



**INSTITUTO POLITÉCNICO NACIONAL**

**UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS AVANZADAS**

**U P I I T A**

***Trabajo Terminal***

***“Diseño y construcción de un prototipo de robot didáctico ortogonal. Parte A”***,

**que para obtener el título de**

***“Ingeniero en Mecatrónica”***

**Presenta**

**David Castro Sánchez**

**Asesores**

**M. en C. Alfonso Campos Vázquez**

**M. en C. Alejandro Escamilla Navarro**



***México, D. F. a 19 de agosto 2011***



**INSTITUTO POLITÉCNICO NACIONAL**  
**UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS AVANZADAS**

**U P I I T A**

***Trabajo Terminal***

***“Diseño y construcción de un prototipo de robot didáctico ortogonal, Parte A”,***

que para obtener el título de

***“Ingeniero en Mecatrónica”***

**Presenta**

-----  
**David Castro Sánchez**

**Asesores**

-----  
**M. en C. Alfonso Campos Vázquez**

-----  
**M. en C. Alejandro Escamilla Navarro**

**Presidente del Jurado**

**Profesor titular**

-----  
**M. en C. Cecilia Fernández Nava**

-----  
**M. en C. Víctor Darío Cuervo Pinto**



***México, D. F. a 19 de agosto 2011***

## **Agradecimientos**

Agradezco al pueblo de México que aporta al gasto público que permitió que fuese posible mi educación. De la misma forma mis agradecimientos se dirigen al Instituto Politécnico Nacional, mi Alma máter, que por un largo periodo de tiempo me cobijó entre los brazos de la ciencia y la tecnología formando a un profesionista al servicio de la patria. Así como a mi querida escuela, UPIITA.

Agradezco profundamente a mi familia que siempre estuvo pendiente de mi educación.

A mis padres, Víctor Castro Domínguez y Gloria Sánchez Ávila, que les debo mi existir y un amor incomparable, son el motor y la dirección justa de mis pasos. Un gran esfuerzo siempre es recompensado y he aquí la culminación de uno de nuestros grandes sueños.

A mi madrina, Perfecta Beatriz Sánchez Martínez, que siempre veló por mi bienestar cuando las penumbras de la vida llegaron a mi puerta. Agradezco de manera infinita el enorme esfuerzo que implicó apoyarme para la culminación de mis estudios.

A mis hermanos, Víctor y Viridiana Castro Sánchez, que de manera directa hicieron suyos mis alegrías así como mis pesares, gracias por estar ahí cuando más los necesitaba y por acompañarme por esta gran travesía.

Agradezco muy especialmente a mi tía Rufina Sánchez Martínez, que estuvo al pendiente desde el principio de mi formación hasta estos días. De manera similar al resto de mi familia en Acapulco que siempre vieron por mi bienestar, a mis tíos y primos con los que compartí miles de experiencias y satisfacciones, gracias.

De manera general, pero poniendo todo el corazón en las palabras, quiero agradecer a mi familia residente en Chilpancingo, Gro. que permaneció siempre al pendiente mis pasos por mi formación académica, aun cuando la distancia y el tiempo nos impedían estar juntos.

Por la parte académica, agradecer a todos mis maestros la semilla de conocimiento que con la culminación de este trabajo se expresa la cosecha de sus frutos.

A mi maestro, asesor y tutor Alfonso Campos Vázquez le extiendo mi agradecimiento por la paciencia, los conocimientos, la participación activa y las atenciones para la culminación de este proyecto, así como un agradecimiento especial a su Sr. Padre por la asesoría magistral en un punto crucial del trabajo.

A mi asesor Alejandro Escamilla Navarro por la asesoría técnica en el desarrollo del proyecto, pero de manera especial por la confianza en el proyecto y el apoyo para emprender mis ideas.

De la misma manera, a mi secretario Víctor Darío Cuervo Pinto, que con bases firmes me ayudó en gran medida en la corrección de la tesis aquí presentada. Así como las observaciones de la presidenta del jurado Cecilia Fernández Nava que reforzaron el valor de lo aquí presentado.

Agradezco de manera especial a Cecilia Lara Anaya, su valiosa paciencia con este proyecto, así como la ayuda durante su redacción.

A mi amigo, compañero y hermano Christopher Virgen Vázquez, por acompañarme en este largo camino que ha llegado a su fin.

A mi amigo Jaime Cruz Cruz por su ayuda técnica en el desarrollo de la etapa de potencia, así como a todos los amigos que hice en la carrera que siempre me dieron aliento para continuar.

“¡No sólo no hubiéramos sido nada sin ustedes, sino con toda la gente que estuvo a nuestro alrededor desde el comienzo; algunos siguen hasta hoy! ¡Gracias totales!” – Soda Stereo.

## Dedicatoria

Dedico primeramente este proyecto a mi madre Gloria Sánchez Ávila, por creer en él y permitirme soñar. Por apoyarme siempre en todas mis decisiones, por darme alas para volar y siempre estar ahí para mí, aun en mis equivocaciones. Simplemente te amo y agradezco todo lo que haces por mí día a día.

Porque tu dolor fue mi dolor, tu esperanza fue mi esperanza y tu esfuerzo mi inspiración, dedico este trabajo terminal a mi padre, Víctor Castro Domínguez que en los momentos más difíciles fue un impulso para seguir adelante, así como siempre ha sido mi gran ejemplo a seguir.

Con todo el cariño posible dedico esta tesis a mi madrina Beatriz Sánchez, porque en cada paso que daba pensaba en demostrar lo mucho que valía su esfuerzo, ahora plasmado en la terminación de mis estudios.

Dedico esta tesis a toda mi familia: a mis hermanos, tíos, tías, primos y primas. Para todos es este gran logro.

Dedico de manera especial este trabajo al amor de mi vida, Cecilia Lara Anaya. Porque solo es un pasito más, porque siempre existirán gusanitos y mariposas, porque los cuentos de hadas siempre terminan en final feliz, porque *I love you 'till the end*.

A mis hermanos de lucha, los PUERCORPS, porque siempre hay algo que mejorar y siempre seremos *'más puercos que nunca'*.

Dedico este trabajo a quien fue dirigido, a ti compañero que estas formándote al igual que yo lo hice. Nunca dejes que alguien te niegue tus sueños, siempre lucha por hacerlos realidad.

'Y así después de esperar tanto, un día como cualquier otro decidí triunfar. Decidí no esperar a las oportunidades sino yo mismo buscarlas, decidí ver cada problema como la oportunidad de encontrar una solución, decidí ver cada desierto como la oportunidad de encontrar un oasis, decidí ver cada noche como un misterio a resolver, decidí ver cada día como una nueva oportunidad de ser feliz. Aquel día descubrí que mi único rival no eran más que mis propias debilidades, y que en éstas, está la única y mejor forma de superarnos. Aquel día dejé de temer a perder y empecé a temer a no ganar, descubrí que no era yo el mejor y que quizás nunca lo fui. Me dejó de importar quién ganara o perdiera; ahora me importa simplemente saberme mejor que ayer. Aprendí que lo difícil no es llegar a la cima, sino jamás dejar de subir. Aprendí que el mejor triunfo que puedo tener, es tener el derecho de llamar a alguien «Amigo». Descubrí que el amor es más que un simple estado de enamoramiento, «el amor es una filosofía de vida». Aquel día dejé de ser un reflejo de mis escasos triunfos pasados y empecé a ser mi propia tenue luz de este presente; aprendí que de nada sirve ser luz si no vas a iluminar el camino de los demás. Aquel día decidí cambiar tantas cosas... Aquel día aprendí que los sueños son solamente para hacerse realidad. Desde aquel día ya no duermo para descansar... ahora simplemente duermo para soñar'

– Walter Elias Disney.

## Índice de tablas

Tabla 2.1 Cronograma de actividades para Trabajo Terminal I.....	17
Tabla 2.2 Cronograma de actividades para Trabajo Terminal II.....	18
Tabla 3.1 Tabla de especificaciones de la gama MHY2 de pinzas neumáticas SMC®.....	23
Tabla 3.2 Asignación de valor numérico a los estados lógicos del codificador.....	41
Tabla 3.3 Asignación de señal de control PWM por rangos de error.....	44
Tabla 4.1 Mediciones de voltaje cuando se suministra una señal de PWM.....	49
Tabla 4.2 Mediciones de la velocidad angular con señales de PWM.....	55
Tabla 5.1 Gastos del proyecto cotizados en Febrero del 2011 con actuadores de vida útil media.....	58
Tabla 3.2 Gastos del proyecto generados en el desarrollo del proyecto.....	58

## Índice de figuras

Figura 2.1 Esquema a bloques y primer estudio del prototipo realizado con AutoCad 2009®.....	12
Figura 2.2 Estructura de la interfaz gráfica.....	13
Figura 2.3 Estructura del control de mando.....	14
Figura 2.4 Desglose de las unidades de mando.....	14
Figura 2.5 Estructura del sistema mecánico.....	15
Figura 2.6 Desglose de las unidades de señal.....	16
Figura 3.1 Diagrama de flujo del prototipo.....	20
Figura 3.2 Esquema de la pinza y el diagrama de cuerpo libre.....	22
Figura 3.3 Figura del circuito neumático para el robot ortogonal.....	24
Figura 3.4 Diagrama de pines del microcontrolador PIC18F4550 en la tarjeta de control.....	27
Figura 3.5 Principales conectores en la tarjeta de control del prototipo.....	29
Figura 3.6 Activación de electroválvulas, inversores <i>Schmitt Trigger</i> y alimentación.....	30
Figura 3.7 Circuito Puente H implementado.....	32
Figura 3.8 Circuito optoacoplador para la señal de control de motores.....	33
Figura 3.9 Esquemático completo de la tarjeta de control.....	34
Figura 3.10 Diagrama de flujo de la operación de control.....	35
Figura 3.11 Pirámide de los componentes conectados al <i>Host</i> .....	37
Figura 3.12 Diagrama del HUB.....	38
Figura 3.13 Diagrama de las velocidades de transferencia USB.....	38
Figura 3.14 Diagrama del cable USB.....	38
Figura 3.15 Flujo de información en la comunicación USB.....	39
Figura 3.16 Señales de salida de un codificador de cuadratura.....	40
Figura 3.17 División de las señales de salida de un codificador de cuadratura.....	40
Figura 3.18 Esquema de un motor de CD.....	41
Figura 3.19 Modelo a bloques del control de posición.....	43
Figura 4.1.1 Unidad de mantenimiento para la alimentación neumática.....	45

Figura 4.1.2 Bloque de electroválvulas montado en la mesa de trabajo.....	46
Figura 4.1.3 Conexiones neumáticas en los manipuladores.....	46
Figura 4.2.1 Circuito PCB de la tarjeta de control.....	47
Figura 4.2.2 Tarjeta de control implementada y conectada al prototipo.....	48
Figura 4.4.1 Gráfica del comportamiento del control de posición.....	56

## Diseño y construcción de un prototipo de robot didáctico ortogonal

**Palabras clave:** robot, manipuladores, ortogonal, interacción, comunicación USB, Java, Mecatrónica.

**Abstract:** In this document it is defined the Orthogonal Robot (actually, an Orthogonal Robotic System) as the next morphology: a robotic system that is composed by two manipulators working in perpendicular planes with a common origin, where each has its restricted joints to move in a plane that is orthogonal to that, in which, the joint is moved above or below. In addition to the above definition (which is described in this document) the prototype has a HMI for control and management thereof, also, could be observed this behavior with the use of a webcam.

### Resumen

En este proyecto se define al robot ortogonal (en realidad sistema robótico ortogonal) con la siguiente morfología: un sistema robótico que tiene dos manipuladores interactuando en planos perpendiculares con un origen común, donde cada uno, tiene sus articulaciones restringidas a moverse en un plano que es ortogonal a aquel en el que se mueve la articulación anterior o siguiente. Aunado a la definición anterior (que en este documento se describe) el prototipo cuenta con una interfaz hombre-máquina para el control y manejo del mismo, donde además se observará el comportamiento de éste con el uso de una cámara web.

El prototipo fue diseñado y construido con base en los conocimientos adquiridos a lo largo de la estadía escolar, entre los que destacan los tópicos de Programación, Control de motores, Electrónica y temas de la materia de Dispositivos Programables. La idea del desarrollo de este proyecto surge de la necesidad de una interacción física entre el alumno y un sistema robótico que funcionase como material didáctico, desarrollado para programar sentencias a manera de lista de instrucciones que serían ejecutadas en él.

El trabajo propuesto es un apoyo didáctico, que puede ser utilizado en el reforzamiento de temas para la materia de Programación, por ejemplo: manejo de vectores y apuntadores, programación orientada a objetos, comunicación USB, entre otros. Consideramos que el hecho de observar la ejecución de una sentencia programada, afecta positivamente en el proceso de aprendizaje, haciéndolo significativo. De manera similar, en la materia de Robótica permite ejemplificar temas como marcos coordenados, cinemática directa e inversa y fundamentos de cooperación.

El robot ofrece una gran accesibilidad a la programación del microcontrolador. Esto brinda una plataforma de trabajo estable, para el desarrollo de la mayoría de los temas concernientes al microcontrolador, que se imparten en la materia de Dispositivos Programables. Además, esta accesibilidad aunada a la integración de la electrónica correspondiente, y motores con codificadores de cuadratura, permiten realizar prácticas correspondientes a los tópicos de control digital angular de posición o velocidad, impartidos en la materia de Control de motores.

Al integrar la cámara web con la que cuenta el sistema, se pueden poner en práctica algoritmos que se estudian en la materia de Visión Artificial, de los cuales pueden surgir proyectos alternos, como la regeneración tridimensional de un objeto, o el control servo-visual en alguna configuración cinemática del robot.

La interfaz gráfica (GUI) se desarrolló en NetBeans, en lenguaje Java, la cual se presenta estable y funcional. La comunicación entre la GUI y la tarjeta controladora es efectiva e igualmente funcional. Los datos de control mandados son posicionados de manera ordenada en una cadena, mismos que el microcontrolador clasifica e interpreta para su ejecución.

Con respecto a los componentes neumáticos se encontró que es importante regular el caudal para el óptimo desempeño de los actuadores y se corroboró que es alta la inversión económica que se necesita en estos componentes.

De los componentes mecánicos se observó que presenta deficiencias en sus movimientos lineales, esto a causa de una manufactura que no permitió las tolerancias permitidas de excentricidad entre los barrenos y los ejes guía, se propone adaptar rodamientos que mejoren el movimiento.

### **Objetivo general**

Diseñar y construir un prototipo didáctico con base en una interacción, en planos perpendiculares, de un manipulador robótico de tres grados de libertad y uno similar de dos grados de libertad.

### **Objetivos particulares**

- Diseñar los componentes de los bloques funcionales del prototipo, dentro de los costos y configuración requeridos.
- Manufacturar los elementos diseñados para un buen desempeño de los actuadores, para que el sistema integrado sea funcional.
- Implementar algoritmos de visión artificial que integrados ejecuten una clasificación y segregación de colores.
- Desarrollar una interfaz gráfica sencilla desde la cual se pueda probar la funcionalidad del trabajo propuesto.
- Establecer comunicación entre el control de mando del prototipo y la interfaz gráfica para la transferencia de datos.
- Realizar la integración de todos los componentes que conforman el prototipo para que en sintonía sinérgica tenga un proceso funcional.
- Diseñar al menos una práctica de un tópico de la materia de Programación dirigida a alumnos que cursan la carrera de Ingeniería Mecatrónica.

## Contenido

Índice de diagramas	I
Índice de tablas	I
Resumen	III
Objetivos	IV
1. Introducción	3
1.1 Antecedentes.....	4
1.2 Definiciones.....	6
1.3 Apoyos didácticos.....	7
2. Descripción del trabajo propuesto	12
3. Desarrollo del trabajo propuesto	19
3.1 Descripción del proceso funcional del prototipo.....	19
3.2 Diseño neumático.....	22
3.2.1 Cálculo de sujetador neumático.....	22
3.3 Diseño de la tarjeta de control.....	26
3.3.1 Microprocesadores y microcontroladores: caracterización.....	26
3.3.2 Microcontrolador PIC18F4550 en la tarjeta de control.....	27
3.3.3 Conectores principales en la tarjeta de control.....	29
3.3.4 Activación de electroválvulas, inversores <i>Schmitt Trigger</i> y alimentación.....	30
3.3.5 Circuito de conmutación.....	31
3.4 Diseño del control.....	35
3.4.1 Esquema del control en el microcontrolador.....	35
3.4.2 Protocolo de comunicación USB e implementación en PIC18F4550.....	37
3.4.3 Codificadores incrementales de cuadratura e implementación.....	40
3.4.4 Modelo matemático de un motor de CD.....	41
3.4.5 Modelo a bloques del control de posición a implementar.....	43
4. Validación del trabajo propuesto	45
4.1 Implementación del equipo neumático.....	45
4.2 Implementación de la tarjeta de control.....	47
4.3 Descripción de la programación en el microcontrolador.....	50
4.3.1 Descripción de los descriptores de <i>hardware</i> generales del microcontrolador.....	50
4.3.2 Descripción del desarrollo de la comunicación USB.....	50
4.3.3 Descripción de la lectura de los codificadores.....	51
4.3.4 Descripción de la función de control.....	52

---

4.3.5 Descripción del bucle general.....	53
4.4 Comportamiento del control de posición angular.....	55
5. Conclusiones	57
5.1 Conclusiones del diseño.....	57
5.2 Conclusiones de la implementación.....	57
5.2 Posibles mejoras al sistema.....	58
5.3 Comparativa de costos.....	59
Referencias	60

#### APÉNDICE 1. Práctica de comunicación USB con el robot didáctico ortogonal

#### Anexos

- A. Pinzas neumáticas
- B. Motorreductor con codificador
- C. Cilindros neumáticos
- D. Electroválvulas
- E. Microcontrolador PIC18F4550
- F. Transistor BJT547
- G. MOSFET IRF630
- H. MOSFET IRF9Z20
- I. *Schmitt-Trigger* HD74LS14P
- J. Optoacoplador dual PC847
- K. Programa del microcontrolador
- L. Reportes generados por Altium Designer por el PCB

## Capítulo 1. Introducción

Se propone realizar un diseño funcional para lograr la interacción de dos manipuladores robóticos en configuración ortogonal, que será utilizado como un elemento didáctico que favorezca el desarrollo de habilidades en los alumnos de la materia de Programación, entre otras materias cursadas por los alumnos de Ingeniería Mecatrónica, que permitirá un primer acercamiento a la robótica en una interfaz fácilmente comprensible. El prototipo en funcionamiento tomará datos de un problema específico que se busque resolver. El alumno podrá enlistar una secuencia de movimientos en la interfaz del prototipo, para que realice las sentencias deseadas y, desde ahí mismo, ejecutarlas para resolver el problema planteado.

Un tema importante en la formación del Ingeniero Mecatrónico o Robótico es la interacción de dos manipuladores. Dado que en la mayoría de las aplicaciones de robótica en la industria, los robots son usados como un elemento dentro del proceso de producción. Donde éstos se encuentran secuenciados y la coordinación entre todas las partes es vital para el proceso mismo. Consideramos que aprender acerca de la interacción de dispositivos robóticos es parte esencial para una correcta implementación en procesos industriales.

“El constructivismo es un conjunto de teorías que se basan en los resultados de la investigación sobre el aprendizaje desde la perspectiva cognoscitivista (...) el aprendizaje es un proceso a través del cual la información llega a ser conocimiento. La información se concibe solo como una representación estructurada de datos. El aprendizaje, entonces, ocurre cuando los alumnos generan conocimiento, no solo cuando reciben la información”, (Ogalde & Bardavid, 2008, págs. 36,38). Retomando esta idea, consideramos que esta propuesta será benéfica para el alumnado que lo utilice, puesto que pondrá en práctica el conocimiento en aulas, haciéndolo significativo para que así, no quede solo como información.

Se considera que el proyecto surge de una necesidad relacionada a la andragogía; con respecto a observar de manera física, el comportamiento de un fenómeno que ha sido programado por un alumno. Lo anterior representa un estímulo para él. Una autora refiere esta acción a las teorías cognoscitivistas, sosteniendo que no basta una asociación entre estímulos, sino que cuando la persona está en contacto con éste, la respuesta manifiesta es una exteriorización del aprendizaje logrado antes, en todos los procesos relacionados con la interacción con el estímulo. Donde el objetivo prioritario es posibilitar que los alumnos realicen aprendizajes significativos por sí solos, es decir, que sean capaces de aprender a aprender. Lo cual supone una modificación de los esquemas de conocimiento que el alumno posee, para lo que supone una intensa actividad por parte de éste (Arriola, 2009).

Si un primer acercamiento a la programación de sistemas robóticos se hace con un manipulador grande, y se cometiera un error, éste implicaría un alto costo de mantenimiento. Se necesitarían corregir los daños. Además existiría un alto riesgo de lesión física en el alumno a causa de alguna contusión. Ahora, si ese mismo error ocurriera en la plataforma que planteamos en este documento, el problema que significaría sería de una escala mucho menor. Así, se podrían obtener ahorros en el mantenimiento industrial y gastos médicos.

La programación secuencial (a manera de lista de instrucciones) permitirá una mejor comprensión de las ejecuciones de estas sentencias, en contra de sistemas más complejos que requieren de mayor dedicación, cuando se programa por primera vez.

El objetivo de todo instructor es lograr que aquella persona a la que está capacitando aprenda lo más posible. Este prototipo es una tecnología educativa que apoyaría al docente, a nivel superior, puesto que (Ogalde & Bardavid, 2008) hacen entender a ésta como un conjunto de procedimientos o métodos, técnicas, instrumentos y medios, derivados del conocimiento científico. La tecnología educativa, concebida en esta forma, es un elemento valioso para que el maestro realice sus funciones específicas de planeación, conducción y evaluación del proceso de enseñanza-aprendizaje. En el caso particular de un docente universitario, quien, en el mejor de los casos, es un buen especialista en el campo disciplinario que enseña, pero que generalmente no ha tenido una preparación didáctico-pedagógica igualmente sólida. La opción que ofrece este prototipo consideramos resultaría atractiva no solo al docente de nivel superior, sino también al discente generando un aprendizaje significativo y motivante.

Es necesario explicar que este prototipo funge como un material didáctico. (Herraiz, 2001) identifica como un medio o recurso que nos puede servir para aplicar una técnica concreta en el ámbito de un método de aprendizaje determinado. Discerniéndolo de un método y técnica didáctica. Entiéndase por método didáctico al modo, camino o conjunto de reglas que utilizamos para obtener un cambio en el comportamiento de quien aprende, que potencie o mejore su nivel de competencia para desempeñar una función productiva. Las técnicas didácticas son las prácticas, estrategias o pasos para lograr el aprendizaje: la lluvia de ideas, el juego de roles, el debate dirigido, entre otros.

### **1.1 Antecedentes**

Desde que iniciamos la carrera en Ingeniería Mecatrónica, hasta este momento que estamos por concluir, nos hemos percatado de la necesidad de tener apoyos didácticos ya elaborados (robots de arquitectura abierta, maquinas CNC de arquitectura abierta, entre otros). Los cuales permitirían al alumno enfocarse en la programación de éstos, así como en el desarrollo de aplicaciones más complejas, tales como Control Automático o aplicaciones de Visión Artificial, que por el hecho de construirlos uno mismo no se proyectan hasta estos niveles de complejidad, por motivos de tiempo al tomar las materias relacionadas con estos temas.

Dado lo anterior, sugerimos que, un apoyo didáctico que facilite el proceso de enseñanza-aprendizaje a alumnos de la materia de Programación, entre otras, se consideré deba tener elementos robóticos, donde su programación se realice en un entorno didáctico con funciones sencillas y de fácil comprensión.

De tal manera que los conceptos de programación sean más inteligibles, y éstos se vean plasmados en movimientos en las articulaciones de los manipuladores; si se agrega un sistema de dos manipuladores ortogonales que puedan interactuar para generar un apoyo didáctico, el campo de ejercicios y prácticas sobre esta plataforma se amplifica permitiendo abarcar más temas del plan de estudios de la materia en sí.

En la U.P.I.I.T.A. se han realizado algunos estudios de robots para aplicaciones didácticas como los que se mencionarán a continuación:

- “Brazo manipulador para aplicaciones didácticas” de N. Martínez, et alii, que expresan que su objetivo fue desarrollar un brazo manipulador para aplicaciones didácticas en pruebas de trayectorias predefinidas, empleando las leyes de control PI y PID (Martínez, Murillo, & Pineda, 2005).
- “Sistema didáctico para la enseñanza-aprendizaje de la visión artificial” de A. Ríos y R. E. Ortega, donde resumen a su trabajo como “la implementación de un sistema didáctico de visión artificial para el reconocimiento y comparación de objetos basados en MATLAB. Además se diseñó y construyó un robot de 4 grados de libertad tipo SCARA para la manipulación de los objetos”. (Ortega & Ríos, 2005)
- “Robot didáctico con tecnología abierta” de J. C. Martínez, et alii, donde exponen que su objetivo fue desarrollar un robot con tres grados de libertad, todos rotacionales, para aplicaciones educativas (Martínez, Martínez, & Silva, 2008).
- “Robot portátil de tres grados de libertad en configuración angular para fines didácticos” de J. C. Vargas, cuyo objetivo fue el diseño, construcción e implementación de un robot portátil de 3 grados de libertad; en configuración angular; para fines didácticos, tratando de mejorar el transporte de este tipo de brazos mediante la separación de sus eslabones, base del robot y colocando estas piezas en una maleta (Vargas, 2008).

El prototipo planteado tiene características que lo hace diferente de los anteriores proyectos. Una de estas características es la interacción entre dos manipuladores, dado que los anteriores no lo presentan, al conformarse por un solo manipulador. Además, se muestra diferente en su sistema mecánico basado mayoritariamente en un sistema neumático, en comparación a los sistemas de motores eléctricos de los proyectos antes mencionados; esta característica impacta en la morfología del robot planteado, que de manera recíproca es diferente.

Hasta donde se ha investigado, la compañía De Lorenzo of America, Corp., es una de las más grandes empresas transnacionales dedicadas al desarrollo de sistemas técnico-didácticos, dentro de su catálogo no se encuentra un dispositivo como el propuesto en este trabajo (De Lorenzo of America). Al no encontrar hasta el momento una referencia comercial clara de un dispositivo parecido se decide desarrollarlo.

## 1.2 Definiciones

**Robot.**– El término robot fue introducido primeramente en nuestro vocabulario por el dramaturgo checo Karel Capek en su obra *Rossum's Universal Robots* en 1920, la palabra robota es la palabra checa para trabajo (en checo 'robot' significa 'sirvo' o 'trabajo forzado'). Una definición oficial de robot proviene del *Robot Institute of America* (RIA): un robot es un manipulador multifuncional reprogramable diseñado para mover material, partes, herramientas, o dispositivos especializados a través de movimientos variables programados para la realización de una variedad de tareas (Spong, Hutchinson, & Vidyasagar, 2006, págs. 2,3).

**Robot ortogonal.**– un sistema robótico que tiene dos manipuladores interactuando en planos perpendiculares con un origen común, donde cada uno, tiene sus articulaciones restringidas a moverse en un plano que es ortogonal a aquel en el que se mueve la articulación anterior o siguiente.

**Mecatrónica.**– Una definición a menudo consultada de mecatrónica fue presentada por Harashima, Tomizuka y Fukada en 1996. Definida como: “La integración sinérgica de la ingeniería mecánica, con electrónica y control computacional inteligente en el diseño y manufactura de productos industriales y procesos” (Bishop, 2008).

**Prototipo.**– Esto es un modelo original del producto que funcione. Un desarrollo reciente importante es el prototipado rápido que se apoya en técnicas CAD/CAM y en diversas técnicas de manufactura para producir prototipos de manera rápida, en forma de un modelo de una pieza física sólido y a un bajo costo (Schmid, 2002, pág. 12).

### 1.3 Apoyos didácticos

¿Qué estudios se han realizado sobre los apoyos didácticos y su impacto en el proceso de enseñanza-aprendizaje? La presente interrogante surge a partir del propósito de este trabajo, pues el dispositivo creado, como se menciona previamente, pretende fortalecer y facilitar el aprendizaje del campo mecatrónico.

Consideramos de vital importancia que en primera instancia se defina el proceso enseñanza-aprendizaje y quiénes son sus actores.

El acto didáctico conlleva cuatro elementos básicos: docente, discente, contenidos y contexto. El docente se encargará de enseñar al discente, por medio de los apoyos y materiales didácticos adecuados a su contexto. En la anterior fórmula hay un elemento sustancial: enseñar. Esta ardua labor que realizan los profesores para Lorenzo Tébar, está inevitablemente unida a los procesos de aprendizaje que, siguiendo sus indicaciones, realizan los estudiantes. El objetivo de docentes y discentes siempre consiste en el logro de determinados objetivos educativos y la clave del éxito está en que los estudiantes puedan y quieran realizar las operaciones cognitivas convenientes para ello, interactuando adecuadamente con los recursos educativos a su alcance (Tébar, 2003, págs. 134,139).

Actualmente se han realizado inmensos estudios sobre el aprendizaje. Son teorías descriptivas que presentan planteamientos muy diversos, pero en todas ellas aún podemos encontrar algunas perspectivas clarificadoras de estos procesos tan complejos. Entre éstas se destacan:

- La perspectiva conductista: B.F. Skinner, W. Watson y Pavlov.
- Teoría del procesamiento de la información: Phye
- Aprendizaje por descubrimiento: J. Bruner
- Aprendizaje significativo: D. Ausubel
- Enfoque cognitivo: Gagné
- Constructivismo. J. Piaget
- Socio-constructivismo: Vigotski

Recordemos que el docente tiene como base a los apoyos didácticos para que el proceso enseñanza-aprendizaje pueda resultar significativo y funcional.

El apoyo didáctico se denomina como aquél medio o recurso que se usa en la enseñanza y aprendizaje; su fundamental fin es representar, aproximar o facilitar al alumno la observación, comparación, investigación o comprensión de la realidad (Paredes, 2000).

Es importante que no se mal interprete este propósito o se vea como un sinónimo de sustitución del docente, al contrario, como su nombre lo indica es un apoyo para el maestro y éste lo utilizará como una herramienta que le permitirá fortalecer los aprendizajes de sus alumnos, a su vez, funciona como un aliciente puesto que respeta los ritmos de trabajo y aprendizaje de los discentes. “El alumno aprende cuando él quiere, no cuando lo decide el maestro”.

Según Cousinet, citado por (Zabala, 1995, págs. 125-127), hay ciertas características que hacen funcional a un apoyo didáctico entre las que destacan las siguientes:

- Son motivacionales
- Facilitan la comprensión de un concepto
- Se realiza la introspección del proceso gracias a la manipulación de un material, pues éste no es de uso exclusivo del profesor
- Impulsan la abstracción e independización de un material

¿Por qué un robot didáctico ortogonal favorece el proceso de enseñanza-aprendizaje a nivel superior en los alumnos de la Ingeniería en Mecatrónica? La delimitación a esta interrogante surge al recordar mis experiencias didácticas dentro de la UPIITA y cuestionarme sobre cómo fueron éstas.

Para ello, es necesario enfatizar que el aprendizaje es un potencial que está presente en la vida humana y se genera en cualquier momento, siempre y cuando existan las condiciones personales y ambientales para ello. Ya J.J Rousseau lo mencionaba al decir que el ser humano aprende gracias a su medio, pues éste se encargará de propiciar las experiencias necesarias para que este aprendizaje sea significativo. Es por ello que al aprender modificamos comportamientos, es decir, a nivel actitudinal cambiando nuestra forma de vida adquiriendo nuevos conocimientos.

(King, 2001) define el aprendizaje desde un enfoque cognoscitivo:

“Aprendizaje es ampliar nuestras capacidades en términos de conocimientos y comportamientos como resultado de la experiencia y que perdura con el tiempo”, es decir, es permanente (Arriola, 2009).

La elaboración de este robot didáctico favorece el aprendizaje, en primera instancia, para sus creadores, ya que hemos involucrado y correlacionado múltiples ramas de la mecatrónica, tales como *Mecánica, Control, Electrónica y Programación*; abarcando ámbitos de la enseñanza a nivel actitudinal, conceptual y procedimental en casos como *la visión artificial, programación de microcontroladores, diseño y manufactura de componentes mecánicos, interfaz gráfica como elemento de programación orientada a objetos, y en la tarjeta de control siendo el resultado de un diseño electrónico.*

Por lo anterior, como individuos obtuvimos cambios en nuestro aprendizaje, en suma, de los siguientes elementos:

- Como resultado de la experiencia.
- Producido por la interacción con el ambiente.
- Satisface una necesidad (burocrática y personal)
- Conductual, ya que o es perjudicial o benéfico.
- Es permanente.

A posteriori a nivel personal, la fabricación de este robot beneficiará a los estudiantes de nivel básico y medio superior con sus correspondientes adecuaciones para generar interés por la tecnología y hacer que se involucren con las funciones más básicas; aunque el primordial propósito del dispositivo es apoyar a la Educación Superior enfocada a la Ingeniería en Mecatrónica teniendo su sustento en las siguientes asignaturas:

- Programación Básica: Servirá para estudiar el tópico de comunicación USB, donde se ejecuta una instrucción enviada desde una computadora.
- Programación Avanzada: Se podrá generar una interfaz gráfica con comunicación USB, desde la cual se envíen las instrucciones que manipulan al prototipo.
- Control de Máquinas Eléctricas: se podrán utilizar las señales de los codificadores de los motores del prototipo para generar con ellos prácticas de control digital en dichos motores.
- Visión Artificial: se podrán estudiar algoritmos de visión artificial con la cámara integrada en el prototipo, así como realizarse programas más avanzados como la reconstrucción visual tridimensional de un objeto, que se genera con el giro del mismo y captura de imágenes.

Se ha mencionado que su objetivo es favorecer la Educación Superior y los actores involucrados en el proceso de enseñanza-aprendizaje son adultos. Es importante mencionar:

“La andragogía es la disciplina que se ocupa de la Educación y el aprendizaje del adulto. Este concepto fue propuesto por la UNESCO en sustitución de la palabra pedagogía, para designar la ciencia de la formación del Ser Humano haciendo referencia a la Educación permanente” (Arriola, 2009).

Hay una intensa relación entre la motivación y el aprendizaje. Los intereses, las experiencias previas, las necesidades, las preocupaciones y las expectativas son, en la formación de adultos, un conjunto de elementos que no sólo condicionan el proceso de aprendizaje, sino de la propia definición, planificación y organización del curso. El adulto va a aprender si está motivado para ello, es decir, si el aprendizaje responde a lo que le concierne e interesa.

Existen dos tipos de motivaciones que mueven a los adultos a aprender:

- Motivaciones Externas:
  - Búsqueda de soluciones a problemas o necesidades concretas.
  - Adaptación profesional.
  - Promoción profesional.
- Motivaciones Internas:
  - Satisfacción personal, aumento de autoestima y auto concepto.
  - Aumentar seguridad a nivel personal, familiar y profesional.
  - Mejorar la capacidad de adaptación y a la tecnología de uso cotidiano.

El adulto es el protagonista en todo proceso de desarrollo que lo afecte, por lo que su participación activa en la planificación, programación, realización y evaluación de las actividades educativas, en condiciones de igualdad con sus compañeros y maestros, ha de ser el elemento clave de una buena praxis formativa (Arriola, 2009).

Hasta aquí se ha justificado por qué el robot didáctico ortogonal funge como elemento para lograr un aprendizaje en la Educación Superior (Educación para Adultos) abordado, a grosso modo, de forma ecléctica, pues abarca los niveles personales y académicos que los adultos involucran para la obtención de un aprendizaje.

## Referencias del capítulo

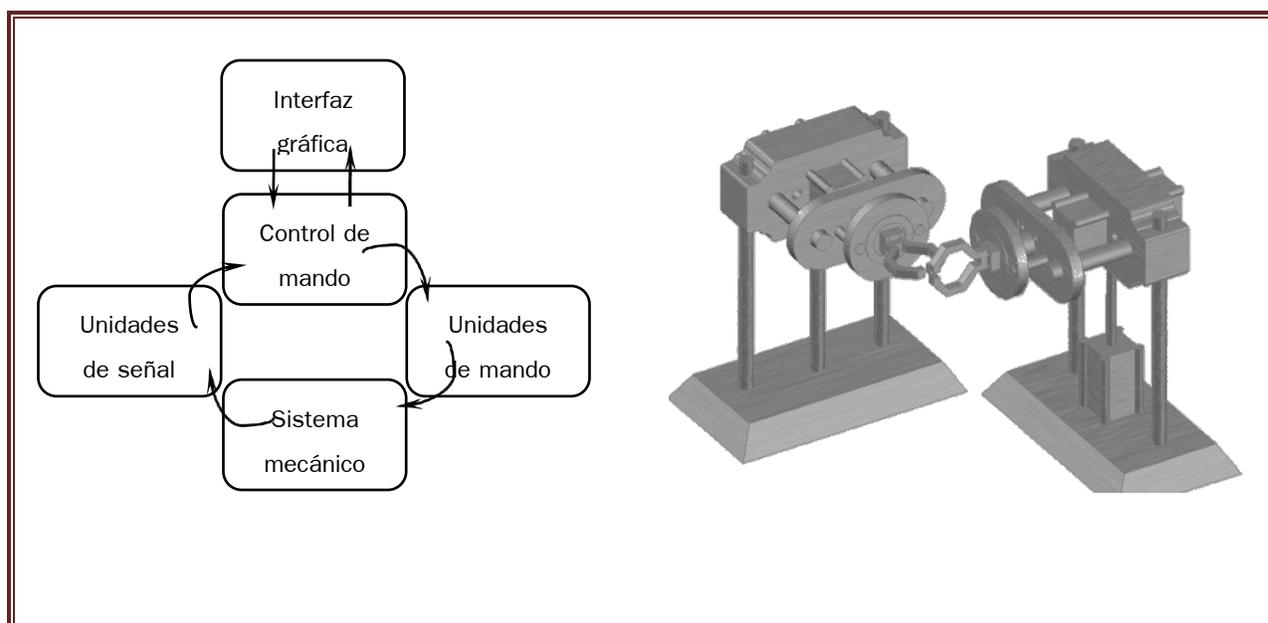
- Arriola, M. A. (2009). *Desarrollo de competencias en el proceso de instrucción*. Ciudad de México: Trillas.
- Bishop, H. R. (2008). *The Mechatronics Handbook – Mechatronic systems, sensors, and actuators: fundamentals and modeling*. Boca Raton, Florida: CRC Press.
- De Lorenzo of America. (s.f.). *Equipos Didácticos: Mecatrónica*. Recuperado el primero de febrero de 2010, de De Lorenzo of America Corportion Web site: <http://www.delorenzo.com.mx/prods/mecatronica/index.htm>
- Herraiz, M. L. (2001). *Formación de formadores: manual didáctico*. Ciudad de México, México: Limusa.
- Martínez, D. O., Martínez, J. C., & Silva, V. M. (2008). *Robot didáctico con tecnología abierta*. Ciudad de México, México: UPIITA – IPN.
- Martínez, N., Murillo, O., & Pineda, J. C. (2005). *Brazo manipulador para aplicaciones didácticas*. Ciudad de México, México: UPIITA – IPN.
- Ogalde, I., & Bardavid, E. (2008). *Los materiales didácticos: medios y recursos de apoyo a la docencia*. Ciudad de México, México: Trillas.
- Ortega, R. E., & Ríos, A. (2005). *Sistema didáctico para la enseñanza-aprendizaje de la visión artificial*. Ciudad de México, México: UPIITA – IPN.
- Paredes, J. (2000). *Materiales didácticos en la práctica educativa: un análisis etnográfico*. Madrid: Universidad Autónoma de Madrid.
- Schmid, S. R. (2002). *Manufactura, ingeniería y tecnología*. Ciudad de México, México: Pearson Educación.
- Spong, M., Hutchinson, S., & Vidyasagar, M. (2006). *Robot Modeling and Control*. New York: John Wiley and Sons.
- Tébar, L. (2003). *El perfil del profesor mediador*. Madrid: Editorial Santillana.
- Vargas, J. C. (2008). *Robot portátil de tres grados de libertad en configuración angular para fines didácticos*. Ciudad de México, México: UPIITA – IPN.
- Zabala, A. (1995). *La práctica educativa. Cómo enseñar*. Barcelona: Editorial Graó.

## Capítulo 2. Descripción del trabajo propuesto

Se propone un sistema robótico con un manipulador de dos grados de libertad, TR, donde su configuración permite tomar dos posiciones en su movimiento traslacional, y en su movimiento angular posiciones en ángulos enteros, en un intervalo de cero a ciento ochenta grados; el anterior presentará interacción con otro manipulador de tres grados de libertad, TTR, que en sus configuraciones traslacionales les permita tomar dos posiciones, y en el movimiento angular posiciones en ángulos enteros, en un intervalo de cero a ciento ochenta grados.

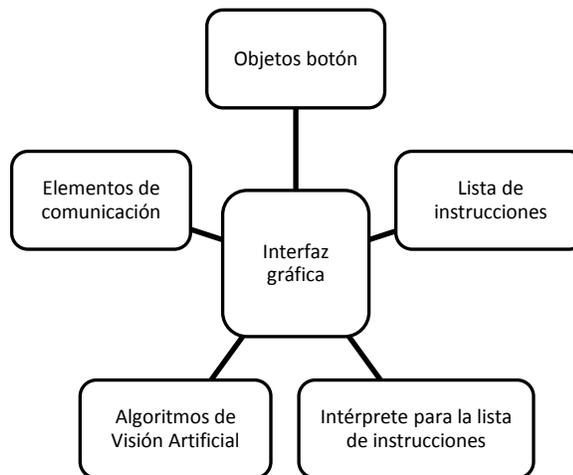
Se plantea deducir los parámetros de la cinemática directa e inversa del prototipo.

El prototipo que se busca desarrollar, para generar el apoyo didáctico antes mencionado, se presenta a continuación desglosado en sus subsistemas, los cuales a su vez integran sus propios bloques funcionales como se describirá más adelante.



**Figura 2.1** Esquema a bloques y primer estudio del prototipo realizado con AutoCad 2009©

La interfaz gráfica presenta dentro de su estructura los siguientes elementos:



**Figura 2.2** Estructura de la interfaz gráfica

La lista de instrucciones que se genera sirve para el control y la programación de los movimientos de los manipuladores, estas instrucciones son simples para que sean inteligibles para los usuarios del prototipo. Se concentran también las funciones de control sobre las cámaras para la visión artificial. Estas instrucciones aparecen en un cuadro de texto, una vez presionados los objetos botón específicos, también se podrán escribir en el cuadro de texto antes citado.

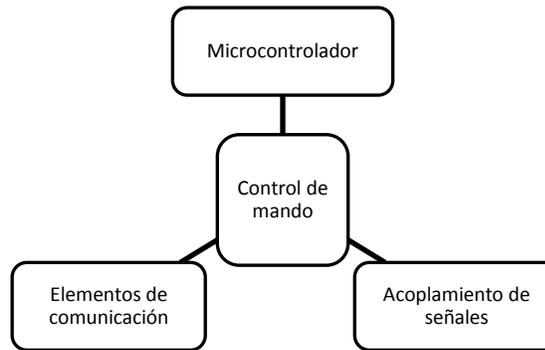
Los objetos botón son los controles visuales. La función de un bloque de ellos es declarar una línea de instrucción; un par de botones definen si el sistema ejecuta línea a línea las instrucciones o a manera de bloque de instrucciones, mientras que la funcionalidad de un par más, es el iniciar y finalizar la ejecución de la lista de instrucciones cuando se encuentra en modo de bloque. Así como algunos otros botones sirven para establecer parámetros iniciales.

Para que la interfaz gráfica funcione, necesita de un ciclo principal que administre todos los eventos con los que cuenta la interfaz, este ciclo funcionará como un Intérprete para la lista de instrucciones.

Los algoritmos de visión artificial se encuentran dentro de la interfaz gráfica, debido a que, la plataforma en la cual se desarrolla es lo suficientemente apta para que en ella se ejecuten los algoritmos (inicialmente estudiados) que conlleven a una correcta clasificación de colores para las aplicaciones que el usuario le sean convenientes.

La interfaz gráfica además tiene las funciones para que se establezca una comunicación con el control de mando, bajo el protocolo que en su momento se consideró conveniente, dado que es una pieza fundamental para tener una interfaz hombre-máquina.

El control de mando presenta dentro de su estructura los siguientes elementos:



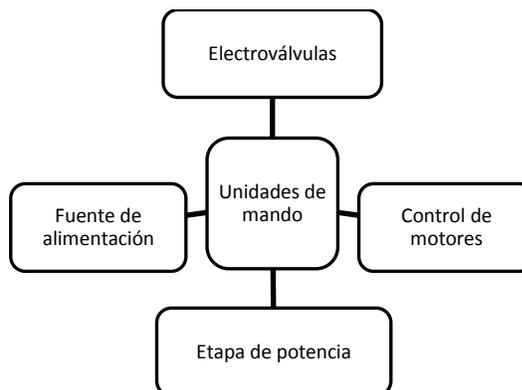
**Figura 2.3** Estructura del control de mando

El microcontrolador es la unidad fundamental del control de mando, el programa cargado en este microcontrolador administra las señales que son enviadas a las unidades de mando. Es tarea del microcontrolador traducir la lista de instrucciones, que son mandadas desde la interfaz, en las señales necesarias para el correcto funcionamiento del prototipo. Por otro lado, también administra las señales que son mandadas de las unidades de señal, que según la programación que se obtenga, ejercerá el control sobre el actuador que se esté midiendo. Por último, el control de mando tiene programado el control de posición angular para los motores que se requieran.

Dadas las funciones que efectúa el microcontrolador, es necesario que las señales que se introduzcan en él sean las adecuadas, en cuanto a la intensidad de corriente y de potencial eléctrico; por lo tanto está provisto de un sistema de acopladores de señal e inversores *Schmitt-Trigger* para esta función.

Como ya se ha presentado, la capacidad de comunicación de este dispositivo es esencial, por lo tanto es necesaria una unidad de comunicación con el mismo protocolo que la interfaz gráfica.

Las unidades de mando se desglosan en los siguientes elementos:



**Figura 2.4** Desglose de las unidades de mando

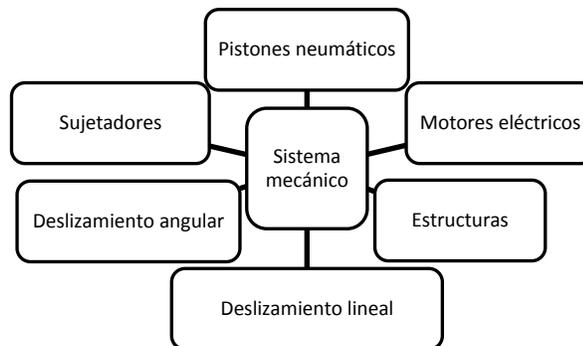
Las electroválvulas son los elementos de mando para las unidades de trabajo neumáticas, en concreto de los pistones neumáticos y sujetadores. Las electroválvulas reciben señales provenientes del control de mando para ejercer la función a la que están destinadas.

El control de motores emite las señales de potencia para accionar los motores con los que cuenta el prototipo, de igual manera recibe la instrucción de activar al motor desde el control de mando.

Una etapa de potencia es requerida para adecuar las señales de control, a la energía adecuada para hacer funcionar los motores eléctricos y los solenoides de las electroválvulas.

En este apartado del prototipo se introduce la fuente de alimentación externa a la tarjeta de control desde la cual se abastece al prototipo exceptuando a la computadora. La fuente tiene los potenciales eléctricos adecuados para cada elemento que lo necesite.

El sistema mecánico presenta dentro de su estructura los siguientes elementos:



**Figura 2.5** Estructura del sistema mecánico

El sistema mecánico en general, lo componen los manipuladores y sus estructuras, el prototipo busca como tal ser capaz de mover un cubo, de masa no mayor de doscientos gramos, con dimensiones aproximadas de 100×100×100 milímetros, y en condiciones ambientales normales en la Ciudad de México; sobre estos principios se realizaron los diseños para el prototipo.

En este proyecto, los pistones neumáticos son las unidades de trabajo que ejercen la mayor fuerza, dado que son los generadores de movimiento para las primeras articulaciones, se buscó que tuvieran una carrera corta y la presión neumática fuese baja.

Los motores eléctricos en un diseño preliminar se proponen accionados por corriente directa. Éstos tienen la función de dar el giro a los sujetadores, en el plano perpendicular al movimiento lineal (que se presenta en la articulación anterior) para colocarse en la posición deseada.

Las estructuras son los soportes sobre los cuales las partes móviles se sujetan, para darle un sustento firme, al momento de generar su movimiento.

El deslizamiento lineal se refiere al movimiento del sistema mecánico que se requirió diseñar e implementar para minimizar la carga flectora sobre el eje del pistón y la fricción entre los ejes, éstos sostienen la estructura que soporta al motor.

El deslizamiento angular se refiere al movimiento del sistema mecánico que se diseñó e implementó para minimizar la carga flectora sobre el eje del motor y la carga a mover en el sujetador, se pretende utilizar rodamientos.

Los sujetadores son pinzas neumáticas que sirven para sostener una pieza incluso si el soporte del sujetador se encuentra en movimiento, son alimentados con la presión mínima necesaria para sostener la pieza.

Las unidades de señal se desglosan en los siguientes elementos:



**Figura 2.6** Desglose de las unidades de señal

Los sensores de inicio y fin de carrera en un primer diseño se definen de tipo mecánico, que muestrean el estado de inicial de los motores; estas señales son enviadas al control de mando para que realice su objetivo.

El codificador es un dispositivo opto electrónico que muestrea la posición de un disco que se adapta a un motor, con la finalidad de monitorear la posición angular de su eje. Es necesario configurar un codificador para cada uno de los motores.

## Parte A

En la descripción del trabajo propuesto se presenta en su totalidad el prototipo de robot didáctico ortogonal. El trabajo propuesto fue dividido, se generaron las partes A y B. Siendo este trabajo la **parte A**, donde se muestra solo el desarrollo e implementación de los **sistemas neumáticos, electrónicos y de control**, necesarios para que funcione el prototipo. El resto de los sistemas se expondrán en la parte B, que es complementaría a la parte A, que en conjunto describen de manera total el trabajo desarrollado.

### Cronograma de actividades

El trabajo propuesto se realizó con base en la calendarización que se presenta a continuación, la primera etapa se relaciona al trabajo desarrollado en Trabajo Terminal I, referente al diseño de los bloques funcionales propuestos.

Id	Nombre de tarea	Comienzo	Fin	Duración	Gantt Chart											
					Feb 2011	Mar 2011	Apr 2011	May 2011								
1	Primera entrega de protocolo (Revisión / Resultado)	2/7/2011	2/16/2011	8d	[Gantt bar from 2/7 to 2/16]											
2	Modificaciones y corrección	2/17/2011	2/18/2011	2d	[Gantt bar from 2/17 to 2/18]											
3	Segunda entrega de protocolo (Revisión / Resultado)	2/22/2011	3/1/2011	6d	[Gantt bar from 2/22 to 3/1]											
4	Modificaciones y corrección	3/2/2011	3/3/2011	2d	[Gantt bar from 3/2 to 3/3]											
5	Diseño mecánico [Análisis]	3/7/2011	3/15/2011	7d	[Gantt bar from 3/7 to 3/15]											
6	Selección de los componentes mecánicos y materiales	3/15/2011	3/25/2011	9d	[Gantt bar from 3/15 to 3/25]											
7	Primera entrega de reporte	3/28/2011	3/28/2011	1d	[Gantt bar at 3/28]											
8	Diseño electrónico [Análisis y pruebas]	3/29/2011	4/8/2011	9d	[Gantt bar from 3/29 to 4/8]											
9	Segunda entrega de reporte	4/11/2011	4/11/2011	1d	[Gantt bar at 4/11]											
10	Diseño del control de mando	4/12/2011	4/21/2011	8d	[Gantt bar from 4/12 to 4/21]											
11	Tercera entrega de reporte	4/25/2011	4/25/2011	1d	[Gantt bar at 4/25]											
12	Rediseño de los sistemas	4/26/2011	5/6/2011	9d	[Gantt bar from 4/26 to 5/6]											
13	Cuarta entrega de reporte	5/9/2011	5/9/2011	1d	[Gantt bar at 5/9]											
14	Correcciones generales	5/10/2011	5/12/2011	3d	[Gantt bar from 5/10 to 5/12]											
15	Preparación del documento	5/13/2011	5/18/2011	4d	[Gantt bar from 5/13 to 5/18]											
16	Entrega de reporte final	5/19/2011	5/20/2011	2d	[Gantt bar from 5/19 to 5/20]											

Tabla 2.1 Cronograma de actividades para Trabajo Terminal I

En la segunda etapa de desarrollo del prototipo, referente a la implementación, se calendarizaron las actividades como se puede apreciar en el cronograma siguiente.

Id.	Nombre de tarea	Comienzo	Fin	Duración	Gantt Chart																
					Feb 2011	Mar 2011	Apr 2011	May 2011													
1	Programación de la interfaz gráfica	2/7/2011	3/18/2011	30d	6/2	13/2	20/2	27/2	6/3	13/3	20/3	27/3	3/4	10/4	17/4	24/4	1/5	8/5	15/5	22/5	29/5
2	Implementación - Tarjeta Controladora	2/7/2011	2/18/2011	10d																	
3	Manufactura de piezas mecánicas	2/21/2011	3/28/2011	26d																	
4	Ensamble mecánico	3/29/2011	4/5/2011	6d																	
5	Establecer COM-USB Tarjeta-PC	4/6/2011	4/19/2011	10d																	
6	Establecer COM-USB Interfaz-Tarjeta	4/20/2011	5/3/2011	10d																	
7	Implementar CTRL DIG Posición	5/4/2011	5/17/2011	10d																	
8	Acople sinérgico	5/18/2011	5/28/2011	6d																	
9	Correcciones y ajustes	5/26/2011	6/3/2011	7d																	

Tabla 2.2 Cronograma de actividades para Trabajo Terminal II

## **Capítulo 3. Desarrollo del trabajo propuesto**

En este capítulo del trabajo se muestra primeramente la descripción del proceso funcional del prototipo, donde se plantea el flujo de tareas que desempeña el prototipo para la ejecución de una instrucción. Seguido de la descripción, se presenta el diseño de los bloques funcionales correspondientes a este trabajo (Parte A), correspondientes a la neumática, la electrónica y el control.

### **3.1 Descripción del proceso funcional del prototipo**

El proceso global de funcionamiento que sigue el prototipo, para la ejecución de una instrucción dada por el usuario desde la interfaz gráfica, se puede observar a manera de diagrama de flujo con las condiciones y ciclos, tal como describe en el diagrama siguiente.

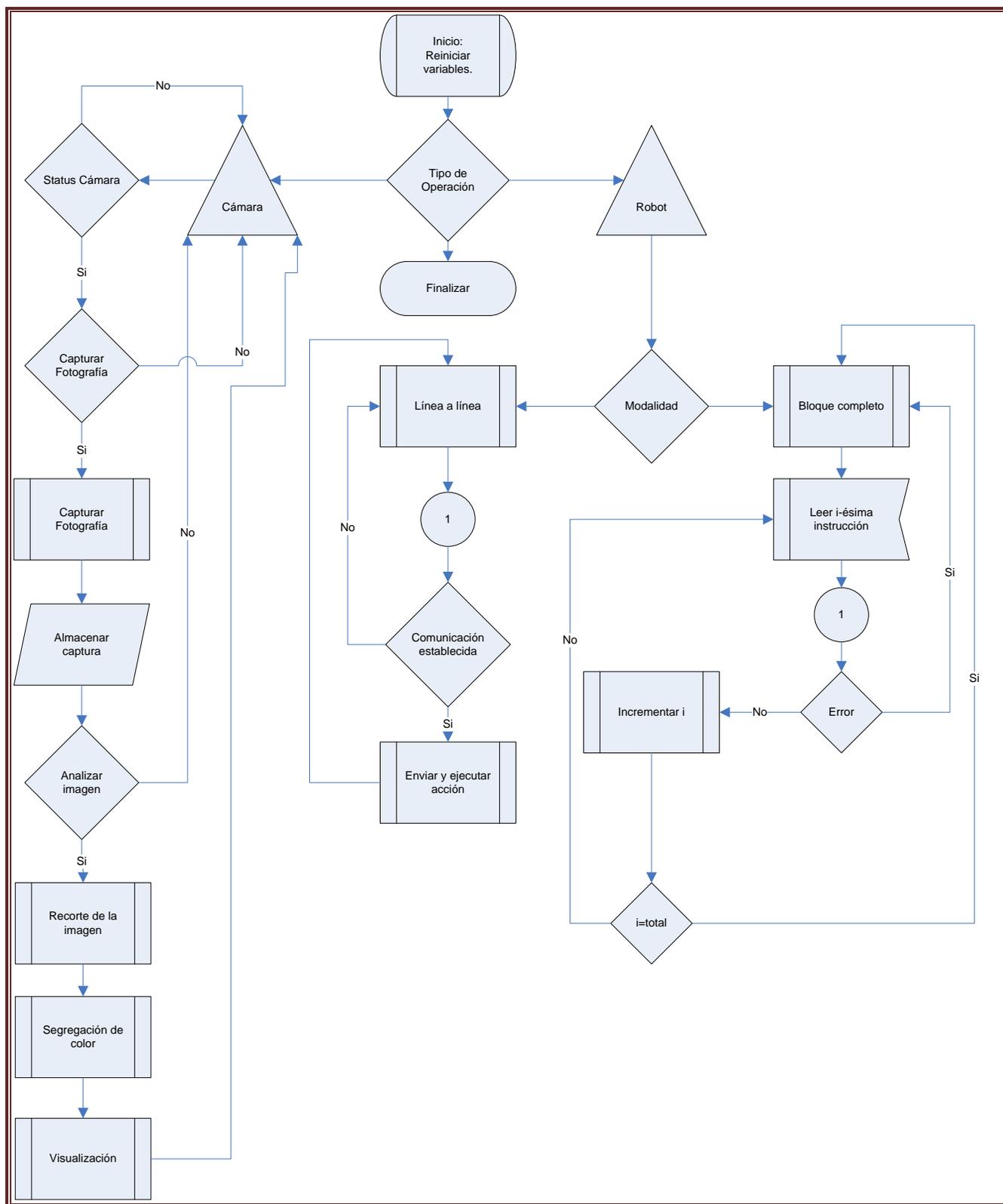


Figura 3.1 Diagrama de flujo del prototipo

El flujo de tareas que se muestra en el diagrama, una vez iniciado el sistema, puede ser cíclico tantas veces sean necesarias para el usuario del sistema robótico. Por lo tanto, el usuario es quien decide cuando terminar el ciclo general.

El proceso inicia cuando se ejecuta la interfaz gráfica y se alimentan todos los sistemas que componen al prototipo; en ese momento éste ejecutará algunas operaciones para establecerse en sus condiciones iniciales.

Aparecerá en un cuadro de texto la solicitud al usuario que para definir el modo de operación.

En caso de pertenecer a las funciones de visión se verificará el estado de las cámaras, un estado positivo realizará dicho comando y el caso contrario retornará al cuadro de texto. Las principales funciones por parte de la visión consistirán en configurar la cámara, tomar una imagen, recortar la imagen y segregar en un número definido de colores.

En las operaciones de movimiento en el robot, el usuario podrá elegir entre ejecutar línea a línea o por bloques completos de instrucciones.

En la modalidad de línea, al presionar un botón de instrucción en el robot, entrará a la etiqueta marcada con el número 1, verificará el estado de la comunicación con la tarjeta controladora y de ser positivo, se codificará un mensaje para ser enviado por la interfaz y al recibirlo el microcontrolador lo decodificará y en sus salidas mostrará las señales pertinentes para que sea energizada la unidad de mando del actuador que se haya mandado a accionar.

Al trabajar bajo la modalidad de bloques completos, el sistema asignará un número progresivo a cada línea del bloque, ejecutará línea a línea tal como se planteó en la anterior modalidad al menos que se presente un estado de error o todas las líneas sean ejecutadas.

Para terminar la sesión de trabajo el usuario cuenta con una opción de salida rápida del programa, se sugiere terminar los procesos en espera y apagar los sistemas; no pasará a mayores si los procesos no han sido concluidos al cerrar la interfaz gráfica o reiniciar la tarjeta de control, dado que, al iniciar nuevamente el sistema se reiniciarán las variables, por lo que se ejecuta una rutina de establecimiento en ceros (*HOMING*).

### 3.2 Diseño neumático

El diseño neumático del prototipo, así como el diseño de las piezas mecánicas parten del efector final de los manipuladores. El primer dato a conocer es la fuerza que requiere el sujetador para sostener el objeto planteado, para determinar las características del mismo.

#### 3.2.1 Cálculo del sujetador neumático

Analizando las fuerzas que actúan sobre el objeto a manipular, se encontró la fuerza mínima de sujeción  $F_g$ .

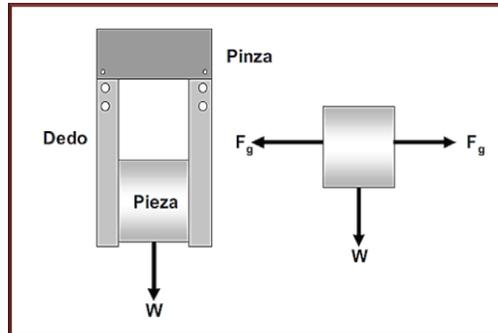


Figura 3.2 Esquema de la pinza y el diagrama de cuerpo libre

Teniendo como referencia el caso que se presenta en la figura 3.2, en el cual la única fuerza que produce el deslizamiento de la pieza es el peso de la misma, se obtienen las siguientes ecuaciones:

$$\sum f_x = 2F_g - 2N = 0$$

$$\sum f_y = 2f_R - \omega = 0$$

$$F_g = N$$

$$2f_R = \omega$$

Sustituyendo el valor de  $f_R = \mu_k N = \omega_x$  en la ecuación anterior:

$$2\mu_k N = \omega$$

Se concluye:

$$2\mu_k F_g = \omega$$

La última ecuación no es válida para el caso en el cual, la fuerza que tiende a sacar la pieza de trabajo de la pinza es superior al peso del objeto. Tal situación se podría presentar debido a las fuerzas generadas por la aceleración, que se puede producir durante el transporte del objeto y/o a los cambios de orientación del movimiento durante dicha aceleración.

Para considerar dicho efecto, se modifica la última ecuación adicionando un factor  $k$  que cobije estos eventos. Se obtiene la siguiente ecuación:

$$2\mu_k F_g = k\omega$$

Donde  $k$  toma el valor de 3.0, si la fuerza de aceleración se aplica en el mismo sentido de la fuerza de gravedad; si la aceleración se aplica en sentido contrario a la fuerza de gravedad,  $k$  toma el valor de 1.0, y si la aceleración se aplica en dirección horizontal,  $k$  toma el valor de 2.0.

Para establecer la fuerza mínima en un movimiento en sentido de la fuerza de gravedad, esto es  $k = 3$ , para nuestro sistema de sujeción con un objeto cúbico de 200 gramos y un coeficiente de fricción estática entre los materiales de 0.2, la fuerza es de 15 N.

En conclusión se requiere un sujetador que proporcione 15 N de fuerza astrictiva y de 0.5 cm a 2 cm de desplazamiento por dedo para su correcto funcionamiento.

Se proponen pinzas angulares para la sujeción del objeto.

Al presentarse este trabajo como un prototipo, se planteó la posibilidad de encontrar los actuadores neumáticos que presentasen una vida útil media. Al no encontrar sujetadores bajo la anterior propuesta, se recurrió a la cotización y compra del mismo.

Al cotizar el sujetador entre las principales compañías del ramo, se encontró que la gama MHY2 de SMC® y las líneas HGW y HGWC de FESTO® presentan una serie de productos que son especializados para estas tareas; al considerar los tamaños, pesos y fuerzas astrictivas de los productos, ponderando también el costo de los mismos, se obtuvo como resultado que, las pinzas neumáticas MHY2-10D de SMC® presentaba la mejor alternativa de compra. Se presentan las características esenciales del actuador en la siguiente tabla (consultar anexo para mayor información).

Specifications					
Fluid	Air				
Operating pressure	0.1 to 0.6 MPa				
Ambient and fluid temperature	-10 to 60°C				
Repeatability	±0.2 mm				
Max. operating frequency	60 c.p.m.				
Lubrication	Not required				
Action	Double acting				
Auto switch (Option)	Solid state switch (3-wire, 2-wire)				

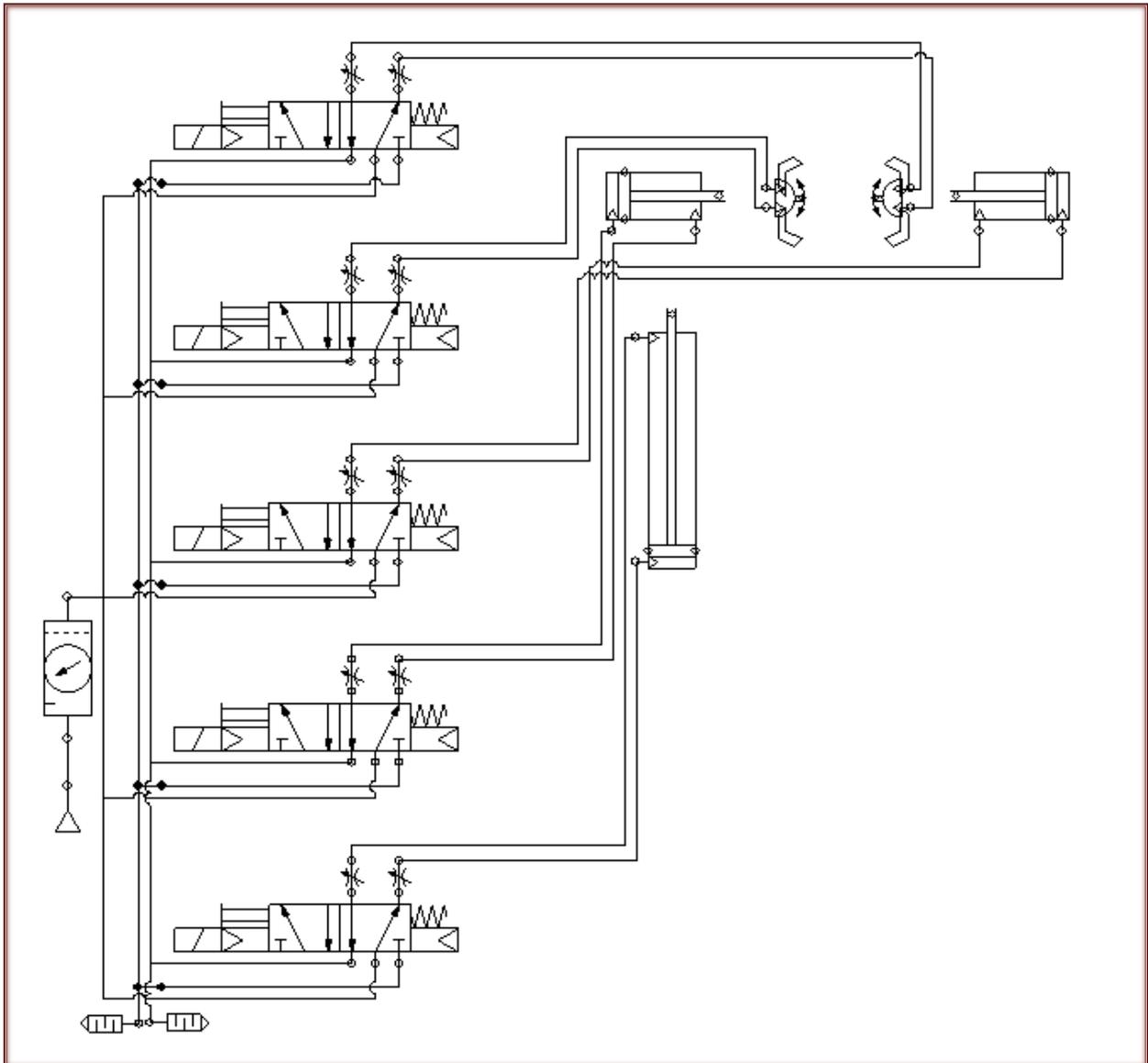
 Note) Refer to page 12-13-1 for further information on auto switches.

Model					
Model	Bore size (mm)	Effective gripping force <sup>(1)</sup> (N·m)	Opening/Closing angle (Both sides)		Weight <sup>(2)</sup> (g)
			Opening side	Closing side	
MHY2-10D	10	0.16	180°	-3°	70
MHY2-16D	16	0.54			150
MHY2-20D	20	1.10			320
MHY2-25D	25	2.28			560

 Note 1) At the pressure of 0.5 MPa  
Note 2) Except auto switch

**Tabla 3.1** Tabla de especificaciones de la gama MHY2 de pinzas neumáticas SMC®

Los actuadores lineales se plantearon y se adquirieron de vida útil media, así como las electroválvulas, quedando el diseño del circuito neumático, tal como se muestra en el siguiente diagrama, realizado en Automation Studio® 5.2:



**Figura 3.3** Diagrama del circuito neumático para el robot ortogonal

El circuito inicia en una toma de aire directa del compresor, la cual se manda regulada en presión de 10 a 20 bars. Seguido de esto pasa por una unidad de mantenimiento neumático, que consta de un filtro, un lubricador y un regulador de presión. Cabe señalar que el prototipo necesita de estos elementos, es decir, se debe contar con una toma de aire tratado a una presión de 0.6 MPa.

Una vez que se dispone de la toma de aire, se conecta por manguera no. 6, a un bloque para válvulas de 5/2 vías con placa base con mando tipo servopilotaje. La placa soporta seis válvulas marca FESTO, el código de la placa es el PRMY-5-1/8-6, mencionado en el anexo de electroválvulas, ésta tiene la función de distribuir la alimentación a las electroválvulas, así como el escape de aire en las salidas de las antes mencionadas.

Las electroválvulas usadas en este proyecto son de tipo monoestable, de 5/2 vías con servopilotaje, de la marca FESTO con código de producto MYH-5/2-2,3-L-LED, mencionadas en el anexo de electroválvulas.

Los anteriores productos conmutan con una señal de 24 VCD, estas electroválvulas son diseñadas para montaje en placa base, por lo que sus salidas se encuentran en el anterior, directamente ligadas a su posición en la placa base. En las salidas de las electroválvulas se tienen conectados racores QS 1/8 a manguera no. 4 en dos de ellas y en las otras, conexiones similares a manguera no. 6.

Para que el prototipo genere los movimientos lineales, utiliza tres cilindros neumáticos redondos de la marca FESTO en su línea económica, ESEU-10, de igual forma anexados en el trabajo. Que se conectan a las salidas de las electroválvulas por medio de mangueras no. 6 como se muestra en el esquema.

Finalmente, los sujetadores neumáticos seleccionados en este proyecto, son los que considerablemente tienen una mejor relación costo – beneficio – calidad. En concreto, son los ya mencionados previamente MHY2-10D de la marca SMC®, que son las pinzas neumáticas más pequeñas de la línea y las más ligeras. Se conectan a las salidas de la válvula por medio de mangueras no. 4, en una configuración que les permite estar abiertas en el estado nominal de la electroválvula.

Este circuito neumático efectúa todos los movimientos de trayectorias lineales en el dispositivo y la sujeción de la pieza. Si se ha tratado con elementos neumáticos, se conoce que establecer control de posición y velocidad es un proceso sumamente complejo, por ello se recurrió a tener solo dos posiciones y, la velocidad, será controlada mediante la limitante de flujo de aire que llegue a los actuadores neumáticos, con la implementación de reguladores de caudal bidireccionales entre las salidas y los actuadores neumáticos.

### 3.3 Diseño de la tarjeta de control

En esta parte del trabajo se presenta el desarrollo de la tarjeta de control. En ésta se encuentra primordialmente el microcontrolador que procesa todas las señales que requiere el prototipo para funcionar.

Además, se encuentran los elementos electrónicos que adecuan las señales provenientes de los sensores, así como la etapa de potencia para los motores eléctricos. De manera similar, elementos electrónicos para generar las señales eléctricas necesarias para la activación de las electroválvulas; finalmente tiene elementos de conexión rápida con los dispositivos antes mencionados, con la fuente de energía y con la computadora a través de una comunicación USB (*Universal Serial Bus*, por sus siglas en inglés).

#### 3.3.1 Microprocesadores y microcontroladores: caracterización

Un microcomputador se compone de tres bloques fundamentales: la CPU (*Central Processing Unit*), la memoria, y la entrada y salida. Los bloques se conectan entre sí mediante grupos de líneas eléctricas denominados buses.

La CPU es el “cerebro” del microcomputador y actúa bajo el control del programa almacenado en la memoria. La CPU se ocupa básicamente de traer las instrucciones del programa desde la memoria, interpretarlas y hacer que se ejecuten. La CPU también incluye los circuitos para realizar operaciones aritméticas y lógicas elementales con los datos binarios, en la denominada Unidad Aritmética y Lógica (*ALU: Arithmetic and Logic Unit*).

En un microcomputador, la CPU no es otra cosa que el microprocesador, el circuito integrado capaz de realizar las funciones antes mencionadas. Un microcontrolador es un microcomputador realizado en un circuito integrado (*chip*).

Los microcontroladores se han desarrollado para cubrir las más diversas aplicaciones. Se usan en automoción, en equipos de comunicaciones y de telefonía, en instrumentos médicos e industriales de todo tipo, en electrodomésticos, en juguetes, etc.

Los microcontroladores están concebidos fundamentalmente para ser utilizados en aplicaciones puntuales, es decir, aplicaciones donde el microcontrolador debe realizar un pequeño número de tareas, al menor costo posible. En estas aplicaciones, el microcontrolador ejecuta un programa almacenado permanentemente en su memoria, el cual trabaja con algunos datos almacenados temporalmente e interactúa con el exterior a través de las líneas de entrada y salida de que dispone. El microcontrolador es parte de la aplicación: es un controlador incrustado o embebido en la aplicación (*embedded controller*). (Pallás & Valdes, 2007)

### 3.3.2 Microcontrolador PIC18F4550 en la tarjeta de control

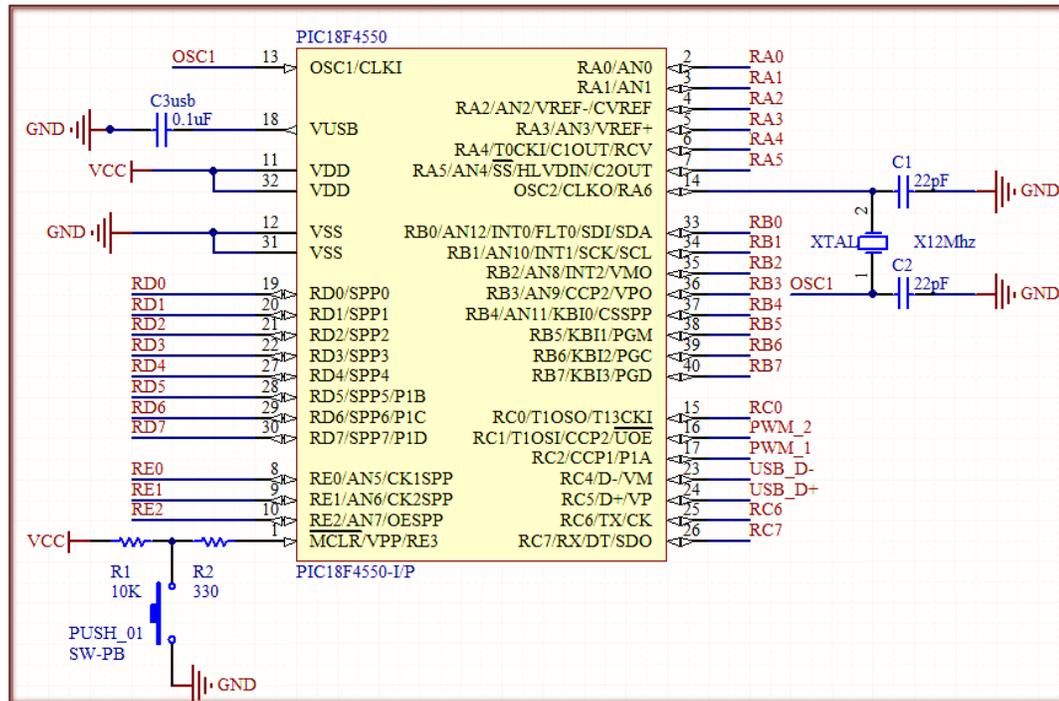


Figura 3.4 Diagrama de pines del microcontrolador PIC18F4550 en la tarjeta de control

En la figura anterior se muestra el diagrama general de pines del microcontrolador, comúnmente abreviado MCU, en concreto es el PIC18F4550 de la marca Microchip®. Este microcontrolador fue elegido con base en los siguientes módulos o características con las que cuenta:

- Módulo de comunicación USB configurable por descriptores de *hardware*.
- Estructura de oscilador flexible configurable en oscilador enganchado en fase de alta precisión - *HP PLL* (*High-Precision Phase-Locked Loop*) necesario para la comunicación USB.
- Un módulo CCP (*Capture/Compare/PWM*) y un módulo ECCP (*Enhanced CCP*) que serán configurados como moduladores por ancho de pulso - *PWM* (*Pulse Width Modulation*). Se necesitan dos PWM en hardware para el proyecto, lo cual cubre al mínimo necesario.
- Tres interrupciones externas, importantes para el estudio de microcontroladores.
- Veintiocho entradas/salidas digitales en la configuración mostrada en la figura, en este proyecto se requieren diecinueve de éstas, siendo este microcontrolador el más próximo en la familia PIC18XX50 en esta característica.

En el esquema se muestra un cristal de 12 MHz, conectado a las entradas del oscilador del microcontrolador en paralelo con capacitores de 22pF recomendados por el fabricante. Pese a que cuenta con la posibilidad de operar el periférico del USB a una frecuencia diferente a la del reloj del microcontrolador, la tarjeta de control trabajará a una frecuencia de 48 MHz.

La frecuencia se establece en 48 MHz para poder establecer una comunicación USB de alta velocidad (*USB full speed*). Esto se logra por medio de la configuración de los siguientes parámetros:

- Definir el modo del oscilador como cristal de alta velocidad con PLL habilitado - HSPLL (*High-Speed Crystal/Resonator with PLL Enabled*).
- Estableciendo el PLL con un valor igual a tres, definido por la frecuencia del cristal (12 MHz) en la hoja de datos.
- Activando el bit de configuración USBDIV. Este bit divide la frecuencia de salida del PLL en dos.
- Establecer la configuración por defecto del divisor de reloj del MCU, esto es un divisor igual a dos (CPUDIV1).

En la terminal 1 (RE3) se muestra un pequeño circuito que permite el reinicio del dispositivo mediante la función *Master Clear* (MCLR) una vez accionado el pulsador, que habrá de interrumpir la señal VCC (5V) suministrando un potencial cero. Destacando que esta terminal es de activación en bajo.

En la terminal 18 (VUSB) se muestra un capacitor a tierra que es necesario según las tablas del fabricante para la correcta implementación del protocolo de comunicación USB. Las terminales 11,12, 31 y 32 son destinados para la alimentación del dispositivo.

En adición se muestra la distribución de los puertos del MCU que serán utilizados para las entradas y salidas del prototipo RDO. El PIC18F4550 cuenta con 35 entradas/salidas (I/O) digitales; en la configuración mostrada se puede disponer de hasta 32 I/O digitales, de las cuales 13 de ellas podrían configurarse como entradas analógicas. En esta tarjeta se restringe el uso a 27 I/O digitales, de las cuales 3 de ellas podrían configurarse como entradas analógicas para fines externos al prototipo RDO que el usuario requiera.

Cabe destacar que tanto el diseño esquemático como el PCB se realizaron en el software Altium Designer Summer 2009 ® bajo licencia de evaluación, por lo cual no es posible comerciar las tarjetas controladoras hasta que sea adquirida la licencia; de la misma forma se hace notar que los diagramas electrónicos que se muestran en este trabajo se realizaron en esta plataforma.

El uso de etiquetas en las conexiones es para evitar el enredo de las mismas al presentarlas completas e incluso en forma de *bus*.

### 3.3.3 Conectores principales en la tarjeta de control

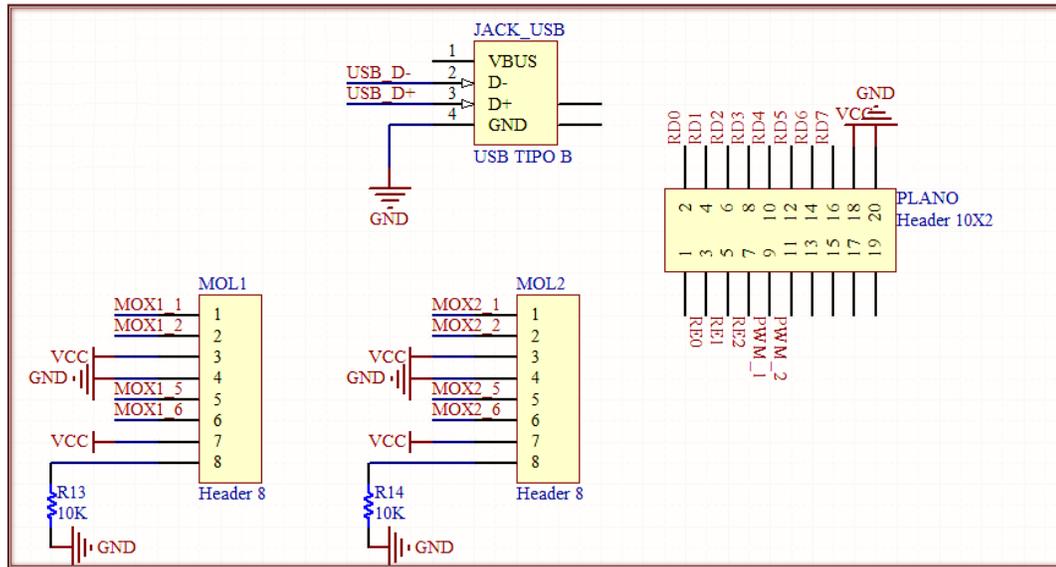


Figura 3.5 Principales conectores en la tarjeta de control del prototipo

El conector USB usado en la tarjeta de control es de Tipo B, que permite una conexión rápida con un cable de conexión USB Tipo B a conexión USB Tipo A para ser conectado a la computadora. Puesto que a la tarjeta se le proporciona una alimentación propia externa, no es necesaria la alimentación que pudiese proveer la computadora a través del puerto USB, por lo tanto permanece sin conexión la terminal 1 (VBUS) en este conector, destinada para este fin. La terminal de información 2 (D-) estará conectada a la terminal especializada 23 (RC4, D-) del MCU; de manera similar la terminal de información 3 del conector en la 24 (RC5, D+). Por último se empara la referencia del USB con la referencia de toda la tarjeta.

A continuación se describe la conexión del Molex del motor del Brazo 1 del RDO, etiquetado como MOL1. En la terminal 1 etiquetada como MOX1\_1 se conecta al cable negro del motor (GND), la terminal 2 MOX1\_2 al cable rojo (VC 12V), la terminal 3 al cable café (VCC) para alimentar los sensores de efecto Hall del *encoder*, la terminal 4 al cable verde (GND) como complemento a los sensores, la terminal 5 MOX1\_5 a la salida A del *encoder*, la terminal 6 MOX1\_6 a la salida B del *encoder* y, entre la terminal 7 y 8 se conecta un interruptor mecánico para detectar el origen del giro del motor. Sirva la explicación del MOL1 para describir de manera análoga el MOL2 que no aparece en el diagrama, pero que es idéntico a éste.

En adición a los puertos requeridos para el funcionamiento del prototipo RDO y, diseñando siempre para la comodidad e interactividad con el usuario, se pone a disposición en una tira doble de *headers* el puerto D del PIC18F4550, el puerto E, la alimentación de 5V y las terminales de los puertos CCP; teniendo el cuidado de quitar los *jumpers* que se encuentran a un costado del puente H descrito más adelante. Esto con la finalidad de que, el estudiante pueda utilizar la tarjeta con comunicación USB y la opción de entradas analógicas en el puerto E, para algún otro objetivo diferente al que está planteado inicialmente, evitándole gastar tiempo, dinero y esfuerzo en hacer una tarjeta adicional.

### 3.3.4 Activación de electroválvulas, inversores Schmitt Trigger y alimentación

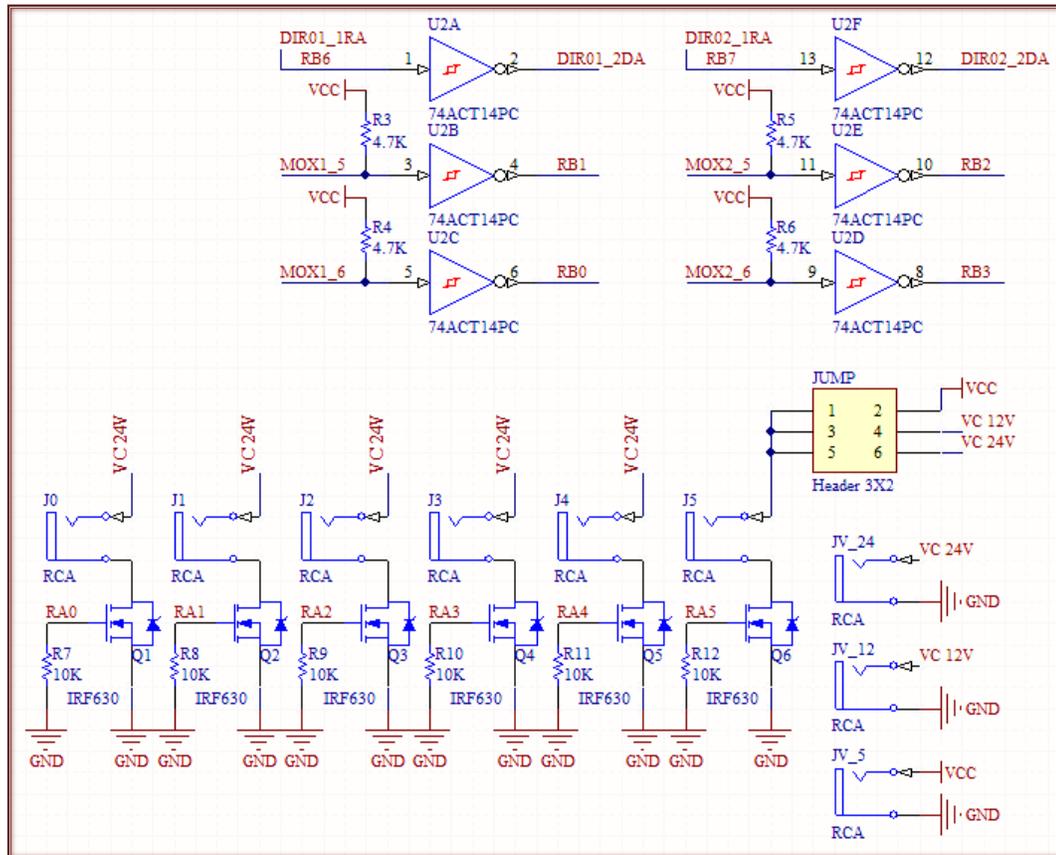


Figura 3.6 Activación de electroválvulas, inversores Schmitt Trigger y alimentación

Con el afán de generar una tarjeta de control con conexiones rápidas, los voltajes de alimentación y las electroválvulas se conectan por medio de terminales RCA que presentan una conexión muy rápida y sencilla.

En el lado izquierdo de la figura antes mostrada se encuentra una hilera de transistores MOSFET IRF630. Estos transistores tienen conectado en su compuerta una terminal del puerto A del PIC con una resistencia de *pull-down* de  $10K\Omega$ , que se asociará a una electroválvula, con excepción del MOSFET asociado al RA5 que se explicara más adelante. En los conectores RCA asociados a los IRF630 antes mencionados, se conectan los cables que a su vez conectan con las electroválvulas para ser activadas. El conector en su terminal interna, se conecta directamente a la terminal positiva de la alimentación de 24 VCD; la terminal complementaria del conector RCA se conecta al drenaje del MOSFET. Por último, la fuente del MOSFET se conecta directamente a tierra. El conjunto de estas conexiones configura al MOSFET como interruptor, éste conecta al drenaje con fuente, al activar en alto el puerto asociado.

El MOSFET asociado al RA5 ha sido diseñado para la activación de un actuador externo al prototipo RDO, en caso que el estudiante así lo requiera; señalando que la alimentación puede ser configurable mediante el *jumper* a un costado del MOSFET en los tres posibles valores que se mencionan a continuación.

Los potenciales eléctricos de alimentación que se requieren para el funcionamiento de la tarjeta de control, deben estar conectados en los RCA que aparecen en la última figura mostrada en el lado inferior derecho. Los potenciales necesarios se describen a continuación:

- 24 VCD: Esta alimentación es necesaria para encender las electroválvulas y solo está alambrado en los conectores RCA. Se requiere que la fuente que suministre este potencial entregue una corriente pico de 800 mA.
- 12 VCD: Esta diferencia de potencial se encuentra ligado al suministro de energía de los motores y de ser necesario por el alumno al MOSFET asociado al RA5, se requiere un mínimo de 2.5 A.
- 5 VCD - VCC: Es la alimentación principal de los circuitos integrados, dado que funcionan con tecnología TTL y de ser necesario por el alumno, servirá para activar un actuador asociando al puerto RA5, se requiere un mínimo de 1 A.

Los inversores *Schmitt - Trigger*, representados en la parte superior de la última figura mostrada, son utilizados para asegurar la estabilidad de la señal proveniente de los *encoders* de ambos motores, por lo tanto son necesarios cuatro de éstos para la realización de dicha tarea. En adición, si consideramos que para el giro de un motor se necesita de una señal y el negado de esta señal, para ambos motores se necesitarían dos inversores, completando el empaquetado del HD74LS14P, adjuntado en las hojas de datos, que presenta seis de estos inversores. La conexión es directa para la señal de control de giro en los motores y para la señal proveniente del sensor de efecto Hall, el fabricante del motor establece que ha de utilizarse una resistencia de *pull-up* de 4.7 en la salida de dichos sensores para su correcto funcionamiento.

### 3.3.5 Circuito de conmutación

La velocidad que alcanza un motor de imán permanente depende de la magnitud de la corriente que pasa por el devanado de la armadura. Por lo tanto, para controlar la velocidad se puede utilizar el control del voltaje que se aplica a la armadura. Sin embargo, dado que el empleo de fuentes de voltaje de valor fijo es frecuente, el voltaje variable se logra mediante un circuito electrónico y el uso de la técnica llamada modulación por ancho de pulso - PWM- (Bolton, 2003) .

Por otro lado, para lograr la inversión de giro de estos elementos, es necesario realizar una conmutación en la alimentación de voltaje fijo de CD. Para lograr esto, generalmente es utilizado un puente inversor monofásico (Rashid, 2004, págs. 232,233), comúnmente conocido como circuito H de conmutación o puente H. En la figura 3.7 se presenta la configuración del circuito diseñado para esta aplicación, el cual cuenta con un grupo de transistores destinados a la excitación de los transistores que se encargan de manipular a los actuadores eléctricos.

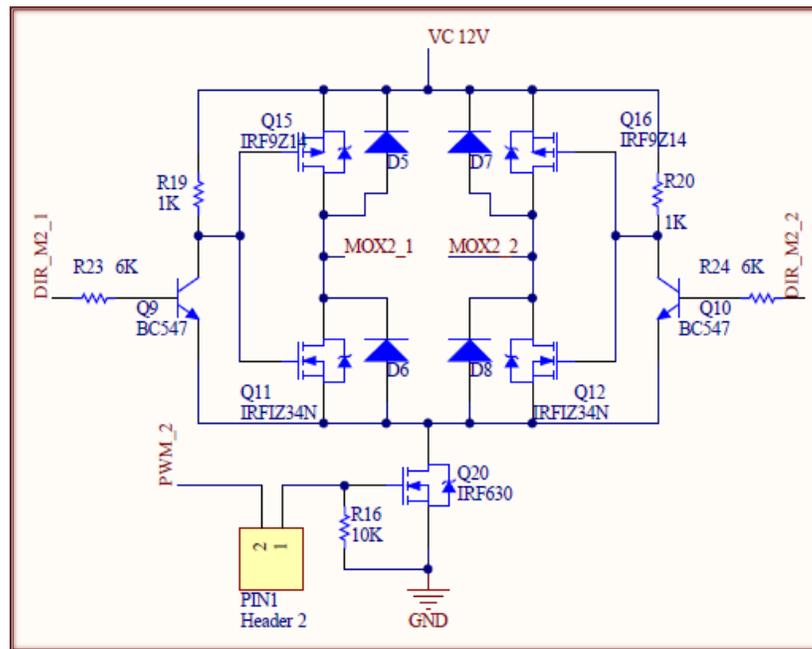


Figura 3.7 Circuito Puente H implementado

La descripción de las conexiones en el puente H implementado para el motor 1, correspondiente al brazo de 2 GDL, es idéntica a la descripción del motor 2; por lo que solo se presenta una.

El circuito de conmutación se alimenta con 12 V.

La señal de control para el giro del motor es administrada por el microcontrolador. De esta señal se obtiene una invertida al pasar por el *Schmitt-Trigger* descrito anteriormente. Estas dos señales pasan por un proceso de optoacoplamiento (descrito más adelante) para que activen los transistores BJT que saturan a los MOSFET.

En conocimiento de lo anterior, se hace notar que existen solo dos pares lógicos de entrada, que controlan la conmutación del motor. Estos pares son (0,1) y (1,0).

Para realizar el circuito mostrado en la figura 3.7 se utilizaron MOSFET canal P con matricula IRF9Z20 y MOSFET canal N con matricula IRF630, ambos de corriente máxima de 9 A. Para saturar los transistores se implementó un circuito adicional con transistores BJT547 bajo las siguientes condiciones.

Para el cálculo de las resistencias de colector y base se tomó en cuenta que la beta del transistor oscilaba en un valor promedio de 185 y se requiere una corriente de saturación en el colector de 0.005 A.

$$R_C = \frac{V_{cc}}{I_{Csat}} = \frac{12V}{0.005A} = 2400\Omega \approx 2200\Omega$$

$$I_{Bmin} = \frac{I_{Csat}}{\beta} = \frac{0.005}{185} = 27\mu A$$

Multiplicamos la  $I_{Bmin}$  por dos para asegurar la saturación del transistor.

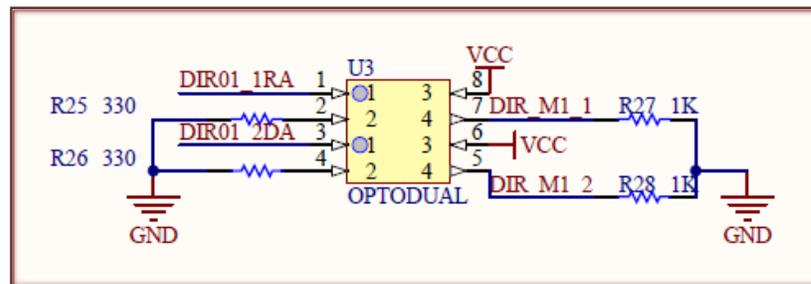
$$I_B = (2)I_{Bmin} = (2)27\mu A = 54\mu A$$

$$R_B = \frac{4.3V}{54\mu A} = 72920\Omega \approx 68000\Omega$$

Las aproximaciones en las resistencias se realizaron con base a la lista de resistencias comerciales.

Por último, es importante destacar que el circuito de conmutación está interrumpido a la referencia por un MOSFET canal P IRF630, el transistor está configurado como interruptor, teniendo como entrada el modulador por ancho de pulso, este último es generado por el microcontrolador. De manera que, el circuito permite el paso de corriente por el motor solo cuando existe una señal controlada. Se hace notar que el MOSFET cuenta con una resistencia de *pull-down* en su compuerta, y que la señal puede ser interrumpida físicamente por el usuario con la implementación de un *jumper*; éste es una protección que fue diseñada en caso de que el alumno necesite los módulos del microcontrolador para actividades externas al uso del robot didáctico.

En la figura 3.8 se puede apreciar el uso del optoacoplador dual PC827 de SHARP®, con el cual se aísla la etapa de control con la etapa de potencia, ya que operan en dos niveles distintos de voltaje de CD, y al ser cargas inductivas las operadas en potencia, almacenan energía que puede ser liberada en un momento posterior, causando problemas a los dispositivos de baja potencia.



**Figura 3.8** Circuito optoacoplador para la señal de control de motores

Una vez descritos todos los circuitos que comprenden a la tarjeta de control, se presenta el esquemático de la misma, desarrollado en Altium Designer Summer 2009 ® en la página siguiente. En él se puede apreciar algunos circuitos que fueron omitidos hasta esta parte, dado que eran redundantes pero necesarios para el desempeño del robot didáctico.

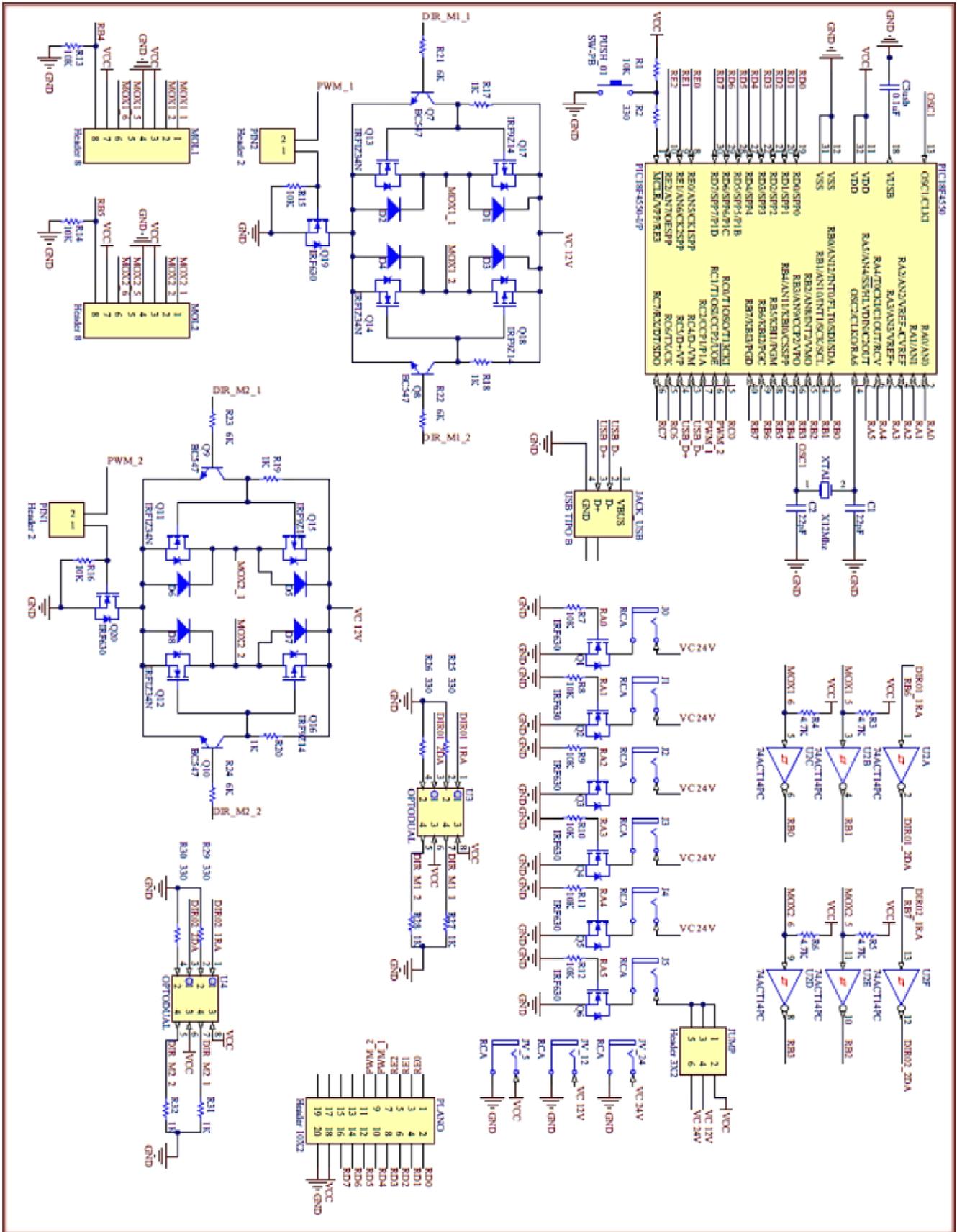


Figura 3.9 Esquemático completo de la tarjeta de control

### 3.4 Diseño del control

En esta parte del trabajo se describen los elementos que ejecutan el control de los movimientos en el robot didáctico. Por lo descrito en el desarrollo de la tarjeta de control, es claro ver que el accionamiento de los elementos neumáticos es de forma directa, ya que solo pueden tomar dos posiciones, el control es un interruptor en lazo abierto. Por otra parte, el control de posición angular digital para los motores eléctricos de CD es de tipo PD (proporcional derivativo), el cual es un tema interesante de la materia de control de máquinas eléctricas, que será tratado desde el punto de vista de la implementación más sencilla, esto con el fin de exponer el tratamiento de los elementos que rodean a la misma.

#### 3.4.1 Esquema del control en el microcontrolador

De manera general, se describe a continuación el esquema de trabajo que sigue el microcontrolador para la ejecución de las instrucciones que le sean ordenadas por el usuario.

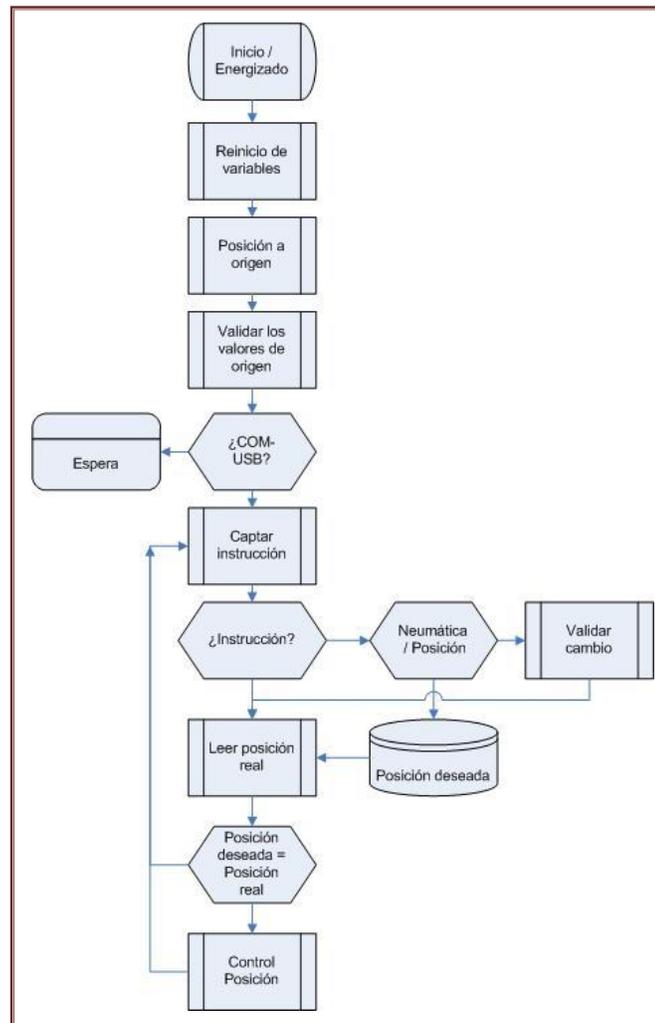


Figura 3.10 Diagrama de flujo de la operación de control

El proceso toma inicio desde que el microcontrolador es energizado por la tarjeta de control. Los valores de dirección de los puertos son reiniciados a los valores predeterminados, hasta que llega a la línea de comando donde se les asigna una dirección. Hasta ese momento se puede definir qué puertos serán entradas o salidas, es por esto que, si se está trabajando con el robot didáctico y se reinicia el microcontrolador, éste perderá la posición que tenía y volverá a su posición de reposo.

En el paso siguiente, el control genera las señales para establecer las posiciones iniciales de los actuadores neumáticos. De la misma forma, genera una señal de control para que el primer motor gire hasta su posición de origen, para después mover el segundo motor hasta su posición. La posición de origen se encuentra desfasada de la posición de referencia (Posición real =  $0^\circ$ ) en  $90^\circ$ .

Terminada la rutina, se establecerá que la variable que contiene a la posición de referencia, tiene un valor de  $90^\circ$  anti horarios. De igual manera cambiará el valor de la variable de la posición deseada a  $0^\circ$ . Lo que generará que instantes del primer movimiento, se genere un segundo de acomodo final.

Una vez posicionados los motores en el origen, el microcontrolador se configurará para establecer comunicación USB con la computadora. Enviará por este método una solicitud de conexión, si está correctamente configurado se conectará a los pocos segundos, de lo contrario no realizará otra actividad hasta su conexión. Esta decisión no puede tomarse de otra manera, dado que las instrucciones que el usuario manda desde la interfaz gráfica se envían por este medio.

Una vez establecida la comunicación entrará a un bucle sin fin, del que solo saldrá al quitarle la alimentación a la tarjeta de control. La primera acción del bucle será la de verificar si existen datos en *buffer* en entrada, de existir una instrucción se puede dividir el flujo en dos partes, se traduce la información para verificar si se trata de un movimiento neumático o un movimiento angular; de no existir datos en el *buffer* verificará inmediatamente la posición de los codificadores.

Si se ha verificado una sentencia de movimiento neumático, el microcontrolador genera de manera directa la activación o el apagado de la electroválvula relativa a este movimiento, para regresar al bucle general posteriormente.

Si se ha verificado una sentencia de movimiento angular, el microcontrolador sumará o restará, según la orientación del giro, los grados indicados por el usuario y guardará en la variable de posición deseada destinada al motor correspondiente. Después de esto regresará al bucle general con la tarea de la verificación de posición.

Para verificar la posición real, es decir, aquella que se está midiendo a partir de la posición de referencia, es necesario aplicar un tipo de máscara en los puertos donde se lee la señal de los codificadores, para secuenciar los valores que se están obteniendo, para que su análisis sea más comprensible; el cual se explica más adelante con mayor detalle.

En el bucle general del programa se tiene una comparación que ejecuta la función del control de motores, en concreto, verificar si la posición deseada corresponde a la posición real en la que se encuentran los motores, pues de ser así se cicla el programa principal en verificar si ha llegado alguna nueva instrucción. Pero en el caso que esto no se cumpla, el control de motores realizará la resta entre estos valores, que almacenará en una variable de error. El error es de mucha importancia en este proyecto, la magnitud de éste será proporcional a la señal de control que se le introduzca para su movimiento, está señal es el PWM. El signo del error será la pauta para la señal de conmutación en el puente H.

De tal manera que siempre se esté verificando, si existe una nueva instrucción en el *buffer* de entrada de la comunicación USB, para ejecutar la instrucción recibida; así como los cambios en las señales de los codificadores se ejecute el control de posición para que se mantengan los motores en los ángulos deseados.

A continuación se muestran temas específicos que son el sustento para la implementación del control.

### 3.4.2 Protocolo de comunicación USB e implementación en PIC18F4550

A pesar de que el puerto USB nos ofrece más ventajas que sus predecesores, su complejidad para implementarlo es enorme, ya que su funcionamiento está basado en protocolos de software.

Sólo puede haber un *Host* en el bus que maneja a todos los componentes conectados como se indica en la figura 2.27

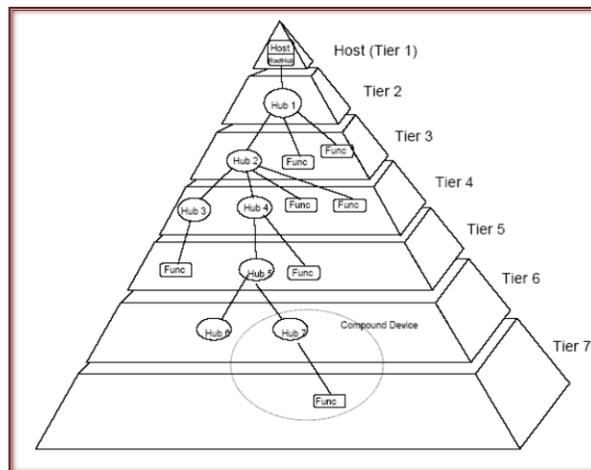


Figura 3.11 Pirámide de los componentes conectados al Host

El HUB es un elemento *plug and play* en la estructura USB y es un concentrador al cual, se le pueden agregar más dispositivos USB, incluyendo otro HUB. Ver figura 3.12.

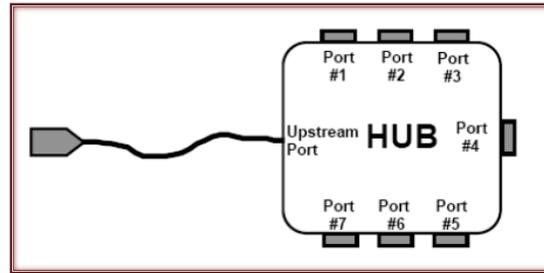


Figura 3.12 Diagrama del HUB

La velocidad de transferencia depende de la velocidad del HUB que se esté utilizando, figura 2.29.

Las velocidades típicas son: 480Mb /s en *high speed*, 12Mb/s *full-speed* y 1.5Mb/s en *low-speed*.

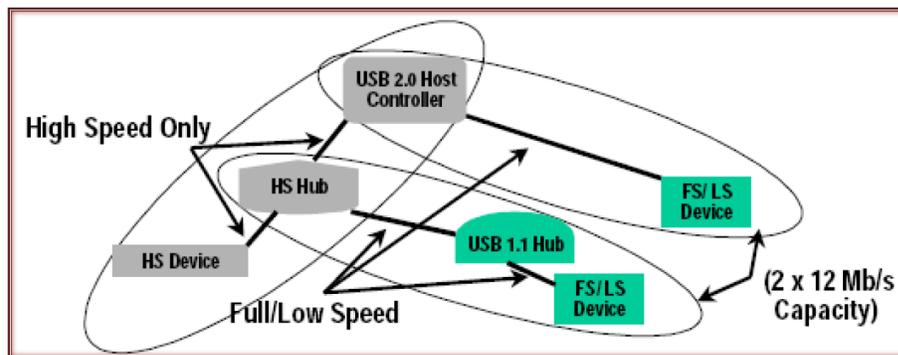


Figura 3.13 Diagrama de las velocidades de transferencia USB

El cable que típicamente es usado no es largo, debido a la velocidad de transferencia y tiene la estructura mostrada en la figura 3.14. Sus terminales son del tipo diferencial y consta de 4 hilos.

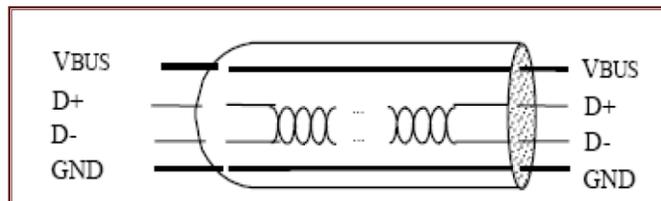


Figura 3.14 Diagrama del cable USB

La arquitectura USB comprende cuatro tipos básicos de transferencia de datos:

- Control de transferencias (*Control Transfers*): Es usado para configurar un dispositivo al momento de que se conecta. Los datos entregados pueden perderse.
- Transferencia de Bloque de Datos (*Bulk Data Transfers*): Entrega el dato por volumen, el ancho de banda puede variar. Es usado en escáner o cámaras. La ráfaga de datos es secuencial.

- Interrupción de la transferencia de datos (*Interrupt Data Transfers*): Se utiliza para la entrega oportuna pero fiable de datos, por ejemplo, caracteres, coordenadas de eco perceptible a los humanos o respuestas de retroalimentación.
- Transferencia de datos isócrona (*Isochronous Data Transfers*): Ocupa una cantidad de ancho de banda USB prenegociado con una latencia de entrega prenegociada. (También se llama flujo de transferencias en tiempo real).

Aunque el PIC no puede funcionar como *host*, ya que se requiere de una gran capacidad de manejo de datos para administrar a cada componente del BUS, es suficiente que se pueda administrar como un dispositivo (*device*), para esto se requiere almacenar en memoria los protocolos necesarios para enlazarse al *host*.

Estos protocolos se le llaman descriptores y sirve para informarle al *host* todo lo necesario para que pueda administrarlo.

Recordemos que los PICs de la serie 18FX5X tienen tres modos de funcionamiento:

1. USB Dispositivo de Interfaz Humana (HID): Es de velocidad baja, no requiere driver.
2. USB *Communication Device Class* (CDC): Velocidad Media, requiere driver. Crea un Puerto Serie Virtual.
3. USB *Custom Driver*: Velocidad Alta, requiere driver. Este es el modo que usa WinUSB (para Windows Vista) y el mpusbapi (Windows 2000 y posterior).

Dentro de los protocolos hay que especificar el tipo de transferencia de datos a usar (*endpoints*), VID&PID (identificación del fabricante e identificación del producto por sus siglas en inglés), nombre y serie del producto que se conecta, para que el *host* identifique al *driver* y pueda instalarlo con el fin de que el dispositivo pueda formar las *pipes* o túneles para su comunicación con el *host*.

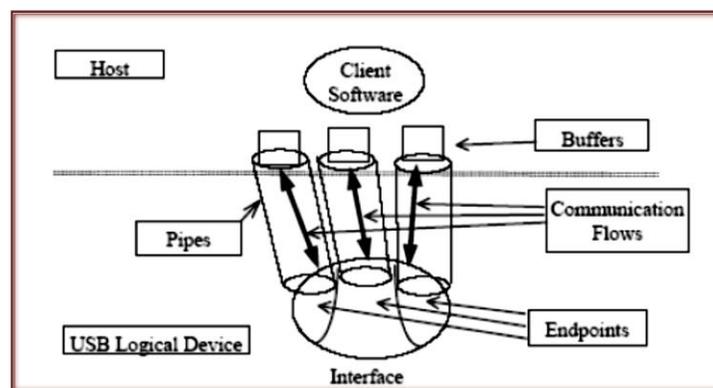


Figura 3.15 Flujo de información en la comunicación USB

### 3.4.3 Codificadores incrementales de cuadratura e implementación

Los codificadores incrementales se encuentran con frecuencia en sistemas de control modernos para convertir desplazamiento lineal o rotatorio en un código digital o señales de pulsos. Los codificadores incrementales proveen un pulso por cada incremento de resolución pero hace distinciones entre dichos incrementos. (Kuo, 1996)

Las señales que nos ofrecen los codificadores incrementales de cuadratura son señales cuadradas periódicas, estas señales cuadradas están desfasadas  $90^\circ$  una respecto a la otra.

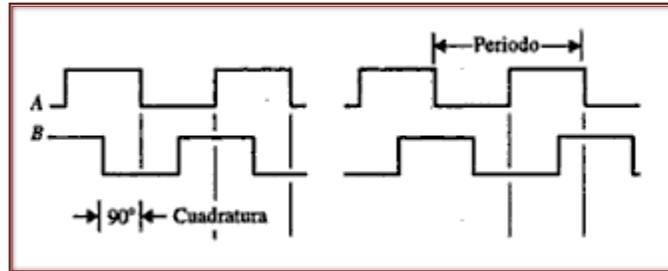


Figura 3.16 Señales de salida de un codificador de cuadratura

En el caso particular de este proyecto, los codificadores de cuadratura miden  $4^\circ$  por periodo. Por lo que es necesario establecer una relación entre ambas señales, esto es dar una secuencia a los cambios de los estados lógicos de las señales. Si se fracciona el periodo de las señales en cuatro partes, como se muestra en el diagrama siguiente y se le asigna un valor, se podrá percibir más claramente el aumento o decremento de la posición.

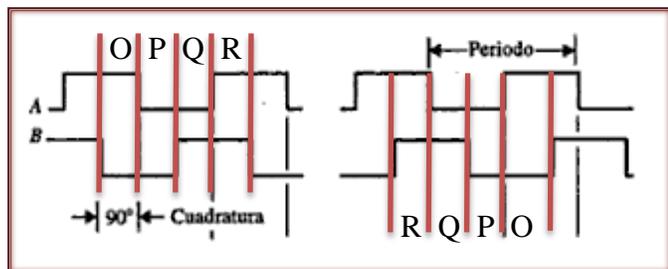


Figura 3.17 División de las señales de salida de un codificador de cuadratura

Se puede observar que si la secuencia de O hacia R de lado izquierdo es simétrica a la del derecho, la secuencia de R hacia O. Estas señales se dan cuando gira en sentido horario u antihorario, es por esto que podemos definir el sentido del giro más la posición angular en ángulos enteros con una resolución de un grado.

Por último se presenta la siguiente tabla que ejemplifica la lectura del codificador en el microcontrolador.

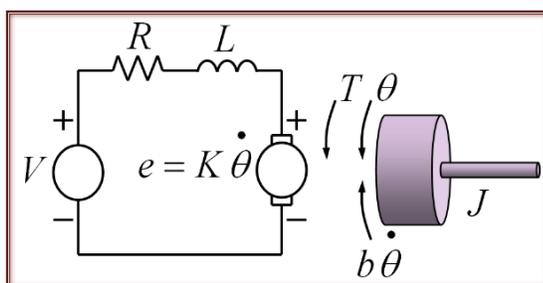
Canal A	Canal B	Posición en el gráfico	Asignación numérica
1	0	0	1
0	0	P	0
0	1	Q	3
1	1	R	2

**Tabla 3.2** Asignación de valor numérico a los estados lógicos del codificador

Por lo que se puede deducir que, si el número asignado decrece estará girando en un sentido, y en caso de incrementar estará girando en el sentido contrario.

### 3.4.4 Modelo matemático de un motor de CD

Se considera importante conocer el modelo matemático de un motor de CD, para el caso en el que se busque realizar un control que dependa de estos parámetros, se agrega solo como referencia para el lector, dado que el control implementado no depende del modelo matemático.



**Figura 3.18** Esquema de un motor de CD

Dónde:

$b$  = Coeficiente de amortiguamiento

$R$  = Resistencia del motor

$L$  = Inductancia del motor

$J$  = Momento de inercia

$\dot{\theta} = \omega$  = Velocidad angular eje del motor

$e$  = Contra FEM

$T$  = Par generado

$K_b$  = Constante de contra FEM

$K_i$  = Constante de Par

De la ecuación eléctrica del motor:

$$V(t) = Ri(t) + L \frac{di(t)}{dt} + K_b \dot{\theta}$$

Dónde:

$$e = K_b \dot{\theta}$$

Por lo tanto:

$$-V(t) + Ri(t) + L \frac{di(t)}{dt} + e = 0 \dots (1)$$

De la ecuación mecánica del motor:

$$T = J \frac{d\omega}{dt} + b\omega \dots (2)$$

Las ecuaciones 1 y 2 se relacionan mediante:

$$e = K_b \dot{\theta} \quad y \quad T = K_f i(t)$$

De donde:

$$K_b = K_f = K$$

Entonces:

$$-V(t) + Ri(t) + L \frac{di(t)}{dt} + K \dot{\theta} = 0 \dots (1')$$

$$-Ki(t) + J \frac{d\omega}{dt} + b\omega = 0 \dots (2')$$

Aplicando transformada de Laplace a (1') y (2')

$$-V(s) + RI(s) + LSI(s) + K\omega(s) = 0 \dots (1'')$$

$$-KI(s) + JS\omega(s) + b\omega(s) = 0 \dots (2'')$$

Despejando I(s) de (2'')

$$I(s) = \frac{JS\omega(s) + b\omega(s)}{K}$$

Sustituimos en (1'')

$$V(s) = R \frac{(JS\omega(s) + b\omega(s))}{K} + LS \frac{(JS\omega(s) + b\omega(s))}{K} + K\omega(s)$$

$$V(s) = R\omega(s) \left( \frac{JS + b}{K} \right) + LS\omega(s) \left( \frac{JS + b}{K} \right)$$

$$V(s) = \omega(s) \left[ \left( \frac{JS + b}{K} \right) (R + LS) + K \right]$$

$$\frac{V(s)}{\omega(s)} = \frac{(JS + b)(R + LS)}{K} + K = \frac{(JS + b)(R + LS) + K^2}{K}$$

Función de transferencia  $\frac{\dot{\theta}}{V}$

$$\frac{\omega(s)}{V(s)} = \frac{K}{(JS+b)(R+LS)+K^2} \dots \frac{\dot{\theta}}{V}$$

### 3.4.5 Modelo a bloques del control de posición a implementar

Un controlador automático compara el valor real de la salida de una planta con la entrada de referencia (valor deseado), determina el error, y produce una señal de control que reducirá el error a cero, o a un valor muy pequeño. La forma como el controlador automático produce la señal de control, se denomina acción de control (Ogata, 1993). En los controladores industriales es muy común encontrar los seis tipos de acción básica de control: de dos posiciones o de encendido-apagado, proporcional, integral, proporcional e integral, proporcional y derivativo, y proporcional, integral y derivativo. Para los fines de este trabajo se utilizan solo las acciones proporcional y derivativa (PD).

El modelo a bloques del control de posición se describe en la siguiente figura, esbozada en SIMULINK®.

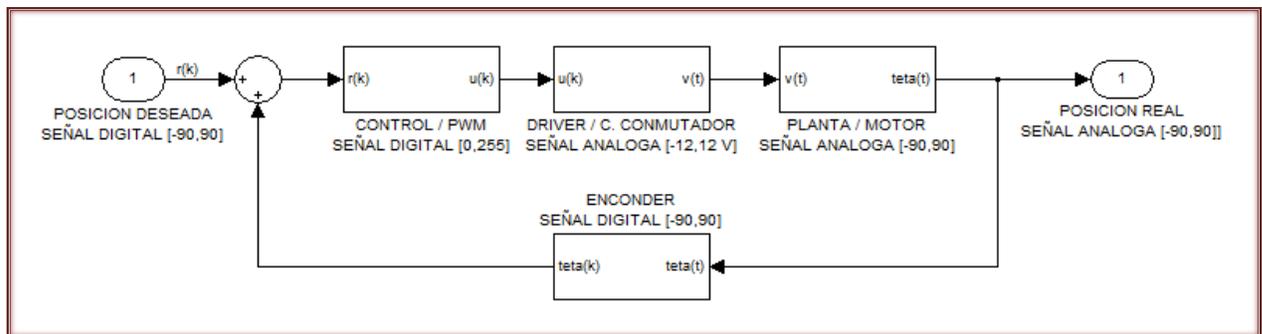


Figura 3.19 Modelo a bloques del control de posición

Si el microcontrolador ya tiene la capacidad de medir la posición con resolución de un grado, y se tiene implementada la electrónica necesaria para la generación de señales de potencia en los motores y la conmutación del giro, faltaría expresar en este modelo la función de control, que se encontrará implementado en el microcontrolador.

El control de posición en primera instancia calcula el error, como la diferencia entre la posición real y la posición deseada, almacenándolo en una variable que utilizará para realizar algunos comparativos.

El sentido de giro se expresa como:

$$G(bool) = \text{sign}(Pos_{real} - Pos_{deseada})$$

Esta última se restringe a los valores máximo y mínimo que puede tomar el motor.

Por último se establecen los valores de control que dependen de rangos en el error, la acción de control se presenta cuando el valor de salida del controlador es proporcional a la velocidad de variación de la señal de error. Esta acción muestra la ventaja de anticiparse al error aunque presenta la desventaja de producir ruido.

Finalmente se presenta la tabla de los valores discretos que toma la señal de salida conforme se presenta el error.

Mínimo error	Máximo error	Señal de control (PWM)
80	90	150
60	79	130
40	59	110
30	39	100
20	29	95
10	19	85
3	9	80
1	2	35

**Tabla 3.3** Asignación de señal de control PWM por rangos de error

La primera observación que se puede hacer es que nunca se propone saturar el controlador, dado que son carreras relativamente cortas, el desplazamiento angular a alta velocidad genera oscilaciones.

Subsecuentemente, la cantidad de rangos así como de las señales depende del usuario y de la sintonización que le parezca más óptima.

Por último, en el último rango se propone mantener energizado al motor para que presente cierta resistencia al movimiento en caso de que un factor externo indujese un par no deseado. Aunque de ocurrir esto el control está diseñado para regresar a la posición deseada una vez desaparecido el par no deseado.

## Referencias del capítulo

- Bolton, W. (2003). *Mecatrónica: Sistemas de Control Electrónico en Ingeniería Mecánica y Eléctrica*. Ciudad de México, México: Alfaomega.
- Kuo, B. C. (1996). *Sistemas de control automático*. Ciudad de México, México: Prentice Hall.
- Ogata, K. (1993). *Ingeniería de Control Moderna*. Naucalpan de Juárez, México : Prentice-Hall.
- Pallás, R. A., & Valdes, F. (2007). *Microcontroladores: Fundamentos y Aplicaciones con PIC*. Barcelona, España: Marcombo.
- Rashid, M. (2004). *Electrónica de Potencia: Circuitos, Dispositivos y Aplicaciones*. Ciudad de México, México: Pearson Educación.

## Capítulo 4. Validación del trabajo propuesto

En este capítulo se dan a conocer tanto la implementación como algunos resultados de los bloques funcionales ya descritos en el capítulo anterior. Para la demostración de los resultados, se ocupó como testigo fotografías de los bloques funcionales y algunas mediciones que se realizaron.

### 4.1 Implementación del equipo neumático

En desarrollo del sistema neumático no se realizaron algunas consideraciones que, al momento de implementar salieron al descubierto, estas consideraciones son las siguientes:

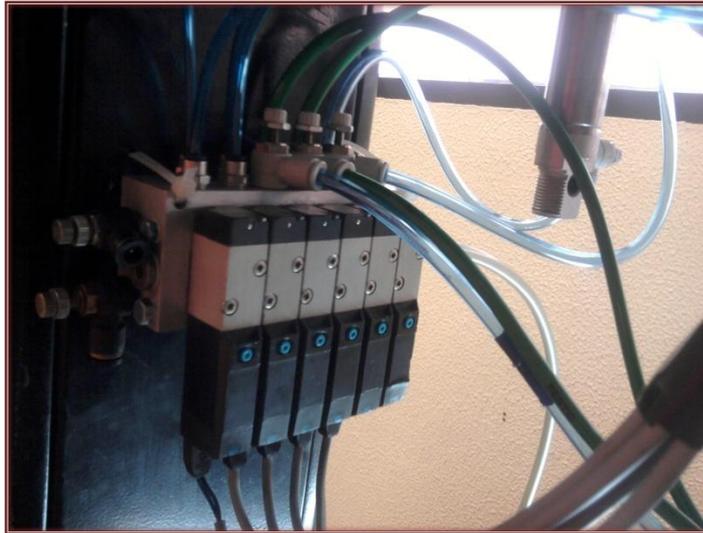
- No se consideró la necesidad de reguladores de caudal a la salida de los escapes de la placa base. Se presentó un problema que consistía en la no regulación de estos accesorios neumáticos que estaban montados en las salidas a escape de las electroválvulas; se supuso que el problema es relativo a la conexión múltiple en el escape de la base. Se planteó como solución, implementar reguladores en los escapes de la placa base; encontrando que lo anterior corrige el problema.
- No se consideró la tensión que pueden sufrir las mangueras de No. 4 conectadas a los sujetadores, que afectan directamente al giro del motor, agregando un par no deseado. Se planteó como solución cambiar de manguera rígida a manguera en espiral, especial para este tipo de trabajos, esto corrigió la problemática.

No obstante de lo anterior, la implementación del sistema neumático no presentó ningún inconveniente y se muestran algunas evidencias.



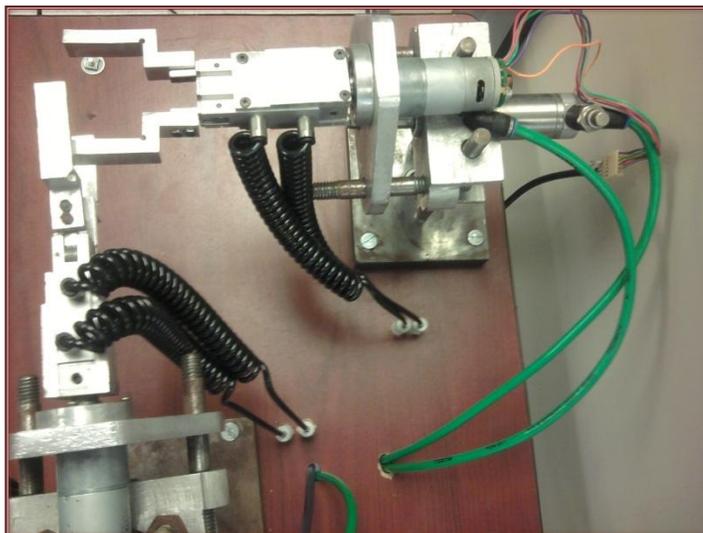
**Figura 4.1.1** Unidad de mantenimiento para la alimentación neumática

Se hace notar que la unidad de mantenimiento debe ser provista por el laboratorio en donde se emplee el prototipo.



**Figura 4.1.2** Bloque de electroválvulas montado en la mesa de trabajo

Las electroválvulas se conectan a la tarjeta de control por cables FESTO de 2 m. Las salidas de las electroválvulas tienen conectado un regulador de caudal a Racor QS manguera No.6, el regulador complementario se encuentra en el pistón correspondiente, como también se aprecia en la siguiente figura. Las mangueras negras en espiral permiten un giro suave con muy poca tensión.



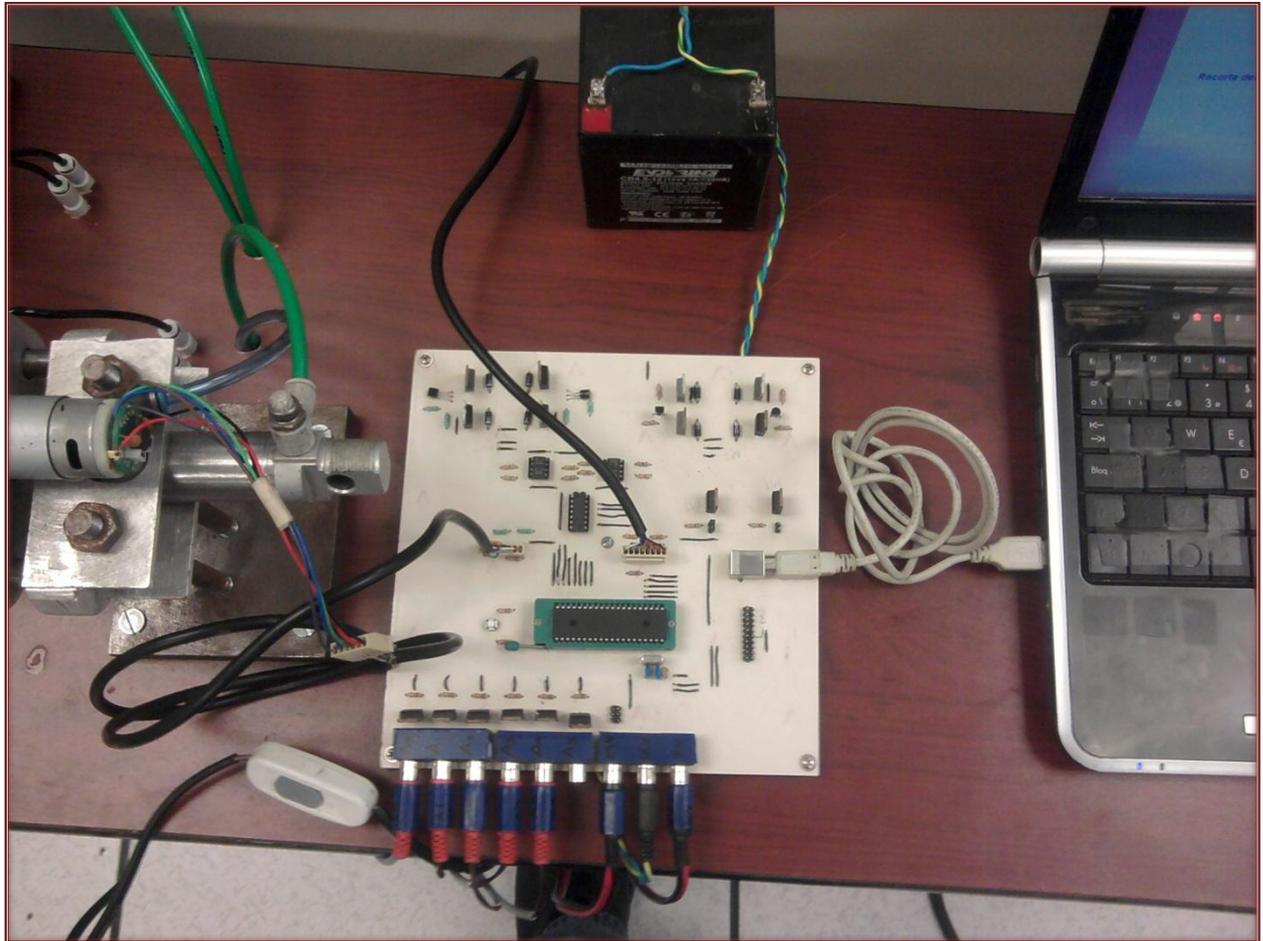
**Figura 4.1.3** Conexiones neumáticas en los manipuladores

Por último se señala el cople de Racor QS M4 a Racor QS M4 de donde se conectan las mangueras No. 4, que no aparecen en el diseño, pero que surgen por la longitud comercial de este tipo de mangueras.



La PCB se perforó según especificaciones físicas de los elementos electrónicos que lo componen y se soldó con caudín y soldadura electrónica.

El resultado del proceso se muestra a continuación en conjunto con las conexiones eléctricas con los elementos eléctricos y electrónicos externos a ella.



**Figura 4.2.2** Tarjeta de control implementada y conectada al prototipo

Se puede señalar que el único cambio que sufrió fueron unas perforaciones para colocar unos apoyos con tornillos, con el fin de aislarla de la superficie de la mesa.

Se realizaron algunas mediciones sobre la tarjeta de control, realizando labores de giro de un motor para verificar la óptima generación de señales. Se midió el voltaje sin conectar el motor cuando se le suministraba un valor definido de PWM, así como también se midió el valor con el motor conectado. Los resultados se presentan en la siguiente tabla.

PWM	V SIN CARGA	V CON CARGA	PWM	V SIN CARGA	V CON CARGA
255	12.51	10.70	120	5.93	5.28
250	12.30	10.35	115	5.69	5.03
245	12.03	10.21	110	5.44	4.80
240	11.78	10.10	105	5.20	4.62
235	11.54	9.93	100	4.96	4.35
230	11.29	9.70	95	4.70	4.15
225	11.05	9.50	90	4.46	3.85
220	10.80	9.30	85	4.22	3.59
215	10.56	9.10	80	3.97	3.33
210	10.31	8.90	75	3.73	3.10
205	10.07	8.85	70	3.48	2.90
200	9.83	8.60	65	3.24	2.70
195	9.58	8.50	60	3.00	2.47
190	9.34	8.30	55	2.75	2.05
185	9.09	8.11	50	2.51	1.70
180	8.85	7.95	45	2.27	1.30
175	8.61	7.75	40	2.02	0.90
170	8.36	7.55	35	1.78	0.80
165	8.12	7.37	30	1.53	0.60
160	7.88	7.10	25	1.29	0.50
155	7.63	6.95	20	1.04	0.43
150	7.39	6.71	15	0.80	0.35
145	7.15	6.45	10	0.55	0.25
140	6.91	6.15	5	0.31	0.16
135	6.66	5.88	0	0.00	0.00
130	6.42	5.70			
125	6.17	5.45			

**Tabla 4.1** Mediciones de voltaje cuando se suministra una señal de PWM

De donde se puede deducir que los circuitos electrónicos ocupados para esta tarea funcionan correctamente, dado que las señales que se presentan coinciden con los máximos y mínimos esperados.

### 4.3 Descripción de la programación en el microcontrolador

En los siguientes apartados se describe el desarrollo de los segmentos funcionales en el microcontrolador, estos son: los descriptores de comunicación USB, el control de posición angular y la activación de los elementos neumáticos. El código final se puede observar completo en la sección de anexos.

El código presentado en los siguientes bloques funcionales se desarrolló en PCWH Compiler®, versión 4.023, en modo de evaluación que en ningún momento se presenta como una infracción a los derechos de autor.

#### 4.3.1 Descripción de los descriptores de hardware generales del microcontrolador

Las siguientes líneas de código son parte del encabezado del programa y sirven para configurar el hardware del microcontrolador.

```
#INCLUDE <18F4550.H>
#FUSES HSPLL,NOWDT,NOPROTECT,NOLVP,NODEBUG,USBDIV,PLL3,CPUDIV1,VREGEN,MCLR,NOPBADEN
#USE DELAY (CLOCK=48M)
#BYTE PORTA = 0x0F80 // Registro del puerto A
#BYTE PORTB = 0x0F81 // Registro del puerto B
#BYTE PORTC = 0x0F82 // Registro del puerto C
#BYTE PORTD = 0x0F83 // Registro del puerto D
#BYTE PORTE = 0x0F84 // Registro del puerto E
```

La primera línea cargan las librerías propias del modelo del microcontrolador, que necesita el compilador para cargar los valores de registros y descriptores en general; en la segunda se configuran los fusibles con los que se configura el microcontrolador, tal como el modo del oscilador en configuración de cristal de alta velocidad con PLL habilitado marcado solo con HSPLL, el deshabilitar el *Watchdog Timer* marcado como NOWDT, utilizar el divisor PLL con un valor de 3 marcado como PLL3, entre otros. La tercera línea se usa para definir la frecuencia del reloj en las funciones de retardo de tiempo. Las subsecuentes líneas se escriben para definir variables con el valor del registro del puerto, esto con el fin de escribir en ellos sin funciones intermediarias, en caso de ser requerido.

#### 4.3.2 Descripción del desarrollo de la comunicación USB

Las siguientes líneas también forman parte del encabezado del programa. En las primeras se encuentran algunas definiciones necesarias para la configuración de la comunicación USB en hardware, las subsecuentes cargan las librerías que contienen las funciones precargadas para la comunicación, cabe denotar que se eligió este compilador por poseer estas funciones. En cada línea se encuentra comentado de lado derecho, la función de la misma de manera concreta.

```
#DEFINE USB_HID_DEVICE FALSE // Deshabilitamos el uso de las directivas HID
#DEFINE USB_EP1_TX_ENABLE USB_ENABLE_BULK // Habilitar EndPoint1 para entrada por bloque/interrupción por transferencia
#DEFINE USB_EP1_RX_ENABLE USB_ENABLE_BULK // Habilitar EP-1-(EndPoint1) para OUT bulk/interrupt transfers
#DEFINE USB_EP1_TX_SIZE 64 // Tamaño de bytes para alojar el mensaje del buffer TX del EP-1-
#DEFINE USB_EP1_RX_SIZE 64 // Tamaño de bytes para alojar el mensaje del buffer RX del EP-1-
#include <PIC18_USB.H> // Microchip PIC18Fxx5x Hardware layer for CCS's PIC USB
#include <USB_DESC_SCOPE.H> // Descriptores del Pic USB
#include <USB.C> // Handles us setup tokens and get descriptor reports
```

En la función principal (*main*) debe de contener las siguientes funciones para poder conectarse al host, a excepción de la primera que reserva espacio de memoria para el almacenamiento de la información recibida.

```
INT8 USB_IN [10]; // Variable p/almacenar el mensaje del USB
USB_INIT(); // Inicializamos el USB
USB_TASK(); // Habilita periférico usb e interrupciones
USB_WAIT_FOR_ENUMERATION(); // Esperamos hasta que el PicUSB sea configurado por el host
```

Una vez que se tiene conectado al host estas funciones nos ayudan a recuperar los datos recibidos; la primera línea revisa que este enumerado el microcontrolador, la segunda línea verifica si hay datos en buffer, de encontrarlos, la tercer línea los mueve hacia una variable ya determinada.

```
IF(USB_ENUMERATED()){ // Si el Pic está configurado via USB
    IF (USB_KBHIT(1)){ // Si el endpoint de salida contiene datos del host
        USB_GET_PACKET(1, USB_IN, 10); // Tomar el paquete de 64 bytes del EP-1- y guardar en la variable de entrada
```

### 4.3.3 Descripción de la lectura de los codificadores

Las siguientes líneas se usan para comparar los puertos en donde se recibe la señal de los codificadores, desde este momento se notifica al lector que se está tratando del código para un motor, considerando que el código para el otro motor es similar.

En la primera parte se definen las variables a ocupar, en este caso una contiene la cuenta del número propuesto para la configuración lógica de las entradas y un clon de ella es utilizado para revisar si existen cambios en las entradas. Así mismo se definen las variables donde se almacenará la posición deseada y la real. El error de posición es la variable *EP\_2*.

```
INT8 COUNT02=0, SEQ_INDEX02=0;
UNSIGNED INT16 POS_R2=2048, POS_D2=2048, EP_2=0;
INT1 ERROR=0;
```

Esta función asigna un número a la configuración de estados lógicos de entrada provenientes del codificador.

```
VOID SYNC_01_M2(){
    IF(INPUT_STATE(PIN_B2)){
        IF(INPUT_STATE(PIN_B3)) COUNT02=2;
        ELSE COUNT02=1;
    }
}
```

```

ELSE{
  IF(INPUT_STATE(PIN_B3)) COUNT02=3;
  ELSE          COUNT02=0;
}
}

```

La siguiente función compara sí el valor ha disminuido o incrementado para hacer lo propio con la variable de posición real.

```

VOID SYNC_O2_M2(){
  SWITCH(COUNT02){
    CASE 0:
      IF(SEQ_INDEX02==3)    POS_R2--;
      ELSE IF(SEQ_INDEX02==1)  POS_R2++;
      ELSE          ERROR=1;
      BREAK;
    CASE 1:
      IF(SEQ_INDEX02==0)    POS_R2--;
      ELSE IF(SEQ_INDEX02==2)  POS_R2++;
      ELSE          ERROR=1;
      BREAK;
    CASE 2:
      IF(SEQ_INDEX02==1)    POS_R2--;
      ELSE IF(SEQ_INDEX02==3)  POS_R2++;
      ELSE          ERROR=1;
      BREAK;
    CASE 3:
      IF(SEQ_INDEX02==2)    POS_R2--;
      ELSE IF(SEQ_INDEX02==0)  POS_R2++;
      ELSE          ERROR=1;
  }
  SEQ_INDEX02=COUNT02;
}

```

Por lo que de haberse perdido de un cambio de estado mandará una señal de error, pero es sumamente complicado que esto ocurra por la rapidez de ejecución del bucle principal en contra de la velocidad de cambio de las salidas del codificador.

#### 4.3.4 Descripción de la función de control

Como se observó en el diseño del control, la señal de salida PWM cambia conforme cambia la magnitud del error. También se ha establecido los criterios de cambio de giro. Se establecen en las siguientes líneas de código las funciones para configurar el PWM, recordando que la señal que controla el giro se encuentra en el PIN\_B7.

```

OUTPUT_HIGH(PIN_B7);           // Establecer condición inicial de giro
SETUP_TIMER_2(T2_DIV_BY_16, 255, 1); // Configurar el Timer2 para la frecuencia de los PWM
SET_PWM2_DUTY(0);             // Establecer tiempo en alto inicial del PWM2
SETUP_CCP2(CCP_PWM);         // Configurar el CCP2 para trabajar como PWM i.e. CCP2->PWM2

```

A continuación la función Control que ha sido ya descrita, de donde se puede notar que para evitar el signo negativo primero se comparan las variables.

```

VOID CONTROL_M2(){
  IF(POS_D2>POS_R2){
    OUTPUT_LOW (PIN_B7);
    EP_2=POS_D2-POS_R2;
  }
  IF(POS_R2>POS_D2){
    OUTPUT_HIGH (PIN_B7);
    EP_2=POS_R2-POS_D2;
  }
  IF( EP_2>=80 ) SET_PWM2_DUTY(105);
  IF((EP_2>=60)&&(EP_2<80)) SET_PWM2_DUTY(100);
  IF((EP_2>=40)&&(EP_2<60)) SET_PWM2_DUTY( 95);
  IF((EP_2>=20)&&(EP_2<30)) SET_PWM2_DUTY( 90);
  IF((EP_2>=10)&&(EP_2<20)) SET_PWM2_DUTY( 85);
  IF((EP_2>= 3)&&(EP_2<10)) SET_PWM2_DUTY( 80);
  IF((EP_2>= 1)&&(EP_2< 2)) SET_PWM2_DUTY( 50);
}

```

### 4.3.5 Descripción del bucle general

Ya se han comentado algunas de las funciones del bucle general, pero se agregan nuevamente para que se visualice el proceso descrito en el capítulo 3, en la estructura del control. El bucle se hace infinito declarando el *while* como verdadero.

```

WHILE (TRUE){
  IF(USB_ENUMERATED()){           // Si el Pic está configurado via USB
    IF (USB_KBHIT(1)){           // Si el endpoint de salida contiene datos del host
      USB_GET_PACKET(1, USB_IN, 10); // Tomar el paquete de 64 bytes del EP-1 - y guardar en la variable de entrada
    }
  }
}

```

En esta parte se verifica que el tercer elemento que se recibió en la cadena contenga un 1 o un 2 (49, 50 en ASCII respectivamente). Para después verificar que instrucción se le ha enviado en el sexto carácter de la cadena.

Las instrucciones que mandan una salida al puerto A son relativas a los movimientos lineales o de sujeción.

Las instrucciones H o A en el sexto carácter pertenecen a movimientos angulares; es en ese momento que se realiza una adición o una resta a la posición deseada.

```

IF (USB_IN[2]==50){
    IF(USB_IN[5]==72) { POS_D2 = POS_D2 + (((USB_IN[8]-48)*10)+(USB_IN[9]-48)); }
    IF(USB_IN[5]==65) { POS_D2 = POS_D2 - (((USB_IN[8]-48)*10)+(USB_IN[9]-48)); }
    IF(USB_IN[5]==69) { OUTPUT_HIGH (PIN_A4);}
    IF(USB_IN[5]==82) { OUTPUT_LOW (PIN_A4);}
    IF(USB_IN[5]==101){ OUTPUT_HIGH (PIN_A2);}
    IF(USB_IN[5]==114){ OUTPUT_LOW (PIN_A2);}
    IF(USB_IN[5]==79) { OUTPUT_HIGH (PIN_A1);}
    IF(USB_IN[5]==67) { OUTPUT_LOW (PIN_A1);}
    }
}
}

```

Se lee el codificador y se enumera, de existir un cambio se busca en qué sentido fue y se actualiza la posición, en caso de que no coincida por la posición deseada se realiza la función de control. Por último se monitorea el error desde el puerto D.

```

SYNC_01_M2();
IF(COUNT02!=SEQ_INDEX02){ SYNC_02_M2(); }
IF(POS_D2!=POS_R2){ CONTROL_M2(); }
OUTPUT_D(EP_2);
}

```

Quedando descrito así la programación del microcontrolador. Cabe destacar que permite un error de hasta 2 grados; este error hasta el momento no ha sido sobrepasado y se han obtenido buenos resultados con esta programación.

#### 4.4 Comportamiento del control de posición angular

En este apartado se presentan los resultados obtenidos del control de posición angular digital, implementado en el microcontrolador PIC18F4550 que en conjunto con la electrónica presente en la tarjeta de control realizan el cambio de posición angular.

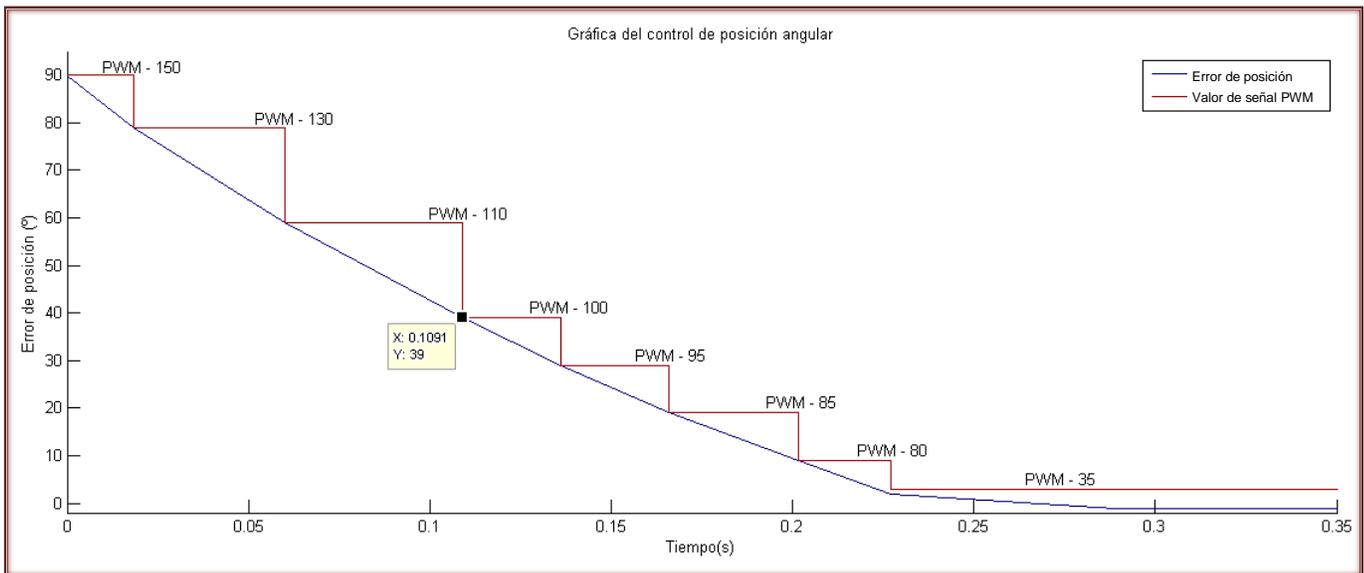
Primero se presentan algunas mediciones que se realizaron con un osciloscopio digital, donde se midió la frecuencia de un canal de un codificador. El dato obtenido fue la base para el cálculo de la velocidad angular, considerando que un periodo de esta señal equivale al tiempo del desplazamiento angular de cuatro grados. Por lo tanto se muestran una tabla de correspondencia.

PWM	FRECUENCIA	REV/SEG	PWM	FRECUENCIA	REV/SEG
255	230.700	2.563	125	108.240	1.203
250	225.600	2.507	120	106.573	1.184
245	218.225	2.425	115	101.500	1.128
240	215.590	2.395	110	96.710	1.075
235	213.110	2.368	105	92.400	1.027
230	208.010	2.311	100	83.900	0.932
225	205.030	2.278	95	75.090	0.834
220	199.845	2.221	90	71.970	0.800
215	195.400	2.171	85	63.200	0.702
210	190.400	2.116	80	58.420	0.649
205	186.950	2.077	75	54.300	0.603
200	180.120	2.001	70	50.360	0.560
195	175.920	1.955	65	42.910	0.477
190	170.603	1.896	60	38.370	0.426
185	166.770	1.853	55	33.320	0.370
180	163.040	1.812	50	25.761	0.286
175	158.910	1.766	45	17.450	0.194
170	152.550	1.695	40	11.740	0.130
165	148.176	1.646	35	4.036	0.045
160	144.020	1.600	30	NULL	NULL
155	140.460	1.561	25	NULL	NULL
150	136.600	1.518	20	NULL	NULL
145	130.300	1.448	15	NULL	NULL
140	126.340	1.404	10	NULL	NULL
135	119.460	1.327	5	NULL	NULL
130	113.770	1.264	0	NULL	NULL

Tabla 4.2 Mediciones de la velocidad angular con señales de PWM

La velocidad obtenida con la señal de PWM establecida en un valor de 35, se obtuvo midiendo el tiempo con un cronometro, dado que es el punto de flexión en el cual vence la inercia propia del motor. Al realizar esta prueba repetidamente se encontró que partiendo del reposo esta señal no vence la inercia, así como en otras repeticiones llegó a detenerse al estar girando a una velocidad mayor. Se encontró este punto como referencia para dejar energizado el motor cuando se encuentra en el margen permitido de error de posición ( $2^\circ$ ).

Así mismo con los valores obtenidos, se graficó una aproximación del comportamiento del control de posición, que concuerda con el comportamiento real del control con diferencias máximas en el tiempo de 0.5 segundos. La gráfica se muestra en la siguiente figura.



**Figura 4.4.1** Gráfica del comportamiento del control de posición

En donde se puede apreciar que la señal de control se encuentra ligada a la razón de cambio del error, de tal manera que decrece en líneas rectas con pendientes, éstas últimas decrecen en rangos establecidos previamente dependientes del error, ligadas a la señal de PWM.

Para evitar oscilaciones en el estado estacionario, el control permite un rango de error de  $2^\circ$  encontrando que de esta forma el motor se posiciona con mayor rapidez en el valor deseado. Este error es permisible en la ejecución de las tareas y no es acumulable cuando se ejecutan cambios de posición en el motor.

Por último, se ha encontrado que esta forma de control es la más fácil de implementar en comparación con la ecuación general del control PD.

## Capítulo 5. Conclusiones

### 5.1 Conclusiones del desarrollo

La comunicación establecida es el protocolo USB que será configurado en hardware en el microcontrolador seleccionado, caso concreto, el MCU PIC8F4550 que ofrece grandes recursos a ser explotados.

El diseño del control de mando se aterrizó sobre la manufactura de un PCB doble capa que contenga a todos los elementos mencionados en el diseño electrónico, se trató de ser ordenados y claros.

Los sistemas neumáticos se ofertan en una amplia gama de modelos en cuanto a un tipo específico, y la competencia entre las marcas que están en el mercado generará calidad en los productos. Encontré que los accesorios y/o refacciones originales neumáticos son de un costo elevado para el ingreso promedio de un alumno; por lo que, tener un sistema como el planteado permite practicar sin gastar en ello.

La realización de este diseño no contempla el máximo rango de la aplicación que tiene todo el prototipo de robot didáctico ortogonal, tal cual es el caso de lograr generar un applet para alojarse en un servidor y desde un portal en internet la manipulación del mismo, pero con la ideología del código abierto que poseemos esperamos algún día verlo realizado.

### 5.2 Conclusiones de la implementación

En lo que corresponde a los elementos neumáticos me he llevado una gran experiencia al trabajar con ellos, sobre todo en el aspecto de la calibración; es importante considerar que cuando se realizan ajustes en equipos neumáticos, utilizándolo en alguna locación, podrá presentarse que no apliquen para otra, aunque se cuente con una presión neumática similar, dado que los procesos neumáticos no son comúnmente precisos.

De la tarjeta de control puedo concluir que el diseñar, plasmar ideas e incluso simular, sobre todo en grandes circuitos, aun se hayan contemplado casi todos los aspectos aún quedarán pequeños detalles que se oculten de nuestra percepción. Al implementar la tarjeta de dos capas me pude percatar que soldar por ambos lados, circuitos muy juntos entre sí, se convierte en una destreza no muy común y que requiere cierta práctica con antelación; por lo que se implementó en un PCB de una capa. Así como cuando por un descuido, no se alambró una pista y tuve que realizar otra placa para un total de cuatro tarjetas de control, pero que finalmente me dejan mucho conocimiento del tema.

En cuanto a lo técnico cabe destacar la precisión que se obtuvo en la salida del circuito conmutador al obtener casi el 100% de la alimentación de entrada, lo cual me da la certeza de que se realizó un buen cálculo.

Con respecto al control, concluyo que los tipos de control industrial que hay son ocupados porque generan la forma más segura, estable y económica en algunos casos de manipular los procesos. Encontrando que la tarea primordial en la implementación de control es establecer una relación de parámetros para su sintonización.

### 5.3 Posibles mejoras al sistema

Considero que la principal mejora del sistema se encuentra cuando el alumno propone qué otra actividad hacer con él, dado que es prototipo listo para ser explotado en las materias que ya he citado en un principio. De verdad espero que este proyecto tenga continuidad y se pueda utilizar tanto en clases como en proyectos alternativos.

Considero que una mejora que pudiese tener el prototipo sería cambiar los actuadores neumáticos lineales por unos con mayor tecnología que puedan desarrollar multiposición controlada. En mi paso por la institución no tuve oportunidad de practicar con ellos, porque no estuvieron a mi alcance debido a que la Institución no cuenta con ellos y yo no tuve la oportunidad de adquirirlos por situaciones económicas. Considero que aumentaría el potencial del dispositivo si contase con este tipo de actuador.

Una aplicación que podría generarse en un futuro podría ser la resolución del problema planteado con el cubo de colores mediante la programación directa en un FPGA.

Otra mejoría podría surgir cambiando la comunicación de USB a Ethernet para practicar con este tipo de comunicación, dado que no es usual en alumnos de ingeniería mecatrónica y que, en los últimos años a despegado en aplicaciones industriales.

### 5.3 Comparativa de costos

Al inicio se plantearon los posibles costos que se generarían para la realización del dispositivo, visualizados en la siguiente tabla:

Material	Cantidad	Precio	Actuadores y accesorios	Cantidad	Precio
Aluminio para fundición	15 kg	\$300.00	Electroválvula 5/2 N	5	\$450.00
Acero AISI 1018	6 kg	\$120.00	Cilindro neumático	3	\$2,400.00
Placas fenólicas	2	\$80.00	Motor eléctrico C.D.	2	\$80.00
Material electrónico	55 piezas	\$100.00	Codificador incremental	2	\$400.00
Circuitos integrados	3	\$180.00	Pinzas neumáticas	2	\$500.00
Cables y conectores	8 pares	\$1,100.00	<b>Gasto de actuadores</b>		\$3,830.00
Cloruro férrico	0.25 lt	\$20.00			
Soldadura y pasta	100 gr.	\$50.00			
Cubo Rubik	2	\$260.00			
Tornillos y tuercas	64 pares	\$250.00			
<b>Gasto de Material</b>		\$2,460.00	<b>Gasto total</b>		\$6,290.00

Tabla 5.1 Gastos del proyecto cotizados en Febrero del 2011 con actuadores de vida útil media

Pero en la realización de este proyecto nos encontramos con la problemática de la inexistencia de actuadores de vida útil media, entre estos, las pinzas neumáticas y motores que acoplaran de manera funcional con los codificadores. Por lo que el gasto se incrementó en \$4,450.00 reduciendo costos al encontrar cilindros neumáticos de vida útil media, pero incrementándose al obtener pinzas neumáticas nuevas y motorreductores con codificador:

Material	Cantidad	Precio	Actuadores y accesorios	Cantidad	Precio
Aluminio para fundición	15 kg	\$330.00	Electroválvula 4/2 N	5	\$450.00
Aluminio 6061 en placa	5 kg	\$350.00	Racor QS 1/8 y 1/4	18	\$540.00
Acero AISI 1018 en placa	8 kg	\$160.00	Cilindro neumático	3	\$350.00
Cilindro acero AISI 1018	4.5 kg	\$90.00	Maguera M6, M4	5 m	\$500.00
Placas fenólicas	2	\$80.00	Regulador de caudal	6	\$600.00
Material electrónico	50 piezas	\$95.00	Motorreductor con codificador	2	\$1,600.00
Circuitos integrados	3	\$180.00	Pinzas neumáticas	2	\$3,700.00
Cables y conectores	8 pares	\$1,100.00	<b>Total de actuadores</b>		\$7,740.00
Cloruro férrico	0.25 lt	\$20.00			
Soldadura y pasta	100 gr.	\$50.00			
Cubo rubik	2	\$260.00			
Tornillos y tuercas	70 pares	\$285.00			
<b>Total de Material</b>		\$3,000.00	<b>Gasto total</b>		\$10,740.00

Tabla 5.2 Gastos del proyecto generados en el desarrollo del proyecto

## Referencias

- Arriola, M. A. (2009). *Desarrollo de competencias en el proceso de instrucción*. Ciudad de México: Trillas.
- Bishop, H. R. (2008). *The Mechatronics Handbook – Mechatronic systems, sensors, and actuators: fundamentals and modeling*. Boca Raton, Florida: CRC Press.
- Bolton, W. (2003). *Mecatrónica: Sistemas de Control Electrónico en Ingeniería Mecánica y Eléctrica*. Ciudad de México, México: Alfaomega.
- De Lorenzo of America. (s.f.). *Equipos Didácticos: Mecatrónica*. Recuperado el primero de febrero de 2010, de De Lorenzo of America Corporation Web site: <http://www.delorenzo.com.mx/prods/mecatronica/index.htm>
- Herraiz, M. L. (2001). *Formación de formadores: manual didáctico*. Ciudad de México, México: Limusa.
- Kuo, B. C. (1996). *Sistemas de control automático*. Ciudad de México, México: Prentice Hall.
- Martínez, D. O., Martínez, J. C., & Silva, V. M. (2008). *Robot didáctico con tecnología abierta*. Ciudad de México, México: UPIITA – IPN.
- Martínez, N., Murillo, O., & Pineda, J. C. (2005). *Brazo manipulador para aplicaciones didácticas*. Ciudad de México, México: UPIITA – IPN.
- Ogalde, I., & Bardavid, E. (2008). *Los materiales didácticos: medios y recursos de apoyo a la docencia*. Ciudad de México, México: Trillas.
- Ogata, K. (1993). *Ingeniería de Control Moderna*. Naucalpan de Juárez, México : Prentice–Hall.
- Ortega, R. E., & Ríos, A. (2005). *Sistema didáctico para la enseñanza–aprendizaje de la visión artificial*. Ciudad de México, México: UPIITA – IPN.
- Pallás, R. A., & Valdes, F. (2007). *Microcontroladores: Fundamentos y Aplicaciones con PIC*. Barcelona, España: Marcombo.
- Paredes, J. (2000). *Materiales didácticos en la práctica educativa: un análisis etnográfico*. Madrid: Universidad Autónoma de Madrid.
- Rashid, M. (2004). *Electrónica de Potencia: Circuitos, Dispositivos y Aplicaciones*. Ciudad de México, México: Pearson Educación.
- Schmid, S. R. (2002). *Manufactura, ingeniería y tecnología*. Ciudad de México, México: Pearson Educación.
- Spong, M., Hutchinson, S., & Vidyasagar, M. (2006). *Robot Modeling and Control*. New York: John Wiley and Sons.
- Tébar, L. (2003). *El perfil del profesor mediador*. Madrid: Editorial Santillana.
- Vargas, J. C. (2008). *Robot portátil de tres grados de libertad en configuración angular para fines didácticos*. Ciudad de México, México: UPIITA – IPN.
- Zabala, A. (1995). *La práctica educativa. Cómo enseñar*. Barcelona: Editorial Graó.

**Instituto Politécnico Nacional**  
**Unidad Profesional en Ingeniería y Tecnologías Avanzadas**  
**Ingeniería Mecatrónica**

**Práctica 1. Comunicación USB en el microcontrolador PIC18F4550**

En esta práctica el alumno reforzará los conocimientos vistos en el aula concerniente al tópico de Comunicación USB tanto en la materia de Programación Básica como en la materia de Dispositivos Programables.

El propósito de estos ejercicios se centra en la implementación del protocolo de comunicación USB, por lo que se desarrollará con base a este objetivo.

La estructura se presenta a continuación.

- Definiciones del protocolo de comunicación USB
- Instalación del controlador (*driver*) del microcontrolador en la PC
- Creación de un programa que realice el envío de datos por el puerto USB
- Realización de ejercicios para la verificación de la comunicación

**Introducción**

No hace muchos años apareció en el mercado el estándar USB para comunicaciones serie y ya hoy todos los ordenadores lo incorporan, llegando a sustituir incluso a los famosos conectores serie.

La estructura de este bus ha sido definida por un consorcio de líderes de la industria para permitir la conexión de múltiples periféricos de ordenador a baja y media velocidad, entre los que incluyen módems, impresoras, ratones, teclados, escáneres, *joysticks*, etc., todos ellos con la mismo puerto del PC. La especificación, basada en una arquitectura abierta, está llegando a ser una característica estándar en todos los PC's de nuevo diseño.

El USB (Universal Serial Bus o Bus Serie Universal) es un bus externo estándar que soporta la transferencia de datos a velocidades de 12 Mbit/s en el USB 1.1 (1,5 Mbit/s para dispositivos que requieren poca velocidad, como *joysticks* y ratones) hasta 480 Mbits/s (USB 2.0). A un único puerto USB, que soporta *Plug and Play* (PnP) y la

conexión/desconexión “en caliente”, es decir sin apagar el ordenador, se pueden conectar hasta 127 periféricos, con una configuración llamada “tiered-star”, lo que significa que algunos dispositivos USB actúan como un hub y proporcionan las puertas para la conexión de otros; solamente uno de ellos se ha de conectar al PC.

Con este bus los usuarios no se tienen que preocupar acerca de qué puerto elegir, no tienen que instalar ninguna tarjeta de expansión ni configurar puentes físicos, parámetros de IRQ, canales DMA y direcciones de E/S, ya que todo se realiza automáticamente. Otra ventaja añadida es que este bus proporciona la alimentación a los periféricos hasta una distancia de 5 metros, lo que hace el diseño de éstos más sencillo.

A principios de 1996 unos pocos fabricantes de ordenadores empezaron a incluirlo en sus productos y, desde el lanzamiento del chipset 440 LX, de Intel en 1997, el USB se ha extendido considerablemente, llegando a reemplazar a los puertos serie y paralelo. Hoy, ya son más de 400 fabricantes que lo han adoptado. Las especificaciones completas de este bus, así como una lista de FAQ sobre el mismo, se pueden encontrar en la siguiente dirección de Internet (USB Forum): <http://www.usb.org>.

*“El estándar USB para comunicaciones serie”*

Huidobro, J. M. (2005). *Sistemas Telemáticos (Electricidad-Electrónica: Sistemas de Telecomunicación e Informáticos)*. Madrid, España : Thompson Paraninfo.

Es deber del alumno presentar en el marco teórico del reporte de la práctica, un resumen complementario al presentado en la introducción, que contenga el funcionamiento del USB y sus principales componentes.

### **Instalación del controlador del microcontrolador en el SO Windows 7**

A continuación se describe el proceso de instalación del driver del PIC18F4550 en el sistema operativo Microsoft Windows 7 ® mediante el asistente de instalación. La comunicación también es efectiva en Windows XP y Windows Vista; el proceso es similar en las plataformas anteriores, comparado con el descrito para Win7.

Antes de conectar el cable USB de la tarjeta a la PC realizar el siguiente proceso.

Proceso de instalación:

- 1.- Abrir el programa “Ejecutar” de Windows, un acceso rápido es mediante las teclas WINDOWS + R.
- 2.- Ejecutar el Asistente de instalación de *Hardware*. Teclear “hdwwiz” y dar clic en el botón ACEPTAR.
- 3.- Establecer permisos de administrador En la barra de tareas de Windows, dar clic con el botón derecho sobre la pestaña del asistente. Dar clic derecho en “Ejecutar como administrador”.
- 4.- Dar clic en el botón SIGUIENTE, de la nueva ventana.
- 5.- Seleccionar la opción correspondiente a “Instalar el *hardware* seleccionado manualmente de una lista (avanzado)” y dar clic en el botón SIGUIENTE.
- 6.- Optar por la opción predeterminada “Mostrar todos los dispositivos” y dar clic a SIGUIENTE.
- 7.- Dar clic al botón USAR DISCO.
- 8.- Abrir la carpeta [.../Controladores/Release/...] donde se ubica el controlador proporcionado por el docente.
- 9.- Seleccionar el archivo mchpusb.inf.
- 10.- Dar clic en el botón ABRIR y dar clic en ACEPTAR de la ventana anterior.
- 11.- Dar clic en SIGUIENTE para posteriormente dar clic en FINALIZAR.

Posteriormente, con la tarjeta de control sin alimentación conectar el cable USB con la tarjeta y la PC. Seguido de esto, energizar la tarjeta de control, en ese momento deberá aparecer un mensaje en el escritorio del SO avisando que se ha conectado correctamente.

De presentarse el caso que no haya sido exitosa la comunicación, pedir al encargado del laboratorio proporcione el archivo RDO.hex (archivo original del programa del microcontrolador para el robot didáctico) con el que se deberá reprogramar el PIC, teniendo la precaución de quitar la alimentación eléctrica al reprogramar el microcontrolador. Esto solucionará el problema.

## Programa para el envío de datos por comunicación USB

Para la realización de esta tarea, se empleará el *software* MATLAB 7.9 ®. A continuación se describe el código que deberá ser generado para establecer la comunicación.

Antes de escribir el código se debe de guardar el archivo \*.m a utilizar en la carpeta [.../Controladores/...] proporcionada por el docente, ya que contiene las librerías del controlador y sus descriptores (mpusbapi.dll, mpusbapi.c).

En seguida, cargar el programa descrito a continuación, el cual se puede obtener del docente sí así lo decide.

Este programa envía una cadena de datos al microcontrolador, para este caso en específico, la codificación del mensaje se realiza por la posición y carácter en la cadena. Al enviar el carácter (1) en la tercera posición y una (E) en la sexta, el robot expandirá horizontalmente el brazo 1. De manera similar al enviar (1) y (R), el robot retrae horizontalmente el brazo 1. Si la comunicación es efectiva realizará ambos movimientos.

```
clear all; % Limpiar workspace
close all; % Cargar la librería
loadlibrary mpusbapi _mpusbapi.h alias libreria

%libisloaded libreria % Confirma que la librería ha sido
% cargada
libfunctions('libreria', '-full') % Muestra en la línea de comandos las
% funciones de la librería
libfunctionsview libreria % Muestra en un cuadro lo mismo que la
% instrucción anterior

data_in = eye(1,64,'uint8'); % Se declara el vector de datos de
% entrada (el que se recibe del PIC)
data_out = eye(1,64,'uint8'); % Se declara el vector de datos de
% salida (el que se envía al PIC)
% TODOS LOS DATOS SE DECLARAN COMO
% UINT8 de lo contrario no hay
% comunicación.

vid_pid_norm = libpointer('int8Ptr',[uint8('vid_04d8&pid_000b') 0]);
out_pipe = libpointer('int8Ptr',[uint8('\MCHP_EP1') 0]);
in_pipe = libpointer('int8Ptr',[uint8('\MCHP_EP1') 0]);
% Se establecen los valores de
% identificación del fabricante
% (Microchip -VID- 04d8) y del producto
% (PIC18FXX5X -PID- 000b) así como las
% "pipes" de entrada y salida mediante
% el EndPoint -1-
```

```
[conectado] = calllib('libreria', 'MPUSBGetDeviceCount', vid_pid_norm)

if conectado == 1
    % Es importante seguir ésta secuencia
    % para comunicarse con el PIC:
    % 1. Abrir túneles
    % 2. Enviar/Recibir dato
    % 3. Cerrar túneles

[my_out_pipe] = calllib('libreria', 'MPUSBOpen', uint8(0), vid_pid_norm,
out_pipe, uint8(0), uint8(0)); % Se abre el túnel de envío
[my_in_pipe] = calllib('libreria', 'MPUSBOpen', uint8(0), vid_pid_norm,
in_pipe, uint8(1), uint8(0)); % Se abre el túnel de recepción

data_out(2) = 49; % Se le asigna al tercer byte
                % de la cadena el valor 49
                % equivalente al 'l' ASCII
data_out(5) = 69; % En la sexta posición una 'E'

calllib('libreria', 'MPUSBWrite', my_out_pipe, data_out, uint8(64), uint8(64),
uint8(10)); % Se envía el dato al PIC

[aa,bb,data_in,dd] = calllib('libreria', 'MPUSBRead', my_in_pipe, data_in,
uint8(64), uint8(64), uint8(10)); % Se recibe el dato que envía el PIC

pause(3);

data_out(2) = 49; % Se le asigna al tercer byte
                % de la cadena el valor 49
                % equivalente al 'l' ASCII
data_out(5) = 82; % En la sexta posición una 'R'

data_in(1) % Se muestra el primer valor
           % del vector recibido

calllib('libreria', 'MPUSBWrite', my_out_pipe, data_out, uint8(64), uint8(64),
uint8(10)); % Se envía el dato al PIC

calllib('libreria', 'MPUSBClose', my_in_pipe); % Se cierra el túnel de
                                                % recepción
calllib('libreria', 'MPUSBClose', my_out_pipe); % Se cierra el túnel de
                                                % envío
end

unloadlibrary libreria % Importante descargar la librería de
                       % memoria, de lo contrario genera
                       % errores

close all
clear all % Limpiar el workspace
```

## Ejercicios sobre la comunicación USB

1. Generar el programa que envíe las instrucciones codificadas de los movimientos del brazo 2, en intervalos de 2 segundos. Según la tabla siguiente, en ese orden.

Movimiento	Carácter en la 3er posición	Carácter en la 6ta posición
Expansión Horizontal	'2' = 50	'E' = 69
Retracción Horizontal	'2' = 50	'R' = 82
Expansión Vertical	'2' = 50	'e' = ¿?
Retracción Vertical	'2' = 50	'r' = ¿?
Abrir pinzas	'2' = 50	'O' = ¿?
Cerrar pinzas	'2' = 50	'C' = ¿?

2. Generar un programa que al ser ejecutado y reiniciada la tarjeta de control, el brazo 2 se expanda horizontalmente, sujete un objeto, se expanda verticalmente, se contraiga horizontalmente, se expanda horizontalmente, se contraiga verticalmente y suelte la pieza. Realizando así una secuencia.
3. Con base en el primer programa, generar otro programa que contenga un bucle, en el que al presionar la tecla correspondiente a la sexta posición realice la acción correspondiente.
4. Genere las secuencias del primer y segundo programa a manera de funciones, que al oprimir la tecla '1' y '2' se ejecuten automáticamente.

Escriba un reporte que contenga el resumen antes mencionado y las siguientes observaciones.

- ¿Qué es una "pipe"?
- ¿Cuál es la función de un EndPoint?
- ¿Cuál fue la mayor dificultad en la implementación?
- ¿Qué programa generó mayor dificultad, cuál fue la dificultad y cómo se resolvió?
- Escribir conclusiones individuales

Entréguese el reporte según el formato del docente.