores de remache para mantener los servomotores en su lugar y no se muevan.



Figura 2.3 – Paso 2 para armar el brazo robótico

Paso 3: Usar 2 espaciadores o arandelas para fijar bien los 2 servomotores que servirán de articulación usando 2 tornillos 3/8" de pulgada. Aprételes suavemente. Ver la figura 2.4.

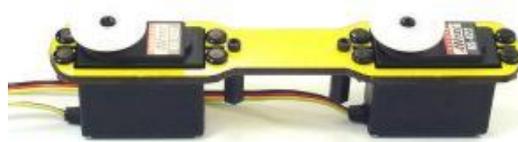


Figura 2.4 – Paso 3 para armar el brazo robótico

Paso 4: Colocar el otro panel del brazo sobre la estructura del antebrazo cuidadosamente. Usar dos tornillos y apretarlos abajo, despacio y con mucho cuidado. Tal como se muestra en la figura 2.5.



Figura 2.5 - Paso 4 para armar el brazo robótico

Paso 5: Insertar las tuercas de 3 mm en las bisagras del servomotor como se muestra en la figura a continuación. Instalar la cinta de doble cara para sostener fijo al servomotor con la bisagra. Ver la figura 2.6.



Figura 2.6 - Paso 5 para armar el brazo robótico

Paso 6: Instalar las bisagras según se muestra en la figura 2.7. Quitar la cubierta verde de la cinta y presiónela firmemente en un lugar. La bisagra se debe alinear con el borde del servo, directamente a través del eje de salida del servo.



Figura 2.7 - Paso 6 para armar el brazo robótico

Paso 7: Quitar el tornillo del servomotor y rotar el servomotor del codo por 90 grados y reinstalar el tornillo sobre el servomotor según lo mostrado en la figura 2.8. Recuerde colocar el eje de salida a su centro de la rotación según lo ilustrado en el paso 2.



Figura 2.8 - Paso 7 para armar el brazo robótico

Paso 8: Montar los soportes a dos servomotores según lo mostrado en la figura 2.9. Usar ocho sujetadores y cuatro soportes.

Verificar que el servo soporte esté instalado debajo de las lengüetas del montaje.



Figura 2.9 - Paso 8 para armar el brazo robótico

Paso 9: Como se necesita rotar estos servos a 90 grados, debe quitar los tornillos del centro del servomotor. Como antes, coloque el eje de salida a su centro de la rotación y vuelva a instalar el servomotor (véase el paso 2). Haga esto a ambos servomotores que sirven de hombro y van juntos.

Paso 10: Utilizar las bisagras correspondientes de cada uno de éstos servomotores y póngalos uno en cada servomotor, luego póngalos juntos y atornillemos para que no se separen éstos servomotores. Ver la figura 2.10.





Figura 2.10 - Paso 10 para armar el brazo robótico

Paso 11: Unir el antebrazo (el gráfico que se muestra en la figura 2.10) al montaje anterior de 2 servomotores (Figura 2.7). Instalar el rodamiento de bolitas a la bisagra como se indica en la figura 2.11.



Figura 2.11 - Paso 11 para armar el brazo robótico

Paso 12: Instalar los dos tornillos para sostener las 2 piezas citadas en la figura 2.11 y dejarlos levemente flojos. Alinear el servomotor del codo según lo mostrado en el paso 2, y apretar los tornillos del servomotor con el antebrazo totalmente horizontal. Agregue el cable del servomotor que servirá para darle la alimentación adecuada, tal como se muestra en la figura 2.12.

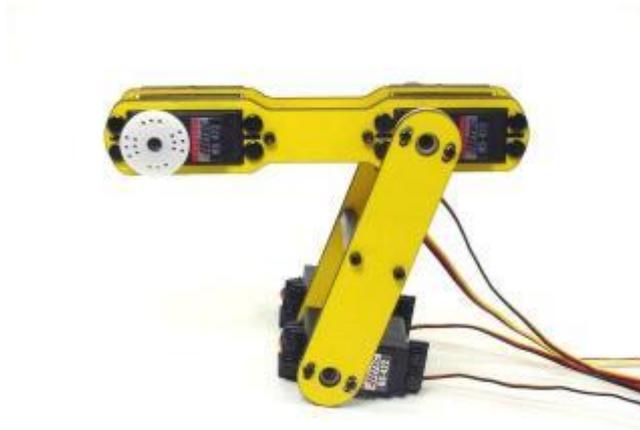


Figura 2.12 - Paso 12 para armar el brazo robótico

Paso 13: Armar la base y colocar un servomotor dentro del panel, en el cual solo se pueda apreciar la parte del servomotor a rotar. Ver figura 2.13.



Figura 2.13 - Paso 13 para armar el brazo robótico

Paso 14: Colocar 3 pilares hexagonales de soporte, 6 tornillos de nylon y 3 tuercas de bellota para poder colocar la otra tapa, y así tener bien formada y fijada la base, tal y como se muestra en la figura 2.14.

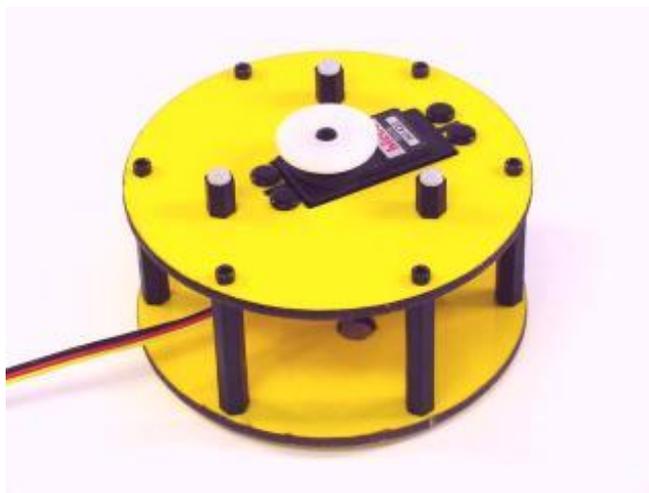


Figura 2.14 - Paso 14 para armar el brazo robótico

Paso 15: Ahora el brazo se puede montar en la base que rota (Figura 2.15), para esto usamos los sujetadores del remache. Utilizar un destornillador plano fuera del sujetador del remache más cercano a la curva del soporte, bastante lejos para poner en los sujetadores inferiores. Entonces apenas empujarla nuevamente dentro de lugar.



Figura 2.15 - Paso 15 para armar el brazo robótico

Paso 16: El brazo ya está casi completo, ver la figura 2.16, sólo faltaría armar lo que es el gripper o pinza e instalarlo.



Figura 2.16 - Paso 16 para armar el brazo robótico

Paso 17: Utilizando los componentes de la figura 2.17, coger uno que es transversal y separarlo con una tuerca hexagonal y después colocar el panel del servomotor en el cual se pone dos sujetadores para que no se mueva el servomotor Hitec Hs-81.

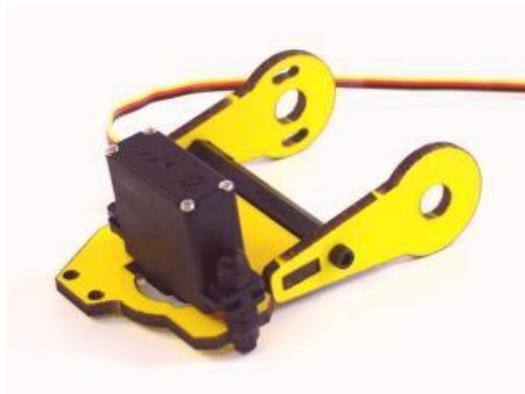


Figura 2.17 - Paso 17 para armar el brazo robótico

Paso 18: Perforar el servomotor en su base rotatoria con la broca de 1/16" de pulgada dentro de las posiciones indicadas. Entonces instalar cada miembro de la pinza al servomotor usando dos tornillos. Tener especial cuidado de no exceder apretando éstos tornillos. Ver la figura 2.18.



Figura 2.18 - Paso 18 para armar el brazo robótico

Paso 19: Unir los miembros cruzados del agarrador pasivo como en la figura 2.19. Insertar un tornillo de nylon en el miembro cruzado pasivo engranado, instalar dos arandelas de nylon 4-40, insertarlas en la placa principal del agarrador, agregar un miembro cruzado del agarrador pasivo, y finalizarlo con una tuerca de bellota del nylon 4-40. Esto no se debe apretar abajo completamente. Debe estar algo libre para que no haya mucha fricción y pueda moverse más fácilmente.



Figura 2.19 - Paso 19 para armar el brazo robótico

Paso 20: Instalar los cuatro miembros cruzados del agarrador pasivo según lo mostrado en la figura 2.20. Utilizar dos tornillos del nylon de 4-40, cuatro arandelas de nylon y dos tuercas de nylon de bellota para este paso. Comenzar con el tornillo. Insertarlo en uno de los miembros cruzados pasivos, agregar dos arandelas de nylon. Insertar esto en la placa principal del agarrador, después agregar otro miembro cruzado y la tuerca de bellota. Éstos no se deben apretar abajo completamente.



Figura 2.20 - Paso 20 para armar el brazo robótico

Paso 21: Instalar uno de los dedos del agarrador según lo mostrado en la figura 2.21. Utilizar un tornillo de nylon 4-40 insertado en el miembro cruzado engranado, agregar una arandela de nylon, agregar el dedo del agarrador, agregar dos arandelas y al final la tuerca de nylon de bellota. Esto no se debe apretar abajo completamente.



Figura 2.21 - Paso 21 para armar el brazo robótico

Paso 22: Vuelva a repetir el paso 21 para insertar el otro miembro en forma de dedo; luego insertar los dedos de goma y la pinza queda como se muestra en la figura 2.22.



Figura 2.22 - Paso 22 para armar el brazo robótico

Paso 23: Colocar la pinza terminada, como se muestra la figura 2.22, y ensamblarla en el brazo total; tal como se indica en la figura 2.23.



Figura 2.23 - Paso 23 para armar el brazo robótico

2.3 CONTROL DE SERVOMECANISMOS.

En este trabajo, para el control de los servomotores y darles una precisión en sus giros y posicionamiento correcto del brazo robot, se ha empleado la técnica del PWM, con la cual se le dice a cada servomotor cuanto debe girar y hacia donde, dependiendo del ancho que tenga el pulso enviado al servomotor.

PWM es uno de los sistemas más empleados para el control de servomotores. Este sistema consiste en generar una onda cuadrada en la que se varía el tiempo que el pulso está en nivel alto, manteniendo el mismo período (normalmente), con el objetivo de modificar la posición del servomotor según se desee.

El sistema de control de un servo se limita a indicar en que posición se debe situar. Esto se lleva a cabo mediante una serie de pulsos tal que la duración del pulso indica el ángulo de giro del motor. Cada servo tiene sus márgenes de operación, que corresponden con el ancho del pulso máximo y mínimo que acepta el servomotor. Los valores más generales se corresponden con pulsos de entre 1 ms y 2 ms de ancho, que dejarían al motor en ambos extremos (0° y 180°). El valor 1.5 ms indicaría la posición central o neutra (90°), mientras que otros valores del pulso lo dejan en posiciones intermedias. Estos

valores suelen ser los recomendados, sin embargo, es posible emplear pulsos menores de 1 ms o mayores de 2 ms, pudiéndose conseguir ángulos mayores de 180°. Si se sobrepasan los límites de movimiento del servo, éste comenzará a emitir un zumbido, indicando que se debe cambiar la longitud del pulso. El factor limitante es el tope del potenciómetro y los límites mecánicos constructivos.

El período entre pulso y pulso (tiempo de OFF) no es crítico, e incluso puede ser distinto entre uno y otro pulso. Se suelen emplear valores aproximados a 20 ms (entre 10 ms y 30 ms). Si el intervalo entre pulso y pulso es inferior al mínimo, puede interferir con la temporización interna del servo, causando un zumbido, y la vibración del eje de salida. Si es mayor que el máximo, entonces el servo pasará a estado dormido entre pulsos. Esto provoca que se mueva con intervalos pequeños.

Mientras el pulso se mantenga en ese ancho de tiempo, el servomotor se ubicará en la posición central de su recorrido. Si el ancho de pulso disminuye, el servomotor se mueve de manera proporcional hacia un lado. Si el ancho de pulso aumenta, el servomotor gira hacia el otro lado. Generalmente el rango de giro de un servo es entre 90° y 180° de la circunferencia total, o un poco más, según la marca y modelo.

Es importante destacar que para que un servo se mantenga en la misma posición durante un cierto tiempo, es necesario enviarle continuamente el pulso correspondiente. De este modo, si existe alguna fuerza que le obligue a abandonar esta posición, intentará resistirse. Si se deja de enviar pulsos (o el intervalo entre pulsos es mayor que el máximo) entonces el servo perderá fuerza y dejará de intentar mantener su posición, de modo que cualquier fuerza externa podría desplazarlo.

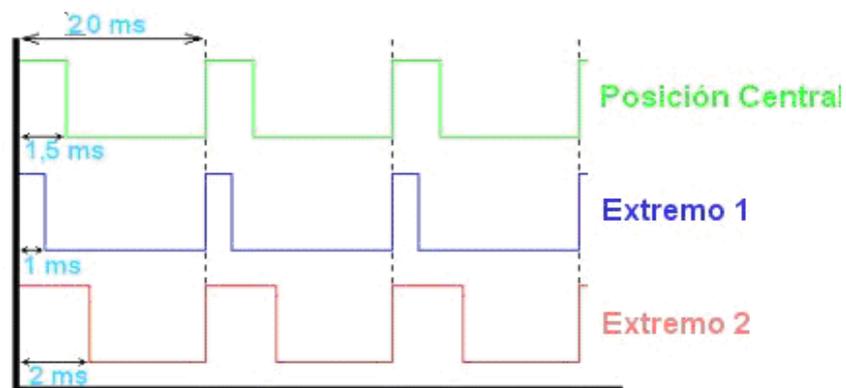


Figura 2.24 – Diferentes pulsos cuadrados (PWM)

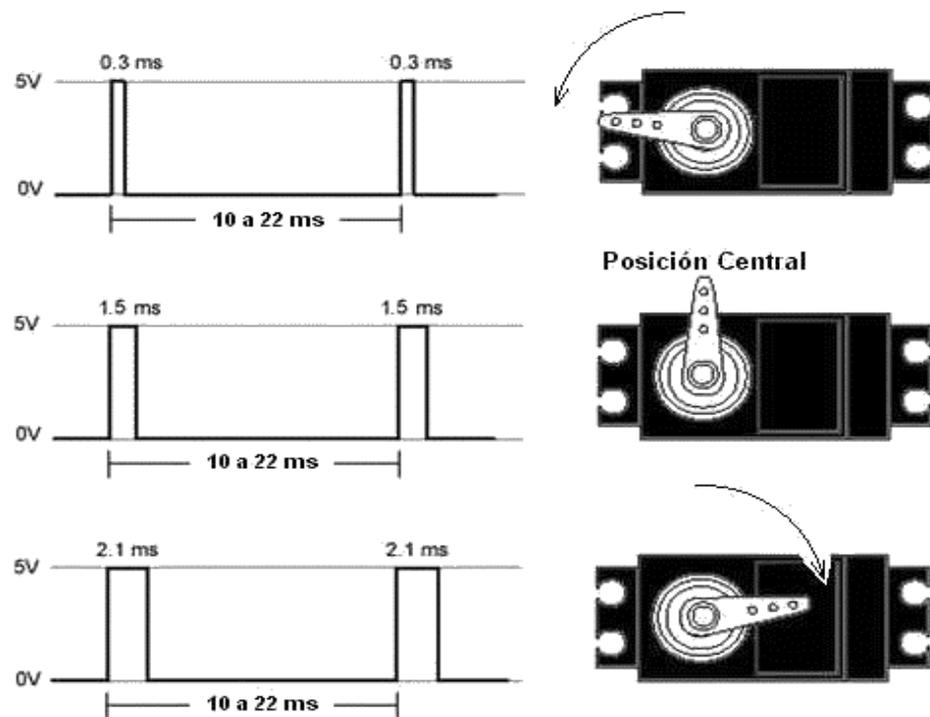


Figura 2.25 – PWM para el movimiento de un servomotor

2.3.1 GENERACIÓN PWM SIN INTERRUPCIONES.

La generación de los pulsos modulados por PWM sin interrupciones, trata específica y literalmente de la generación continua de pulsos modulados, a modo de un tren de pulsos con un periodo de 20ms tal como se muestra en la figura 2.26.

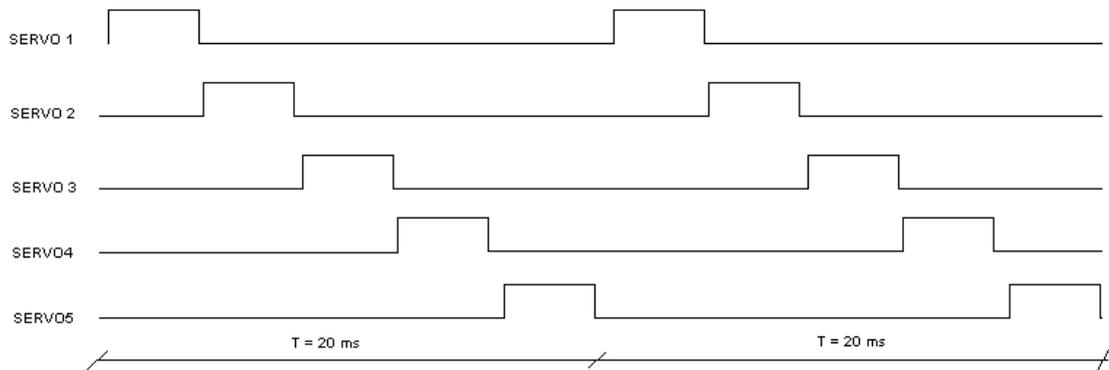


Figura 2.26 - Pulsos generados mediante PWM

Esto se debe a que el mecanismo de control del servomotor, al no detectar un pulso de tiempo en alto (máximo de 2ms) al inicio de un periodo de 20ms, hará que el servomotor crea que es un estado de desactivación, y el torque se reduce a cero, provocando movimientos erróneos y no deseados por el usuario, quien es el que controla el brazo robot por medio de la PC.

Los datos que controlan los movimientos son recibidos desde el maestro_i2c.

El rango de datos esta entre 0 y 255 y deben ser relacionados con **(1)**.

(1) $\text{TIEMPO PWM} = (2000/255) * \text{DATO DEL SERVO} + 500$

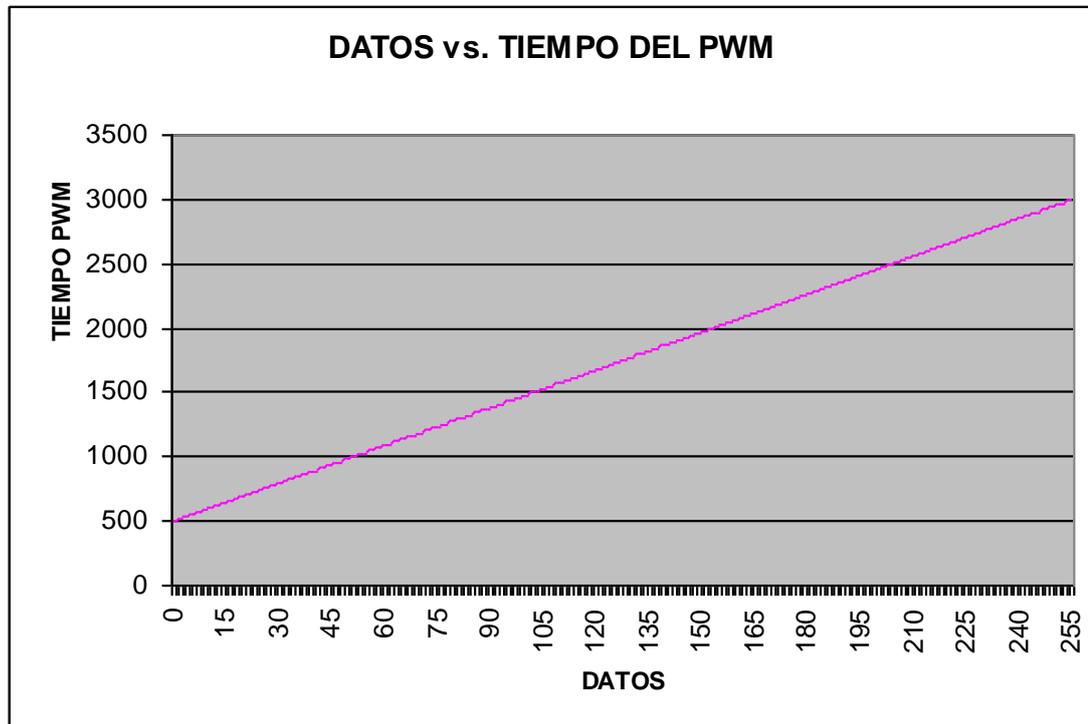


FIGURA 2.27 - Gráfica de relación lineal

Dicha relación se muestra en la [figura 2.27](#) y como se observa se interpretan los datos mediante una relación lineal, en un rango de tiempo de 500us hasta 2500us. El control del tiempo de todos los servos, se lo hace por medio del temporizador de 16 bits del PICX16F877A (TMR1), el cual sirve como reloj general y en conjunto con los módulos CCP (módulos de comparación, captura y modulación) en modo de comparación, que generan los pulsos de cada servomotor.

*****CONFIGURA COMPARADOR Y TEMPORIZADOR*****

```
CONF_COMPARADOR
```

```
BANK0
```

```
MOVLW    B'00001000'
```

```
MOVWF    CCP1CON    ; COMPARADOR EN MODO DE INTERRUPCION
```

```
MOVLW    B'00001000'
```

```
MOVWF    T1CON      ; TMR1 MODO ASÍNCRONO
```

```
          ; PARA EL DISPARO DEL COMPARADOIR
```

```
RETURN
```

```
,*****
```

El TMR1 debe ser configurado en modo asíncrono. Los datos son linealizados y convertidos en valores de 16 bits, correspondientes a los cinco servomotores. Estos valores se cargan en los registros CCPR1H y CCPR1L y son comparados continuamente, con el valor de los registros TMR1H Y TMR1L. En el instante que son iguales estos valores, se produce una interrupción y se genera el pulso reseteando TMR1. Este proceso se repite para cada servo generando señal por señal continuamente y actualizando los datos por medio del maestro_i2c.

```
FOR SERVO=1 TO 5
```

```
SERVO_1
```

```
BSF      PORTE,0
```

```
CLRF     TMR1H
```

```
CLRF     TMR1L
```

```
MOVLW    B' 00000001' ;salida del servo 1
```

```
MOVWF    PORTD
```

```
MOVLW    0X01
```

```

MOVWF    MENSAJEOUT
MOVWF    FLAG_INDICE
MOVF     SER1H,W

```

Cargamos los valores respectivos de cada servo para compararlos y generar la interrupción y tratar la modulación de las salidas de los servos.

```

MOVWF    CCPR1H
MOVF     SER1L,W
MOVWF    CCPR1L
MOVF     SERVO2,W
CALL     LINEALIZAR
MOVF     ACCBHI,W
MOVWF    SER2H
MOVF     ACCBLO,W
MOVWF    SER2L
BTFSS    FLAG_INT_CCP,0

```

;EXPLORAMOS SI SERVO1 FUE SETEADO

```

GOTO $-1
CLRF     FLAG_INT_CCP
NEXT

```

Este lazo se repite para cada pulso de manera secuencial y cíclica, es decir, que se genera primero el pulso del servo y hasta el quinto servo repitiéndose indefinidamente.

2.4 SERVOMOTORES A UTILIZARSE.

Un servomotor es un motor eléctrico que consta con la capacidad de ser controlado, tanto en velocidad como en posición.

Un servomecanismo es un actuador mecánico “generalmente un motor, aunque no exclusivamente”, que posee los suficientes elementos de control como para que se puedan monitorear los parámetros de su actuación mecánica, como su posición, velocidad, torque, etc.

En realidad se utilizan muchos tipos de servos (o servomotores) en equipos industriales y comerciales, desde una disquetera en nuestra computadora o en el video casetera casera, a las unidades de almacenaje de entrada y salida de datos de grandes sistemas de computación (como discos magnéticos), y hasta en los ascensores en edificios. El motor de un ascensor, junto con su equipo de control y detectores de posicionamiento, no es ni más ni menos que un servomotor. El mecanismo que saca el porta - CD de la lectora de CD de su computadora es un servomotor. La figura 2.28 que se muestra a continuación nos muestra como se ve un servomotor.



Figura 2.28 - Servomotor

Estos servomotores se componen, en esencia, de un motor de corriente continua, un juego de engranajes para la reducción de velocidad, un potenciómetro ubicado sobre el eje de salida (que se usa para conocer la posición) y un circuito para el control. En la figura 2.29 se muestran las partes de un servomotor.



Figura 2.29 - Diferentes componentes de un servo



Figura 2.30 - Servomotor internamente



Figura 2.31 - Otro modelo interno de un servomotor

En las figuras 2.30 y 2.31 se puede observar como es un servomotor por dentro y las diferentes partes que lo integran.

Hay una gran variedad de servomotores como los que se muestran en la figura 2.33, tales como Futaba, Hitec, etc, pero los utilizados en esta tesis son los HITEC HS-422 que van en todas partes del brazo robótico a excepción del HS-81 que es el que va en el gripper o pinza.

Hay que tomar en cuenta que los servomotores vienen hechos de fábrica para que sólo puedan moverse 180 grados, pero si se necesita que el servomotor pueda moverse un poco más de sus

límites o hacer que gire vueltas completas, lo adecuado sería desarmarlo y cambiar el potenciómetro, o quitar el potenciómetro, dependiendo del caso que se desee [9].

.Los servomotores tienen 3 cables de conexión externa, los cuales, como se indican en la figura 2.32 son:

- ✓ **Rojo:** para la alimentación. (+ Vcc, aproximadamente + 5 V.).
- ✓ **Negro:** para la conexión a tierra. (GND).
- ✓ **Blanco o Amarillo:** es la línea de control usada por el PWM, para indicar cuantos grados debe moverse.

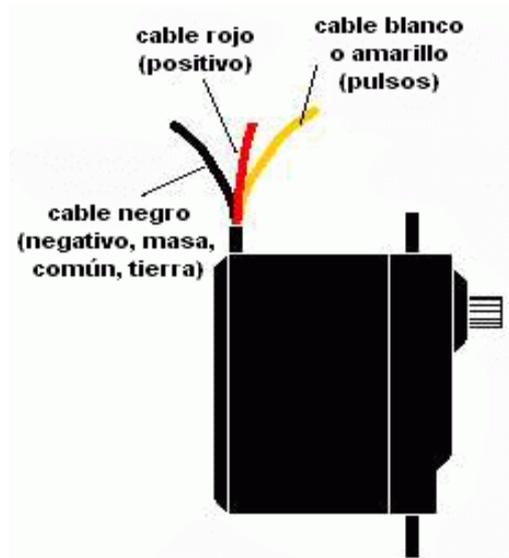


Figura 2.32 - Cables de conexión de un servomotor

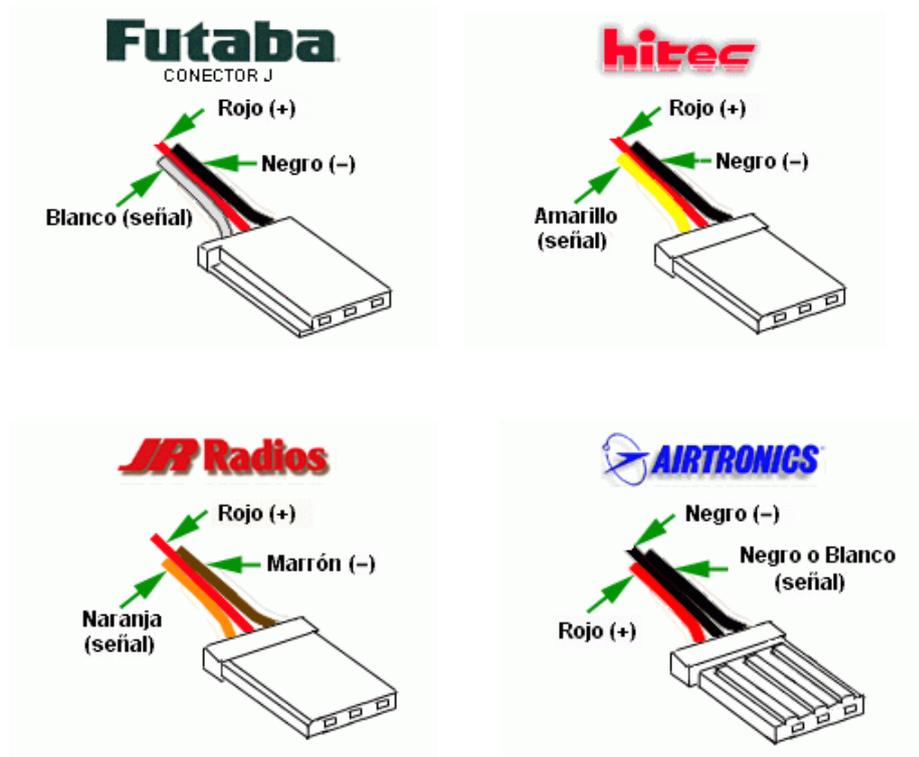


Figura 2.33 - Algunos modelos y marcas de servomotores

2.4.1 CARACTERÍSTICAS DEL SERVOMOTOR HITEC HS-422.



Figura 2.34 - Servomotor HITEC HS-422

Tamaño	Standard
Rango	0° a 180°
Voltaje	4.8 – 6.0 Vdc
Torque	57 oz. – in.
Peso	1.66 oz.
Velocidad	0.16 seg. / 60 grados
Dimensión	1.6" x 0.8" x 1.4"

Tabla 2.1 – Características del Servomotor HITEC HS-422**2.4.2 CARACTERÍSTICAS DEL SERVOMOTOR HITEC HS-81.****Figura 2.35 - Servomotor HITEC HS-81**

Tamaño	Micro
Rango	0° a 160° aproximadamente
Voltaje	4.8 – 6.0 Vdc
Torque	42 oz. – in.
Peso	0.58 oz.
Velocidad	0.09 seg. / 60 grados
Dimensión	1.17" x 0.47" x 1.16"

Tabla 2.2 – Características del Servomotor HITEC HS-81

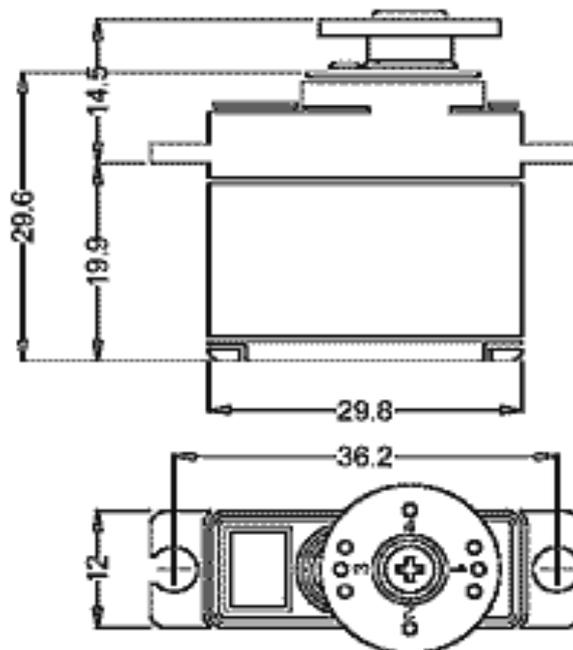


Figura 2.36 - Dimensiones del servomotor HITEC HS-81



Figura 2.37 - Otra perspectiva del servomotor HITEC HS-81

CAPÍTULO 3

MEDIO DE COMUNICACIÓN PARA LA TRANSMISIÓN DE DATOS

En este proyecto se realizan 3 comunicaciones entre 2 equipos o elementos para poder realizar las transmisiones y recepciones de datos; la primera se

trata de la comunicación entre la computadora y el PicModem; la segunda entre el PIC maestro_i2c y PIC esclavo_i2c en el controlador y la última la comunicación inalámbrica entre los módulos de transmisión y recepción

1. La comunicación entre la computadora y el PicModem se la realiza utilizando las señales de control CTS y RTS, la señal de transmisión de datos y la señal de recepción de datos que son propias del puerto serial de la computadora y que en el PicModem se las utilizó para que se pueda controlar el envío y recepción de datos de la computadora hacia el transmisor; puesto que, cuando el transmisor recibe datos este los envía de manera continua e inmediata, pero esto puede hacer que el receptor se sobrecargue y empiece a generar errores en los datos recibidos; por este motivo, hubo que hacer este control en la transferencia de los datos.
2. La comunicación entre el pic maestro_i2c y el pic esclavo_i2c se la controla a través del bus i2; esto quiere decir que cuando el módulo receptor le pasa los datos al pic maestro_i2c, éste se los envía por medio del i2c al pic esclavo_i2c pero con un bit ACK en la trama, que sirve como una pregunta hacia pic esclavo_i2c para saber si el pic esclavo_i2c se encuentra desocupado y poder enviarle los datos; si la respuesta es sí entonces seguirán con la siguiente transmisión pero si

es no, entonces el pic maestro_i2c tendrá que esperar un tiempo para volver a preguntarle. Así mismo, si el pic esclavo_i2c no le contesta durante un cierto momento, entonces el pic maestro le volver a mandar de nuevo la trama.

3. La siguiente comunicación es entre los módulos transparentes de transmisión y recepción de datos. Aquí no hay ningún problema, puesto que lo que se transmite será recibido por el receptor. Estos elementos funcionan en pareja y utilizan una modulación integrada y que fue realizada por el fabricante, la cual es CPCA.

La trama de datos que en este proyecto se usa esta formado por un bit de sincronismo y los datos en sí, así como se muestra en la figura 3.1. Los datos que son enviados inalámbricamente están formados de 5 bytes, cada byte consta de un bit de inicio y un bit de parada como se muestra en la figura 3.2, lo que representa una información de referencia, posicionamiento y cuanto debe moverse cada servomotor; es decir, que el byte 1 representa al servomotor 1, el byte 2 representa al servomotor 2 y así sucesivamente hasta el byte 5.

BYTE DE SINCRONISM	DATO 1	DATO 2	DATO 3	DATO 4	DATO 5
-----------------------	--------	--------	--------	--------	--------

Figura 3.1 – Trama de datos

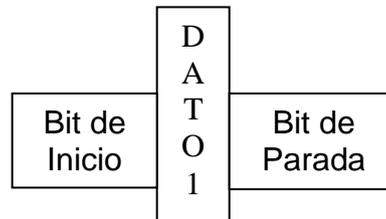


Figura 3.2 – Formato de cada dato

3.1 CARACTERÍSTICAS DEL TRANSMISOR TXM - 433 - LC.

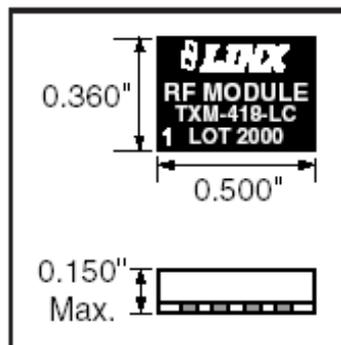


Figura 3.3 - Dimensión del transmisor

Este módulo de transmisión serial fue creado por la empresa Linx Technologies, ver la figura 3.3. Una descripción de este elemento sería que la serie LC se satisface idealmente para el uso en aplicaciones tales como mando a distancia, seguridad, identificación y transferencia de datos periódicamente. Todo su contenido se encuentra en una superficie compacta, como un pequeño paquete, el

transmisor de la serie LC utiliza una arquitectura optimizada llamada arquitectura SAW [10], para alcanzar una mezcla incomparable del funcionamiento, del tamaño, de la eficacia y del costo. Cuando está apareado con un receptor de la serie LC o de la serie LR se forma un acoplamiento sin hilos altamente confiable, capaz de transferir datos seriales en distancias de hasta 300 pies (LC) o de hasta 3.000 pies (LR). No se requiere ningún componente externo de RF (excepto de una antena), haciendo el diseño y la integración sólida y uniforme, incluso para los ingenieros sin experiencia previa sobre RF.

- Entre sus características constan:
 - ✓ Su bajo costo.
 - ✓ No necesita elementos ni componentes externos de RF.
 - ✓ Consumo de energía muy bajo.
 - ✓ Basado en la arquitectura estable SAW.
 - ✓ Soporta transmisión de datos hasta 5000 Bps. ó baudios.
 - ✓ Su rango de energía va desde 2.7 VDC a 5.2 VDC.
 - ✓ Interfase serial directa.
 - ✓ Armónicos bajos.
 - ✓ No produce resonancia.

- Entre las aplicaciones que incluye este transmisor constan:
 - ✓ Control remoto.
 - ✓ Abrir y cerrar puertas o garajes.
 - ✓ Control de luz.
 - ✓ Sistemas médicos de supervisión de llamadas.
 - ✓ Transferencia de datos periódicos.
 - ✓ Automatización industrial o de hogares.
 - ✓ Alarmas de seguridad o de fuego.
 - ✓ Eliminación de cable.
 - ✓ Largo rango de RFID.

- Para saber sobre los grados máximos de aplicación y uso de este tipo de transmisor debe ver la tabla 3.1 que se muestra a continuación.

Fuente de voltaje VCC	-0.3 a +6.0 VDC
Cualquier PIN de entrada o salida	-0.3 a +Vcc VDC
Temperatura de funcionamiento	-30° a +70° C
Temperatura de almacenaje	-45° a +85° C
Temperatura de suelda	+225° C por 10 segundos

Tabla 3.1 – Grados máximos en el uso del transmisor

La forma de operar este transmisor es que usa una técnica de modulación llamada CPCA. Este es un tipo de modulación AM, que también se la puede comparar con la modulación OOK, On Off Key. Este tipo de modulación presenta una lógica baja "0" para representar la ausencia de la portadora y lógica alta "1" para representar que existe una portadora.

Este método nos ofrece muchas ventajas, entre sus principales ventajas tenemos:

- 1.- Rentabilidad debido a la simplicidad del diseño.
- 2.- Ningún requisito mínimo para la transmisión de datos en el espacio.
- 3.- Una energía más alta a la salida del transmisor y así mayor rango de cobertura en países (tales como los EEUU) donde hay espacios donde no llegan bien la señal **[11]**.

Sus datos de funcionamiento y su forma de conectar son como se muestra en la figura 3.4:

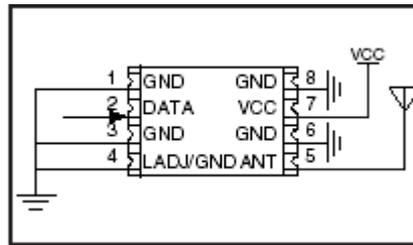


Figura 3.4: Conexión del transmisor TXM – 433 - LC

- Los pines del transmisor están asignados como se muestra en la figura 3.5.

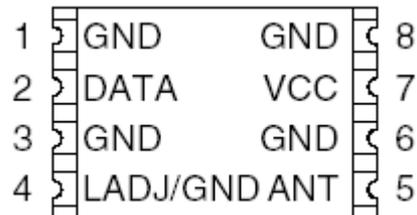


Figura 3.5 - Nombres de cada pin del transmisor

- Descripción de los pines en la tabla 3.2:

# de PIN	Nombre	Descripción
1	GND	Tierra analógica.
2	DATA	Entrada de datos digitales
3	GND	Tierra analógica.
4	LADJ/GND	Nivel de ajuste. Puede ser usado para ajustar la salida de potencia del transmisor.
5	ANT	Salida RF de 50 Ω (ohmios).

6	GND	Tierra analógica.
7	VCC	Fuente de voltaje de entrada.
8	GND	Tierra analógica.

Tabla 3.2 – Descripción de los pines del transmisor

3.1.1 CARACTERÍSTICAS DEL CONECTOR SERIAL DB-09.

Este tipo de conectores pertenecen o forman parte de lo que se conoce con el nombre de interfase V24 ó RS – 232, y que sirve para transportar datos de una manera digital, ya sea enviando o recibiendo información.

Dentro de esta interfase V24, tenemos las siguientes señales:

- Datos: lo que vamos a enviar y/o recibir.
- Control: para que las señales lleguen bien y no se crucen entre sí y el computador no se inhiba.
- Sincronismo: es una señal que le dice en que momento enviar dicha información y en que momento recibirla. Lo hace en orden, sino colapsaría.
- Referencia o tierra.

Dentro de esta interfase tenemos 3 tipos de conectores: el DB-25 y el DB-15, los cuales tienen 25 y 15 pines respectivamente y tienen las 4 señales que se habló en la parte superior; por lo tanto, éstos conectores son **sincrónicos**. Y el otro tipo que existe es el DB-09, que es **asincrónico**, por lo que no tiene la señal de sincronismo, la cual es la que permite la transmisión o recepción de los datos; y tiene solo 9 pines, como se observa en la figura 3.6.

Asincrónico significa que enviaremos carácter por carácter o bit por bit y los conectores pueden ser machos o hembras [12].

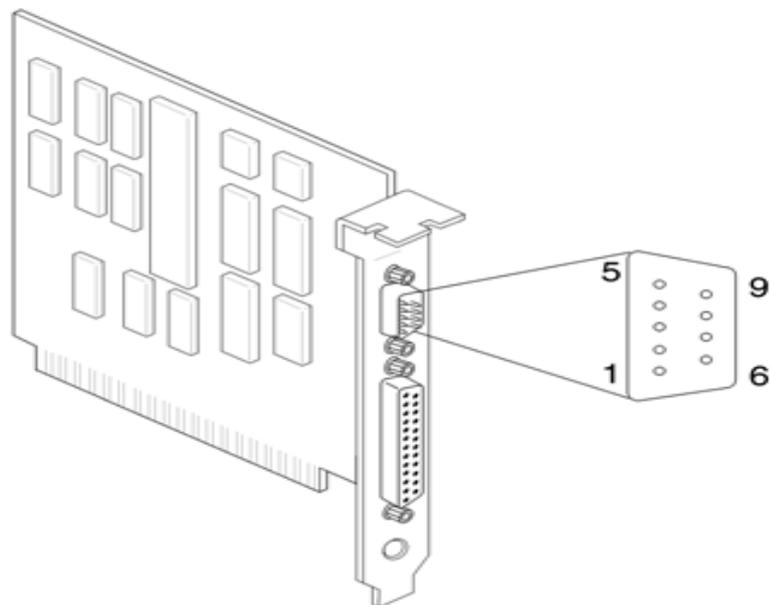


Figura 3.6 - Conector DB – 09

Las señales que tiene cada uno de los pines de este conector DB-09 son las que se muestran en la tabla 3.3:

# PIN	SEÑAL	DESCRIPCIÓN	ENTRADA/SALIDA
1	DCD	Data Carrier Detect	IN
2	RD	Receive data	IN
3	TD	Transmit data	OUT
4	DTR	Data terminal ready	OUT
5	SG	Signal ground	TIERRA
6	DSR	Data set ready	IN
7	RTS	Request to send	OUT
8	CTS	Clear to send	IN
9	RI	Ring indicator	IN

Tabla 3.3 – Descripción de los pines del puerto serial (DB-09)

3.1.2 CONEXIÓN DEL TRANSMISOR AL CONECTOR SERIAL DB-09.

En la conexión de éstos dos elementos hay que tener en cuenta que el cable que sale del pin 3 (pin que transmite) del DB-09 debe ir conectado al pin 2 (entrada de datos) del transmisor. El transmisor se encargará de enviar la información a través de la antena y su modulación propia. Pero hay que tener en cuenta que las salidas del

puerto serial son de 12 voltios digitales, y para poder mandar esa información se necesita un componente que transforme ese voltaje en 5 voltios TTL, por lo cual se usa el MAX – 232.

En la figura 3.7 se muestra el conector DB-09 que se usa para transportar los datos desde la computadora hacia el transmisor, y en la figura 3.8, se encuentra el transmisor que se encargará de enviar los datos inalámbricamente.

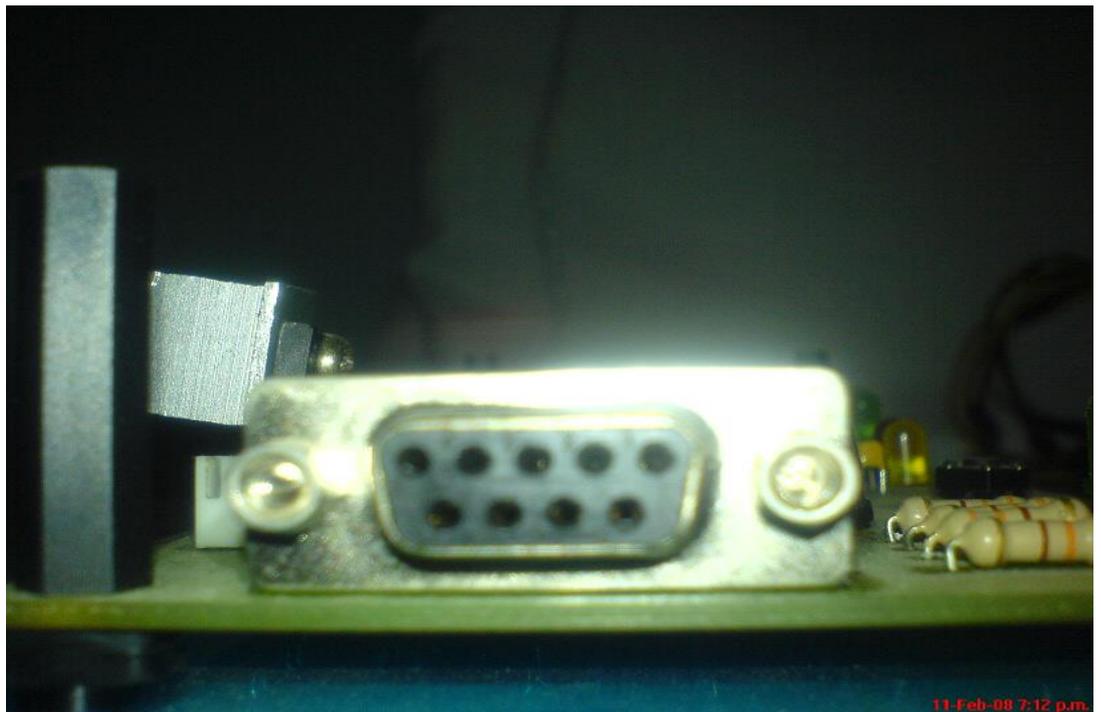


Figura 3.7 – Conector serie



Figura 3.8 – Transmisor TXM – 433 – LC



**Figura 3.9 – Puerto Serie, fuente 5 voltios (7805), PicModem,
MAX-232 y transmisor**

En la figura 3.9 se muestra como esta formado la parte transmisora, en la que consta de la entrada del puerto serial junto al MAX232 con el cual se hace el cambio de tecnología CMOS a TTL, con su propia fuente de alimentación, PicModem y el transmisor. En esta figura podemos ver que primero se utilizo un protoboard para hacer todas las pruebas requeridas para ver su funcionamiento antes de soldar todos los componentes en una placa.

3.2 CARACTERÍSTICAS DEL RECEPTOR RXM-433-LC-S.

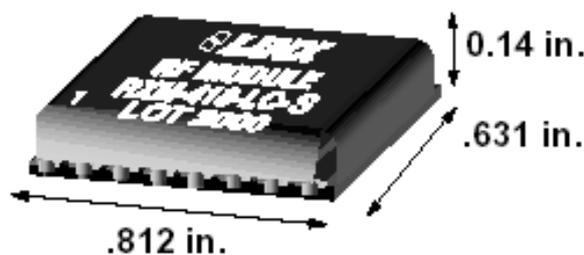


Figura 3.10 - Dimensión del receptor

La serie del LC se satisface idealmente para el uso del volumen en aplicaciones de OEM tales como mando a distancia, seguridad, identificación y transferencia de datos periódica. Todo su contenido se encuentra en una superficie compacta, como un pequeño paquete, el transmisor de la serie del LC utiliza una arquitectura optimizada de la arquitectura SAW [10] para alcanzar una mezcla incomparable del funcionamiento, del tamaño, de la eficacia y del costo. Cuando está

apareado con un transmisor de la serie del LC que empareja o de la serie de LR, se forma un acoplamiento sin hilos altamente confiable, capaz de transferir datos seriales en las distancias en el exceso de 300 pies (LC) o de hasta 3.000 pies (LR).

En la figura 3.10 se muestra el receptor y sus dimensiones para tener una idea de porque su uso.

No se requiere ningún componente externo del RF (excepto una antena), haciendo el diseño y la integración sólida y uniforme, incluso para los ingenieros sin experiencia previa sobre RF.

- Entre sus características constan:
 - ✓ Bajo costo.
 - ✓ No requiere de componentes RF externos.
 - ✓ Bajo consumo de energía.
 - ✓ Basado en arquitectura SAW estable.
 - ✓ Soporta transferencia de datos de hasta 5000 Bps.
 - ✓ Interfase serial directa.
 - ✓ No produce resonancia.

- Entre las aplicaciones que incluye este receptor consta:

- ✓ Control Remoto.
- ✓ Abrir y cerrar puertas o garajes.
- ✓ Control de luz.
- ✓ Sistemas médicos de supervisión de llamadas.
- ✓ Transferencia de datos periódicos.
- ✓ Automatización industrial o de hogares.
- ✓ Alarmas de seguridad o de fuego.
- ✓ Eliminación de cable.
- ✓ Largo rango de RFID.
- ✓ Monitoreo remoto industrial.

Sus datos de funcionamiento y su forma de conectar serían como se muestra en la figura 3.11; tomando en cuenta que estaría funcionando a 3 VDC y si se desee utilizarlo a 5 VDC, que es nuestro caso, se coloca una resistencia externa de 200 ohmios; ya que los otros elementos empleados funcionan también a 5 VDC.

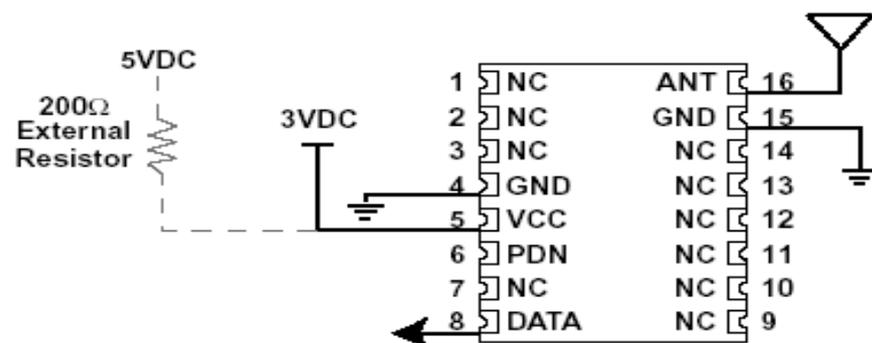


Figura 3.11 - Modo de conexión del receptor

Los pines marcados con **NC**, no tienen conexión física; y son designados únicamente para dar soporte [13].

- Sus grados máximos para aplicación y uso de este tipo de transmisión se muestra en la tabla 3.4:

Fuente de voltaje (Vcc)	- 0.3 a + 4.2 VDC - 0.3 a + 5.2 VDC
Temperatura de operación	- 30 °C a + 70 °C
Temperatura de almacenaje	- 45 °C a + 85 °C
Temperatura de soldadura	+ 225 °C por 10 segundos
Entrada RF, pin 16	0 dBm
Cualquier pin de entrada o salida	- 0.3 a Vcc

Tabla 3.4 – Grados máximos en el uso del receptor

Los pines están asignados como se muestra en la figura 3.12 y 3.13:

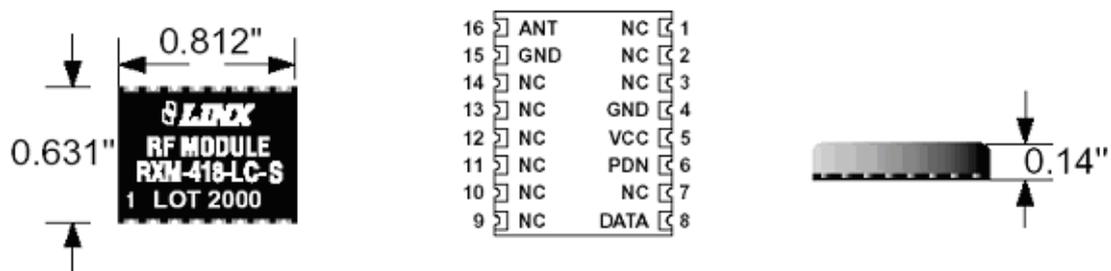


Figura 3.12 - Otra forma de ver el receptor

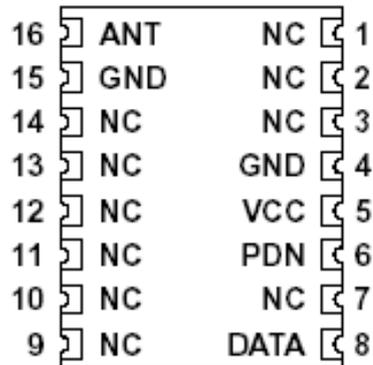


Figura 3.13 - Nombre de los pines del receptor

- La descripción de los pines se muestra en la tabla 3.5:

# de pin	Nombre	Descripción
1, 2, 3, 7, 9, 10, 11, 12,13,14	NC	No conectado
4, 15	GND	Tierra
5	Vcc	Fuente de energía
6	PDN	Pone el receptor en modo para dormir, osea que no recibe datos, cuando recibe un bajo.
8	DATOS	Por donde envía los datos a ser usados.
16	ANT	Medio por el cual recepta los datos transmitidos por el aire.

Tabla 3.5 – Descripción de los pines del receptor

3.2.1 CONEXIÓN DEL RECEPTOR AL PIC 16F877A.

Se conecta el pin 8 del receptor, que es el pin de datos, los cuales ingresan al pic 16F877A (maestro_i2c, parte del controlador principal) por medio del pin 33, el cual se encargará de decodificar la información transmitida desde la computadora y este maestro_i2c pasa la información decodificada por la comunicación I2C hacia el esclavo_i2c, para que éste transmita los datos anchos de pulsos adecuados a cada uno de los servomotores.

Las conexiones del transmisor y receptor se las realiza como se explicó en la parte de arriba, en los puntos anteriores; los 2 pics 16F877A, que son los que funcionan como controlador y en el cual, uno funciona como maestro y el otro como esclavo, son conectados entre sí en los pines 18 y 23, los cuales habilitan el envío y recepción de datos por medio de la comunicación I2C; y luego, desde el pic que funciona como esclavo, el cual enviará las ondas de ancho de pulso hacia cada servomotor.

En la figura 3.14 se muestra la parte receptora, que en sus principios se encontraba ensamblada en otro protoboard para poder hacer todos los análisis necesarios antes de ser armada la placa receptora.

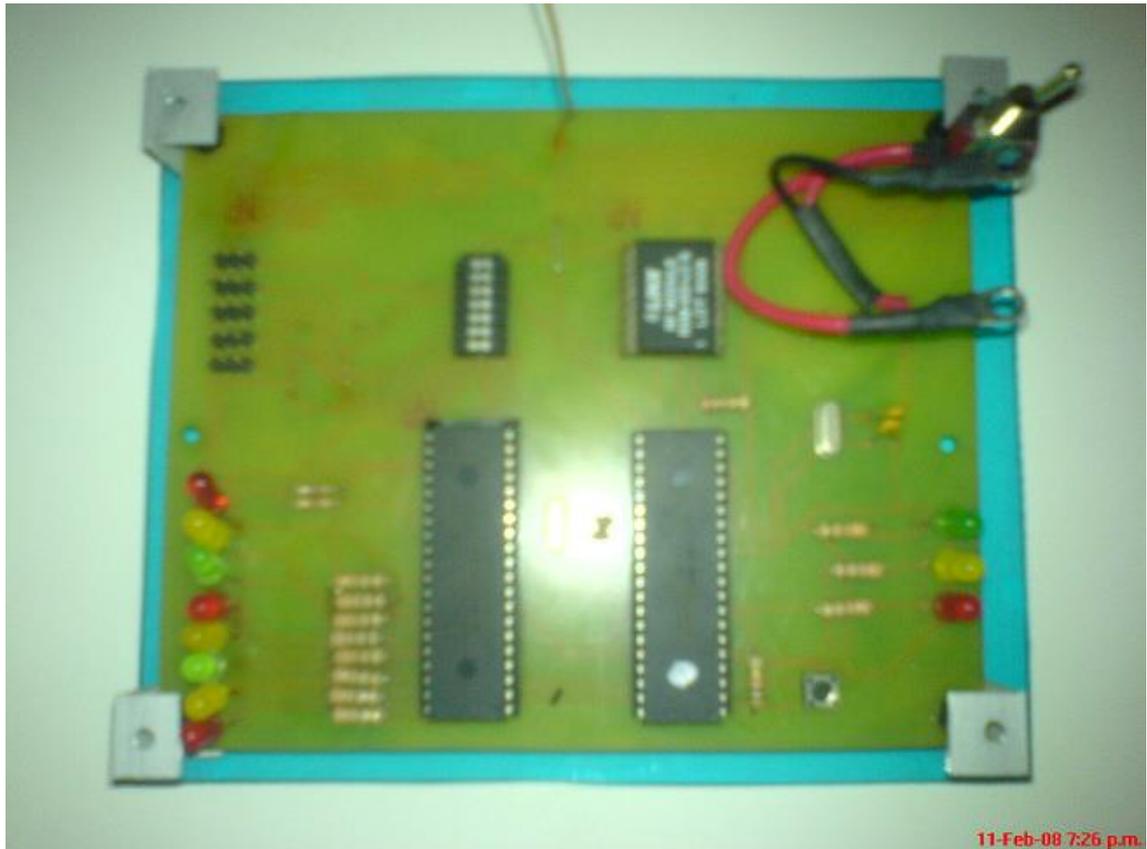


Figura 3.14 – Receptor, controlador (maestro_i2c y esclavo_i2c) y pines para la conexión de los servomotores

Capítulo 4

Programación de Control

De todo lo desarrollado hasta ahora, el brazo robot contiene una estructura mecánica y una red electrónica de control, la cual necesita un programa de control tanto a nivel software como hardware, que sean capaces de reproducir el movimiento de los servomotores desde una interfase gráfica y a través del puerto serie del PC, controlando el envío de datos hacia el hardware. Este hecho conlleva a la reproducción del movimiento de las articulaciones y eslabones del brazo robot.

Para la programación del control del brazo robot es necesario dividir y especificar los tipos de lenguajes de programación utilizados en el desarrollo de la aplicación. A nivel software se utilizará Visual Basic 6.0 de Microsoft para entornos de Windows. Para el nivel hardware se utilizará el MPLAB IDE v7.30 de Microchip para entornos de Windows

4.1 Características de los tipos de programación usados.

La palabra "Visual" hace referencia al método que se utiliza para crear la interfaz gráfica de usuario (GUI). En lugar de escribir

numerosas líneas de código para describir la apariencia y la ubicación de los elementos de la interfaz, simplemente puede arrastrar y colocar objetos prefabricados en su lugar dentro de la pantalla

La palabra "Basic" hace referencia al lenguaje BASIC (Beginners All-Purpose Symbolic Instruction Code), un lenguaje utilizado por más programadores que ningún otro lenguaje en la historia de la informática o computación. Visual Basic ha evolucionado a partir del lenguaje BASIC original y ahora contiene centenares de instrucciones, funciones y palabras clave, muchas de las cuales están directamente relacionadas con la interfaz gráfica de Windows. Los principiantes pueden crear aplicaciones útiles con sólo aprender unas pocas palabras clave, pero, al mismo tiempo, la eficacia del lenguaje permite a los profesionales a cumplir cualquier objetivo que pueda alcanzarse mediante cualquier otro lenguaje de programación de Windows.

El lenguaje de programación Visual Basic no es exclusivo de Visual Basic. La Edición para aplicaciones del sistema de programación de Visual Basic, incluida en Microsoft Excel, Microsoft Access y muchas otras aplicaciones Windows, utilizan el mismo lenguaje. El sistema de programación de Visual Basic, Scripting Edition (VBScript) para

programar en Internet es un subconjunto del lenguaje Visual Basic. La inversión realizada en el aprendizaje de Visual Basic le ayudará a abarcar estas otras áreas.

Si el objetivo es crear un pequeño programa para uso personal o para un grupo de trabajo, un sistema para una empresa o incluso aplicaciones distribuidas de alcance mundial a través de Internet, Visual Basic dispone de las herramientas que necesita.

- Las características de acceso a datos le permiten crear bases de datos y aplicaciones cliente para los formatos de las bases de datos más conocidas, incluidos Microsoft SQL Server y otras bases de datos de ámbito empresarial.

- Las tecnologías Actives, le permiten utilizar la funcionalidad proporcionada por otras aplicaciones, como el procesador de textos Microsoft Word, la hoja de cálculo Microsoft Excel y otras aplicaciones Windows. Puede incluso automatizar las aplicaciones y los objetos creados con la Edición profesional o la Edición empresarial de Visual Basic.

- Las capacidades de Internet facilitan el acceso a documentos y aplicaciones a través de Internet desde su propia aplicación.

- La aplicación terminada es un auténtico archivo ejecutable (programa.exe) que utiliza una biblioteca de vínculos dinámicos (DLL) de tiempo de ejecución que puede distribuir con toda libertad.

- La Edición de aprendizaje de Visual Basic permite a los programadores crear robustas aplicaciones para Microsoft Windows 95 hasta Windows XP e Incluye todos los controles intrínsecos, además de los controles de cuadrícula, de fichas y los controles enlazados a datos.

El lenguaje utilizado para el desarrollo del hardware es el ensamblador y entre algunas de las características del lenguaje se tiene:

- Es más eficiente al generar el código Hexadecimal.
- En cada momento del programa se conoce el estado de los registros.
- Es más fácil de manipular los bits de registros.

- Los programas en ensamblador ocupan menos espacio en memoria.
- Requiere de más cuidado por parte del programador, pues es propenso a que los errores de lógica se reflejen más fuertemente en la ejecución.
- Como es un lenguaje de bajo nivel requiere más líneas de instrucciones para realizar el mismo proceso, en comparación con un lenguaje de alto nivel.

4.1.1 Software y Hardware

El programa realizado en Visual Basic consta de una pantalla principal como se muestra en la figura 4.1; en la cual se puede tener acceso directo a diferentes pantallas, cada una de estas describiendo un modo visual diferente para controlar el brazo robot.

Cada pantalla consta con un conjunto de botones que nos ayudan a controlar el puerto serie. Estos sirven para abrir y cerrar el puerto serie, ó terminar la aplicación. También se tiene un grupo de indicadores que muestran el estado de las líneas de control del puerto serie (RTS Y CTS) y la acción realizada (Transmisión y recepción de

datos). Si su color es rojo su estado es inactivo y si su color es verde su estado es activo.

La pantalla principal muestra tres accesos a diferentes pantallas y un control vinculado a cerrar la aplicación:

- Movimientos libres
- Movimientos 2D
- Secuencia programada
- Cerrar aplicación

El acceso a Movimientos libres se muestra en la figura 4.2. En esta pantalla se muestran cinco barras deslizantes (scrolls), donde cada barra controla el movimiento de una articulación, en nuestro caso cada articulación toma una analogía con un brazo real.



Figura 4.1- Pantalla principal y los controles vinculados a la misma.

El acceso a MOVIMIENTOS 2D se muestra en la figura 4.3. En esta pantalla se muestra el movimiento de un brazo robot tanto en la simulación visual como en su movimiento físico.

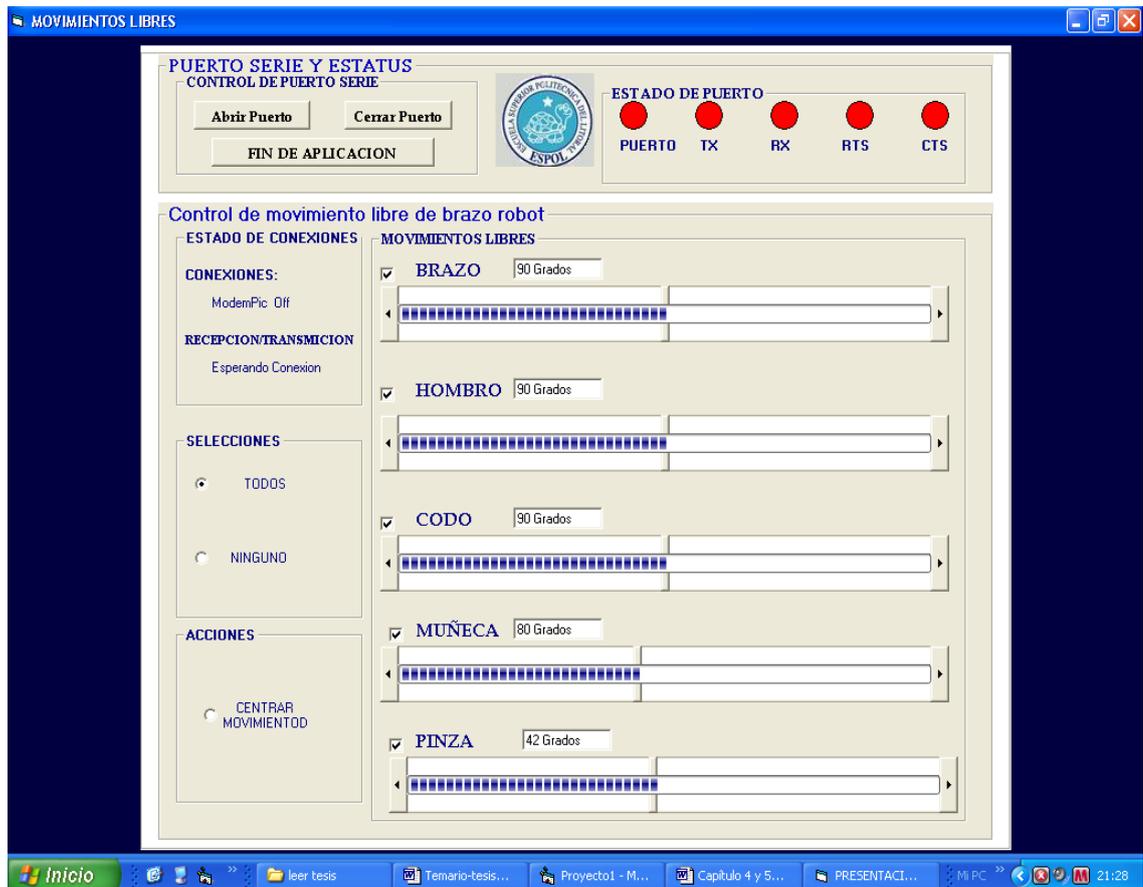


Figura 4.2 - Pantalla de MOVIMIENTOS LIBRES y los controles vinculados a la misma.

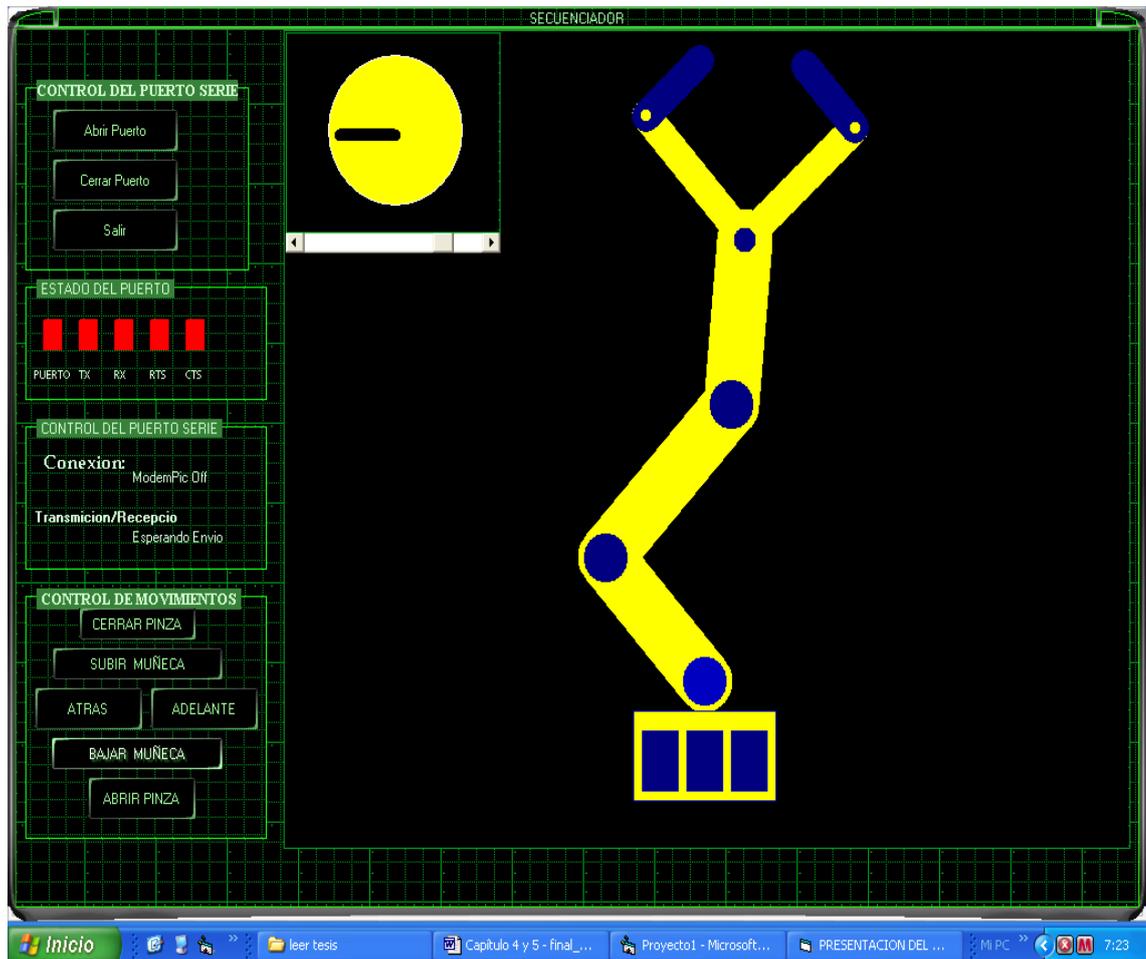


Figura 4.3 - Pantalla de MOVIMIENTOS 2D y los controles vinculados a la misma.

La pantalla SECUENCIA PROGRAMADA se muestra en la figura 4.4.

En esta se muestran los controles para el inicio y termino de secuencia y el control del puerto serie además de sus indicadores de estado del puerto serie.

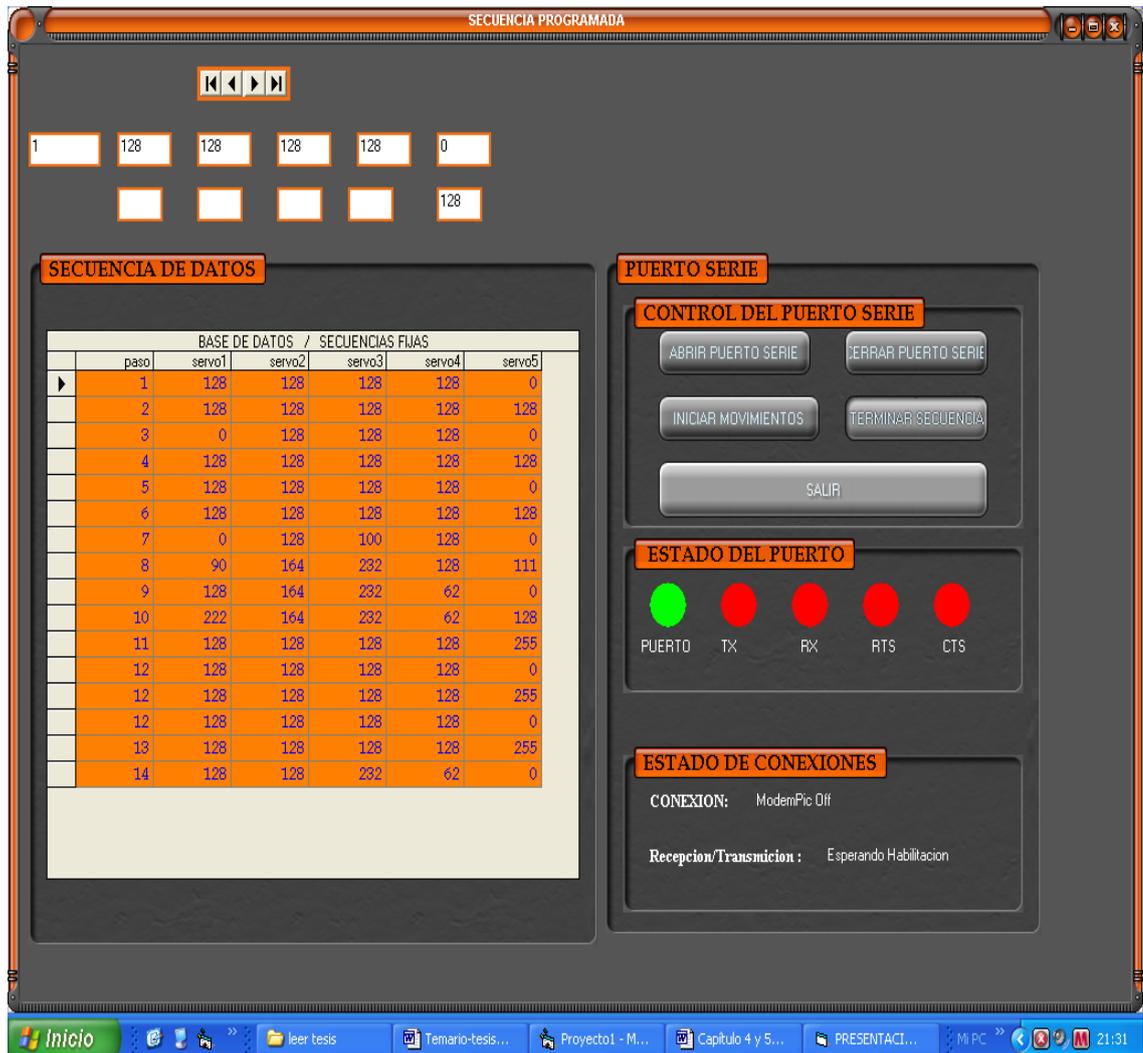


Figura 4.4 - muestra la pantalla SECUENCIA PROGRAMADA y los controles vinculados a la misma

A continuación se describirán los algoritmos de las pantallas utilizadas en el control del brazo robot mediante diagramas de flujo.

Visual Basic:**Pantalla principal**

Cuando algún botón es presionado, se accede a su correspondiente pantalla, tal como se muestra en la figura 4.5. Si ningún botón es presionado se mantiene en estado de espera hasta que el usuario escoja una de las opciones mostradas en la pantalla.

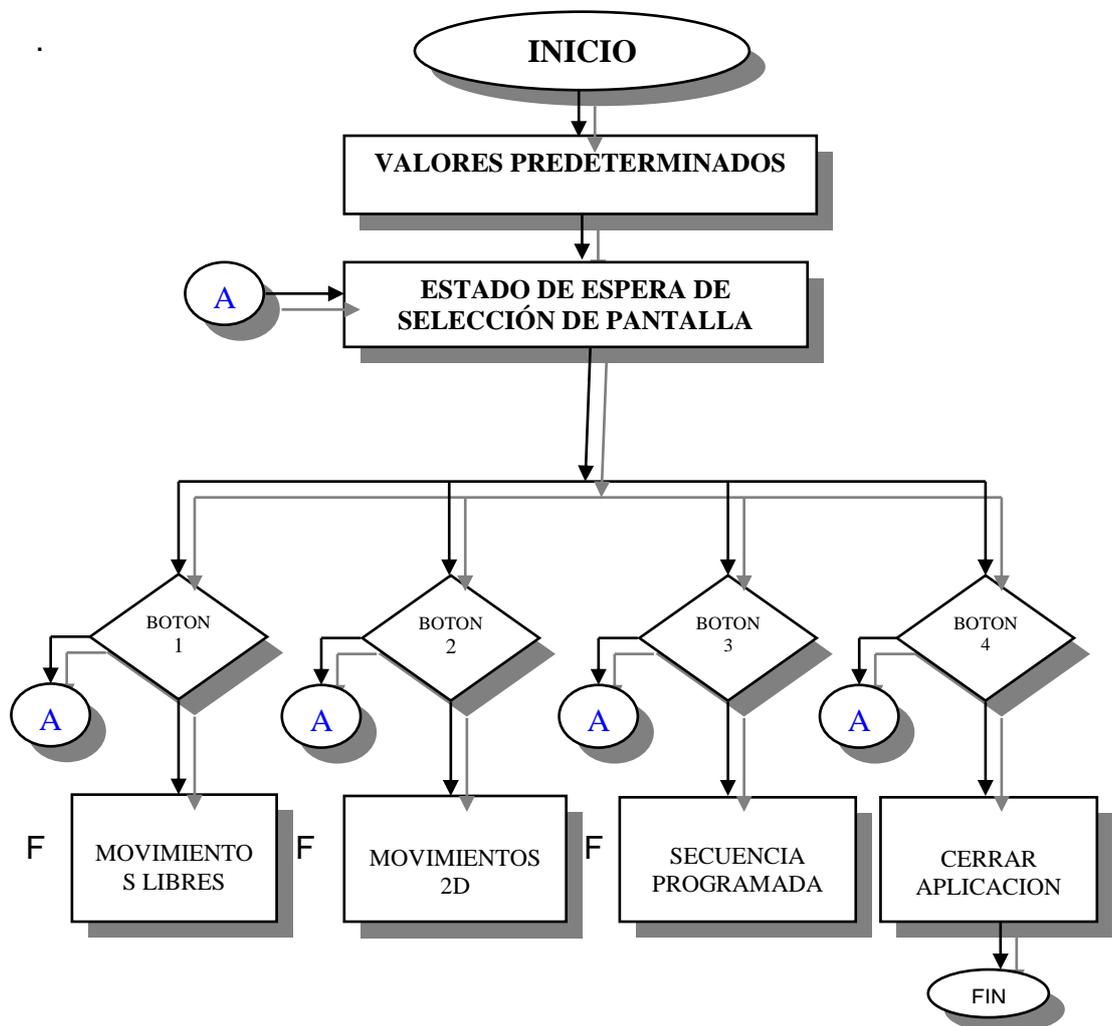


Figura 4.5 – Diagrama de flujo de la pantalla principal

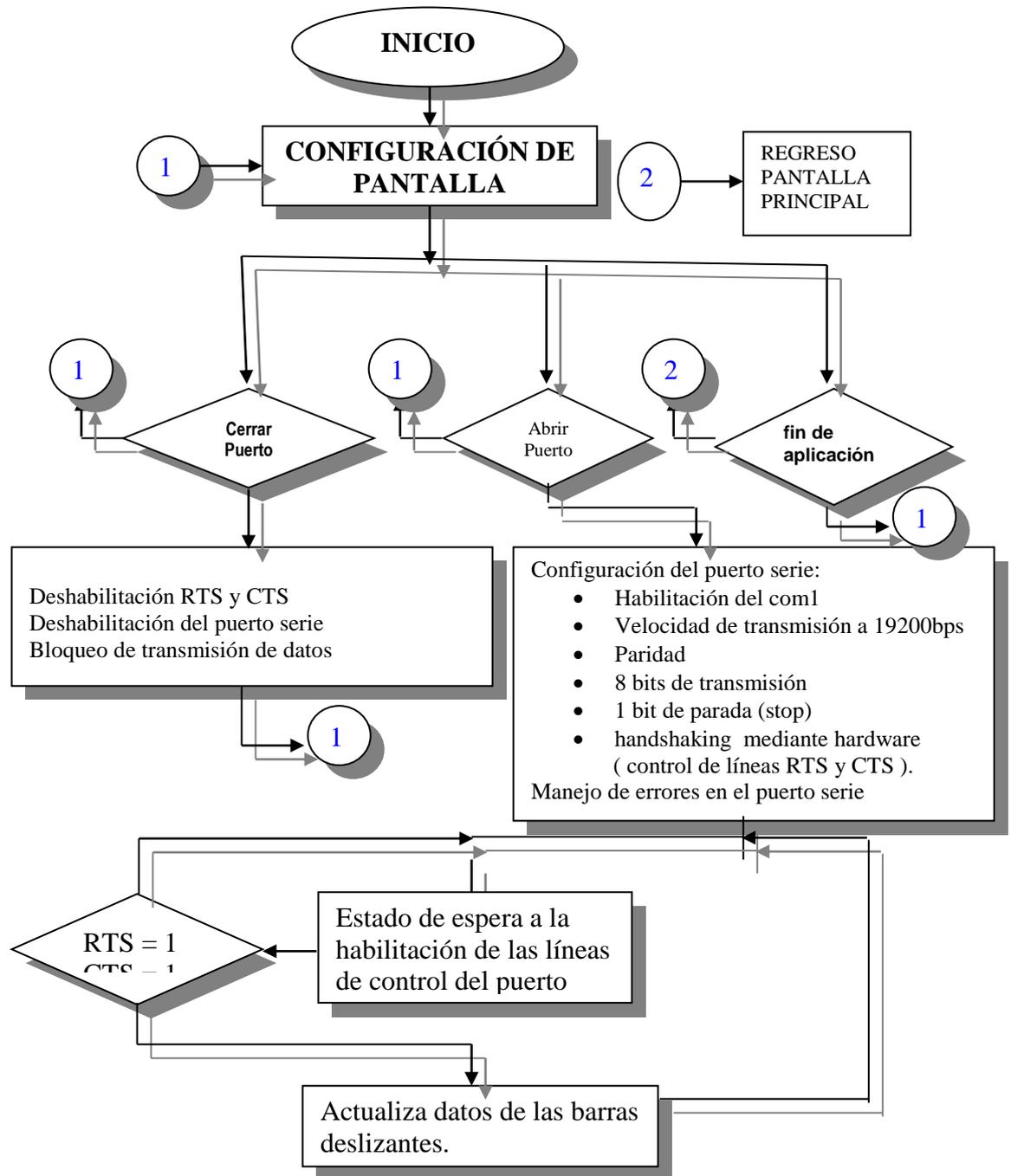
MOVIMIENTOS LIBRES:

Figura 4.6 – Diagrama de flujo de la pantalla de movimientos libres

La figura 4.6 muestra el diagrama de flujo de la pantalla Movimientos libres. La pantalla de movimientos libres consta de tres botones que controlan el puerto serie, tal como se muestra en la figura 4.5. Los botones mencionados son:

- Abrir puerto
- Cerrar puerto
- Fin de aplicación

El control del puerto serie en Visual Basic se lo lleva a cabo mediante el Mscomm (Microsoft Comm Control 6.0) el cual brinda un control completo del puerto serie [15]. El estado del puerto serie se visualiza mediante los indicadores colocados en la sección llamada Estado de puerto, el estado de las líneas de control RTS, CTS, RX Y TX, tal como se muestra en la figura 4.2. Cuando el estado de una línea es activo su color se torna verde, por lo contrario, si su estado es inactivo su color se torna rojo. Este proceso se lleva a cabo a través de un temporizador, el cual lleva a cabo un muestreo cada 100 milisegundos en el puerto serie y procesa el estado de los indicadores para visualizarlos con su respectivo estado (activo o inactivo).

Las articulaciones del brazo robot son controladas mediante cinco barras de deslizamiento horizontal (scrolls) [16], las cuales para

nuestro caso se deslizan en un intervalo de movimiento correspondiente a los grados deseados. Las articulaciones correspondientes a la base, hombro, codo y muñeca tiene un rango comprendido entre 0° y 180° , mientras que la pinza tiene un rango de 0° a 90° . Otra opción que controla las barras deslizantes es la de selección, la cual consta de dos botones de opción (option button) y en cada barra una caja de selección (checkbox) con una función similar a la comentada (habilitación y deshabilitación) pero de manera individual. Los botones de opción son:

- Todos: Habilita el movimiento de todas las barras deslizantes.
- Ninguno: Deshabilita el movimiento de todas las barras deslizantes.

La opción de CENTRAR ARTICULACIONES tiene la función de centrar las articulaciones de todo el brazo. El estado del hardware se muestra en las etiquetas llamada CONEXIONES la cual nos indica el estado de la conexión:

- ModemPic On: indica hardware conectado.
- ModemPic Off: indica hardware desconectado.

MOVIMIENTO 2D

La figura 4.7a y 4.7b muestran el diagrama de flujo que controla la pantalla de movimientos 2D. La parte del control del puerto serie es similar a la pantalla de Movimientos libres con la excepción de que en parte de la programación no se controla barras deslizantes, sino en su defecto, líneas que simulan articulaciones correspondientes al brazo robot. El movimiento es controlado por el control llamado control de movimientos, el cual corresponde a un conjunto de botones [16], que controlan el movimiento de todas las articulaciones. El conjunto de botones se describe a continuación:

- ADELANTE
- ATRÁS
- SUBIR MUÑECA
- BAJAR MUÑECA
- ABRIR PINZA
- CERRAR PINZA

Los botones al ser presionados generan un valor de incremento o decremento, previamente limitado según sea el caso de cada articulación. Dichos valores son procesados para generar las coordenadas que ubican los puntos de las articulaciones tanto en el eje de las abscisas y de las ordenadas. También una barra deslizante se encarga del control del movimiento de la base, la cual gira en torno

a su eje 180°. El muestreo del estado de las líneas de control del puerto serie y el estado de las conexiones del hardware es idéntico en todas las pantallas.

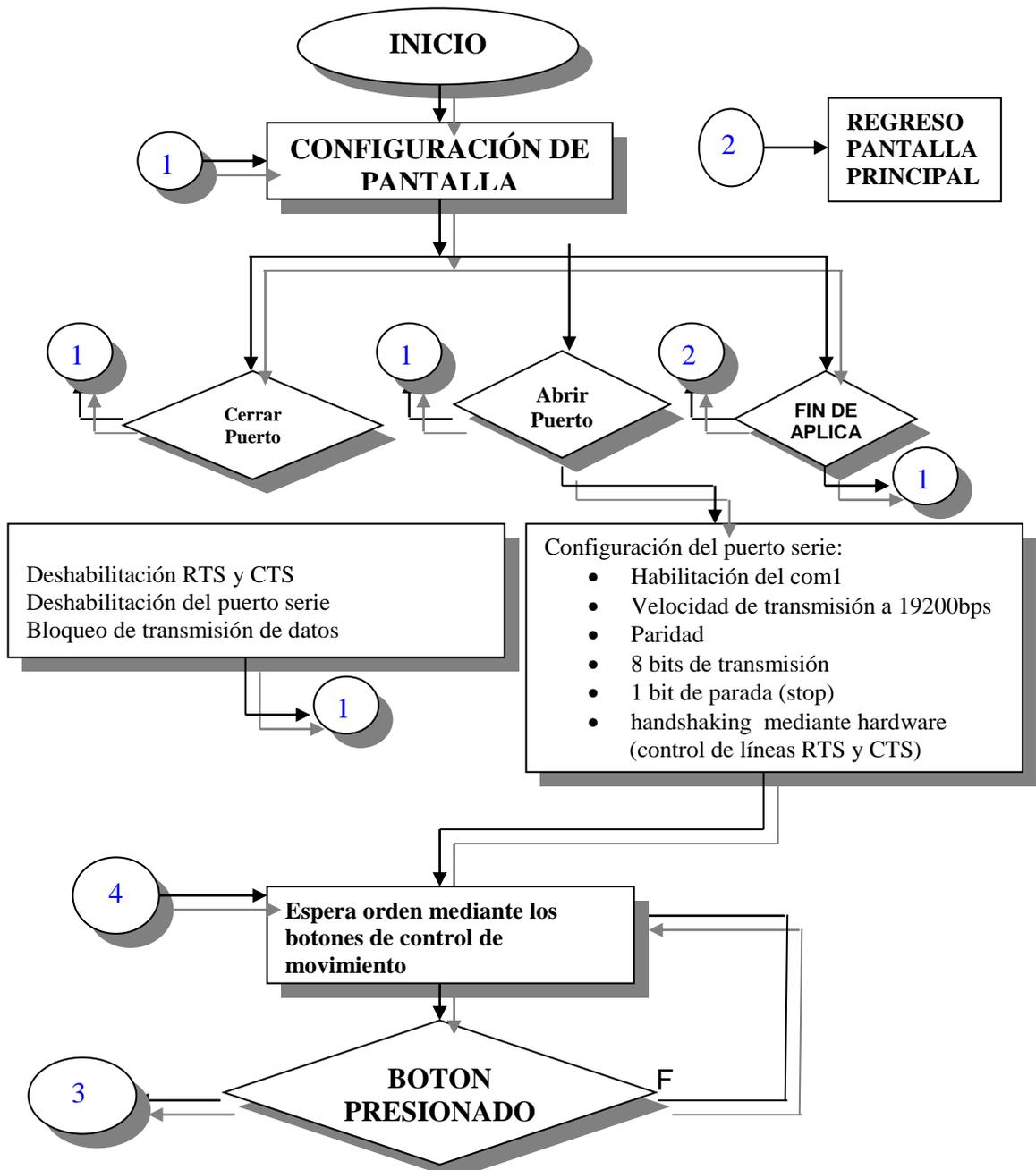


Figura 4.7a – Diagrama de flujo de los MOVIMIENTOS 2D

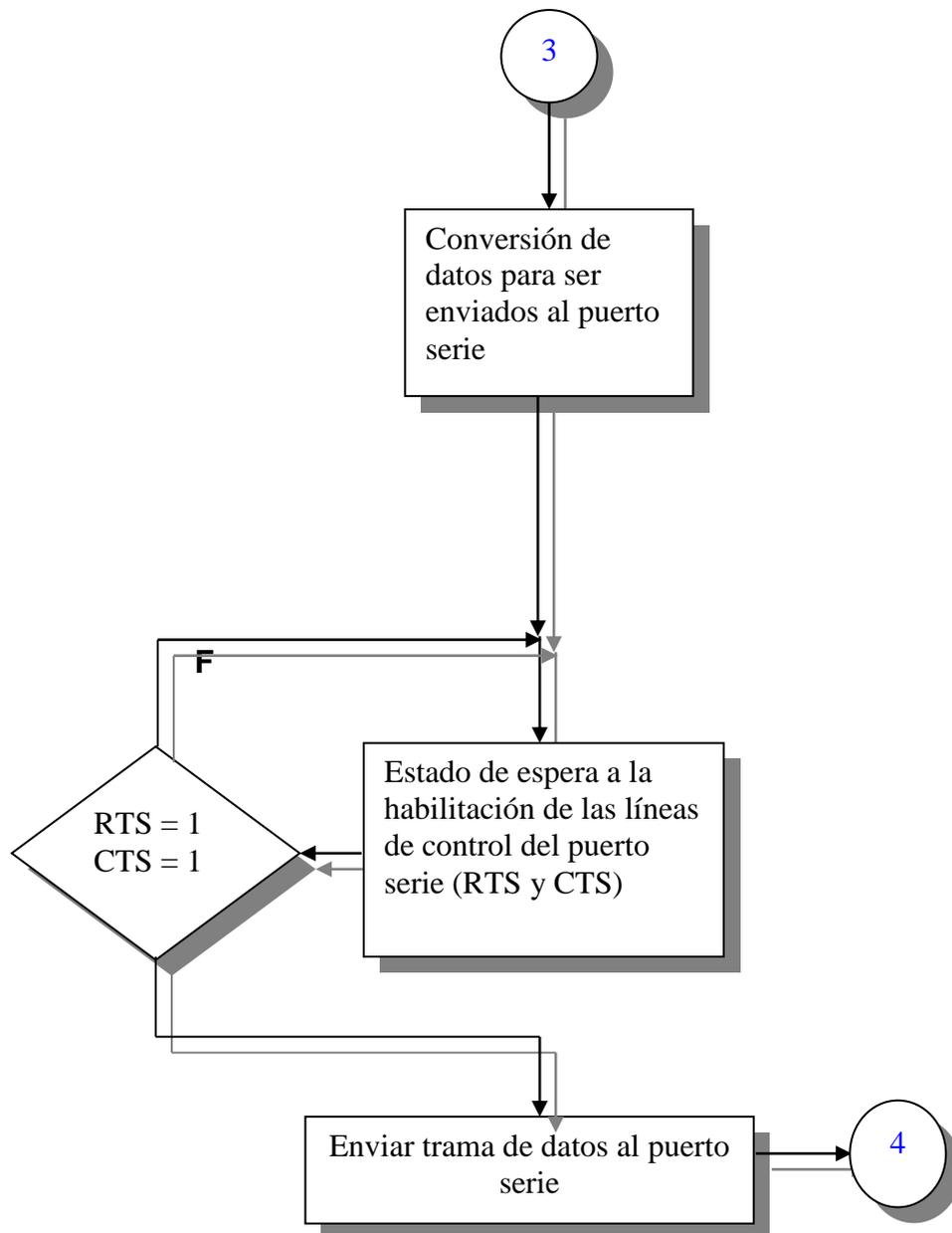


Figura 4.7b – Diagrama de flujo de los MOVIMIENTOS 2D

SECUENCIA PROGRAMADA

La figura 4.8a y 4.8b muestran el diagrama de flujo que controla la pantalla de secuencia programada. Como se menciono anteriormente el control del puerto serie se lleva a cabo de igual manera para todas las pantallas, tal y como se explico con detalles en la pantalla de MOVIMIENTOS LIBRES.

En esta pantalla se usa una base de datos en Microsoft Access que conjuntamente con Visual Basic se enlazan para enviar datos desde dicha base hacia el puerto serie y luego vía RF al control del brazo robot. Para utilizar una base de datos en Visual Basic existe un control llamado control data, el cual se asocia la base de datos mediante la propiedad **RecordSource** y conjuntamente con la propiedad **Connect [18]**, esta nos indica que base de datos se utilizará. Además de los botones del control del puerto serie, esta aplicación consta de los siguientes botones para el inicio de la lectura de la base de datos y su transmisión:

- Iniciar movimiento
- Terminar secuencia

El objeto data posee dos propiedades muy importantes para indicar el inicio y el fin de los registros. La primera es BOF (Begin of file – inicio de archivo) y el segundo EOF (End of file – fin de archivo), ambas propiedades ayudan a saber cuando estamos en el fin o el inicio de la tabla de datos.

Para el incremento o decremento en la lectura de las líneas de las tablas se utilizan las siguientes instrucciones:

- **tabla.Recordset.MoveNext** : El puntero de la tabla avanza una posición en la tabla de datos.
- **tabla.Recordset.MoveFirst**: El puntero de la tabla se ubica en la primera posición en la tabla de datos.
- **tabla.Recordset.MovePrevious**: El puntero de la tabla retrocede una posición en la tabla de datos.
- **tabla.Recordset.MoveLast**: El puntero de la tabla se ubica en la última posición en la tabla de datos.

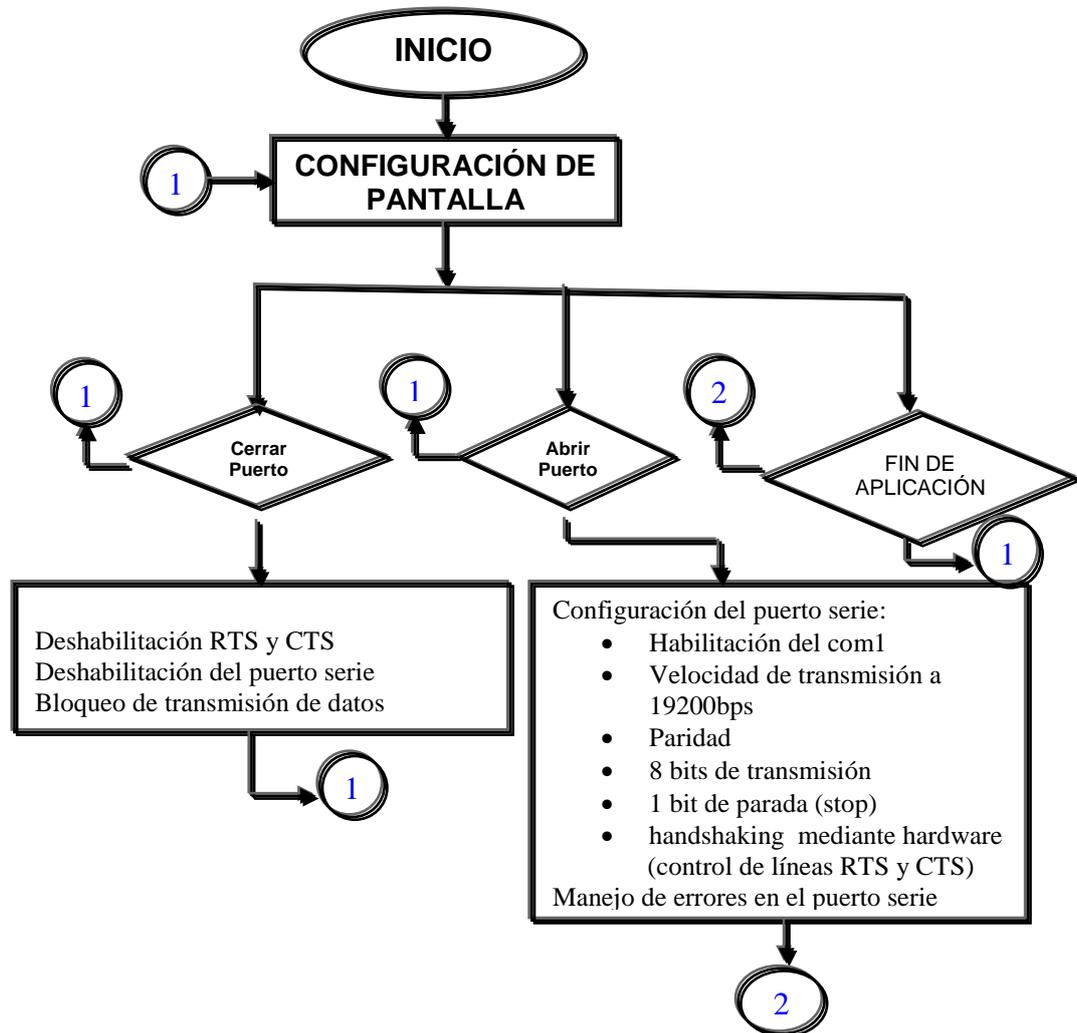


Figura 4.8a – Diagrama de flujo de SECUENCIA PROGRAMADA

El uso de este control ayuda a la lectura y escritura de una base de datos en Access y a su visualización debido a que se conecta con otros controles visuales tales como caja de texto (TextBox). Cuando el usuario halla abierto el puerto y presionado el botón de inicio de secuencia se empezara enviando al puerto serie la primera línea de la tabla de datos y así sucesivamente seguirá leyendo y enviando datos

hasta que se presione el botón de terminar de secuencia. Si se desea salir de la aplicación se debe presionar el botón salir, el cual regresara a la pantalla principal.

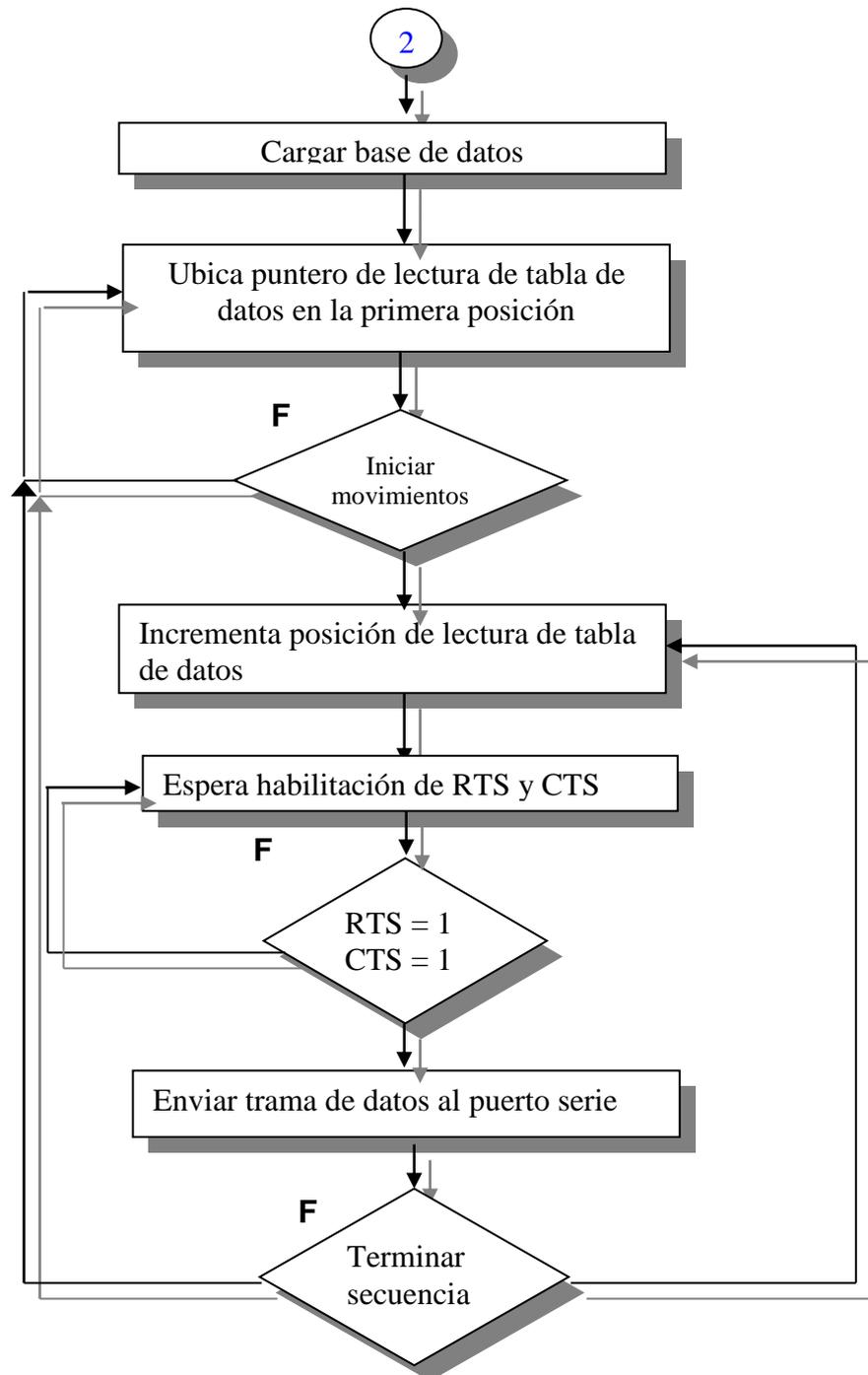


Figura 4.8b – Diagrama de flujo de Secuencia programada

4.2. Descripción básica de los programas usados

MPLAB IDE7.30 (Ensamblador):

Los programas que se desarrollaron en MPLAB IDE, controlan el hardware de la aplicación, tanto para el envío y recepción de datos vía RF, comunicación I2C y el control de los servomotores mediante la modulación de ancho de pulso (PWM). Los programas que se explicaran en detalle son los siguientes:

- PicModem
- Maestro_i2c
- Esclavo_i2c

PicModem:

Este programa, maneja el control de flujo de datos, mediante las líneas del PIC16F877A RC0 y RC1 conectadas a las líneas RTS, CTS, TX y RX, además leds indicadores nos muestran el estado de las líneas RTS, CTS, TX Y RX. Dos indicadores nos indican también el estado de ejecución del programa, es decir, si esta comunicándose

con la PC o enviando los datos vía RF. Estos dos indicadores se conectan en las líneas RC2 Y RC3 del PIC. La descripción del algoritmo se muestra en la figura 4.9. La función principal del PicModem es la de recibir la trama de datos, desde el puerto serie a través del MAX232 el cual convierte los niveles de la Norma 232 a niveles TTL, para luego enviar los datos vía RF, destacando el control de la habilitación y la deshabilitación de las líneas RTS y CTS.

La transmisión desde la PC se lleva acabo a través del modulo USART (Universal synchrhonous asynchrhonous receiver transmitter) implementada en el microcontrolador [20]. La transmisión inalámbrica se lleva acabo mediante una librería llamada UART.asm.

Los datos son enviados en una trama que consta de un byte de sincronismo y cinco datos correspondientes al movimiento de cada servo.

El diagrama de flujo que se muestra en la figura 4.9a y 4.9b muestra el funcionamiento del programa.

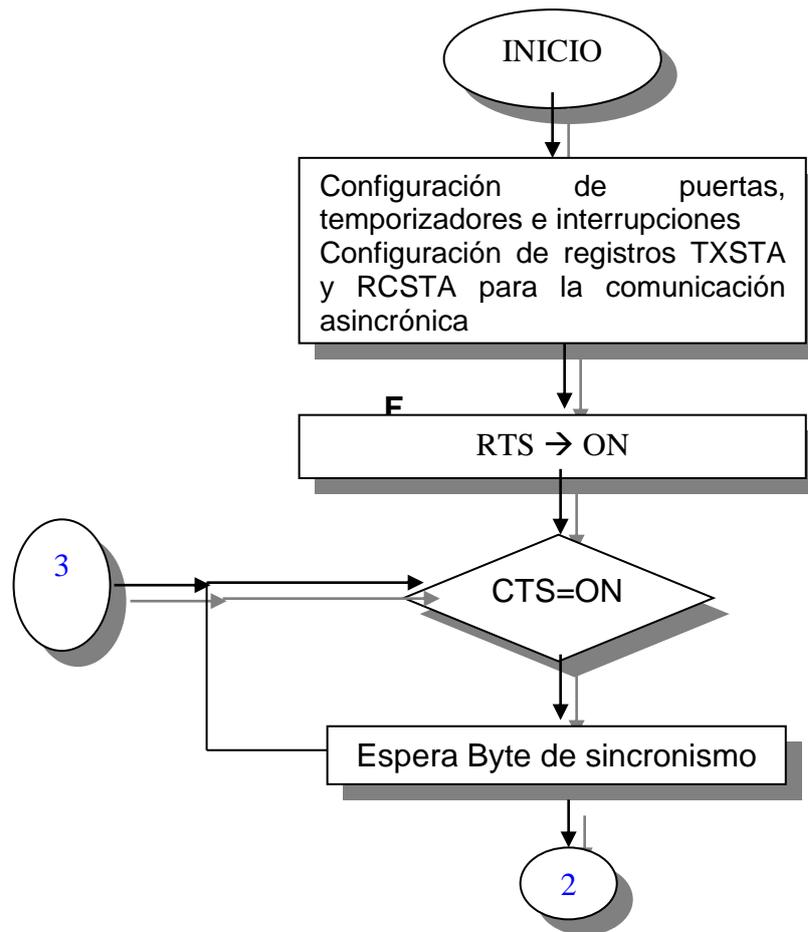


Figura 4.9a – Diagrama de flujo del PicModem

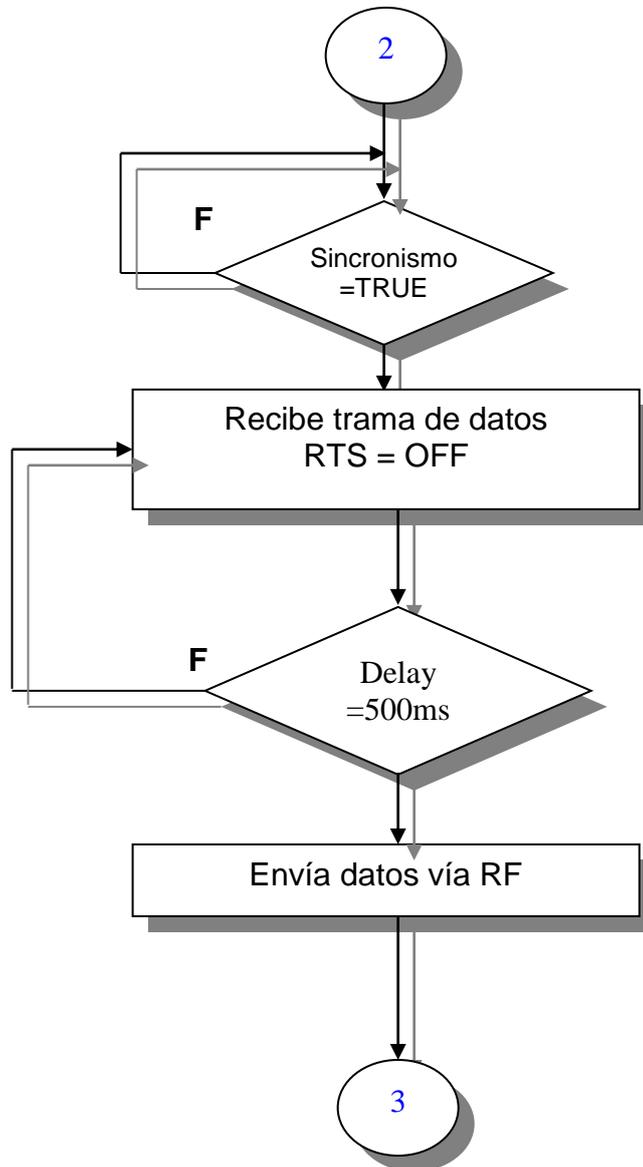


Figura 4.9b – Diagrama de flujo de PicModem

Maestro i2c:

Este programa esta encargado de recibir los datos vía RF, para lo cual comprueba que el primer byte sea el de sincronismo y los cinco datos restantes sean los datos correspondientes de cada servo. La trama de datos debe ser recibida en un tiempo límite de 20ms.

En caso de que no se reciban los datos en este tiempo se descartaran datos y se esperara una nueva trama.

Cuando se ha aceptado una trama como valida se envía los datos correspondientes a cada servo por medio del bus i2c hacia el esclavo.

El diagrama de flujo que se muestra en la figura 4.10a y 4.10b muestra el funcionamiento del programa.

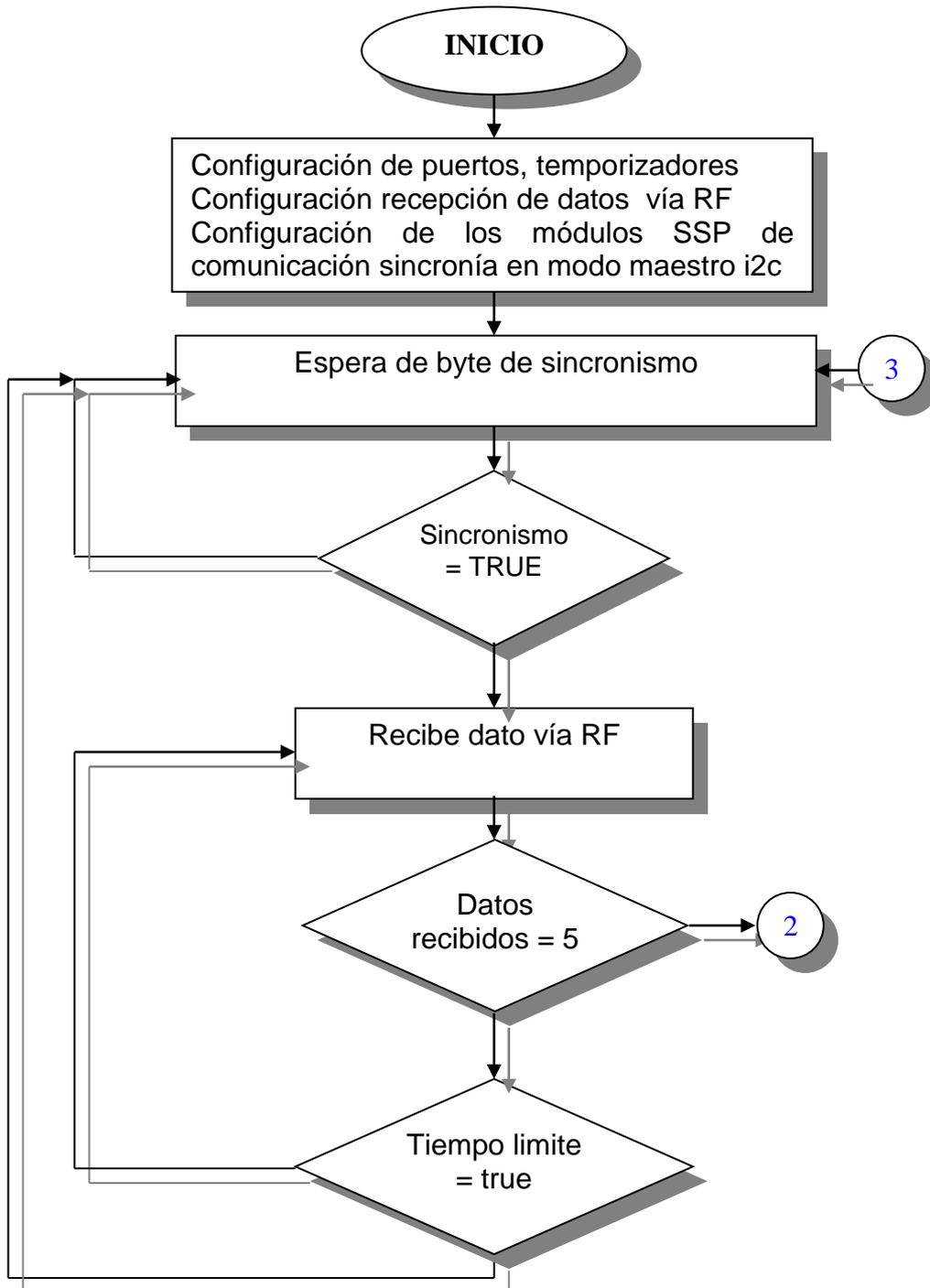


Figura 4.10a – Diagrama de flujo de Maestro i2c

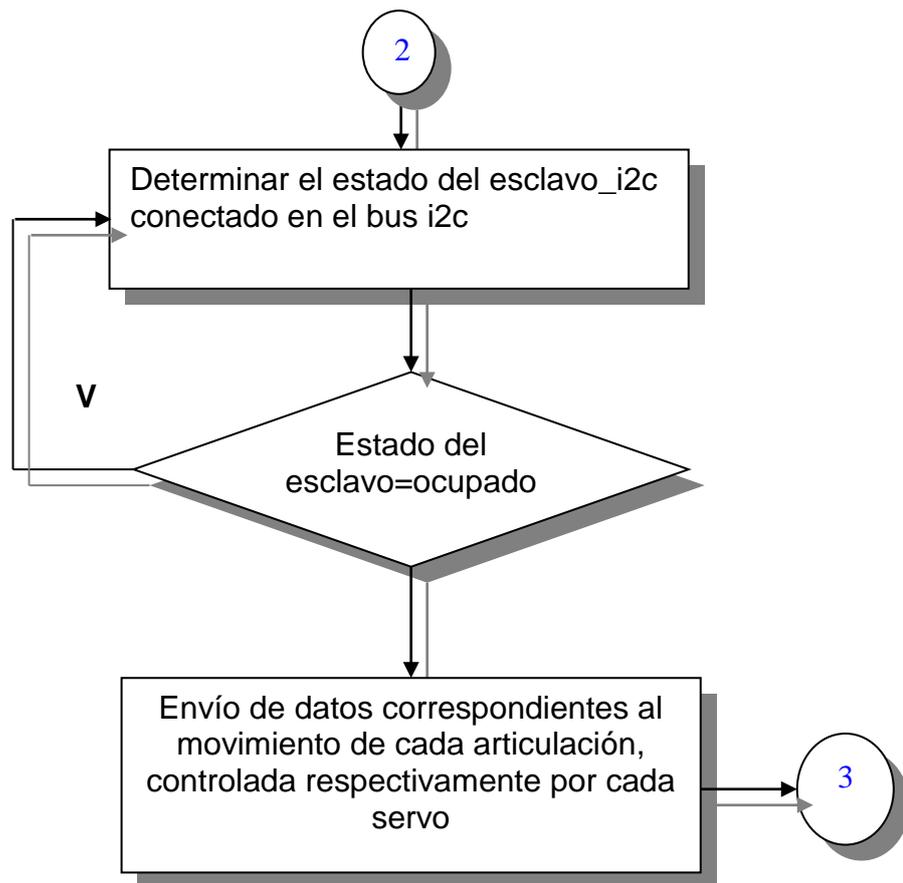


Figura 4.10b – Diagrama de flujo de Maestro_i2c

Esclavo i2c:

Este programa esta encargado del control de movimientos a través de la recepción de datos recibidos del maestro_i2c por medio del bus

i2c. Se recibirán los datos que deben ser tratados para generar el ancho de pulso necesario y respectivo de cada articulación (servomotor). La figura 4.11.a y 4.11b describen el algoritmo de este programa. En este se observa que se ha programado como único esclavo y cabe recalcar que se pueden conectar más esclavos a este maestro, dependiendo del direccionamiento que se escoja al momento de fijar sus parámetros.

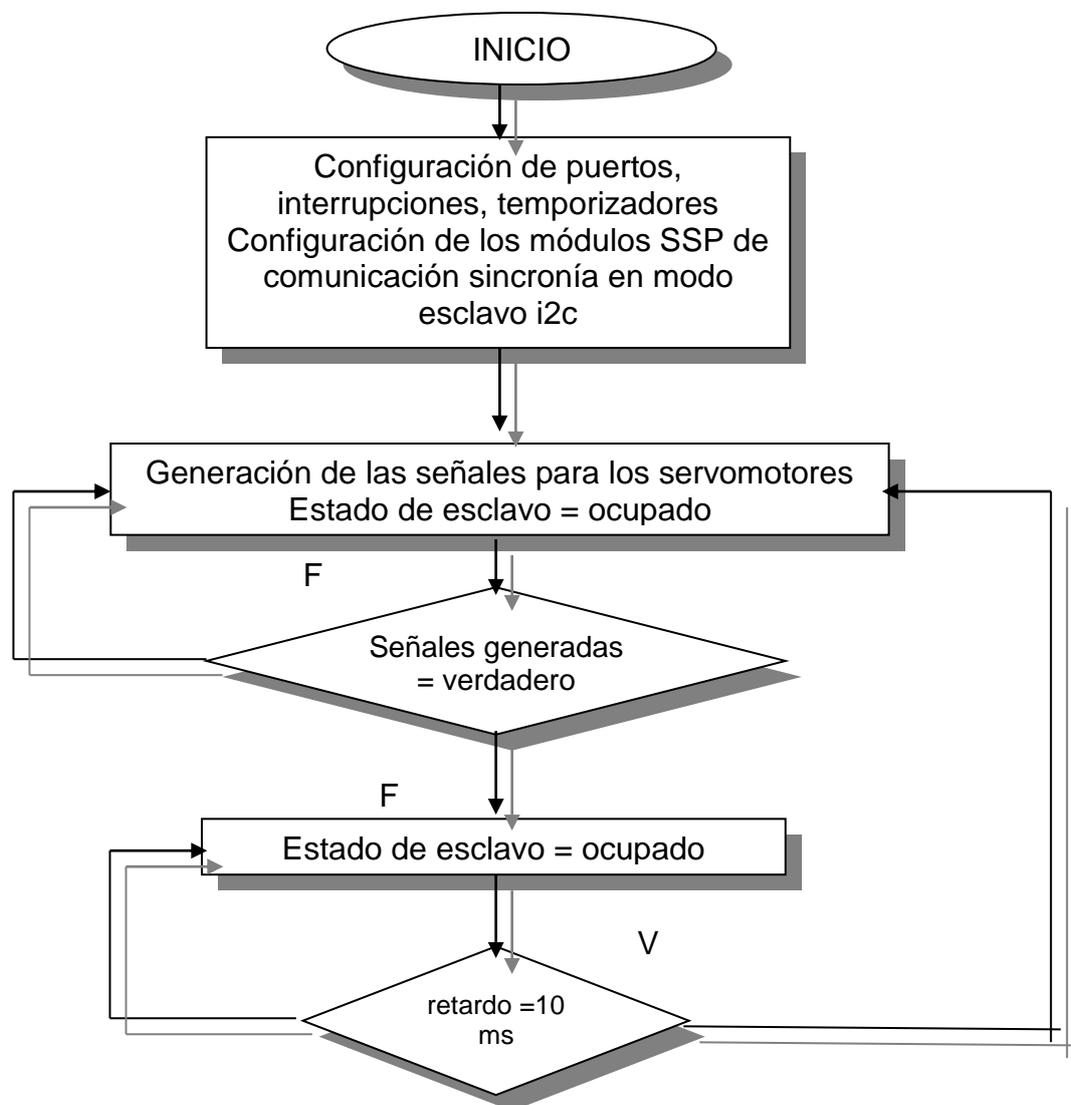
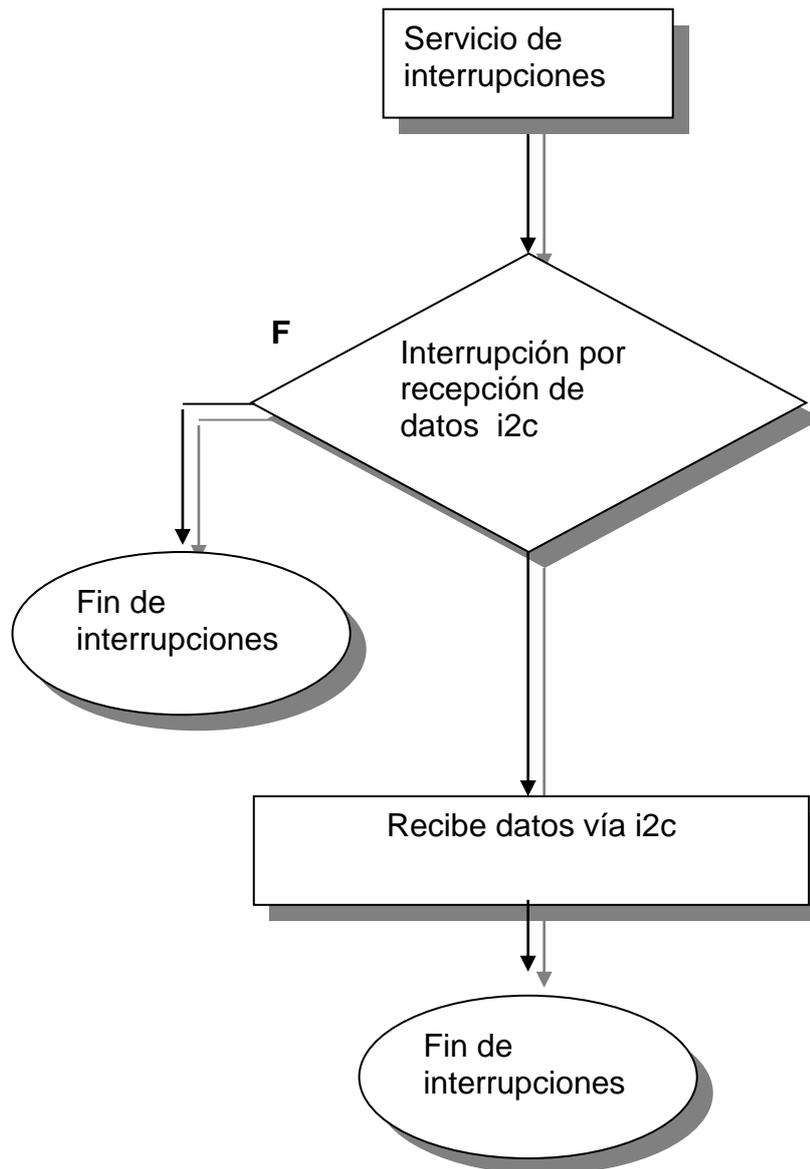


Figura 4.11a – Diagrama de flujo de Esclavo_i2c**Figura 4.11b – Diagrama de flujo de Esclavo_i2c**

Capítulo 5

MANUAL DEL USUARIO



Figura 5.1 – Módulo transmisor, receptor y el brazo robot



Figura 5.2 – Otra vista de los módulos transmisor, receptor, brazo robot, cable serie y una fuente de alimentación

5.1 DESCRIPCIÓN DE LAS TARJETAS CONTROLADORAS PARA EL BRAZO ROBOT

Las siguientes descripciones del modo de operación, conexión y manipulación de las tarjetas que controlan el brazo robot se dividirán en dos partes.

La primera abordará la descripción de la tarjeta PicModem, su conexión, indicadores de estado y otras características que se describirán posteriormente.

Caso similar con la segunda tarjeta (Tarjeta receptora), la cual esta encargada de controlar los movimientos del brazo robot.

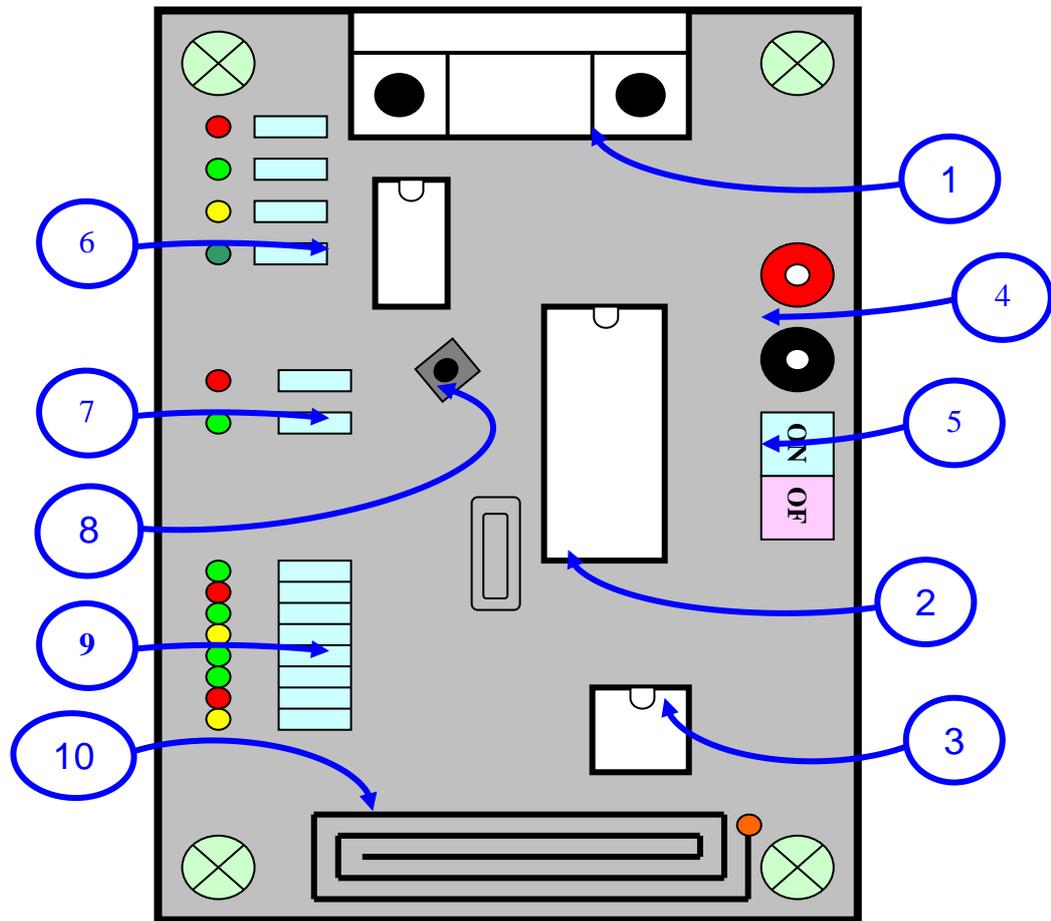
PicModem:

Figura 5.3 - Descripción de tarjeta PicModem

El nombre de los principales y más representativos elementos de la tarjeta que se muestra en la figura 5.3 se detalla a continuación:

1. Conector DB9 hembra.
2. PIC16F877A.
3. TXM-433 LC-S, transmisor inalámbrico RF.
4. Conectores de alimentación (Vcc /Gnd).

5. Interruptor de dos vías (Encendido / apagado).
6. Leds indicadores del puerto serie (RTS, CTS, RX, TX).
7. Leds indicadores de estado de ejecución del programa.
8. Botón de reset general del microcontrolador.
9. Leds indicadores de datos.
10. Antena (50Ω de impedancia).

La descripción de cada punto anterior viene a continuación:

- 1.- Aquí, debe ser conectado un conector DB9 macho / hembra y conectarse al puerto serie de la PC para recibir los datos de la posición de cada servomotor.
- 2.- El PIC16F877A debe ser insertado en esta ubicación, este contiene el programa que controla el envío y recepción de datos¹, tanto desde la PC como vía RF.
- 3.- Transmisor RF, encargado de enviar los datos inalámbricamente **[14]**.
- 4.- La alimentación en estos conectores puede ser de hasta +12Vdc, respetando las polaridades, además, esta tarjeta consta de un regulador de voltaje (LM7805), que es capaz de suministrar hasta un máximo de 5Vdc/500mA.

¹ Nota: Colocar cuidadosamente el PIC16F877A teniendo en cuenta la ubicación del pin1, y la muesca del microcontrolador como referencia para la correcta ubicación del mismo

- 5.- Interruptor de dos vías, destinado a activar la alimentación de la tarjeta ó en su defecto desactivarla.
- 6.- Leds indicadores del estado de las líneas de control. El led rojo representa a la línea RTS, el verde claro CTS, el amarillo TX y por ultimo RX.
- 7.- Leds que indican el estado de ejecución del programa. El led rojo indica que se esta recibiendo datos desde la PC, mientras que el verde indica que la transmisión vía RF es activa. Ambos leds se encienden de manera excluyente, es decir, uno solo a la vez.
- 8.- Reset general para el microcontrolador.
- 9.- Ocho leds que indican el último byte que fue recibido desde la PC en formato hexadecimal
- 10.- Antena de 20cm de longitud.

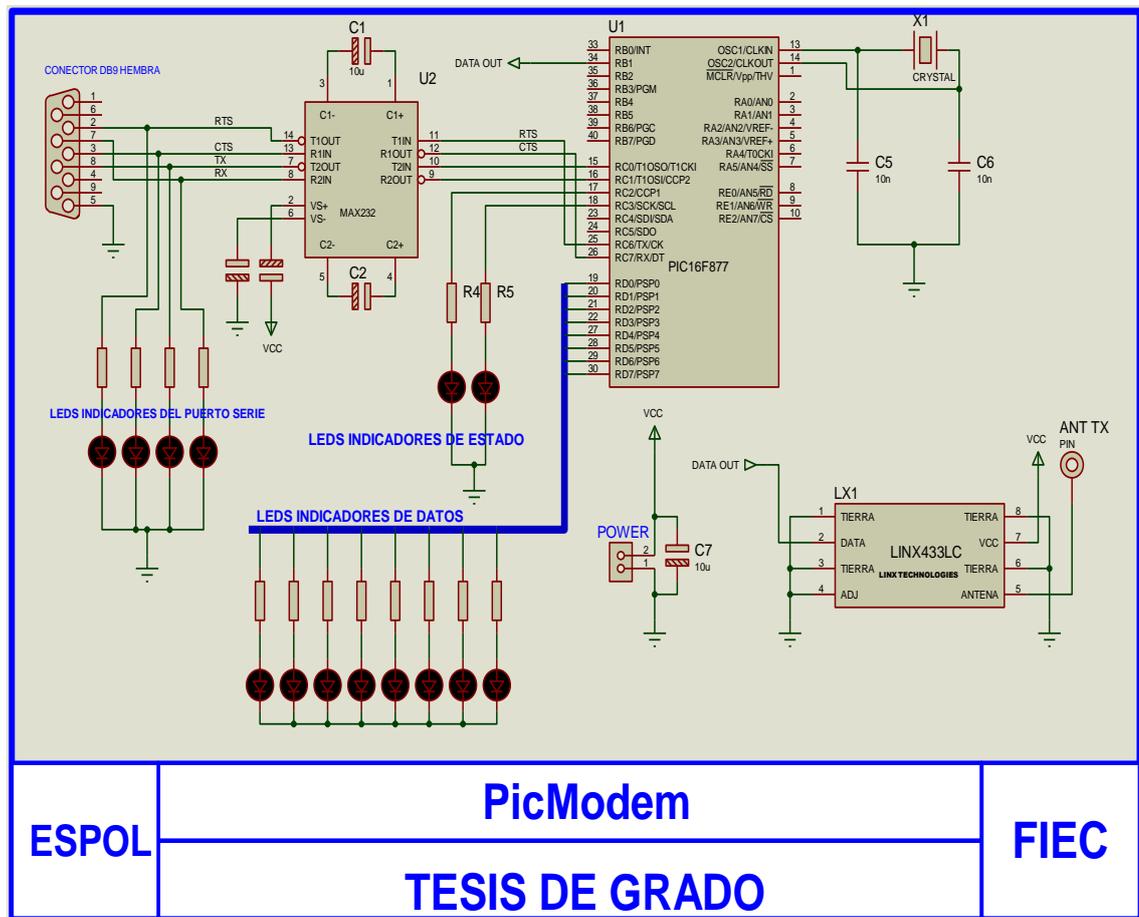


Figura 5.4 - Diagrama eléctrico de tarjeta PicModem

La figura 5.4 muestra las conexiones eléctricas de la tarjeta PicModem donde figuran los elementos mencionados anteriormente. Aquí se puede apreciar en detalle la conexión de los leds indicadores de estado y de datos, el conector DB9 Hembra, el microcontrolador y el transmisor RF.

Tarjeta Receptora:

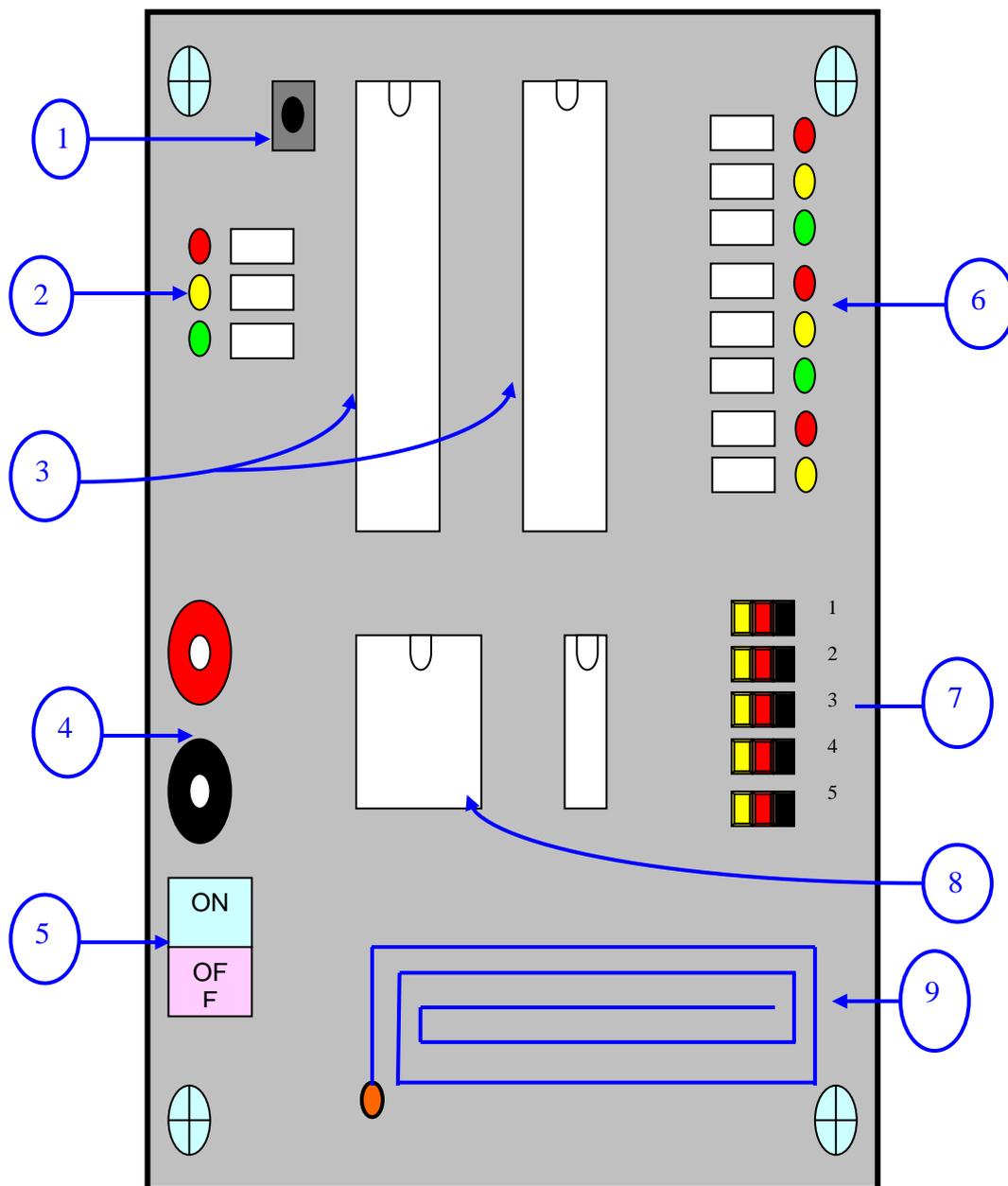


Figura 5.5 - Descripción de TARJETA RECEPTORA

Los elementos más representativos en la tarjeta receptora se muestran en la figura 5.5, la cuál muestra la ubicación espacial de los elementos en la misma.

Los nombres de dichos elementos se los menciona a continuación:

1. Botón de reset general del microcontrolador.
2. Leds indicadores de estado de ejecución del programa.
3. microcontroladores conectados mediante el bus I2C (maestro_i2c y esclavo_i2c)
4. Conectores de alimentación (Vcc /Gnd).
5. Interruptor de dos vías (Encendido / apagado).
6. Indicadores de datos.
7. Salidas de control de servomotores
8. RXM-433 LC-S, receptor inalámbrico vía RF **[13]**.
9. Antena (50 Ω de impedancia).

La descripción de cada punto anterior viene a continuación:

- 1.- Reset general para el microcontrolador.
- 2.- Los leds indican si se esta recibiendo datos vía RF o se están actualizando los datos vía i2c.

- 3.- En el maestro_i2c se carga el programa de recepción de datos vía RF y comunicación i2c, mientras que el esclavo esta programado para generar las salidas moduladas del control de los servomotores.
- 4.- Conectores de alimentación. El conector rojo representa Vcc (+5Vdc) y conector negro representa tierra (Gnd).
- 5.- Interruptor de dos vías, destinado a activar la alimentación de la tarjeta ó en su defecto desactivarla.
- 6.- Ocho leds que indican el último byte que fue recibido desde la PC en formato hexadecimal con lo cual se comprueba también su correcto funcionamiento.
- 7.- Las salidas moduladas de los servomotores son capaces de controlar un rango de rotación de 0° hasta 180° para cada servo. Los pines de color negro hacen referencia a su conexión a tierra, los pines de color rojo a +Vcc y por ultimo los pines amarillos a la señal de control de su respectivo servomotor. Estos pines son los conectores estándares para los servomotores de Hitech, pues cabe resaltar que para otros fabricantes esta referencia tiene modificaciones.
- 8.- Receptor RF, encargado de recibir datos inalámbricamente.
- 9.- Antena de 20 cm de longitud.

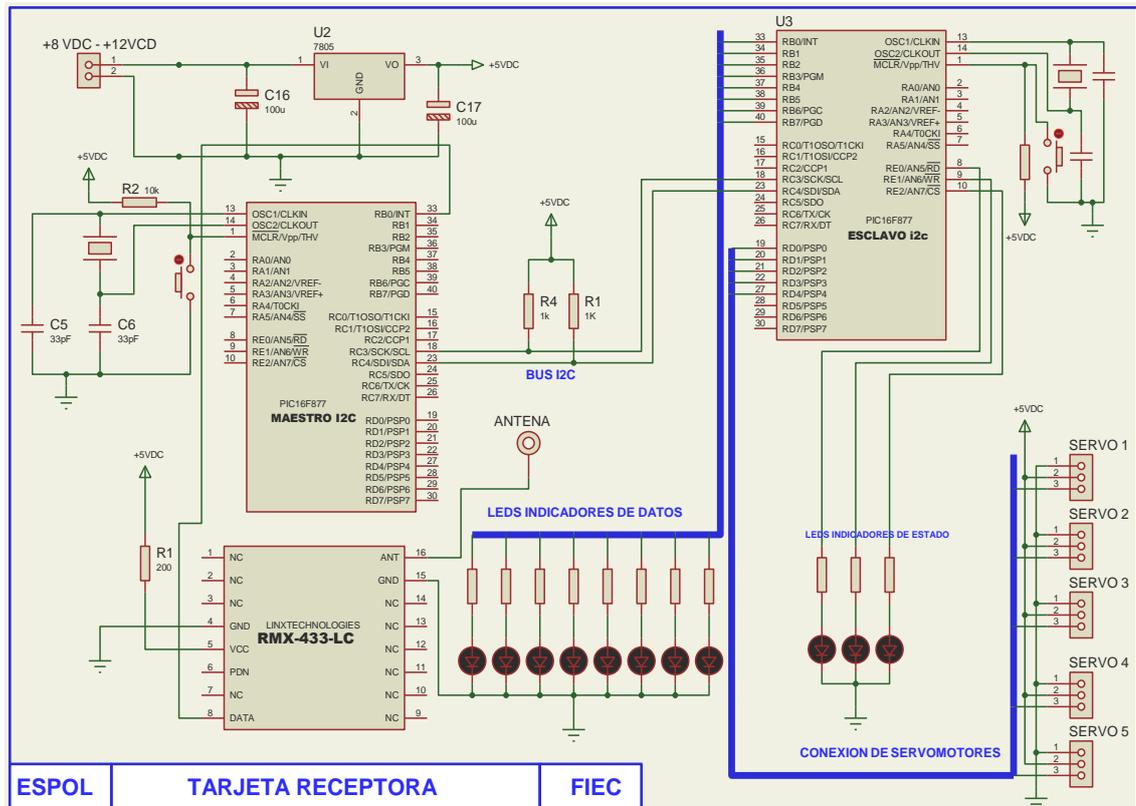


Figura 5.6 - Diagrama eléctrico de TARJETA RECEPTORA

La figura 5.6 muestra las conexiones eléctricas de la tarjeta receptora donde se observa los elementos mencionados anteriormente. Aquí se puede apreciar en detalle la conexión de los leds indicadores, tanto de estado de datos, conectores de los servomotores, el microcontrolador y el receptor RF.

A continuación se explicaran los pasos necesarios para la correcta ejecución del software, tanto como las conexiones de las tarjetas PicModem y tarjeta receptora.

Si se va a instalar la aplicación por primera vez debe:

1. Copiar el contenido de la carpeta OCX en C:\WINDOWS\system.
2. Ejecutar el archivo Control Brazobot.exe
3. Seleccionar entre las opciones de la pantalla principal

Una vez realizado esto, todos los componentes ya están instalados en el sistema y la aplicación los llamara cuando sean requeridos.

La conexión de las tarjetas se describirá a continuación:

1. Conectar entre la placa PicModem y la PC un cable DB9 con terminales Macho / Hembra.
2. Conectar como alimentación una fuente de poder, cuyo valor puede variar de + 8Vdc a +12Vdc.
3. Antes de encender el interruptor de dos vías, comprobar que la polaridad de la fuente sea la misma que la de la tarjeta, es decir el conector rojo conectado a +Vcc y el conector negro a tierra². Una vez que se ha tomado esta precaución encienda la tarjeta.

Para la conexión de la tarjeta receptora, debemos tomar los siguientes pasos para un correcto funcionamiento:

² Nota: Utilizar un multímetro en caso de que su fuente tenga diferentes colores asignados y para comprobar que el valor de la fuente sea de +5Vdc.

1. Conectar cada servo a su respectiva salida, tomando en cuenta que el cable color negro se conecta a tierra, el cable rojo a +Vcc y el cable amarillo a la señal de control. Estos toman referencia a los colores ya definidos en la tarjeta receptora, tal y como se observa en la figura 5.4.
2. Conectar como alimentación una fuente de + 5Vdc/800 mA respetando la polaridad ya explicada anteriormente.
3. Antes de encender el interruptor de dos vías de la tarjeta receptora, comprobar que la polaridad de la fuente sea la misma que la de la tarjeta, es decir, el conector rojo conectado a +5Vdc y el conector negro a tierra. Tomada esta precaución encienda la tarjeta receptora.
4. Sujete el brazo robot a una base fija para evitar que sufra alguna avería o daño permanente debido a alguna caída o maniobra inadecuada.

5.2 CONEXIONES DE LOS DIFERENTES SERVOMOTORES DEL BRAZO ROBOT.

El brazo robot consta de seis servomotores, cada uno de ellos tiene una ubicación específica y definida tanto en el brazo robot como en la

tarjeta receptora. Dichas posiciones están restringidas a un orden específico, tal y como se muestra en la figura 5.3, definida por la propia morfología de la estructura del brazo robot. La posición de cada articulación, es dada por una asignación numérica³ (números de 1 hasta 5), mientras la figura 5.8a muestra la posición del servo 1, el cual representa a la base de nuestro brazo robot y debe ser conectado en la posición 1 de la tarjeta tal y como muestra la figura 5.8b.

Para el movimiento del hombro se conectan los servos 2 y 3, en la segunda posición de la tarjeta. El movimiento del codo esta representada por el servo 3, el mismo que debe ser conectado en la tercera posición de la tarjeta.

La muñeca representa al servo 4 y debe ser conectada en la cuarta posición de la tarjeta y por ultimo la pinza representada por el servo 5 se conecta en la quinta posición.

³ Nota: Cinco asignaciones numéricas, una por cada grado de libertad y por cada salida de la tarjeta receptora.

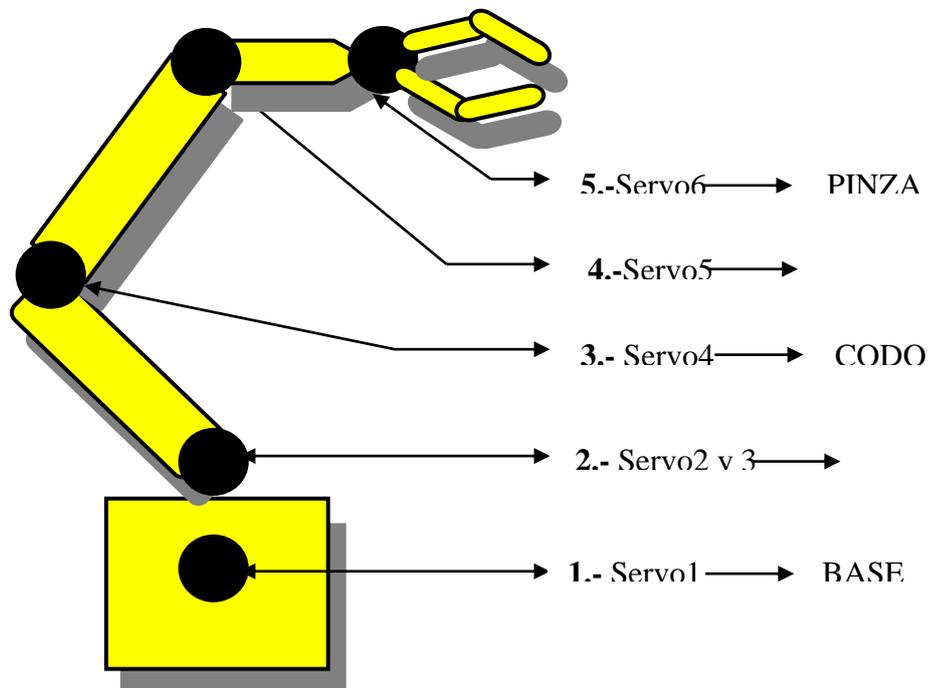


Figura 5.7 – Descripción de cada articulación del brazo robot con una parte del brazo humano

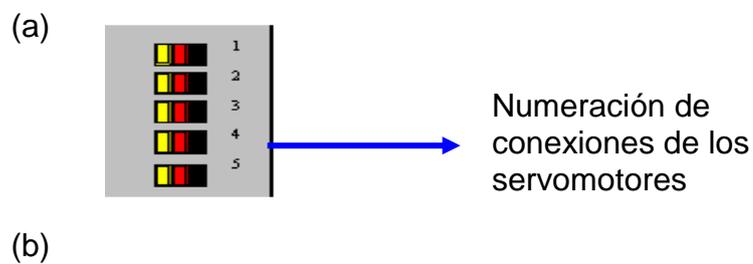


Figura 5.8 - Conexión de los servomotores

(a): Posición de los servomotores con respecto a su morfología.

(b): numeración de conectores para los servomotores y su respectivo orden en la tarjeta receptora

Capítulo 6

Resultados

Se realizaron cálculos teóricos para la generación de los pulsos, basados en una relación establecida entre los datos enviados, desde el maestro hacia el esclavo, y el tiempo que se necesitaba para generar el ancho de pulso de cada servomotor.

Los resultados obtenidos muestran que para desplazamientos cortos se tiene un grado de error relativamente mayor en comparación con los desplazamientos largos, en cuyo caso el error es despreciable. Para fines prácticos dicho error no representa mayor inconveniente, restringiendo el comentario a nuestro caso en particular, en el cual el error promedio es de 0.96%.

La relación representada en la figura 6.1 muestra lo mencionado anteriormente. Los cálculos fueron desarrollados en Microsoft Excell [21].

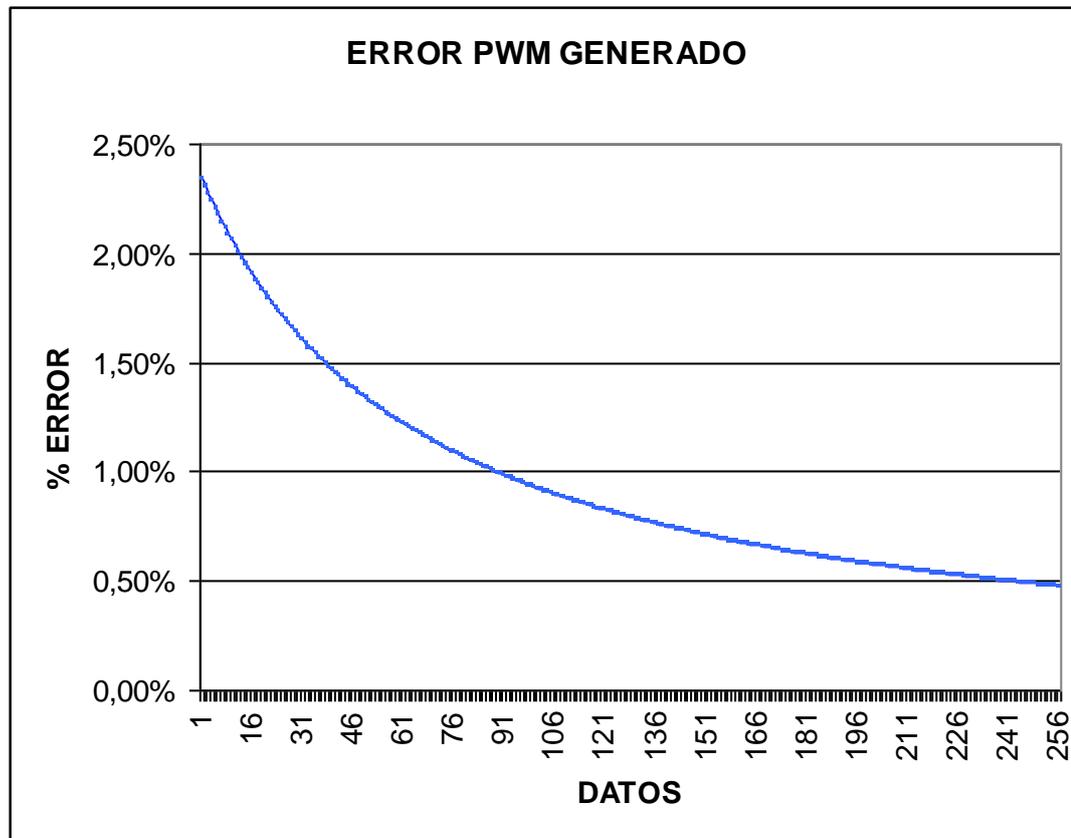


Figura 6.1. Grafico de Error vs. Datos

En cuanto al diseño, el uso de dos microcontroladores en la tarjeta receptora, lo cual a primera vista se observa como un sobre dimensionamiento en el diseño, la justificación de dicha selección esta basada en la velocidad máxima de transmisión de los módulos inalámbricos (4800BPS), la comunicación entre microcontroladores (i2c) y la trama de datos usada para el envío de datos desde la PC. El problema surgió con la primera opción de trama de datos que se decidió utilizar (byte sincronismo + dato servo + dato posición). El tiempo que toma la recepción de la trama

anterior es de 6.25ms, lo cual interrumpiría la generación del pulso, cuyo intervalo de tiempo en alto es de 0.5ms a 2.5ms (2ms de periodo y un periodo total de 20ms), este hecho interrumpía el movimiento continuo de los servos, provocando movimientos erróneos y no deseados por el usuario. La opción escogida fue, conectar un microcontrolador en función de maestro y otro microcontrolador como esclavo, ambos conectados por medio del bus i2c, donde el maestro se encargaría de la recepción inalámbrica y el envío de datos al esclavo, mientras el esclavo se encarga de la generación de los pulsos (PWM), para el control de movimiento del brazo robot, con lo cual se soluciona el problema de generación de PWM y recepción de datos y además permite tener una visión a futuro del proyecto, pues, el bus i2c permite la conexión de otros periféricos, tales como memorias, sensores y otros microcontroladores en modo de esclavo ó en su defecto maestro.

Las dimensiones de la tarjeta en comparación con otras similares en el mercado, no presta mayor beneficio, debido a la tecnología usada en la fabricación de la misma, como por ejemplo, si se hubiesen utilizado elementos de soldadura superficial o realizar una placa de doble cara ó multicapas con dichos elementos, sus dimensiones se reducirían de manera considerable.

Por otro lado su precio final, al utilizar módulos RF y no haber adquirido transceivers, cuyo costo es elevado en comparación con los módulos, reduce el costo final considerablemente sin quitarle prestaciones al diseño.

En cuanto al software desarrollado para el control del brazo robot, se destaca el uso del modulo Mscomm, el cual es un componente de Visual Basic con su librería llamada Microsoft Comm Control 6.0, el cual debe ser agregado como un componente a la aplicación. El uso de este componente brinda la facilidad para el programador de envío y recepción de datos por el puerto serie, permitiendo la asignación de todos los parámetros necesarios para la comunicación serial, tales como velocidad, paridad, numero de bits de stop y el control de flujo de datos (handshacking) que en el caso particular, fue el manejo de las líneas CTS Y RTS para el envío y recepción de datos.

Para el control de los servomotores se pensó en un driver (ULN2003A), pero el uso de este generaba demasiadas corrientes de retorno al circuito. Para resolver este problema se decidió que las líneas del microcontrolador vallan directamente a las entradas de control de los servomotores y además en la placa de los elementos se separe una línea de tierra para el circuito de control y otra línea de tierra para los servomotores con lo cual se resolvió este inconveniente.

Finalmente sé probó todo el sistema integrado donde el usuario define los movimientos que deseen que realice el brazo robot, cuando el operador determina una dirección se presenta inmediatamente la visualización del movimiento que el brazo robot debe realizar, de manera simultánea al cálculo de los ángulos de las articulaciones que se deben enviar al brazo robot para que las ejecute.

Para lograr una mayor eficiencia en el tiempo de ejecución y respuesta, se utilizaron cristales de cuarzo de 4MHZ y de 20 MHZ, los resultados obtenidos, como era de esperarse, el cristal de 20MHZ supera en relación de cinco a uno al de 4 MHZ, mejorando la eficiencia y el tiempo de ciclo de máquina.

Existen dos aspectos de la comunicación I2C que no se han implementado ya que el diseño no lo requería. El primer aspecto es el tema del Arbitraje.

El PIC16F877A es la única unidad principal (modo maestro) por lo que no se tomó en cuenta dicha situación y sin embargo el análisis de dicha situación no representaría mayor dificultad en el desarrollo del algoritmo.

El segundo aspecto no implementado son los estados de espera. El microcontrolador en modo esclavo es lo suficientemente rápidos por lo que

tampoco hubo la necesidad de que el diseño del algoritmo de control para el esclavo tome en cuenta dicha situación.

Como reseña, cabe destacar que el presente trabajo permite una firme introducción al desarrollo de robots, tratando de que este sea un pequeño paso de un largo camino a recorrer, en el cual no pretendemos abarcar todo el espectro de temas concernientes al mismo, tanto en la Ingeniería en Computación como en la Electrónica, sin dejar de esperar un excelente trabajo realizado y que la implementación lograda valide el modelo propuesto.

CONCLUSIONES

1. Para resolver la baja velocidad de los módulos de transmisión RF, fue necesario implementar el bus i2c, para la recepción de datos vía RF, por parte del maestro y la generación de las señales PWM por parte del esclavo, quien controla los servomotores. Este hecho aumenta el volumen de dispositivos usados en la aplicación, pero le da un agregado de versatilidad para futuras expansiones y manejo de periféricos que se conecten, mediante el bus i2c.
2. El uso de servomotores reduce la parte de control de posición de motores DC, permitiendo enfocar el diseño en otros puntos de igual o mayor relevancia, sin restarle ni reducir valor al trabajo.
3. El control de flujo de datos mediante el control de las líneas RTS y CTS, se presentó como la opción más práctica en lo que se refiere a control de flujo de datos, pues con la activación de las líneas o su desactivación se impide la transmisión y recepción de datos.
4. El uso del Mscmm permite la interacción entre el puerto serie y un microcontrolador, de una manera eficaz y sencilla, tanto en transmisión como recepción. La única limitación es la velocidad

máxima del módulo Mscomm (28.8 Kbps), tomando en cuenta que para el presente trabajo se utilizó 9600bps lo cuál no presentó inconveniente alguno.

5. La utilización de Microsoft Access enlazado con Visual Basic, presta facilidades para la visualización de los datos numéricos, mediante TextBox, ComboBox y ListBox, los cuales se conectan directamente con la base de datos desde su ventana de propiedades.
6. Sentir la satisfacción de terminar un proyecto y con el cual se ha aprendido partes fundamentales en el inicio del armado de un robot.
7. Se usa servomotores porque ayuda a dar una posición más exacta en el movimiento de las diferentes partes del brazo robótico con el uso del PWM. En este caso, los servomotores ya venían con una retroalimentación interna para su control.
8. Se utiliza dispositivos inalámbricos, con los cuales se facilita el uso del control de diferentes componentes que pueden estar muy alejados y no tendríamos que tener un cableado muy extenso.

9. Se linealizan los datos recibidos por I2C para determinar el tiempo, posición y giro de cada servomotor debido a que su período de trabajo es de 0.5 mseg a 2.5 mseg.

10. Se utilizó todos los elementos que se tuvo a la mano, para no perder mucho tiempo trayendo cosas del exterior que pueden hacer perder el tiempo o se dificultaría su uso.

RECOMENDACIONES

1. Se recomienda conectar entre la alimentación (+ 5Vdc), del modulo receptor una resistencia de 200 Ohm, en el caso de que la alimentación sea +5 Vdc y una conexión directa a +Vcc si la alimentación es de +3 Vdc; este hecho como norma del fabricante [13].
2. Se deben conectar al bus i2c resistencias PULL - UP, cuyos valores fluctúan entre 1K Ohm y 10K Ohm; este valor depende de la velocidad de transmisión, donde el valor más bajo de resistencia se usa con velocidades más elevadas.
3. Para la conexión de las líneas de control, en el integrado MAX232A, tomar en cuenta que éste integrado consta de dos entradas y dos salidas tipo TTL, dos entradas y dos salidas tipo CMOS, donde las conexiones de tipo TTL se conectan al PIC16F877A y las de tipo CMOS van al puerto serie. La incorrecta conexión dañaría el microcontrolador o el puerto serie, el cual está insertado en el mainboard de la PC. Se deben tomar en cuenta también el valor correcto de los capacitores electrolíticos que deben conectarse en el MAX232, varían su valor en función del fabricante.

4. La programación en ensamblador, facilita la activación de las interrupciones, pudiendo darles en un cierto momento del programa una prioridad alta o baja en su ejecución, posponiendo otras que en su momento no son lo más relevante para el control.
5. En Visual Basic los lazos de control “While” deben poseer la instrucción DoEvents, la cual evita que el programa principal enfoque toda su atención al lazo While, y en su defecto se pueda ejecutar otras opciones al mismo tiempo sin correr el riesgo de que el bucle inhiba el programa.
6. En el momento del ensamblaje del brazo robot, tomar en cuenta la posición del eje y el sentido de rotación de cada servo, para darle a cada articulación y eslabón el movimiento que se desee realizar, según la morfología a simular.
7. Se debe tener precauciones al soldar los módulos RF, pues el fabricante da un límite máximo de +225 °C por un tiempo máximo de 10 segundos, lo recomendable es usar un Cautín de 20W, y al momento de llevar a cabo la soldadura de cada pin en la PCB, tomar lapsos de espera de algunos segundos esperando que se disipe el calor del chip para seguir con el proceso de soldadura. Otra

recomendación es la de usar zócalos, para todos los integrados, pues llevado a cabo esto, se pueden cambiar con facilidad, en el caso específico de los microcontroladores. En ocasiones se necesita cambiar el programa del microcontrolador.

- 8.** Se recomienda usar pasta disipadora de calor para los elementos cuya temperatura tenga que ser disipada por medio de un disipador de calor, tal como en el caso del LM7805; teniendo en cuenta que la pasta debe estar entre el disipador de aluminio y el elemento a refrigerar, removiendo los excesos. El rango de disipación de la pasta es de 40 a 400 ° F.
- 9.** Si se usan cristales de 20MHZ, se recomienda revisar la palabra de configuración del PIC, pues en nuestro caso, este pequeño e ínfimo detalle, nos dio más de un dolor de cabeza y perder tiempo valioso.
- 10.** Verificar los cables de conexión de los servomotores, ya que los diferentes fabricantes saben variar en el orden o en el color de los cables.
- 11.** Utilizar multímetro y si es posible un osciloscopio para verificar todos los voltajes y formas de onda, para estar seguro de lo que se realiza y

nos ayudan a solucionar muchas dudas que aparecen en el momento del ensamblaje.

- 12.** Colocar una tierra independiente para los servomotores para un mejor funcionamiento y en especial al hacer la placa.
- 13.** Es bueno la utilización de simuladores, los cuales ayudan a tener una idea más general de lo se hace y podemos añadir o quitar elementos dependiendo de que respuesta se tiene al simular todo o parte del trabajo.
- 14.** Tener en cuenta los pasos para realizar un buen robot, es decir, los pasos de la Torrebot, que nos ayudará a hacer un mejor robot para el bien de todos.
- 15.** Este proyecto es de gran ayuda para entender mejor el funcionamiento de muchos elementos, los cuales pueden ser usados en otros proyectos o pueden complementarse.

- 16.** Hacer uso de todos los tutoriales y manuales que se tenga sobre cada elemento que se va a usar, para verificar los datos de fabricantes y hacer una buena conexión de los elementos.

- 17.** Por último y la más importante recomendación, es que no existe nada que no se pueda realizar, con esfuerzo y aplicación todo es posible y en ocasiones se obtienen resultados inesperados que nos llenan de satisfacción.

APÉNDICE

APÉNDICE 1 – EVOLUCIÓN DE LA ROBÓTICA (CD)

APÉNDICE 2 – TUTORIAL DE CONSTRUCCIÓN DEL BRAZO ROBOT (CD)

APÉNDICE 3 – INFORMACIÓN SOBRE PWM (CD)

APÉNDICE 4 – INFORMACIÓN DE SERVOMOTORES (CD)

APÉNDICE 5 – INFORMACIÓN SOBRE ARQUITECTURA SAW (CD)

APÉNDICE 6 – MANUAL DEL TRANSMISOR (CD)

APÉNDICE 7 – MANUAL DEL PUERTO SERIAL (CD)

APÉNDICE 8 – MANUAL DEL RECEPTOR (CD)

APÉNDICE 9 - EL INTEGRADO MAX-232 (CD)

APÉNDICE 10 – MANUAL DEL INTEGRADO LM7805 (CD)

APÉNDICE 11 – MANUAL DEL PIC16F877A (CD)

APÉNDICE 12 – DISEÑO DE LAS PLACAS (CD)

APÉNDICE 13 – PROGRAMAS USADOS EN LOS PIC´S (CD)

APÉNDICE 14 – PROGRAMA DE VISUAL BASIC (CD)

GLOSARIO

RTS - Request To Send (Señal de espera para enviar datos).

CTS - Clear to Send (Señal que indica que está listo para recibir datos).

LED - Elemento que produce luz al ser polarizado directamente

Hscroll – Barra deslizante horizontal.

Puerto serie - Entrada y salida de datos por 9 pines de la computadora.

PWM - Modulación por Ancho de Pulso.

RF - Radio Frecuencia.

i2c –Bus de control de transmisión de datos sincronos desarrollado por PHILIPS.

TXSTA – Registro de control de transmisión asincrónica de datos , USART.

RCSTA – Registro de control de Recepción asincrónica de datos, USART.

GUI - Grafic User Interface.

DLL - Dynamic Library Link.

BIBLIOGRAFÍA

- [1] Frédéric Giamarchi, Robots Móviles, Estudio y Construcción, (2000).Editorial PARANINFO THOMSON LEARNING, México.
- [2] Gardner Nigel, “PICmicro MCU C. An introduction to programming The Microchip PIC in CCS C”, Bluebird Electronics, 2002.
- [3] INFORMACIÓN TÉCNICA, (2007).datasheet.Internet,URL:
<http://www.datasheet.com>
- [4] GALICIA, G. (2002) Curso Básico de Microcontroladores PIC.Internet,URL:<http://www.ctrl.cinvestav.mx/Profesores/asl/CursoPi-cs/pics.pdf>.

- [5] SÁNCHEZ VEGA, M. (1954) Estudio comparativo de la concepción mecánica del animal y sus fundamentos en Gómez Pereyra y Renato Descartes. Capítulo VI. Internet, URL: <http://www.filosofia.org/cla/per/1954veg7.htm>
- [6] CDI (Ciencia al Día Internacional). (2001) *Glosario*. No 2, Volumen IV Internet, URL : <http://www.ciencia.cl/CienciaAlDia/volumen2/glosario-i.html>.
- [7] LABOSIN, Laboratorio de Simulación e Investigación Numérica (2001) *¿Qué es un modelo?* Internet, URL: <http://fc.uni.edu.pe/labosin/mural/archivos/mural02.pdf>
- [8] CDI (Ciencia al Día Internacional). (2001) *Glosario*. No 2, Volumen IV Internet, URL: <http://www.ciencia.cl/CienciaAlDia/volumen2/glosario-i.html>

[9] EPSA (Escuela Politécnica Superior de Albacete), Departamento de Informática, Laboratorio de Electrónica (2002). Servomotores. Internet, URL: <http://www.infoabuclm.es/labelec/solar/electronica/elementos/servomotor.htm>.

[10] Datos e información sobre la Arquitectura Saw se lo puede encontrar dentro de CD adjunto, en la carpeta Apéndices.

[11] “LC SERIES TRANSCEIVER MODULE DATA GUIDE”, 2007. Disponible en <http://www.linxtechnologies.com/docs/pdfs/AN00125.pdf>.

[12] APPLICATION NOTE AN-00130, “Modulation Techniques For Low- Cost RF Data Links”, 2007. Disponible en <http://www.linxtechnologies.com/docs/pdfs/AN00130.pdf>.

- [13] "LC SERIES RECEIVER MODULE DATA GUIDE", 2007. Disponible en <http://www.linxtechnologies.com/docs/pdfs/AN00120.pdf>.
- [14] "General Considerations for Sending Data With The LC Series", 2007. Disponible en <http://www.linxtechnologies.com/docs/pdfs/AN00232.pdf>.
- [15] "Aplicaciones cliente servidor comunicación puerto serie: El control Mscomm" 2007.
<http://www.tutoriales.com/VisualBasic/GuíaestudianteCapitulo20>.
- [16] "Objetos Visual Basic: Propiedades, Procedimientos Métodos", 2006.
<http://www.tutoriales.com/VisualBasic/Guíaestudiante\Capitulo2>.
- [17] "Métodos gráficos", 2006.
<http://www.tutoriales.com/VisualBasic/GuíaestudianteCapitulo7>.

- [18] "Manejo de base de datos con Microsoft Access ,2000", 2007. Disponible en <http://ajax.schwarz-interactive.de>.
- [19] "Bases de datos en VISUAL BASIC" 2006. <http://www.tutoriales.com/VisualBasic/Guía de estudiante/Capitulo12>.
- [20] "PIC16F877A Datasheet", Microchip Technology Inc., 2003.
- [21] Resultado de cálculo de errores en la generación del PWM [Anexos CD].