



PONTIFICIA **UNIVERSIDAD CATÓLICA** DEL PERÚ

Esta obra ha sido publicada bajo la licencia Creative Commons  
Reconocimiento-No comercial-Compartir bajo la misma licencia 2.5 Perú.

Para ver una copia de dicha licencia, visite  
<http://creativecommons.org/licenses/by-nc-sa/2.5/pe/>



¡Error! Marcador no definido.**PONTIFICIA UNIVERSIDAD  
CATÓLICA DEL PERÚ**  
**FACULTAD DE CIENCIAS E INGENIERÍA**



PONTIFICIA  
**UNIVERSIDAD  
CATÓLICA**  
DEL PERÚ

DISEÑO E IMPLEMENTACIÓN DE UN BRAZO ROBOT DE DOS  
GRADOS DE LIBERTAD PARA EL TRAZADO DE  
DIAGRAMAS EN UN PLANO

Tesis para optar el Título de Ingeniero Electrónico, que presentan los  
bachilleres:

**Jaime Ricardo Nakamura Lam**  
**Miguel Antonio Chávez Tapia**  
**César Olivera Susaníbar**

ASESOR: Gustavo Kato Ishizawa

**Lima, abril del 2009**

## RESUMEN

El presente trabajo consiste en el diseño y la implementación de un brazo robot de dos grados de libertad para su aplicación en el trazado de diagramas en un plano de trabajo A3. El diseño implicó el desarrollo de un modelo mecánico del sistema, desarrollado parcialmente por Luis Felipe López Apostolovich, alumno de la especialidad de Ingeniería Mecánica de la Pontificia Universidad Católica del Perú y complementado con el diseño del Sr. Alberto Orihuela, egresado del instituto técnico SENATI, quien también apoyó en los procesos de manufactura que fueron realizados en su taller.

El diseño del sistema de control para el brazo robot de dos grados de libertad se dividió en tres partes: una etapa de interfaz de usuario, encargada de interactuar con el usuario mediante un computador personal y comunicarse con la etapa del controlador; la cual, se encarga de recibir del computador personal los valores de trayectoria ingresados por el usuario y generar las señales de control necesarias para accionar los actuadores encargados de realizar el movimiento, así como de recibir información de los sensores acoplados al sistema; y por último una etapa de interfaz de potencia, encargada de recibir las señales de control generadas por el microcontrolador, llevándolas a los niveles de potencia adecuados para los actuadores.

La implementación del sistema se llevó a cabo de la siguiente manera: la interfaz de usuario se realizó programando en el lenguaje Visual C++, para la etapa del controlador se utilizó un microcontrolador ATmega8 de la familia AVR de la compañía ATMEL Corporation, así como un interruptor óptico OPB940 y un potenciómetro rotatorio como sensores del sistema; y para la etapa de interfaz de potencia se utilizaron semiconductores de potencia: transistores Darlington TIP122, transistores MOSFET IRF540 e IRF9540, dimensionados para manejar dos tipos de motores: un motor paso a paso unipolar y motor de corriente continua de imán permanente.

## ÍNDICE

<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>CAPÍTULO 1</b>	
<b>PLANTEAMIENTO GENERAL DEL SISTEMA .....</b>	<b>3</b>
1.1. ETAPAS DEL BRAZO ROBOT .....	4
1.1.1. <i>Interfaz de usuario</i> .....	4
1.1.2. <i>Control de movimiento</i> .....	6
1.1.3. <i>Interfaz de potencia</i> .....	7
<b>CAPÍTULO 2</b>	
<b>ESTUDIO DE SISTEMAS PARA TRAZADOS DE GRÁFICAS EN UN PLANO MEDIANTE UN BRAZO ROBOT .....</b>	<b>9</b>
2.1. DEFINICIÓN DE ROBOT MANIPULADOR .....	9
2.2. ROBOTS MANIPULADORES CARTESIANOS .....	10
2.3. S.C.A.R.A. ....	11
2.4. ROBOTS ARTICULADOS .....	13
<b>CAPÍTULO 3</b>	
<b>DISEÑO E IMPLEMENTACIÓN DEL SISTEMA MECÁNICO PARA UNA HOJA TAMAÑO A3.....</b>	<b>15</b>
3.1. SELECCIÓN DE MATERIALES .....	15
3.2. PROCESO DE MANUFACTURA.....	16
3.3. SELECCIÓN Y ACOPLAMIENTO DE MOTORES .....	17
3.3.1. <i>Actuador para el eje Y</i> .....	17
3.3.2. <i>Actuador para el eje Z</i> .....	18
<b>CAPÍTULO 4</b>	
<b>DISEÑO E IMPLEMENTACIÓN DEL SISTEMA .....</b>	<b>20</b>
4.1. DISEÑO DE UNA INTERFAZ GRÁFICA DE USUARIO PARA LA GENERACIÓN DE TRAYECTORIAS	20
4.1.1. <i>Ingreso y salida de datos del usuario</i> .....	20
4.1.2. <i>Cálculo de trayectorias</i> .....	22
4.1.2.1. Cálculo de los puntos intermedios .....	22
4.1.2.2. Cálculos de cinemática inversa .....	23
4.1.3 <i>Simulación gráfica</i> .....	25
4.1.4. <i>Cálculo de los parámetros de las funciones</i> .....	28
4.1.5. <i>Visualizar las gráficas de las funciones</i> .....	30
4.1.5.1. Visualización por medio de la interfaz gráfica de usuario .....	30
4.1.5.2. Visualización por medio del programa SMA4 .....	31
4.1.6. <i>Visualizar detalles</i> .....	33
4.1.7. <i>Comunicación serial</i> .....	34
4.1.7.1. Abrir el puerto serial.....	34
4.1.7.2. Configurar el puerto serial .....	36
4.1.7.3. Configurar los tiempos del puerto serial .....	37
4.1.7.4. Escritura y lectura del puerto serial.....	38
4.1.7.5. Cerrar el puerto serial .....	40
4.1.8. <i>Programa principal</i> .....	41
4.2. DISEÑO DE LA ETAPA DE CONTROL Y DE ADQUISICIÓN DE SEÑALES .....	44
4.2.1. <i>Adquisición de señales</i> .....	45

4.2.1.1. Adquisición del desplazamiento lineal en el eje z .....	45
4.2.1.2. Adquisición de la señal de ángulos relativos .....	48
4.2.1.3. Conversión de la señal analógica a digital.....	49
4.2.1.4. Acondicionamiento de la señal adquirida del potenciómetro .....	50
<b>4.2.2. Generación de las señales de control y comunicación con el software de la computadora.....</b>	<b>54</b>
4.2.2.1. Circuito de conversión de niveles de voltaje para la comunicación serial.....	55
<b>4.2.3. Circuito Regulador de voltaje de alimentación de la tarjeta de control y comunicación.....</b>	<b>57</b>
<b>4.2.4. Configuración del microcontrolador ATmega8.....</b>	<b>60</b>
<b>4.3. DISEÑO E IMPLEMENTACIÓN DEL CIRCUITO EXCITADOR DE CADA MOTOR .....</b>	<b>65</b>
<b>4.3.1. Circuito excitador de motor paso a paso unipolar .....</b>	<b>66</b>
4.3.1.1. Diseño propuesto.....	66
4.3.1.2. Implementación del circuito .....	69
<b>4.3.2. Circuito excitador de motor de corriente continua de imán permanente.....</b>	<b>71</b>
4.3.2.1. Diseño propuesto.....	72
4.3.2.2. Implementación del circuito .....	78
<b>4.3.3. Interfaz entre microcontrolador y circuito excitador de motor de corriente continua de imán permanente .....</b>	<b>80</b>
4.3.3.1. Diseño propuesto.....	80
4.3.3.2. Implementación de la interfaz .....	81
<b>4.3.4. Cálculo de disipadores para dispositivos de potencia.....</b>	<b>81</b>
4.3.4.1. Transistores TIP122 .....	82
4.3.4.2. Transistores IRF540 e IRF9540.....	83
<b>CAPÍTULO 5</b>	
<b>IMPLEMENTACIÓN DEL SISTEMA Y PRUEBAS .....</b>	<b>85</b>
5.1. IMPLEMENTACIÓN DEL SISTEMA .....	85
5.2. PRUEBAS DEL SISTEMA .....	86
5.3. AJUSTE DE PARÁMETROS DEL SISTEMA .....	86
5.4. RESULTADOS .....	88
<b>CONCLUSIONES .....</b>	<b>94</b>
<b>BIBLIOGRAFÍA.....</b>	<b>96</b>

## Introducción

En los últimos años los avances tecnológicos a nivel mundial en el campo de la electrónica son cada vez mayores y más significativos. Estas nuevas tecnologías son luego aplicadas a la vida cotidiana de distintas formas, mejorando así el nivel de vida de la sociedad en general. Sin embargo, detrás de cada tecnología hay una gran cantidad de esfuerzo, tiempo y recursos en lo que es la investigación previa. En la mayoría de los casos no apreciamos esto o simplemente damos las cosas que tenemos por sentado.

Los países altamente industrializados desarrollan y comercializan aparatos y equipos sumamente avanzados y costosos porque invierten muchos recursos en lo que es la investigación de nuevas tecnologías. Lamentablemente, este no es el caso en el Perú. A nivel nacional, las empresas buscan soluciones en otros países y solo se limitan a utilizar los equipos. Esto se debe a que no se fomentan las investigaciones de forma adecuada.

Esta falta de desarrollo también se aprecia en el campo de la robótica, ya que en nuestro país se encuentra muy limitada. En nuestra propia universidad hay muy pocas investigaciones o tesis con respecto al tema. Por esta razón se decidió diseñar e implementar un brazo robot de dos grados de libertad para el trazado de diagramas en un plano.

El objetivo de este proyecto es servir como un precedente y base para desarrollar brazos robots más complejos. El trazado de planos es una de las tantas aplicaciones que se le podrían dar a un brazo robot, las cuales varían dependiendo de la herramienta que se coloque en el efector final del robot.

Para controlar los movimientos del brazo robot, el usuario será capaz de introducir las coordenadas finales del trazo por medio de una interfaz gráfica. El programa realizará el cálculo de la trayectoria y enviará estos datos al brazo por medio del puerto serial. El software además mostrará una simulación gráfica del movimiento y las gráficas del movimiento realizado.

Se contará también con una etapa de control de los actuadores que recibirá los datos enviados por el puerto serial desde el programa en la computadora. Esta etapa debe asegurarse de que se realice el trazo deseado mediante realimentaciones de posición para cada eje de movimiento. Esta etapa además debe ser capaz de devolver los valores de posición reales obtenidos de las lecturas de los sensores al programa de la computadora usando el mismo puerto serial.

Para poder transmitir las señales de control hacia los actuadores seleccionados, se deben diseñar las interfaces adecuadas para cada tipo de actuador. Básicamente cada interfaz se encargará de darle la potencia adecuada a la señal de control para que sea capaz de manejar a los actuadores, teniendo en cuenta la protección debida del controlador, así como la protección del mismo actuador.

El objetivo principal del presente trabajo es diseñar, desarrollar e implementar un brazo robot de dos grados de libertad. El ingreso de los datos del usuario será por medio de una interfaz gráfica. Además se busca motivar futuras investigaciones en temas de robótica.

## CAPÍTULO 1

### PLANTEAMIENTO GENERAL DEL SISTEMA

El desarrollo del brazo robot manipulador comprende diferentes áreas de trabajo, siendo estas la interfaz final de usuario, el control del movimiento, la interfaz de potencia y el diseño mecánico.

La función principal de la interfaz de usuario es de recibir los datos de entrada por parte del operario, en este caso equivalen a los movimientos del brazo robot, y transmitirlos como datos numéricos hacia el sistema de control del brazo robot. Para desarrollar el programa se utilizará el lenguaje Visual C++, debido a que tiene la capacidad de programar entornos gráficos fácilmente. Esto permite que el programa sea capaz de realizar una gran cantidad de cálculos matemáticos (como por ejemplo el cálculo de las posiciones del brazo robot) y operaciones en tiempo real.

El control de movimiento es de suma importancia, puesto que se puede obtener un control del manipulador acorde a nuestros requerimientos para así cumplir con el objetivo, que es la realización de trazos sobre un plano tamaño A3.

El diseño de la interfaz de potencia hará el enlace entre el control digital y los actuadores. La elección de los actuadores se hace en base a los requerimientos de precisión y de fuerza, incluyendo también mecanismos de protección para los dispositivos.

Los motores a pasos constituyen una elección apta para el trabajo de manipuladores debido a la precisión de sus movimientos y a la simplicidad de la interfaz de control. Aunque presentan la desventaja de un menor torque comparado con otros motores de corriente continua. Por este motivo se seleccionaron motores de ambos tipos para esta aplicación: en un eje se utiliza un motor a pasos, donde la precisión y el torque lo permiten; y en el otro eje se utiliza un motor de corriente continua, en donde el requerimiento de torque es mayor.

La etapa de adaptación de señales de control implica el manejo de voltajes y corrientes elevadas para ambos motores. Esto se puede lograr utilizando transistores de potencia tipo Darlington para el motor a pasos, debido a su flexibilidad de rango de trabajo con estos parámetros y a la simplicidad en el diseño de su circuito de control, además se encuentran disponibles en el mercado local a bajo costo. Además se utilizan transistores MOSFET para el diseño del circuito excitador del motor de corriente continua, debido a la facilidad de manejo de estos dispositivos en la configuración utilizada y al mayor rango de trabajo en comparación con otros tipos de transistores. Sin embargo se deben tener en cuenta las características no deseadas que presentan estos dispositivos para poder evitar su malfuncionamiento y los límites de trabajo de cada uno.

El diseño mecánico y su implementación son puntos muy importantes ya que estos definirán los alcances básicos del robot y serán expuestos con mayor detalle en el capítulo 3 del presente documento.

### 1.1. Etapas del brazo robot

Ya que la implementación del presente proyecto presenta una alta complejidad, se ha decidido dividir del desarrollo del robot manipulador en tres etapas principales: Interfaz de usuario, control de movimiento e interfaz de potencia. La interacción entre dichas partes se aprecia en la figura 1.

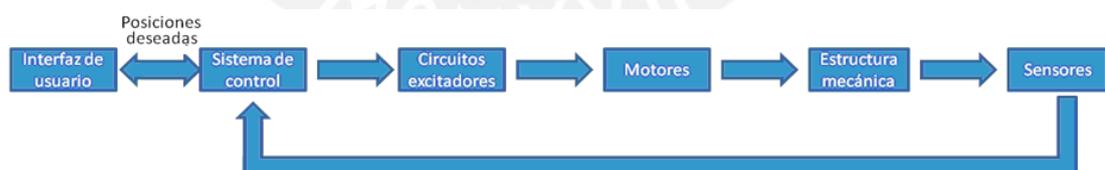


Figura 1. Representación gráfica del modelo teórico.

#### 1.1.1. Interfaz de usuario

La interfaz de usuario consiste en una aplicación, programada por medio del lenguaje Visual C++ para que el usuario sea capaz de introducir a la computadora los movimientos que desea que el robot manipulador realice. Con la finalidad de que la programación del robot manipulador sea lo más

simple posible, se desarrollará una interfaz gráfica que interactúe de forma amigable con el usuario. Esta le mostrará al usuario el estado de las variables del robot manipulador por medio de una simulación y permitirá el ingreso de datos de forma fácil y directa. Adicionalmente, si el usuario desea saber los valores numéricos de las variables de estado, puede acceder a estos datos por medio de una ventana que se activará cuando el usuario lo desee.

El objetivo del robot manipulador solo será el trazar líneas rectas entre dos puntos dentro de un área de trabajo. Una vez que se tienen el punto inicial y el punto final del movimiento deseado, se determinan los puntos intermedios entre el inicial y final, los cuales son almacenados en el programa. Una vez obtenidos estos puntos, por medio de cálculos geométricos se obtienen los ángulos relativos deseados en las uniones en cada punto respectivamente. Estos ángulos se distribuyen en el tiempo para formar una función discreta de la trayectoria deseada.

Una vez que se tiene la función discreta, el programa halla la función continua de las variables que definen el movimiento del brazo robot. Adicionalmente, el usuario debe ser capaz de visualizar las gráficas de las funciones. Para enviar los datos al programa del controlador, se utilizará el puerto serial.

Además, el programa realizará una simulación gráfica del movimiento del robot. En esta simulación se observarán dos vistas del robot, la lateral y la horizontal. De esta forma el usuario podrá visualizar el movimiento del robot manipulador en la computadora antes de que se ejecute la acción. Una ventaja de Visual C++, es que tiene una serie de instrucciones que permiten realizar gráficos simples en la ventana de la aplicación por medio de líneas de código. De esta forma, utilizando las variables de estado de robot (posiciones de las uniones) se puede realizar una simulación del robot. Esta gráfica se actualiza cada vez que una variable cambia, dándole la ilusión de movimiento. También en la simulación se visualizarán los puntos en donde el efector final está en contacto con el área de trabajo.

### 1.1.2. Control de movimiento

Como se mencionó anteriormente, la etapa de control de movimiento será la encargada de controlar el movimiento de cada motor para así lograr el movimiento deseado del sistema y del efector final. Para ello esta etapa recibirá los datos calculados por la interfaz de usuario en la computadora que le indicará las trayectorias a seguir. Empleando estos datos como referencia, el sistema de control deberá calcular la cantidad de movimiento requerida para cada motor y luego traducirá esto a secuencias coordinadas que serán enviadas a la etapa de potencia.

La precisión será dada al sistema mediante el uso de la realimentación de la posición angular relativa entre los eslabones. Para ello se empleará un potenciómetro acoplado al eje que une a los eslabones que permitirá medir la variación en la impedancia (convertidas en voltaje) que representa la variación angular. De esta manera se obtendrá una medida de la posición angular real de cada eslabón en todo momento, con lo cual se podrá corregir los errores respecto de la posición deseada.

El empleo de potenciómetros es factible puesto que el rango de movimiento de cada eslabón respecto al anterior al cual está conectado está restringido por el área de trabajo. Además, se pueden emplear convertidores analógicos – digitales, con lo cual se pueden convertir fácilmente las señales de voltaje entregadas por los potenciómetros en valores binarios a ser usados por el controlador.

Las secuencias generadas por el controlador serán enviadas a la tarjeta de potencia, que se encargará de elevar los niveles de voltaje y corriente y de convertir estos niveles lógicos a los voltajes y corrientes requeridos por los motores para así producir el movimiento mecánico deseado.

Para realizar todas estas funciones se propone el uso de un microcontrolador, y puesto que la velocidad no es un factor fundamental en este proyecto, se pueden reducir costos al emplear un microcontrolador que tenga además un convertidor analógico digital incorporado, ya que no es necesario que los tiempos de conversión sean muy pequeños. El

microcontrolador debe además contar con las suficientes salidas para lograr el control de los motores y la posibilidad de comunicarse con una computadora personal para la recepción de los datos generados por el programa de usuario.

### 1.1.3. Interfaz de potencia

La interfaz de potencia consiste en una serie de dispositivos que en su funcionamiento conjunto serán capaces de controlar los actuadores finales, en este caso los motores de cada eje del brazo robot manipulador. Para ello tendrá comunicación directa con la etapa del controlador, la cual será su fuente de señales. Éstas señales, al ser incapaces de brindar energía suficiente a los motores, serán adaptadas a señales de mayor energía que sí puedan alimentarlos.

Previamente a la etapa de selección de dispositivos a usar y al diseño mismo de la interfaz, se deben elegir los motores adecuados al trabajo en el brazo manipulador. De acuerdo a la exigencia de cada eje en cuanto a torque, cada motor requerirá mayor o menor dimensión. Esto influirá drásticamente en el diseño, en especial en la etapa de selección de dispositivos, por lo tanto se debe realizar de manera correcta la elección para evitar sobredimensionar los motores.

También es objetivo de la interfaz proteger todos los dispositivos de la misma y también de la etapa del controlador. Al manejar la interfaz potencias más elevadas que el resto del sistema existe el riesgo que estas afecten y dañen componentes de las otras etapas, por tanto el aislamiento es requisito indispensable. También debe implementarse protecciones para los mismos dispositivos de la interfaz, teniendo en cuenta que se trabajará con motores. Otro objetivo importante es la comunicación de los sensores de posición con la etapa de controlador, ya que éstos estarán ubicados en los motores, es conveniente que la misma interfaz de potencia sea la encargada de transmitir la señal de los mismos al controlador.

Se propone el uso de motores a pasos para el trabajo en el eje vertical, dado su fácil control y su precisión en movimiento angular, ya que pueden

mantener una posición angular deseada. Para el movimiento del en el eje horizontal se propone el uso de un motor de corriente continua de imán permanente, debido al mayor requerimiento de torque. Para su control se propone un diseño en base a transistores MOSFET, pues son de fácil activación y permiten, en determinada configuración, el manejo de corriente en ambos sentidos y son capaces de soportar potencias bastante elevadas, suficientes para la aplicación actual.



## CAPÍTULO 2

### ESTUDIO DE SISTEMAS PARA TRAZADOS DE GRÁFICAS EN UN PLANO MEDIANTE UN BRAZO ROBOT

Dentro de la amplia variedad de brazos robot aplicados al trazado de diagramas en planos se encuentran varias configuraciones, cada una con características específicas. A continuación se presenta la definición de brazo robot manipulador aplicado a la industria. Luego se muestran las configuraciones de brazos robots más conocidas e información específica de cada una de ellas.

#### **2.1. Definición de robot manipulador**

Una de las definiciones de robot industrial más aceptadas es la brindada por la RIA (siglas en inglés para Asociación de Industrias Robóticas), la cual es: un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas, o dispositivos especiales, según trayectorias variables, para realizar tareas diversas.

De acuerdo a la evolución de la automatización y la robótica, los robots industriales han seguido el avance de la tecnología. Con su evolución también vinieron formas de clasificación de los mismos. Se presenta a continuación algunas formas de clasificación:

##### 1. Manipulador

Sistemas mecánicos capaces de realizar múltiples tareas y con un sencillo sistema de control. La tarea a realizar puede ser controlada de forma manual, mediante una secuencia fija o mediante una secuencia variable.

##### 2. Robots de repetición o aprendizaje

Manipuladores limitados a repetir una secuencia de movimientos previamente ejecutada por un operador.

### 3. Robots con control por computador

Estos sistemas implican tener un lenguaje específico en el computador para controlar al robot mediante una serie de instrucciones.

### 4. Robots inteligentes

Estos sistemas robóticos son capaces de relacionarse con su entorno mediante sensores y programarse en tiempo real en base a los datos recibidos por los sensores.

A continuación se presentan las configuraciones de robots manipuladores más conocidas y utilizadas en la industria.

## 2.2. Robots Manipuladores Cartesianos

Los Robots Manipuladores Cartesianos son robots manipuladores de tres grados de libertad, siendo estos grados de libertad tres ejes lineales, que tienen el espacio cartesiano como área de trabajo. Se desplazan en tres ejes, X, Y y Z, para lo cual cuentan con sistemas de posicionamiento deslizantes y perpendiculares entre sí para cada eje. El movimiento de los mismos puede ser secuencial o paralelo, de ello dependerá la complejidad del sistema encargado de su control. El espacio requerido para su instalación suele ser grande en comparación con su área de trabajo, La figura 2 muestra el esquema de un robot manipulador cartesiano y sus ejes de movimiento. Se le denomina también como configuración de caja debido a su apariencia.

Algunas de las ventajas de los manipuladores cartesianos son:

- Alta velocidad de desplazamiento lineal.
- Requiere de un sistema de control relativamente sencillo.
- Estructura rígida y estable debido a que los ejes están sostenidos a ambos lados.

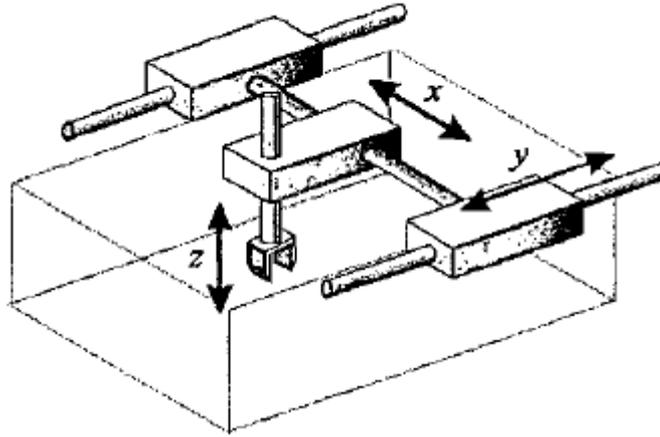


Figura 2. Ilustración esquemática de un robot 3D cartesiano. [2]

Y en contraparte, algunas de las desventajas que presentan estos manipuladores son:

- Grandes requerimientos de espacio para su instalación.
- Debido a las largas guías que requiere cada eje, el sistema es muy susceptible al daño por el polvo en las partes mecánicas.
- El movimiento que se puede lograr con este manipulador es mayoritariamente lineal, y su configuración limita la posibilidad de trabajo en situaciones que requieran trabajos no-paralelos al plano XY del robot.

### 2.3. S.C.A.R.A.

La palabra S.C.A.R.A. corresponde a la abreviación de 'Selective Compliance Assembly Robot Arm', cuya traducción al español es 'Brazo robótico de ensamblaje selectivo y compensado'. Esto significa que el movimiento del brazo robot se limita a dos dimensiones (eje X,Y); mientras que es rígido para el eje Z, trabajando en coordenadas cilíndricas donde el radio se trabaja con dos eslabones. Es una combinación de dos brazos manipuladores: el brazo articulado y el brazo cilíndrico [3].

Las principales ventajas del SCARA son:

- Se necesita poco espacio para la cimentación en relación con el área de trabajo.

- Elevada rigidez en el brazo en dirección vertical.
- Mínima deflexión del brazo al moverse a una velocidad programada, incluso al tener carga.
- Resistencia a colisiones y choques en todo su conjunto.
- Elevada velocidad horizontal y rápidos ciclos para acciones de 'recoger y colocar' (pick and place).
- Control del brazo es más simple (ya que los efectos de la gravedad se consideran constantes para el análisis dinámico).

En contraparte podemos mencionar:

- Requiere un control más complejo, comparado con el Robot Cartesiano.
- Es de coste más elevado

De acuerdo a la naturaleza del movimiento de los brazos SCARA, las aplicaciones más comunes son en las áreas de ensamblaje mecánico (discos duros, componentes electrónicos, etc), embalaje, alimentación a otras máquinas, manipulación de materiales en general y en muchas operaciones donde se requieren movimientos repetitivos, rápidos y precisos.

Un gran número de compañías se especializan en construir diversas versiones de brazos tipo SCARA, dependiendo de la aplicación específica en la que se van a usar. Algunas de estas compañías son: Epson, Hirata, Kuka y Adept.

Se muestra en la figura 3 el modelo KR 10 Scara R850, de la empresa KUKA.



*Figura 3. Robot SCARA KR 10 scara R850 [4]*

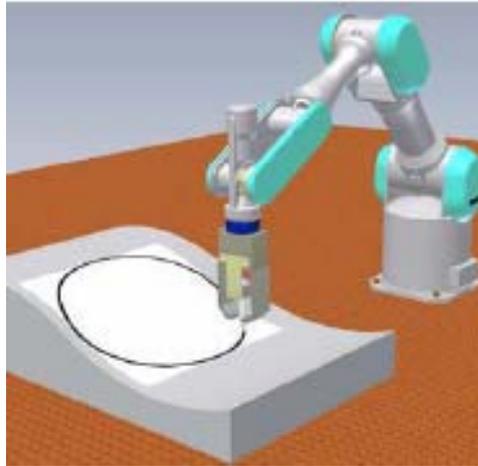
#### **2.4. Robots articulados**

Los manipuladores articulados presentan una capacidad de movimiento similar a la de un brazo humano y son los más usados comúnmente. Básicamente el movimiento del robot se realiza por medio del accionamiento de articulaciones. La configuración básica es tener dos o tres articulaciones asociadas al cuerpo y al brazo, y dos o tres más asociadas a la muñeca. Esta configuración está constituida por dos componentes rectos (brazo y antebrazo humanos) sostenidos por un pedestal giratorio, unidos por articulaciones correspondientes al hombro y codo. En el extremo del antebrazo se encuentra una muñeca, proporcionando articulaciones suplementarias.

Las articulaciones utilizadas en el diseño de los robots suelen implicar un movimiento relativo de las uniones continuas, movimiento que es lineal o rotacional. Las articulaciones lineales implican un movimiento de traslación. Hay como mínimo tres tipos de articulaciones giratorias que pueden distinguirse en los manipuladores de robots.[5]

Dada su gran capacidad de movimiento, su aplicación en trazados no se limita a planos. Se encuentran también aplicaciones en trazados sobre superficies en tercera dimensión. Este tipo de aplicación se muestra en la figura 4. Las aplicaciones del tipo industrial del manipulador articulado

incluyen soldadura, pintura y manejo de carga, que implican actividades repetitivas.



*Figura 4. Robot articulado Mitsubishi PA-10, ejemplo de aplicación de trazado en superficies tridimensionales. [6]*



## CAPÍTULO 3

### DISEÑO E IMPLEMENTACIÓN DEL SISTEMA MECÁNICO PARA UNA HOJA TAMAÑO A3

El sistema elaborado debe ser capaz de controlar el movimiento en los ejes Y y Z del plano de trabajo, el cual es un plano tamaño A3. Tomando esto en consideración y con la colaboración de personas capacitadas en el área de mecánica se elaboró un diseño capaz de realizar estas tareas. Se elaboraron planos de construcción del mismo y el proceso de fabricación se realizó según los mismos. Los planos se adjuntan en el Anexo 1.

A continuación se detallarán los criterios empleados para la construcción del prototipo.

#### **3.1. Selección de materiales**

Debido a las dimensiones del prototipo a realizar no se pudo emplear metal para construir todo el sistema, puesto que esto hubiese elevado el precio y el peso del mismo. Es por estas razones que se eligió madera como material para la construcción de la base del sistema y del plano elevado que sostiene la hoja A3. De esta manera el peso se reduce notablemente, lo cual contribuye a la portabilidad del prototipo.

Para todos los soportes se requiere firmeza pero también se requiere que el peso sea reducido. Por ello se elige el aluminio como material para estas piezas. El aluminio se encuentra en el mercado local en placas, por lo que se les corta y dobla hasta en un ángulo de 90° para fabricar los soportes de las guías de la base del brazo, que se desplaza gracias al tornillo sinfín.

Las guías deben ser rígidas y firmes, pero así mismo deben ser lisas para que ofrezcan el mínimo rozamiento al momento en que se desplace la base del brazo. Por estas razones se eligió usar varas de acero trefilado para elaborar estas guías.

El tornillo sinfín encargado de transmitir el movimiento en el eje Z se escogió de acuerdo a la disponibilidad y los precios en el mercado local. Fue así que se eligió usar un tornillo sinfín de hierro de 1.55mm de paso por vuelta.

Los eslabones del brazo son elaborados a base de tubos cuadrados de aluminio para reducir el peso. Esta elección es de gran importancia para reducir el torque necesario del motor paso a paso, puesto que si los eslabones tienen un peso excesivo el motor paso a paso encargado de generar el movimiento en el eje Y requeriría un torque mayor, y la disponibilidad de motores de alto torque y tamaño reducido en el mercado local es baja.

Los pines de unión y las bocinas deben soportar las cargas a las que sean sometidas. En este caso, las cargas son reducidas y por ello para reducir el peso del sistema se optó por emplear nylon para elaborar estos componentes. El nylon además es muy fácil de trabajar en procesos de torneado y requiere un menor tiempo para su mecanizado, además de tener un costo reducido.

La estructura que sostiene el efector final fue elaborada en acrílico para reducir el requerimiento de torque del motor y la rueda omnidireccional es de plástico debido a las opciones presentes en el mercado.

### **3.2. Proceso de manufactura**

Para la elaboración de las bocinas y pines se trabajó con procesos de torneado. Se escogió este procedimiento debido a la importancia de elaborar piezas precisas y simétricas, y el torno permite realizar trabajos de alta precisión. Además el torno permite realizar piezas perfectamente simétricas debido a la forma en que se trabaja: se hace rotar el material y la cuchilla encargada de remover el material entra en contacto con el mismo eliminando el material deseado.

La elaboración de las demás piezas como los soportes y ángulos involucró el doblado, taladrado y limado de planchas de metal cortadas a la medida o de tubos cuadrados. La elaboración de los soportes de las guías del eje Z y de la base del brazo debió realizarse con especial cuidado para asegurar

siempre la distancia correcta entre los centros de las guías y del tornillo sinfín. Esto es debido a que si no se cumple con este requerimiento el sistema se puede bloquear mecánicamente forzando los motores e impidiendo el funcionamiento normal del mismo.

### 3.3. Selección y acoplamiento de motores

Para la aplicación del brazo robot de dos grados de libertad, se tienen características específicas para el movimiento del brazo en cada uno de los ejes perpendiculares. Estas características de ambos movimientos se ven reflejadas en la selección de los motores para realizarlos.

#### 3.3.1. Actuador para el eje Y

El movimiento en este eje se transmite mediante la rueda omnidireccional en la dirección perpendicular al tornillo sinfín. Para realizar este movimiento se requiere de un tipo de actuador que pueda ser fácil de controlar y que pueda mantener una posición angular específica. De esta forma al realizar un movimiento recto en el eje Z, del tornillo sinfín, se pueda tener una línea recta.

Teniendo en cuenta estos factores, se elige como actuador para este eje un motor paso a paso unipolar. El motor paso a paso permite fácilmente moverse una posición angular definida, mediante la activación secuencial de sus bobinas, y mantener la última posición angular, manteniendo la energía en la última bobina de la secuencia. El factor que un motor paso a paso sea unipolar implica únicamente que su circuito excitador sea más simple.

Trabajando únicamente con el limitado mercado local para la elección del motor, se optó por seleccionar al motor PM25S-048. Entre las características principales del motor contamos con un paso angular de  $7.5^\circ$ , un valor de resistencia de  $65\Omega$  por fase, una alimentación de 12V y un rango de torque que varía entre 1.5 y 5.0 mNm. Además de esto el tamaño del motor fue el idóneo para el acoplamiento en el sistema.

Al realizar las pruebas al motor se midió una corriente de 184mA por fase, además de una elevación de temperatura considerable al trabajar en

secuencia de medios pasos, donde por momentos se energizan dos bobinas simultáneamente. Sin embargo la temperatura desarrollada por el motor se encontraba dentro del rango especificado en su hoja de datos.

Para el acoplamiento del motor se instaló una bocina en su eje para poder acoplar al eje del motor un engranaje de manera que transmita el movimiento rotatorio a la rueda omnidireccional. El motor se sujetó a la estructura que sostiene al efector final mediante los agujeros de montaje del mismo, de manera que se logre la mejor sujeción posible entre ambos.

Para el caso del motor paso a paso utilizado se puede tener una precisión teórica que va a depender del radio de la rueda omnidireccional y del el paso angular del motor. Ambos valores son conocidos, teniendo en cuenta si se maneja al motor con medios pasos y que el radio de la rueda omnidireccional es 20mm:

$$L_{\min} = \theta \times R \quad (\text{ec. 1})$$

Entonces:

$$L_{\min} = \left( \frac{7.5^\circ}{2} \times \frac{\pi}{180^\circ} \right) \text{rad} \times 20\text{mm} = 1.31\text{mm}$$

### 3.3.2. Actuador para el eje Z

Para lograr el movimiento en este eje, a lo largo del área de trabajo, se emplea un tornillo sinfín. En este caso se necesita un torque más elevado que en el caso del eje Y, pues es necesario mover el peso de todo el sistema. Debido a la configuración empleada el sistema no presentará movimiento en el eje Z si no se energiza el motor correspondiente.

Otra característica del sistema para el eje Z es que la transmisión de movimiento rotacional a lineal es pequeña, aproximadamente de 1.55mm por revolución del tornillo. Esto da mayor libertad para el movimiento rotatorio del actuador en el eje Z, ya que permite el movimiento de una vuelta completa del actuador para obtener un desplazamiento lineal apreciable.

Al tomar estas características del movimiento del eje Z, se puede seleccionar como actuador para este eje a un motor de corriente continua. Este motor

debe ser capaz de proporcionar el torque necesario para realizar el movimiento angular del tornillo sinfín.

Para este trabajo, teniendo en cuenta la disponibilidad del mercado local, se eligió trabajar con el motor RS-360SH-09600, del cual se cuenta únicamente con los datos de placa: voltaje nominal 24V y corriente 230mA. No se cuenta con la hoja de datos del motor. El motor está acoplado a una caja reductora de 90°, la cual convierte el giro en un sentido a un giro perpendicular al original. Al realizar las pruebas al motor para determinar si el torque desarrollado por el motor era el necesario para la aplicación descrita, se determinó que sí podía realizar el trabajo determinado.

Para acoplar el motor de corriente continua al sistema se elaboró una bocina la cual se acopló al eje final de la caja reductora y el otro extremo de la bocina se acopló al tornillo sinfín. Para permitir el correcto acoplamiento con la bocina, se eliminó parte de la rosca del tornillo sinfín en un extremo mediante un proceso de torneado.

## CAPÍTULO 4

### DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

#### **4.1. Diseño de una interfaz gráfica de usuario para la generación de trayectorias**

Para esta etapa se utilizó el programa Visual Studio 2008 de Microsoft; específicamente se realizó la programación en el lenguaje Visual C++. El programa desarrollado debió realizar las siguientes funciones: obtener los datos ingresados por el usuario, realizar los cálculos matemáticos de las trayectorias, enviar los datos de la trayectoria al sistema de control de movimiento, mostrar la información por medio de gráficas y finalmente mostrar una simulación gráfica del movimiento.

Se decidió utilizar el lenguaje Visual C++ por la siguiente razón:

- Visual C++ es uno de los lenguajes más rápidos en la actualidad con la capacidad de trabajar con entornos visuales. La velocidad del programa es un parámetro importante en el presente trabajo debido a que el cálculo de las trayectorias requiere de un gran número de cálculos matemáticos, el programa tarda un tiempo considerable en ejecutarse. Además, la simulación gráfica también requiere de una alta velocidad de procesamiento, ya que se desea que los movimientos se realicen en un tiempo determinado. Por tales motivos, mientras más rápido sea el lenguaje de programación utilizado mejores serán los resultados.

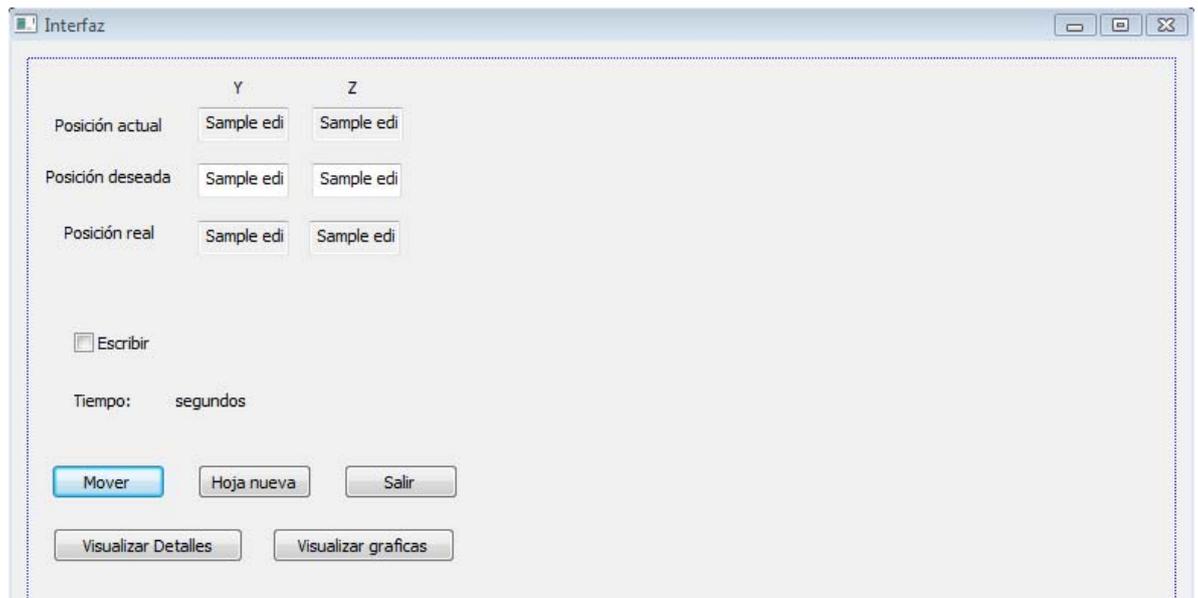
El programa desarrollado está compuesto por varios distintos bloques que se encargan de ciertas tareas específicas. A continuación se explicarán a detalle cada uno de estos bloques para luego describir el funcionamiento del programa en su totalidad.

##### **4.1.1. Ingreso y salida de datos del usuario**

Para empezar se crea un nuevo proyecto por medio de Visual Studio 2008. Este proyecto debe de ser una aplicación MFC (Microsoft Foundation Class)

basada en diálogos. De esta forma se crea una ventana vacía, que será la base sobre la cual se realizará la programación de la interfaz. [9]

Utilizando la barra de herramientas se pueden agregar distintos elementos a la ventana de diálogo que se está creando. Para este proyecto en particular, se utilizaron cuadros de texto, botones de comando y textos estáticos distribuidos como se muestra en la figura 5.



*Figura 5. Ventana de diálogo con los elementos para el ingreso de datos.*

En la parte superior de la ventana se encuentran un grupo de cuadros de texto. Los pertenecientes a la fila superior muestran las coordenadas actuales (teóricas) del efector final del brazo (sistema cartesiano, con los ejes Y,Z respectivamente). En la segunda fila, el usuario es capaz de ingresar las coordenadas finales del movimiento a realizar. Finalmente, en la tercera fila se encuentran las posiciones actuales del brazo robot (reales) que son recibidas del sistema de control de movimiento. Estos datos son recibidos por medio del puerto serial.

Luego se aprecia un pequeño cuadrado para marcar (check box) con el identificador "Escribir". Si es que el cuadrado esta marcado, al realizar algún movimiento, se trazará una recta en el papel por los puntos donde pase el efector final; en caso contrario se realizará el movimiento pero no se

realizará ningún trazo. Adicionalmente hay un texto que señala el tiempo transcurrido desde el inicio del movimiento en segundos.

Finalmente se colocaron los botones de comando de la interfaz. A continuación, se explicará la función de cada uno:

- Mover: Al presionar este botón se inicia el programa principal de la interfaz. El programa se encarga de tomar las coordenadas de la posición deseada, validar los datos, hacer el cálculo de la trayectoria, ejecuta la simulación del movimiento y envía los datos por el puerto serial al sistema de control de movimiento.
- Hoja nueva: Este botón sirve para borrar los trazos realizados previamente y de esta forma empezar un nuevo dibujo en una hoja nueva.
- Salir: Se cierra la ventana y se cierra el programa.
- Visualizar detalles: Al presionar el botón, se abre otra ventana en donde se muestran datos adicionales al usuario (los ángulos entre cada eslabón y las coordenadas X,Y,Z de cada articulación del robot).
- Visualizar gráficas: Sirve para visualizar las gráficas de las variables de control (ángulos entre los eslabones y la posición en el eje Z) vs tiempo en una nueva ventana.

#### 4.1.2. Cálculo de trayectorias

##### 4.1.2.1. Cálculo de los puntos intermedios

Una vez validada la posición final del movimiento se procede a calcular todos los puntos intermedios entre la posición inicial y la final de la recta. El primer paso es obtener la pendiente de la recta (ec. 2).

$$\text{pendiente} = \Delta Y + \Delta Z \quad (\text{ec. 2})$$

donde:  $\Delta = \text{posición final} - \text{posición inicial} \quad (\text{ec. 3})$

En el caso que el módulo de la pendiente sea mayor o igual a 1, la posición Y varía en un milímetro y se calcula la posición Z utilizando la pendiente. Se utiliza la misma lógica en caso que el módulo de la pendiente sea menor que 1, pero con la única diferencia en que la posición Z se modifica en un milímetro y se calcula la posición Y.

Se trabajan con ambos casos para obtener la mayor cantidad de puntos intermedios entre el punto inicial y el final. Además, para que el programa funcione en los casos extremos (solo varían las coordenadas de un eje) es necesario hacer esta distinción.

La figura 6 ilustra el cálculo de los puntos pertenecientes a la recta.

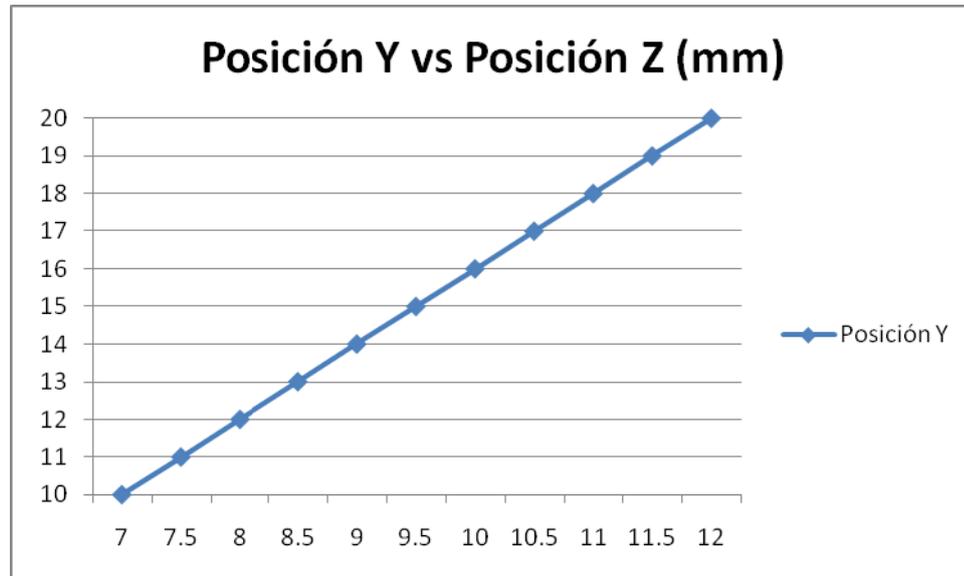


Figura 6. Distribución de puntos en una línea recta.

En este caso la pendiente es igual a 2. Por esta razón la posición Y aumenta en razón a 1mm y la posición Z es la igual a:

$$\text{posición Z} = \text{posición inicial Z} + \Delta Y + \text{pendiente} \quad (\text{ec. 4})$$

Todos los puntos intermedios están separados por un intervalo de tiempo de 1 segundo. De esta forma la velocidad máxima en cada uno de los ejes es de 1mm/segundo. Se fija este límite de velocidad considerando la velocidad máxima de desplazamiento del brazo robot implementado.

#### 4.1.2.2. Cálculos de cinemática inversa

El cálculo de cinemática inversa consiste en obtener los ángulos entre los eslabones del brazo a partir de las coordenadas del extremo del brazo. Una vez conocidos los ángulos se pueden calcular también las posiciones de las juntas del brazo. Estos datos se utilizarán para realizar la simulación del movimiento del brazo y luego serán enviados al sistema de control de movimiento [7].

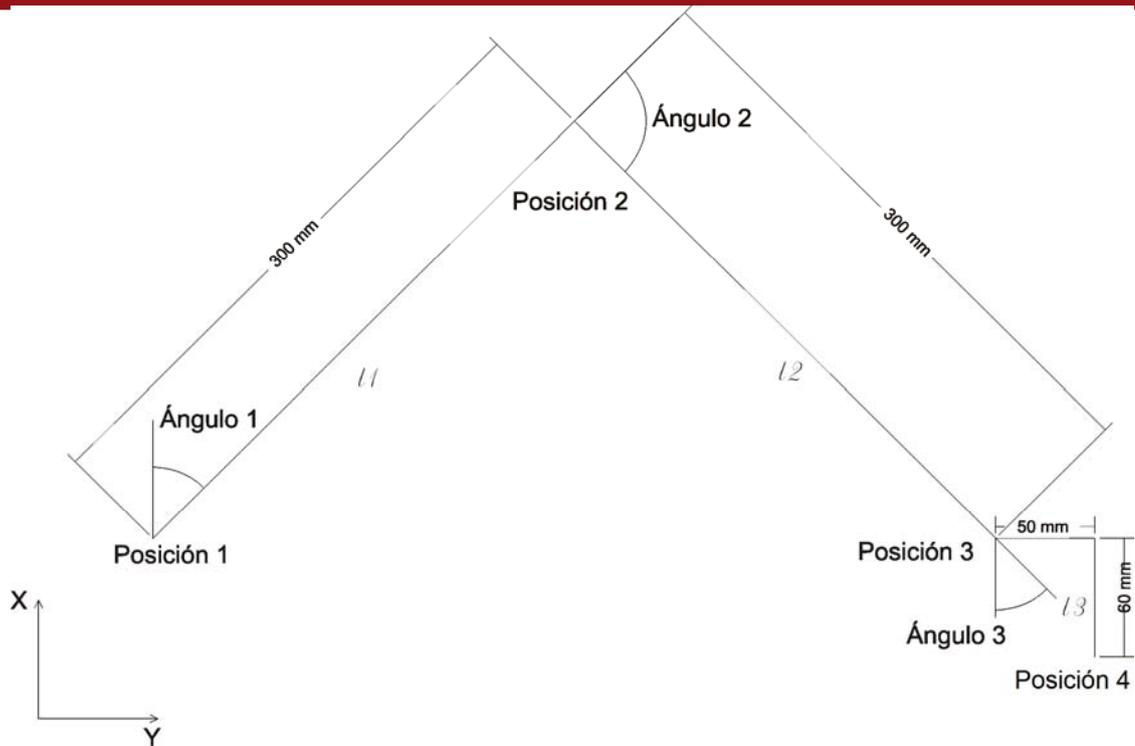


Figura 7. Configuración geométrica del brazo robot.

Dada la configuración y características físicas del sistema, se pueden calcular las posiciones de los distintos eslabones de forma matemática. Las longitudes  $l_1, l_2$  son iguales a 300mm, mientras que  $l_3$  es igual a 60mm.

El sistema debe de cumplir las siguientes condiciones:

- La coordenada Y de la posición 3 es igual a la de la posición 4 menos un desplazamiento de 50mm, debido a las características mecánicas del brazo.
- Las coordenadas X de las posiciones 1 y 3 deben de ser iguales. Esta condición se debe de cumplir para que los ángulos 1 y 3 sean iguales porque las longitudes de los eslabones son iguales (formando un triángulo isósceles).
- Todas las posiciones en el eje Z son iguales.

Aplicando la ley de cosenos en triángulo que se forma con ambos eslabones, se puede calcular el valor del ángulo2:

$$\text{Ángulo2} = \cos^{-1} \left( \frac{\text{Posición3}y^2 - l_1^2 - l_2^2}{2l_1l_2} \right) \quad (\text{ec. 5})$$



Para utilizar las instrucciones gráficas de visual C++ primero se debe de declarar el objeto en donde se realizarán los trazos (en este caso el cuadro de diálogo). Esto se logra por medio de la siguiente instrucción:

```
CClientDC dlgDC(this);
```

Luego de esto ya se pueden realizar las gráficas deseadas, utilizando las instrucciones disponibles en visual C++ que son: MoveTo(), LineTo(), Rectangle(), Ellipse(), SetPixel() y TextOut() [8]. A continuación se explicará brevemente como funcionan estas instrucciones:

- MoveTo(int x, int y) sirve para modificar la posición del cursor gráfico que se utiliza para realizar los gráficos. Sus parámetros de entrada son valores enteros que representan a las coordenadas X e Y de la posición del cursor.
- LineTo(int x, int y) se utiliza para trazar líneas desde la posición del cursor gráfico hacia la posición indicada por los parámetros de entrada.
- Rectangle(int x1, int y1, int x2, int y2) se utiliza para trazar rectángulos. Los parámetros de entrada x1 e y1 indican la posición de uno de los vértices del rectángulo; mientras que x2 e y2 indican la posición del vértice opuesto.
- Ellipse(int x1, int y1, int x2, int y2) sirve, como su nombre lo indica, para graficar elipses. Los parámetros de entrada son los vértices del rectángulo que contiene a la elipse.
- SetPixel(int x, int y, COLORREF crColor) cambia el color de un pixel determinado. Los primeros dos parámetros de entrada son las coordenadas del pixel. El tercer parámetro determina el color del pixel, el cual se define por sus componentes de rojo, verde y azul (RGB).
- TextOut(int x, int y, LPCTSTR lpszString, int nCount) sirve para insertar textos en el gráfico. Los primeros dos parámetros son las coordenadas en donde se situará el texto. El tercer parámetro contiene el texto en sí, mientras que el último parámetro contiene la longitud del texto.

Utilizando estas instrucciones se puede obtener una simulación gráfica del brazo robot, la cual puede ser apreciada en la figura 8.

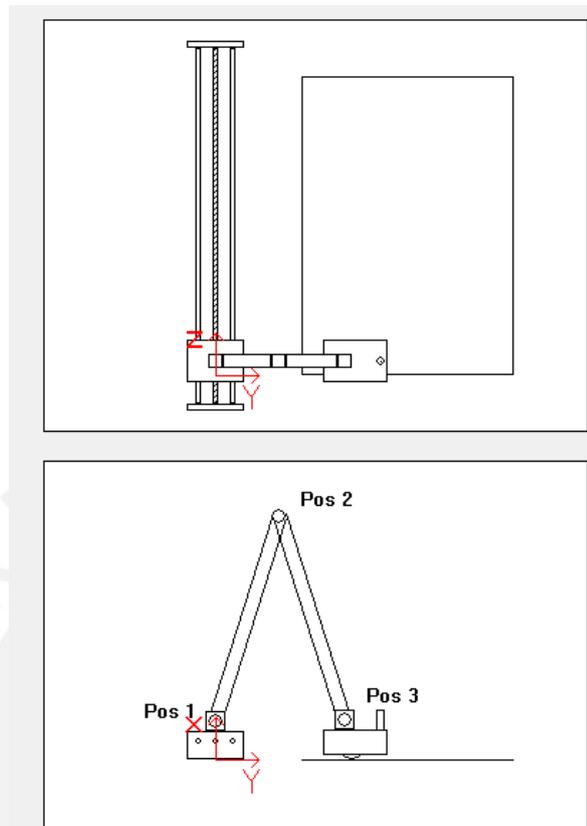


Figura 8. Simulación gráfica del brazo robot.

La simulación gráfica se coloca al lado de los cuadros de entrada de datos de la ventana principal, para que el usuario pueda observar el movimiento del brazo robot y saber la posición actual del puntero. Además, el programa almacena los trazos realizados previamente.

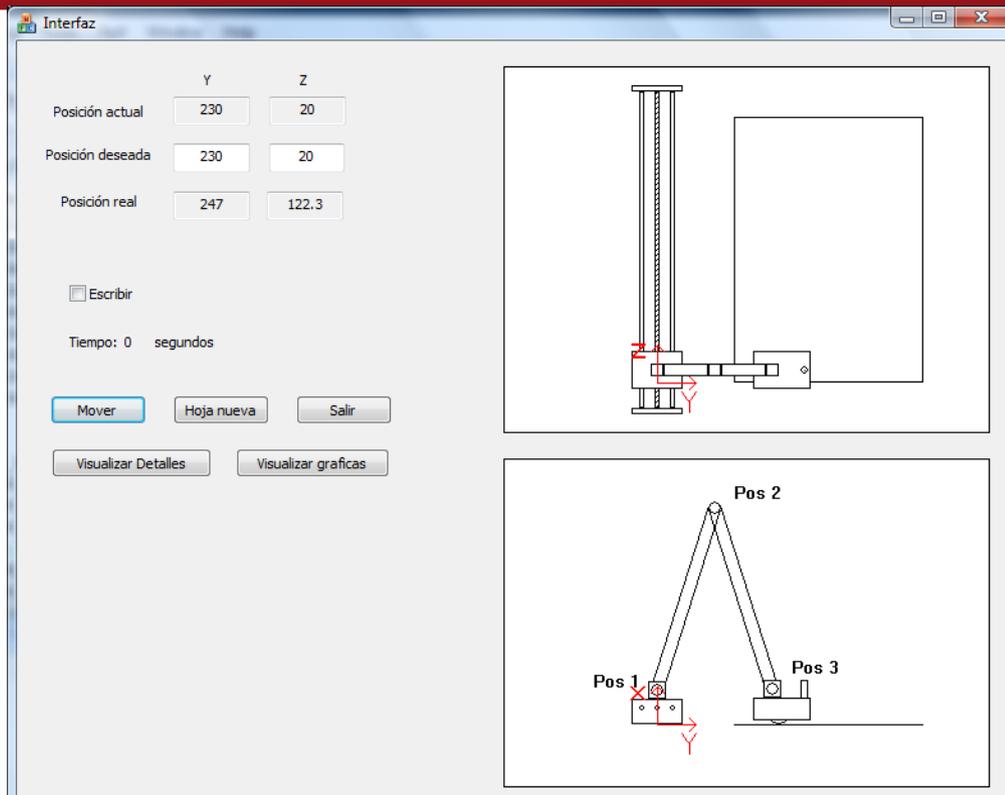


Figura 9. Ventana de diálogo completa.

#### 4.1.4. Cálculo de los parámetros de las funciones

Una vez finalizado el cálculo de los puntos que describen el movimiento del brazo robot se procede a hallar la función continua que describa el movimiento. Esto se hace porque los datos que se tienen hasta este punto son de naturaleza discreta. Para poder estimar los valores de los datos en cualquier instante de tiempo se necesita conocer su función continua.

Una función se define por el valor de sus constantes y de su grado. Para los valores que se manejan, basta utilizar una función de segundo grado para describir la variación de los datos.

$$\text{función} = a_0 + a_1 \times t + a_2 \times t^2 \quad (\text{ec. 11})$$

El valor de  $a_0$  es igual al valor inicial de la variable. Para hallar las constantes  $a_1, a_2$  van variando hasta que la diferencia entre el valor de la función y todos los puntos discretos de la variable sea despreciable.

En el programa se deben de hallar 4 funciones: La que describe el desplazamiento en el eje Z y las que describen la variación de los ángulos 1,2 y 3:

$$\text{ángulo1}(t) = a_0 + a_1 \times t + a_2 \times t^2 \quad (\text{ec. 12})$$

$$\text{ángulo2}(t) = b_0 + b_1 \times t + b_2 \times t^2 \quad (\text{ec. 13})$$

$$\text{ángulo3}(t) = c_0 + c_1 \times t + c_2 \times t^2 \quad (\text{ec. 14})$$

$$\text{posiciónz}(t) = d_0 + d_1 \times t \quad (\text{ec. 15})$$

Para el caso del eje Z, el brazo robot se desplaza a velocidad constante. Por esta razón la función será solamente de primer orden. Ya que la velocidad máxima es de 1mm/segundo, el valor de  $d_1$  varía entre -1 y 1. Se varía el valor de  $d_1$  a razón de 0.1 y se compara con el valor discreto de la posición Z hasta que la diferencia entre ellos sea de 0.1mm.

Para el cálculo de las constantes de los ángulos, las funciones son de segundo orden. De igual forma que en el caso de la posición Z, los índices de las funciones se van variando a razón de 0.0001 porque la variación de los ángulos es mucho menor y se necesita más precisión para hallar sus funciones. Los índices varían entre [-0.2 , 0.2] en el caso de  $a_1$  y  $c_1$  y [-0.3 , 0.3] en el caso de  $b_1$ . Para los índices  $a_2, b_2$  y  $c_2$  su rango es de [-0.1 , 0.1]. Estos límites fueron fijados luego de depurar el programa y ver los valores máximos que tomaban las constantes. Por fines prácticos se fijo un margen de error de  $0.5^\circ$  entre la función continua y los puntos discretos hallados previamente, ya que el tiempo que el programa demora en realizar estos cálculos aumenta considerablemente si se desea más precisión.

Debido al diseño mecánico del brazo, los ángulos 1 y 3 siempre serán iguales. Por esta razón solo es necesario calcular los valores de las constantes del ángulo 1 y luego se copian a la función del ángulo 3.

#### 4.1.5. Visualizar las gráficas de las funciones

##### 4.1.5.1. Visualización por medio de la interfaz gráfica de usuario

El programa es capaz de graficar las funciones en una ventana nueva para que el usuario pueda observarlas. La figura 10 muestra como se ve la ventana que presenta las gráficas.

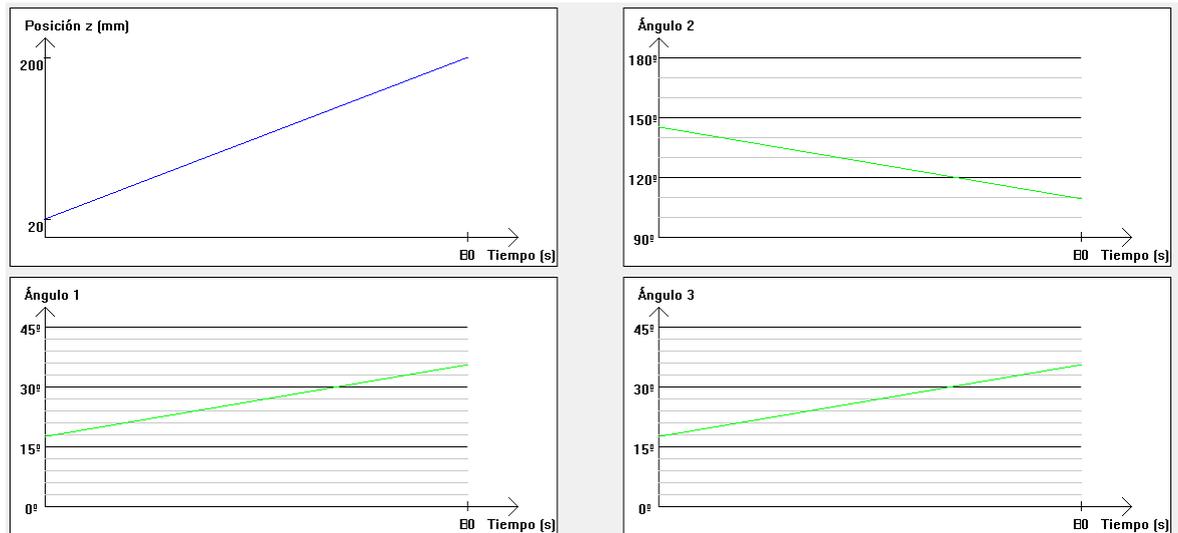


Figura 10. Visualización de las gráficas.

La programación de esta parte de la interfaz se realizó utilizando las instrucciones para realizar dibujos en los cuadros de diálogo del visual C++.[8]

- Se utilizó la instrucción `Rectangle()` para dibujar los marcos con fondo blanco de cada gráfica.
- Para dibujar las líneas y los ejes de las gráficas se utilizaron las instrucciones `MoveTo()` y `LineTo()`.
- Para trazar las gráficas en sí, se utilizó la instrucción `SetPixel()`. Para toda la gráfica, se recorren todos los píxeles que la conforman de inicio a fin.

Para graficar correctamente a la función se debe hacer una conversión entre el valor de tiempo y los píxeles. Esto se logra por a través de la siguiente operación:

$$t_{\text{tiempo}} = (\text{pixel} \times \text{tiempo total}) + \text{número total de pixeles} \quad (\text{ec. 16})$$

El resultado de esta operación es el tiempo que equivale a cada pixel de la gráfica. En el programa desarrollado el número total de pixeles que se utilizan para graficar al eje del tiempo es de 420 pixeles. El tiempo total de la gráfica varía de acuerdo al movimiento realizado.

#### 4.1.5.2. Visualización por medio del programa SMA4

Para visualizar con más detalle las gráficas de las funciones se decidió utilizar el software SMA4 para Windows desarrollado en Japón por Toshio Suzuki, el cual se especializa en graficar funciones. Para que las funciones calculadas en la interfaz sean graficadas por el SMA4 se debe de crear un archivo de tipo .SMP en donde se detallan los parámetros que definen a la gráfica como por ejemplo el tiempo de duración, el rango de la función, el intervalo entre las divisiones mínimas en el eje, el color de la línea y los parámetros de la función.

Los archivos que funcionan con este software tienen un formato ya definido por el autor y debe de ser respetado. Ya que los archivos con los que funciona este programa tienen una gran cantidad de variables, se tomó como base un archivo y se modificaron solo los parámetros necesarios (nombrados previamente) para obtener las gráficas deseadas.

Para obtener el rango que debe de abarcar la función se identifican sus máximos y mínimos, los cuales se pueden hallar por medio de simples comparaciones: Se inicializa el máximo en 0 y el mínimo en máximo valor posible (el cual varía dependiendo del tipo de variable). Luego se van comparando todos los valores de la función:

- Si el valor es mayor que el máximo, este será el nuevo máximo.
- Si el valor es menor que el mínimo, este será el nuevo mínimo.

Luego de conocer los máximos y mínimos se procede a calcular las divisiones mínimas en el eje. Las divisiones serán proporcionales a la diferencia entre el máximo y el mínimo de la función.

Para observar las gráficas con el SMA4, solo se inicia el programa y se presiona el botón de abrir archivo, en la barra de herramientas. Luego se busca en la PC la carpeta del proyecto, en donde están los archivos de la interfaz gráfica de usuario. Finalmente se elige el archivo deseado y se abre (dichos archivos son de tipo .SMP).

A continuación las figuras 11 y 12 ilustran como el programa SMA4 grafica las funciones calculadas.

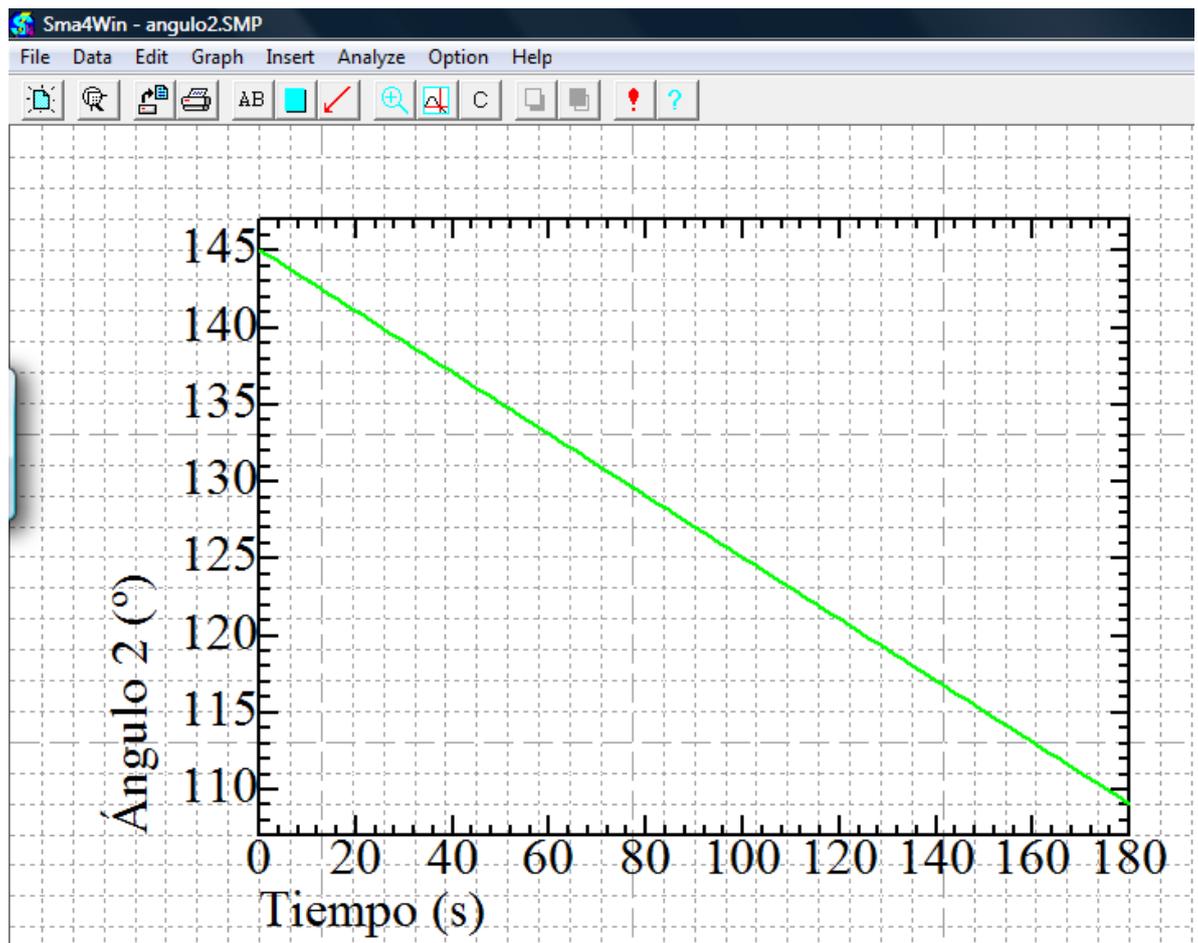


Figura 11. Gráfica del ángulo 2 en el SMA4.

El gráfico de la figura 11 muestra la variación en grados del ángulo 2 en el tiempo. Debido a las características del brazo robot implementado, el rango en el que varía el ángulo 2 es de  $109^\circ$  hasta  $145^\circ$ .

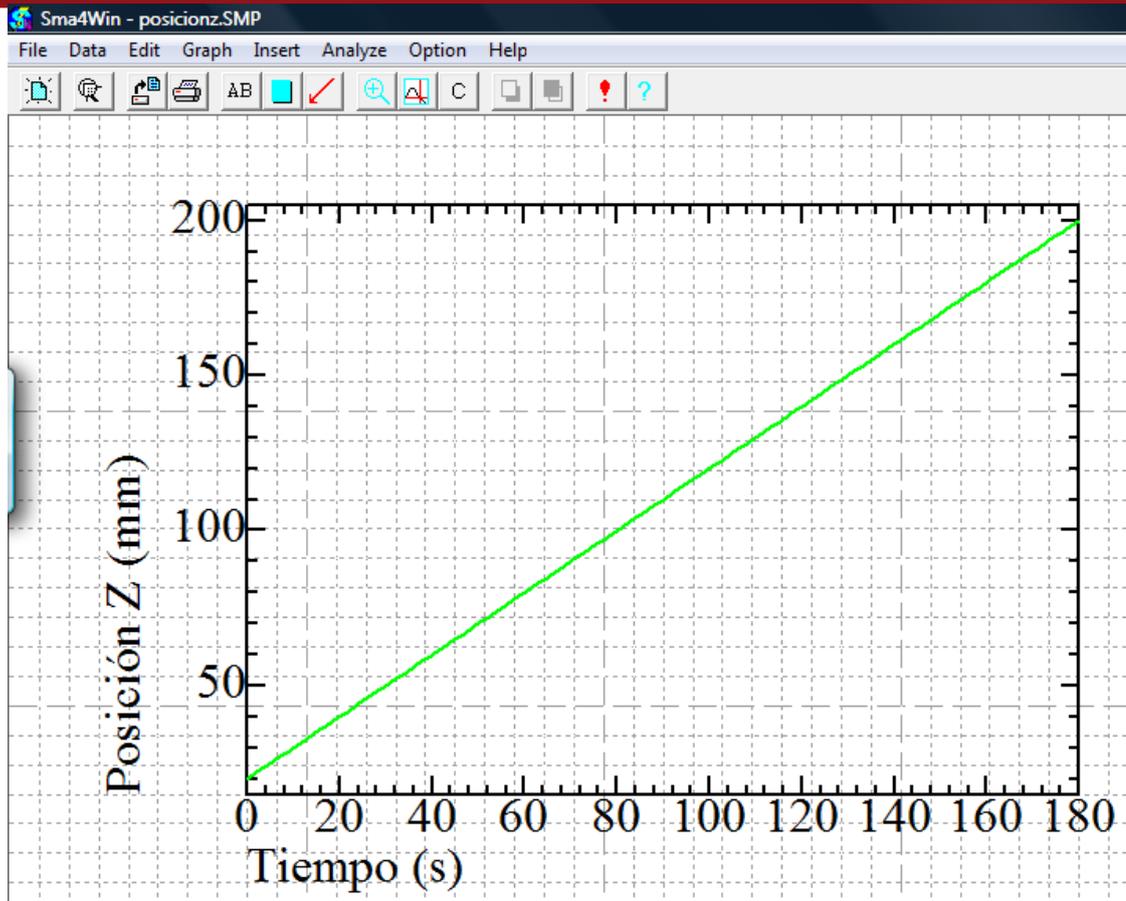


Figura 12. Gráfica de la posición Z en el SMA4.

En el gráfico mostrado en la figura 12, se puede observar que la velocidad de desplazamiento en el eje Z es de 1mm/segundo y se mantiene constante durante todo el trayecto. Además el movimiento se inicia en la posición 20 y finaliza en la 200.

#### 4.1.6. Visualizar detalles

Al pulsar el botón de “Visualizar detalles” se abre una nueva ventana que muestra los valores de todas las posiciones de los ejes del brazo robot y los ángulos entre sus eslabones. Esto se logra simplemente copiando los datos calculados por el programa a la nueva ventana. Además, se identifica el error máximo en el eje Y y el eje Z y se muestra en la ventana. La figura 13 muestra como se muestran los distintos valores en la ventana.



Posición 1		Posición 3		Ángulos	
X	55	X	55	Ángulo 1	17
Y	0	Y	180	Ángulo 2	145
Z	20	Z	20	Ángulo 3	17
Posición 2		Posición 4		Máximo Error	
X	341	X	0	Error Y	0
Y	90	Y	230	Error Z	0
Z	20	Z	20		

Salir

Figura 13. Ventana para visualizar los parámetros del sistema.

#### 4.1.7. Comunicación serial

Se utiliza el puerto serial para que se comuniquen la interfaz gráfica con el microcontrolador ATmega8 del sistema de control de movimiento. Para utilizar el puerto serial programando por visual C++ se debe de crear un archivo con el nombre del puerto que se desea utilizar (en este caso, se utilizará el puerto COM2). Si se desea enviar o recibir datos por el puerto serial, lo que se hace es escribir o leer el archivo respectivamente [10].

A continuación se explicará con mayor detalle el funcionamiento de esta parte del programa:

##### 4.1.7.1. Abrir el puerto serial

Se declaran las variables hSerial de tipo HANDLE y para abrir el puerto serial, se ejecuta la siguiente instrucción:

```
hSerial = CreateFile(m1,
    GENERIC_WRITE | GENERIC_READ,
    0,
    0,
    OPEN_EXISTING,
```

```
FILE_ATTRIBUTE_NORMAL,
    0);
```

- El primer parámetro de la instrucción CreateFile, m1, contiene el nombre del archivo a crear. Ya que se desea abrir el puerto 2, la variable m1 será igual a "COM2".
- El segundo parámetro indica que si se va a enviar o recibir información por el puerto serial. Para el presente proyecto se requiere leer y escribir, así que se declaran ambas funciones.
- El tercer parámetro se fija en 0 para que el puerto que está siendo utilizado por el programa (COM2) no pueda ser utilizado por ningún otro proceso.
- El cuarto parámetro se fija en 0 e indica que el handle que retorna de la función CreateFile no puede ser heredado por otros procesos menores.
- El quinto parámetro indica que solo abra un archivo ya existente. Para la presente aplicación, el archivo ya debe de existir porque se trata del puerto 2 utilizado para la comunicación serial.
- El sexto parámetro, FILE\_ATTRIBUTE\_NORMAL le indica que Windows trate al archivo como un archivo normal, sin ningún detalle extra.
- El último parámetro de la función se fija en 0, ya que al abrir un archivo la función CreateFile ignora este último parámetro.

La instrucción CreateFile además de crear un archivo, le da un valor al archivo hSerial (que es de tipo handle). En caso de que no pueda abrir el puerto COM2, hSerial tendrá un valor inválido (INVALID\_HANDLE\_VALUE) y el programa mostrará un mensaje de error:

```
if(hSerial==INVALID_HANDLE_VALUE)
{
    if(GetLastError()==ERROR_FILE_NOT_FOUND)
    {
        m1="Error al abrir el puerto";
        MessageBox(m1,m2,MB_ICONEXCLAMATION);
    }
}
```

#### 4.1.7.2. Configurar el puerto serial

Para que se pueda realizar una comunicación exitosa, la configuración de entre el puerto serial de la computadora y la del ATmega8 deben ser exactamente iguales. Dichos parámetros son:

- Tasa de transmisión: Es la velocidad a la que se transmiten los bits por el puerto serial en baudios.
- Tamaño de carácter: Es el número de bits por carácter transmitido.
- Número de bits de parada: Es el número de bits que separan a cada byte transmitido.
- Paridad: Es una forma para controlar errores en la comunicación serial y se trabaja con la sumatoria de los bits transmitidos.

Para que se configure el puerto serial, Windows utiliza una estructura llamada DCB. Se declara una variable `dcbSerialParams` de tipo DCB. A continuación se muestra la porción de código responsable de la configuración del puerto serial:

```
dcbSerialParams.DCBlength=sizeof(dcbSerialParams);
if(!GetCommState(hSerial, &dcbSerialParams))
{
    m1="Error al obtener el estado del puerto";
    MessageBox(m1,m2,MB_ICONEXCLAMATION);
}
dcbSerialParams.BaudRate=CBR_9600;
dcbSerialParams.ByteSize=8;
dcbSerialParams.StopBits=ONESTOPBIT;
dcbSerialParams.Parity=ODDPARITY;
if(!SetCommState(hSerial, &dcbSerialParams))
{
    m1="Error al configurar el puerto";
    MessageBox(m1,m2,MB_ICONEXCLAMATION);
}
```

Primero se fija el tamaño de los parámetros para la configuración del puerto serial. Luego se obtiene el estado actual del puerto serial por medio de la instrucción `GetCommState()`. Si ocurre un error con esta instrucción, se mostrará el mensaje del error correspondiente.

Luego se procede a modificar solo los parámetros que deseamos fijar. Se fija la tasa de transmisión a 9600 baudios, 8 bits por carácter, 1 bit de parada y

paridad impar. Finalmente con la instrucción SetCommState() se configura el puerto serial con los parámetros fijados previamente. En caso de que ocurra algún error, se mostrará un mensaje señalándolo.

#### 4.1.7.3. Configurar los tiempos del puerto serial

Uno de los principales inconvenientes de la comunicación con el puerto serial ocurre al momento de recibir algún dato. ¿Cómo sabe el programa cuando debe de leer el puerto serial? La siguiente sección del programa le indica a la computadora cuanto tiempo debe de esperar para leer el puerto serial. Esto se logra utilizando las siguientes instrucciones:

```

COMMTIMEOUTS timeouts={0};
timeouts.ReadIntervalTimeout=250;
timeouts.ReadTotalTimeoutConstant=250;
timeouts.ReadTotalTimeoutMultiplier=10;
timeouts.WriteTotalTimeoutConstant=50;
timeouts.WriteTotalTimeoutMultiplier=10;
if(!SetCommTimeouts(hSerial, &timeouts))
{
    m1="Error al configurar los tiempos del puerto";
    MessageBox(m1,m2,MB_ICONEXCLAMATION);
}

```

- ReadIntervalTimeout indica cuanto tiempo en milisegundos debe de esperar para leer el puerto serial.
- ReadTotalTimeoutConstant indica el tiempo que tarda en retornar.
- ReadTotalTimeoutMultiplier indica cuanto tiempo adicional debe de esperar antes de retornar por cada byte recibido.
- WriteTotalTimeoutConstant y WriteTotalTimeoutMultiplier funcionan de la misma forma, pero con la única diferencia que se aplican para el caso de escritura en el puerto serial.

Como en los casos anteriores, si se detecta un error al configurar los tiempos del puerto se mostrará un mensaje de error detallando el tipo de falla.

Se puede utilizar esta forma de recibir datos por el puerto serial, ya que se conoce el tiempo en el que se deberían de recibir los datos por el puerto. En

caso de que este tiempo no sea conocido, se debe de realizar un programa más complejo que detecte la aparición de datos en el puerto serial.

#### 4.1.7.4. Escritura y lectura del puerto serial

Una vez configurado el puerto serial, este se encuentra listo para enviar y recibir datos. La escritura y lectura del puerto serial se hace por medio de las instrucciones WriteFile() y ReadFile() respectivamente:

```
DWORD dwBytesRead = 0;  
char dato[1];  
  
WriteFile(hSerial,dato, 1, &dwBytesRead, NULL);  
ReadFile(hSerial,dato,1, &dwBytesRead, NULL);
```

Los parámetros de entrada de las funciones WriteFile() y ReadFile() son exactamente iguales y se describirán con mayor detalle a continuación:

- El primer parámetro es el archivo de tipo HANDLE hSerial que fue creado al abrir el puerto serial.
- El segundo parámetro es un arreglo de tipo char en donde se almacenan los datos recibidos (en caso de que se lea el puerto serial) o los datos a ser enviados (en caso de escribir en el puerto serial).
- El tercer parámetro indica el número de bytes a leer o escribir.
- El cuarto parámetro es un puntero que apunta hacia un entero que indica el número de bytes leídos.
- El quinto parámetro se fija en NULL, excepto cuando se trabaja con una comunicación serial asíncrona.

Para que el programa intercambie información con el sistema de control de movimiento, primero se debe de enviar un carácter que indique que variable se va a enviar.

- Si se va a enviar el ángulo entre los eslabones se transmite el carácter "A" (número 65 en el código ASCII).
- Si se va a enviar la posición en el eje Z se transmite el carácter "B" (número 66 en el código ASCII).

Luego de enviar este primer carácter, se recibe el valor real de la variable desde el ATmega8. Los valores recibidos por el puerto serial deben de ser adaptados a las unidades físicas que el programa utiliza (milímetros).

Para controlar el desplazamiento en el eje Z en el brazo robot se utiliza un sensor óptico infrarrojo, con el que se cuentan las revoluciones del tornillo sin fin utilizado para desplazar a la estructura. Cada vuelta del tornillo sin fin es igual a 1.55 milímetros de desplazamiento. Se debe de considerar un offset de 20 al hacer la conversión entre la posición Z y la cuenta de las vueltas del tornillo, ya que la posición inicial en el eje Z es de 20 milímetros desde el borde de la hoja. La fórmula utilizada para convertir la cuenta del sensor en desplazamiento en milímetros es la siguiente:

$$\text{PosiciónZ} = \# \text{ de vueltas} \times 1.55 + 20 \quad (\text{ec. 17})$$

Para medir el desplazamiento en el eje Y se colocó un potenciómetro en el eje que une a los eslabones del brazo y mide la variación del ángulo entre los eslabones. La variación del ángulo ingresa al ATmega8 como una señal analógica que es leída por el convertidor análogo/digital. Lo que debe de hacer el programa es convertir la posición en el eje Y su el valor correspondiente a la salida del convertidor ADC. Por medio de mediciones con el brazo robot, se hizo una tabla que relaciona la posición en el eje Y con el valor obtenido del potenciómetro. Para los valores intermedios entre los puntos medidos, se utilizó una aproximación lineal. A continuación se mostrará en la tabla 1 con los valores medidos experimentalmente.

Tabla 1. Posición Y (mm) y la medición digital del ángulo 2.

Posición Y (mm)	Salida del convertidor análogo digital
230	48
240	58
250	67
260	75
270	84
280	96
290	107
300	118
310	130
320	142
330	153
340	166
350	178
360	192
370	203
380	217
390	231
400	245

Adicionalmente, se comparan los valores teóricos del movimiento y los recibidos por el puerto serial y las diferencias (error estimado) de los ejes Y y Z se almacenan en la computadora.

#### 4.1.7.5. Cerrar el puerto serial

Para cerrar el puerto serial solo se utiliza una instrucción, la cual es la siguiente:

```
CloseHandle(hSerial);
```

El único parámetro de entrada de esta instrucción es el archivo HANDLE utilizado para abrir el puerto serial.

#### 4.1.8. Programa principal

El funcionamiento del programa principal está descrito en el diagrama de flujo presentado en la figura 14.

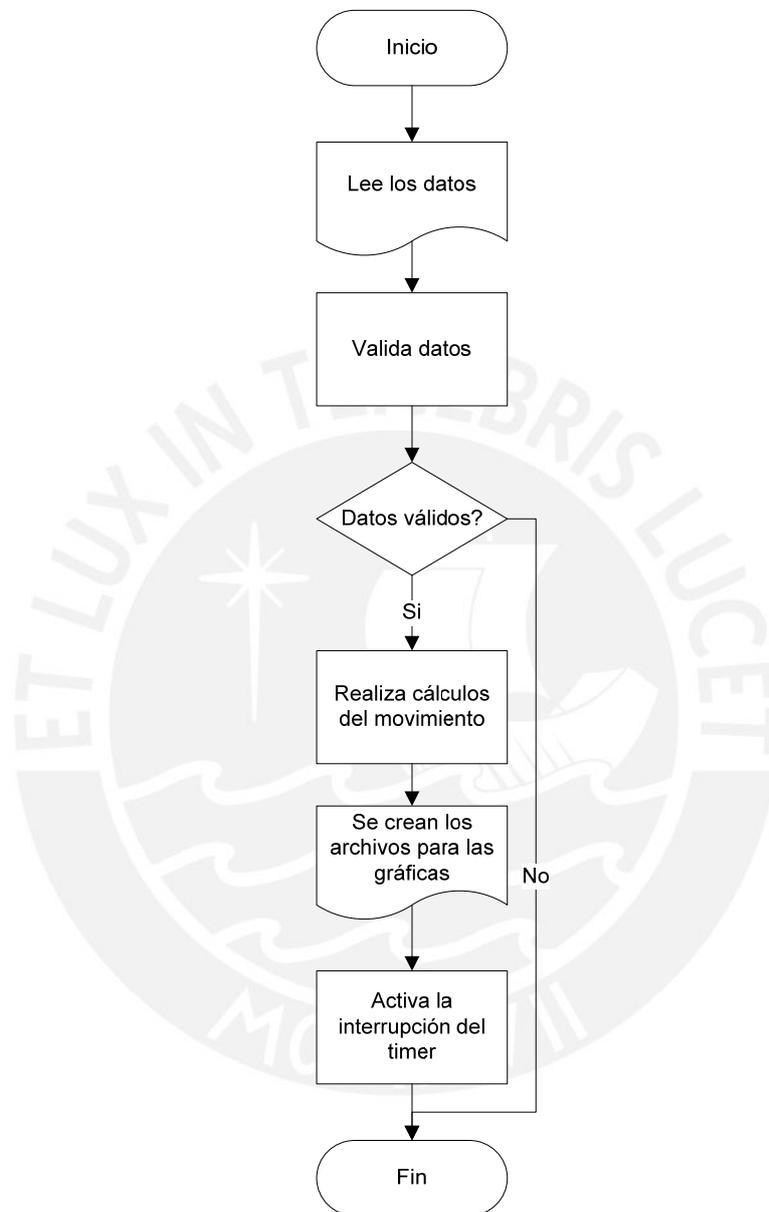


Figura 14. Diagrama de flujo del programa principal.

El programa se ejecuta cuando se presiona el botón “Mover” de la interfaz gráfica de usuario. Primero lee las posiciones ingresadas por el usuario y verifica que sean válidas. La posición Z debe de estar entre 20 y 400; mientras que la posición Y debe de estar entre 230 y 400. En caso de que alguna de ellas sea una posición inválida, se envía un mensaje de error al usuario.

En caso de que las posiciones sean válidas, se procede a realizar los cálculos de los puntos intermedios por donde pasa el efector final, los cálculos de la cinemática inversa y el de los índices de las funciones. Una vez realizados estos cálculos se crean los archivos que sirven para visualizar las gráficas por medio del programa SMA4. Finalmente se activa la interrupción por el timer, la cual se ejecutará cada segundo.

A continuación se explicará el funcionamiento de la subrutina de la interrupción por tiempo a través de la figura 15.



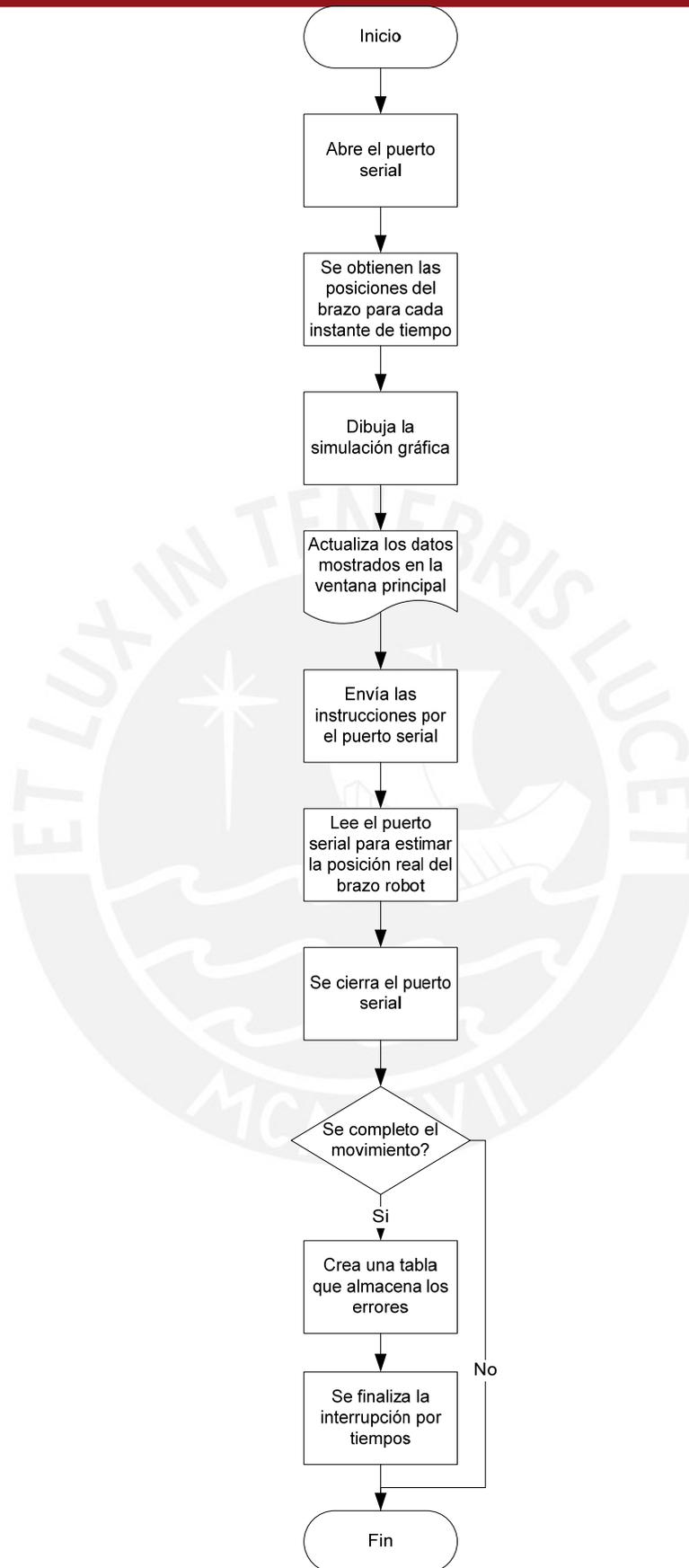


Figura 15. Diagrama de flujo de la interrupción por tiempos.

Cada segundo se grafica la simulación del brazo robot y así se da la sensación de movimiento del brazo. Además se actualiza el tiempo que ha transcurrido desde que se inició el movimiento, la posición deseada del puntero en el plano y la posición real del brazo (recibida por el puerto serial desde el sistema de control de movimiento). Se cierra el puerto serial y se verifica si ya se finalizó el movimiento. De ser así se desactiva la interrupción por tiempo, se crea una tabla en un archivo "error.txt" que almacena los errores del movimiento en una tabla y el programa está listo para recibir el siguiente movimiento. Si el movimiento aún no ha finalizado, se espera hasta la siguiente interrupción y se ejecuta nuevamente esta parte del programa.

#### 4.2. Diseño de la etapa de control y de adquisición de señales

Esta etapa debe encargarse de la comunicación con el software de la computadora y recibir de él las posiciones deseadas para cada eje. También debe ser capaz de responder al software de la computadora con las posiciones actuales en cada instante para así poder registrar el error. Además debe encargarse de adquirir las señales de ángulo relativo entre los dos eslabones y del desplazamiento en el eje Z para realizar las acciones correctivas hasta asegurarnos que lleguen al punto deseado. Estas acciones correctivas serán realizadas actuando sobre los motores encargados de transmitir el movimiento mediante los circuitos excitadores de cada motor.

Por lo explicado anteriormente se identifican tres bloques principales:

- Adquisición de señales.
- Procesamiento y generación de las señales de control.
- Comunicación con el software de la computadora.

La Figura 16 muestra el diagrama de bloques de la solución propuesta.

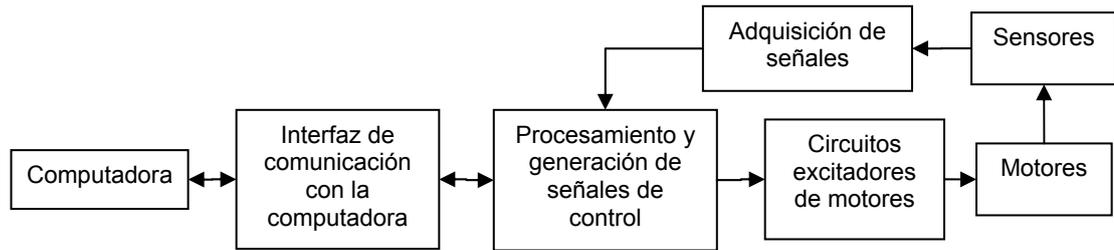


Figura 16. Diagrama de bloques de la solución propuesta

A continuación se detallará el diseño de los bloques de “Adquisición de señales”, “Procesamiento y generación de señales de control” y “la Interfaz de comunicación con la computadora”.

#### 4.2.1. Adquisición de señales

El sistema está diseñado para actuar en función de los ángulos entre los eslabones y el desplazamiento lineal en el eje z, por ello estas son las principales magnitudes a medir.

##### 4.2.1.1. Adquisición del desplazamiento lineal en el eje z

El sistema cuenta con una pequeña pestaña negra acoplada al tornillo sinfín que transmite el movimiento al efector. El desplazamiento lineal es de 1.55mm por vuelta del tornillo sin fin.

La solución propuesta para adquirir la señal del desplazamiento lineal es la siguiente:

- Instalación de un interruptor óptico ranurado activado por la pestaña acoplada al tornillo sinfín. La salida del sensor nos brindará pulsos cada vez que se complete una vuelta, y esta señal podrá ser usada en un microcontrolador para contar el desplazamiento

El interruptor óptico seleccionado es el OPB940 (Figura 17) que es un interruptor óptico con un amplificador con histéresis interno, y nos ofrece las siguientes prestaciones:

- Emisor y receptor en un mismo empaque.

- Voltaje de alimentación del circuito de salida igual a 5V.
- Salida TTL.
- Amplificador interno con lazo de histéresis.
- Empaque opaco para evitar interferencia de fuentes infrarrojas externas.



Figura 17. Interruptor óptico OPB940. Imagen tomada de su hoja de datos.

Este dispositivo se montará en el sistema de tal manera que la pestaña acoplada al tornillo sinfín sea la que, al pasar por la ranura del dispositivo, genere los pulsos que permitirán llevar la cuenta de las vueltas que se han completado para así conocer el desplazamiento lineal.

El OPB40 cuenta con 5 terminales, las cuales se detallan a continuación:

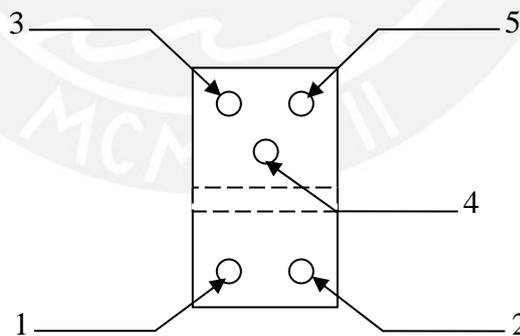


Figura 18. Distribución de pines del OPB940.

Tabla 2. Descripción de pines del OPB940.

# de Pin	Descripción
1	Ánodo
2	Cátodo
3	Alimentación
4	Salida
5	Tierra

El voltaje de alimentación debe ser de 5V según las especificaciones del fabricante. Así mismo, la corriente máxima es de 40mA entre ánodo y cátodo del Led emisor.

El circuito para poner en funcionamiento el OPB940 es el mostrado en la Figura 19.

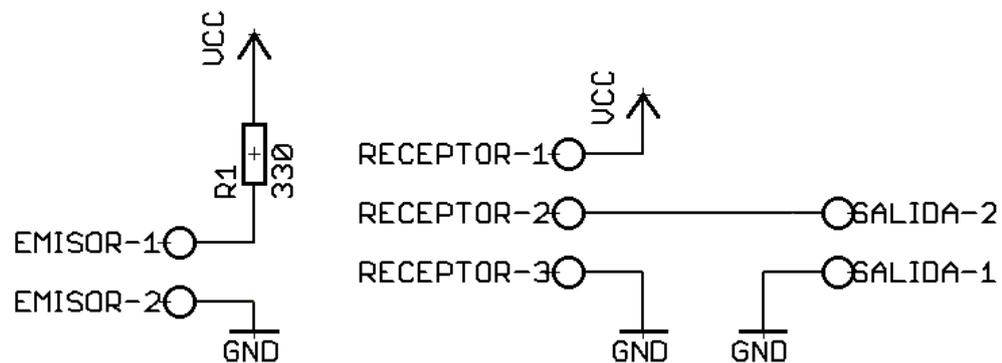


Figura 19. Circuito de alimentación del OPB940

Tabla 3. Descripción de entradas y salidas del circuito de alimentación del OPB940.

Conector	Descripción
Emisor1	Ánodo
Emisor2	Cátodo
Receptor1	Alimentación
Receptor2 / Salida2	Salida
Receptor3 / Salida1	Tierra

En la Tabla 3 se especifica que terminal del OPB40 debe ir conectado a cada conector del circuito de la Figura 19.

La salida de este circuito generará pulsos por cada vuelta del tornillo sinfín, lo que se empleará para contar el desplazamiento lineal del eje Z.

#### 4.2.1.2. Adquisición de la señal de ángulos relativos

Se tienen en cuenta los siguientes criterios:

- El ángulo entre eslabones va a estar comprendido entre  $20^\circ$  y  $72^\circ$  sexagesimales, lo cual nos da un rango de trabajo de  $52^\circ$  sexagesimales.
- La precisión del sensor va a tener una resolución máxima determinada por la capacidad de actuación del sistema de desplazamiento.
- El costo del sistema de adquisición de señales debe ser reducido.

Bajo estas premisas se selecciona el uso de resistores variables rotatorios de respuesta lineal, puesto que ofrecen las siguientes prestaciones:

- Rango de trabajo:  $270^\circ$  sexagesimales de giro de la perilla.
- Respuesta altamente lineal en todo su rango de trabajo.
- Costo reducido.

La señal de salida en la terminal variable de un resistor variable conectado a una referencia de voltaje fija es un voltaje proporcional al ángulo al cual se rote la perilla. Es esta característica la cual se usará para medir el ángulo entre eslabones, fijando el cuerpo del resistor variable a un eslabón y su perilla rotatoria al siguiente eslabón mediante el eje de unión entre eslabones.

El potenciómetro será alimentado con +5V. El rango de trabajo del potenciómetro en la aplicación dada (con una alimentación de +5V) corresponde a una variación de 1V. Esto hace necesario el diseño e implementación de una etapa intermedia de acondicionamiento de la señal

adquirida del potenciómetro antes de ingresar al convertidor análogo a digital.

#### 4.2.1.3. Conversión de la señal analógica a digital

Como se vio anteriormente, la respuesta del resistor variable para medir ángulos relativos será en voltaje. Para poder trabajar esta variable analógica en un microcontrolador, que será usado en la etapa de generación de señales de control, se debe convertir a digital. Para ello se empleará un convertidor analógico-digital que deberá cumplir con los siguientes requerimientos:

- El convertidor debe trabajar en un rango de voltaje fijo, con una referencia asignada y fija, lo suficientemente estable para dotar de la precisión requerida al sistema.
- El convertidor debe brindar una respuesta lo suficientemente estable para simplificar las correcciones por software y evitar que las acciones correctivas se efectúen de manera brusca sobre el sistema.
- El tiempo de conversión debe ser lo suficientemente rápido como para calcular el ángulo relativo entre los eslabones antes de efectuar la siguiente acción correctiva.

Teniendo en cuenta los requerimientos anteriormente mencionados, se propone el uso del convertidor analógico digital interno del microcontrolador empleado para la generación de las señales de control y para la comunicación con el software, ya que actualmente muchos convertidores incluidos en los microcontroladores disponibles en el mercado local ofrecen las prestaciones necesarias:

- Ofrecen la posibilidad de seleccionar el voltaje de alimentación del microcontrolador como voltaje de referencia.
- Ofrecen muchas prestaciones para la eliminación de ruido y ofrecen la posibilidad de ser manejados mediante interrupciones.
- Al estar incluido en el microcontrolador se reduce el costo total de la implementación.

Por lo expuesto anteriormente se tendrá en cuenta el seleccionar un microcontrolador que tenga un convertidor análogo-digital que cumpla con estos requerimientos al seleccionar el microcontrolador encargado de generar las señales de control y de la comunicación.

#### 4.2.1.4. Acondicionamiento de la señal adquirida del potenciómetro

La señal del ángulo relativo adquirida sin acondicionamiento alguno estará en un rango de salida limitado, puesto que solo se trabaja en  $52^\circ$  de los  $270^\circ$  que ofrece el resistor variable rotatorio, y el rango de entrada de la señal del convertidor análogo digital será limitado en su rango de trabajo también. Es por ello que se hace necesaria una etapa de acondicionamiento de la señal del ángulo relativo para eliminar la diferencia del valor mínimo con respecto a cero y multiplicar la señal por una ganancia para así aprovechar todo el rango de trabajo del convertidor análogo digital.

Por lo anteriormente expuesto, los requerimientos de esta etapa son:

- Eliminar el corrimiento de la señal respecto al nivel de 0V (Tierra o GND, por su traducción en inglés “Ground”).
- Multiplicar la señal por una ganancia tal que la señal de salida abarque todo el rango de trabajo del convertidor análogo digital para así aprovechar al máximo su resolución.

Tomando como premisas los requerimientos anteriormente expuestos se propone lo siguiente:

- Diseño e implementación de la etapa de acondicionamiento mediante un circuito sumador y amplificador de voltaje basado en amplificadores operacionales.
- Amplificar la señal proporcional al ángulo relativo hasta obtener como valor máximo el voltaje de referencia del convertidor análogo digital.

El diseño propuesto es el que se muestra en la Figura 20, donde se muestran las alimentaciones y las entradas y salidas, identificadas como IN y OUT respectivamente.

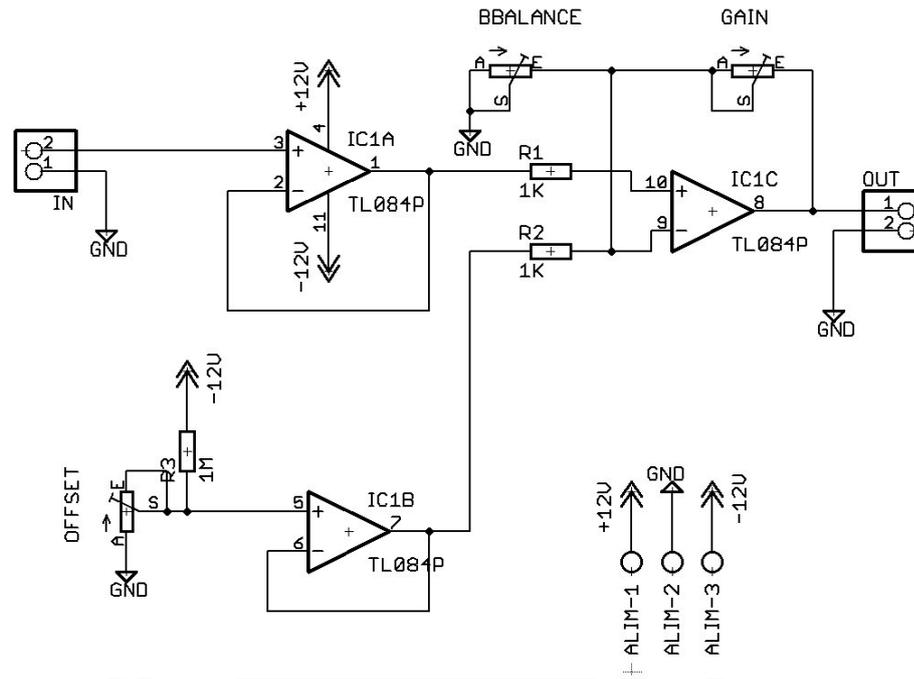


Figura 20. Circuito de acondicionamiento de señal.

El circuito diseñado está conformado por las siguientes etapas:

**Sumador no inversor de voltaje:** Circuito obtenido empleando un amplificador operacional configurado como un sumador no inversor. A la señal de entrada “Vin1” (voltaje proporcional al ángulo relativo) se le suma un voltaje de  $-0.01V$  “Vin2” para eliminar el corrimiento respecto de GND. Este corrimiento se presenta debido a los  $20^\circ$  de giro respecto de la posición inicial de la perilla del potenciómetro. Este desplazamiento inicial de  $20^\circ$  se fija para evitar los cambios abruptos que se presentan en el momento en que se comienza a girar la perilla desde  $0^\circ$ . Empleando esta configuración del amplificador operacional se puede también amplificar cada señal a ser sumada por un factor configurable (sin llevar la salida a alguno de sus dos niveles de saturación). En la configuración empleada estos factores de amplificación son los mismos e iguales a uno. Esta etapa se puede apreciar en la Figura 21.

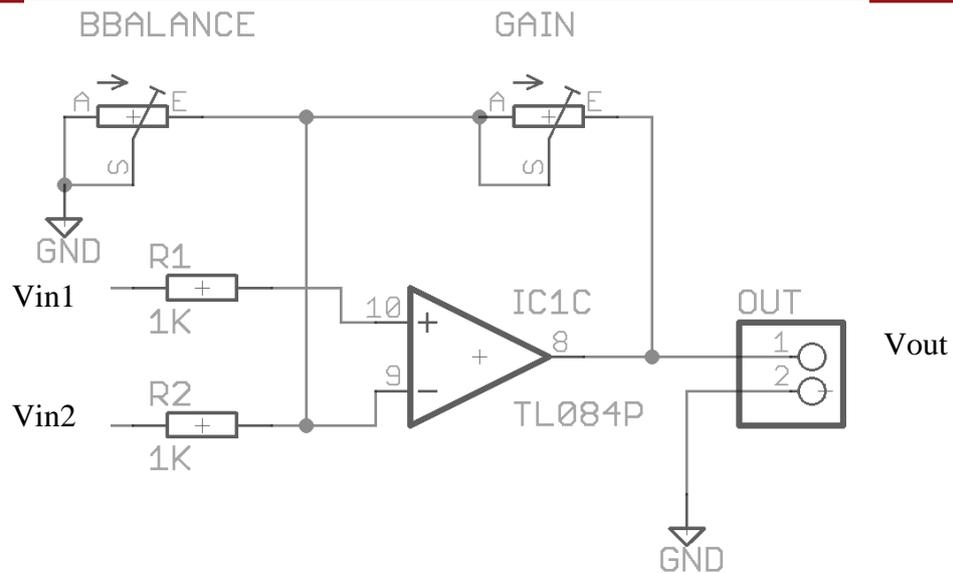


Figura 21. Etapa sumadora no inversora.

Las ecuaciones correspondientes al circuito propuesto son las siguientes:

$$V_{out} = (GAIN/R1) * V_{in1} + (GAIN/R2) * V_{in2} \quad (\text{ec. 18})$$

Esto se cumple si se cumple con la siguiente restricción:

- El valor del paralelo de las resistencias BBALANCE y GAIN debe ser igual al paralelo de R1 con R2. Esto es para asegurar la estabilidad del circuito.

Se tiene así pues que los valores de las resistencias R1 y R2 se fijan iguales a 1K para así regular la ganancia simplemente con el potenciómetro de precisión denominado GAIN. El valor del potenciómetro de precisión BBALANCE se ajusta para asegurar que se cumpla con la restricción para asegurar la estabilidad del circuito.

Los terminales no usados de los potenciómetros se cortocircuitan para evitar problemas de ruido del ambiente de trabajo.

**Voltaje de referencia:** El voltaje de  $-0.01V$  a ser sumado con la señal de entrada se obtiene mediante un divisor resistivo entre una resistencia de  $1M\Omega$  y un potenciómetro en serie alimentados con  $-12V$  (R3 y OFFSET respectivamente en la Figura 22). El valor del potenciómetro de precisión es

ajustado hasta obtener una diferencia de potencial de  $-0.01\text{V}$  entre el potenciómetro y tierra.

**Seguidor de voltaje:** Para evitar el acoplamiento de impedancias del circuito del voltaje de referencia con el sumador de voltaje se añade una etapa intermedia conformada por un amplificador operacional configurado como seguidor de voltaje. Esta etapa es de suma importancia puesto que sin ella se variaría tanto el voltaje de referencia obtenido del divisor resistivo como el factor de amplificación de la etapa sumadora correspondiente a este mismo voltaje y la señal obtenida a la salida no sería la deseada.

La figura 22 muestra el divisor resistivo conectado a la entrada positiva del seguidor de voltaje.

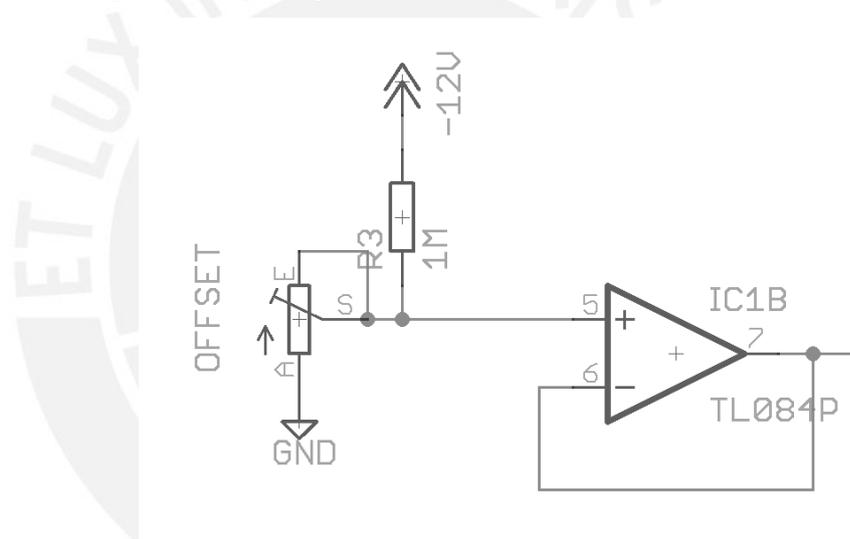


Figura 22. Obtención del voltaje de referencia y seguidor de voltaje.

Se coloca otro seguidor de voltaje entre la entrada del sumador de voltaje y la entrada de la señal de voltaje proporcional al ángulo relativo, tal y como se muestra en la Figura 23, para evitar el acoplamiento de impedancias.

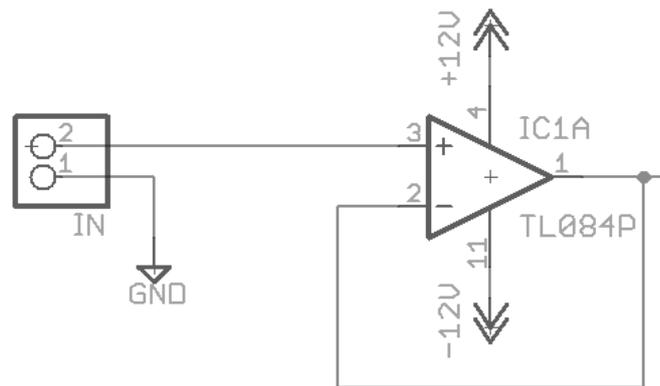


Figura 23. Seguidor de voltaje a la entrada del circuito de acondicionamiento.

#### 4.2.2. Generación de las señales de control y comunicación con el software de la computadora

Esta etapa debe ser capaz de realizar las siguientes tareas:

- Controlar el motor a pasos encargado del movimiento en el eje Z.
- Controlar el motor de corriente continua encargado del movimiento en el eje Y.
- Leer los datos del potenciómetro para efectuar las correcciones necesarias.
- Llevar la cuenta del número de vueltas del tornillo sinfín y efectuar las correcciones necesarias.
- Comunicarse con el software en la computadora.

Para realizar todas las tareas mencionadas se propone el uso de un microcontrolador que además, por lo visto en el punto 4.2.1.3, debe contar con un convertidor análogo digital. Además, por fines prácticos, se elegirá un microcontrolador que cuente con dispositivo de comunicación serial incorporado.

Por las razones anteriormente expuestas se propone el uso del microcontrolador ATmega8 de la empresa ATMEL, que cuenta con 23 líneas programables de entrada/salida, un dispositivo de comunicación emisor-receptor serial síncrono y asíncrono configurable USART, un temporizador

de 16 bits con modo de comparación, un convertidor análogo digital de 10 bits y dos líneas de interrupciones externas, además de otras características que no serán utilizadas en este diseño.

Para realizar la tarea de emitir las señales de control del motor a pasos se necesitan 4 salidas, una para cada controlar cada bobina del motor, las cuales se controlaran mediante el circuito excitador del motor a pasos. Estas salidas estarán asignadas a los pines PB0:3.

Así mismo, para controlar el motor DC se necesita de dos salidas como mínimo, una para encender o apagar el motor y otra para indicar la dirección del mismo. Estas salidas serán las entradas del circuito excitador diseñado para el motor DC y serán asignadas a los pines PB4:5.

La realimentación se dará, como se mencionó en el punto 4.2.1.2, mediante el potenciómetro empleado para el caso del ángulo entre los dos eslabones y mediante un sensor óptico infrarrojo para el caso del tornillo sinfín. Para el caso del potenciómetro, como la señal es de tipo analógico y según se explico en el punto 4.2.1.3 se debe emplear el convertidor análogo digital. La salida del circuito acondicionador de señal será conectada a la terminal central del conector denominado POT0, donde también se encuentran la alimentación y la conexión a 0V o tierra para el potenciómetro. El pin AREF se conecta a tierra mediante el condensador C5 de 10uF para eliminar el ruido que pueda estar presente en el ambiente y que pudiera afectar el resultado de la conversión.

Para el caso del sensor óptico infrarrojo se emplea la línea de interrupción externa PD2, para llevar la cuenta del número de vueltas que ha completado el tornillo sin fin, lo cual se traduce en el desplazamiento lineal del efector, según lo explicado en el punto 4.2.1.1.

#### **4.2.2.1. Circuito de conversión de niveles de voltaje para la comunicación serial**

Para la etapa de comunicación con el software de la computadora se empleará el dispositivo interno USART, siglas de su nombre en inglés

“Universal Synchronous and Asynchronous serial Receiver and Transmitter” o transmisor y Receptor serial síncrono y asíncrono serial universal. Para que este dispositivo se pueda comunicar con la computadora mediante el puerto serial se debe cumplir con el estándar RS232, que es un estándar de conexión entre equipos para la transmisión de señales binarios de manera serial. Este estándar especifica entre otras cosas los niveles de voltaje necesarios para la comunicación entre equipos que cumplan con el mismo. Para realizar esta conversión de niveles de voltaje se empleará el circuito integrado MAX232, que cuenta con dos módulos de transmisión y dos de recepción, de los cuales solo se usará uno de cada uno.

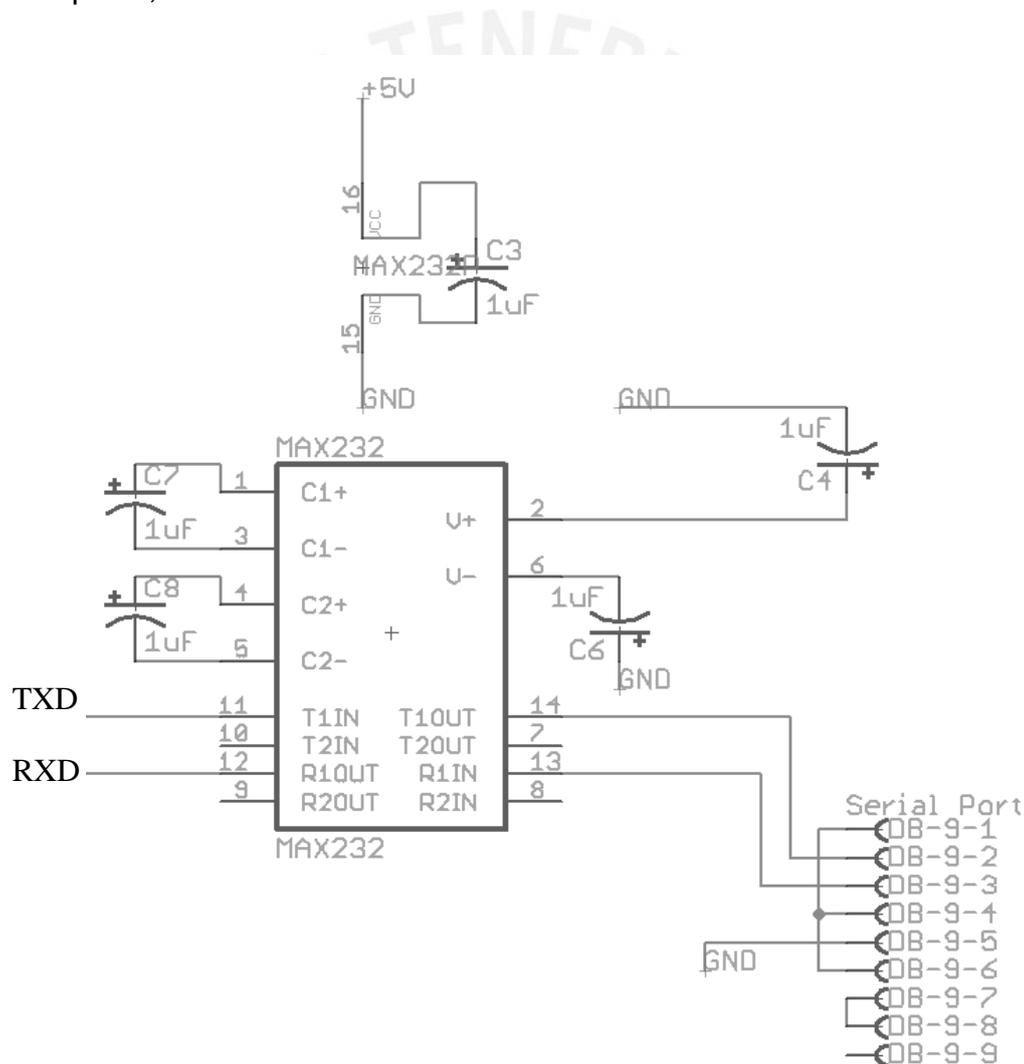


Figura 24. Diagrama esquemático de conexiones del MAX232.

El circuito propuesto para realizar esta conversión es el presentado en la Figura 24. Se puede apreciar que el integrado MAX232 se alimenta con +5V

y se añade el condensador C3 de 1 $\mu$ F para eliminar el ruido que se pueda presentar en la alimentación, y a partir de este voltaje se obtienen los niveles de +/- 10V. Para realizar esta conversión el MAX232 tiene internamente dos fuentes conmutadas, la primera de ellas en conjunto con los condensadores electrolíticos C4 y C7 adaptan el nivel de voltaje tomado de la alimentación de +5V a +10V, la segunda fuente conmutada y los condensadores electrolíticos C6 y C8 invierten los niveles de voltaje para así obtener -10V, estos niveles de voltaje son utilizados para realizar la adaptación de los voltajes RS232 y se encuentran dentro de los rangos permitidos por el estándar RS232.

Los pines T1IN y R2UOT del MAX232 se conectan a los pines TXD y RXD del microcontrolador ATmega8 respectivamente, encargados de la comunicación serial, y a través de los pines T1OUT y R1IN, mediante el conector DB9 denominado Serial Port en la Figura 24, se conecta al puerto serial de la computadora para así realizar la comunicación con el software.

Se observa además que los terminales 1,4 y 6 del conector DB9 están cortocircuitados, así como los terminales 7 y 8 del mismo conector. Estas conexiones se realizan en caso sea necesario el trabajo con algún método de control de flujo de parte de la computadora, pero en caso de no usarse el control de flujo estas conexiones no afectan la comunicación.

#### **4.2.3. Circuito Regulador de voltaje de alimentación de la tarjeta de control y comunicación**

En la Figura 25 se presenta el circuito encargado de regular el voltaje de alimentación de la tarjeta de control y comunicación, conformado principalmente por un regulador de voltaje LM7805. Este regulador es usado para obtener un voltaje estable de +5V a partir de un voltaje de +9V que se obtiene de una fuente externa y se conecta mediante el conector DCJ0202. Este circuito requiere ser alimentado con un voltaje entre 7V y 20V y a la salida presentara un voltaje estable de +5V, eliminando así las posibles fluctuaciones a la entrada.

A la entrada del condensador se coloca un diodo de protección contra corriente inversa, para que en caso se alimente con polaridad inversa el

circuito, el diodo se encargue de bloquear la corriente y evitar así un posible daño al circuito.

Además se colocan dos condensadores de  $0.33\mu\text{F}$  y  $0.1\mu\text{F}$  a la entrada y a la salida del regulador respectivamente para filtrar el ruido interno de control del regulador y que podría afectar a las cargas que con él se alimentan. Estos valores son elegidos por recomendación del fabricante. Finalmente se coloca a manera de indicador de encendido un diodo emisor de luz (LED) conectado a la salida del regulador (+5V) mediante la resistencia R2 de  $330\Omega$  para limitar la corriente que alimenta al LED.

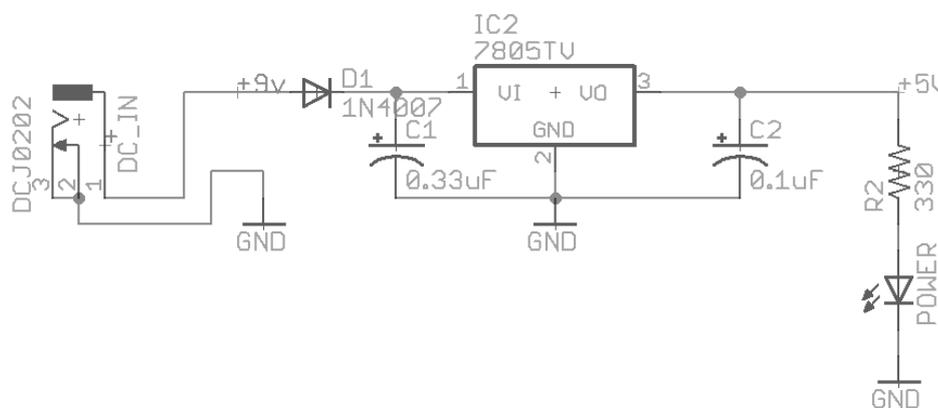


Figura 25. Diagrama esquemático del regulador de voltaje.

En la Figura 26 se muestra el circuito completo del bloque de procesamiento, comunicación y generación de señales de control.

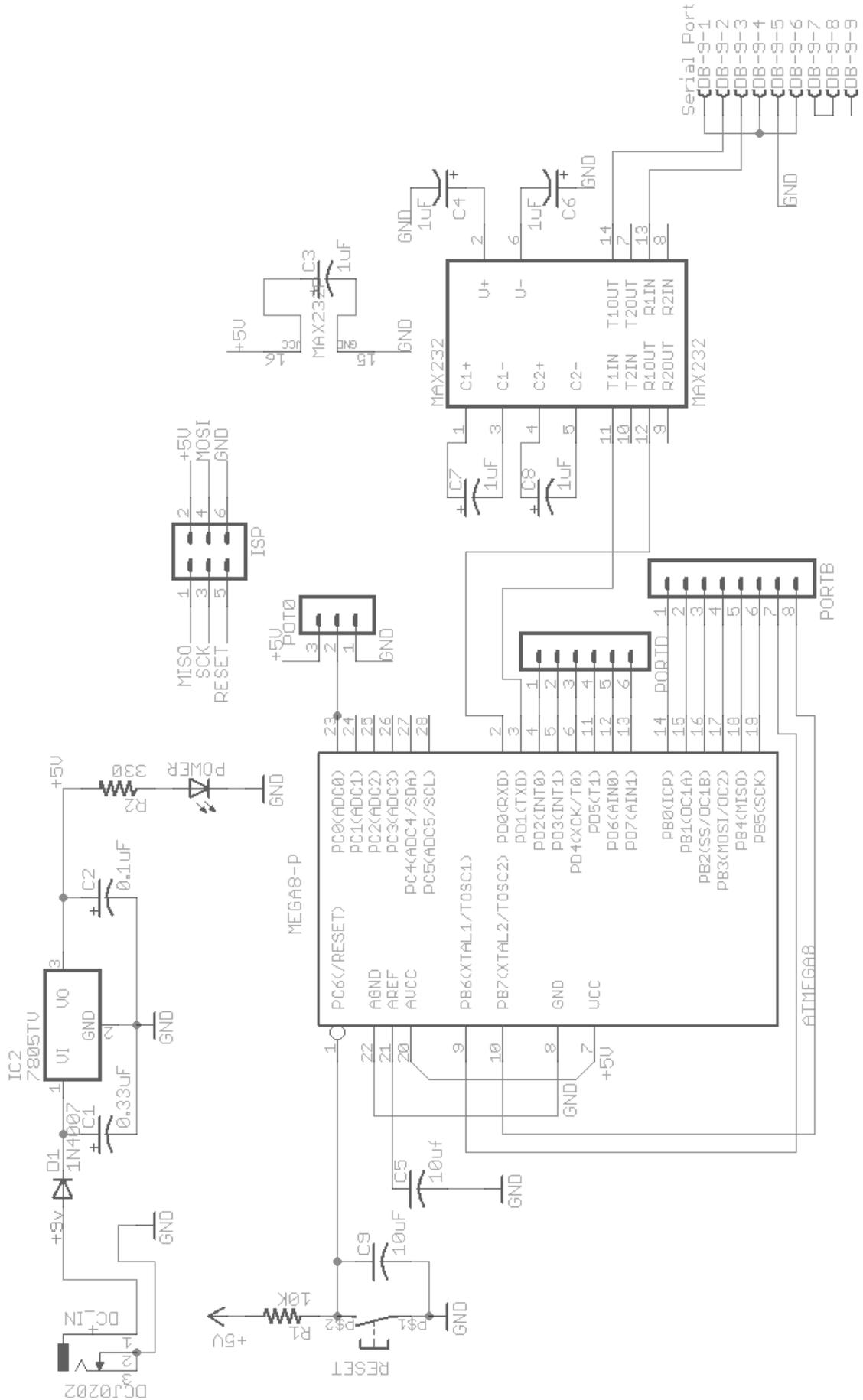


Figura 26. Diagrama esquemático de la tarjeta de control, adquisición de datos y comunicación.

#### 4.2.4. Configuración del microcontrolador ATmega8

Con los circuitos de adquisición de señales, conversión, procesamiento, control y comunicación ya diseñados, la tarea que resta es realizar un programa que se encargue de realizar y administrar todas estas tareas.

El programa propuesto es capaz de controlar el movimiento de los motores mediante los circuitos excitadores para así actuar sobre cada eje para asegurar que se cumpla con llegar a las posiciones deseadas. Este proceso es realizado tomando como realimentación la lectura de los sensores instalados para el ángulo relativo entre los eslabones del eje Y y para el desplazamiento lineal del eje Z. Las posiciones deseadas son generadas por el software de la computadora y son adquiridas por el microcontrolador mediante el USART y son usadas como referencia para efectuar las acciones correctivas necesarias.

La adquisición del ángulo relativo del potenciómetro se realizará, como se especificó en la Sección 4.2.1.3, mediante el convertidor análogo digital. El ATmega8 cuenta con un convertidor análogo-digital de resolución máxima de 10 bits, esto significa que aproxima el voltaje que se encuentre en su respectiva entrada a un valor de 10 bits proporcional a dicho voltaje, tomando como nivel mínimo el nivel de tierra y como nivel máximo (para la configuración usada) el voltaje de alimentación, que es 5V en este caso. El voltaje que se tiene a la entrada del convertidor es el de salida de la etapa de acondicionamiento, por ello variará de 0V a 5V, lo cual nos permitirá usar todo el rango del convertidor.

El convertidor se configurará para que realice las conversiones constantemente desde que es habilitado, de esta manera podemos asegurarnos de conocer siempre el valor del ángulo actual y poder realizar así las acciones correctivas adecuadas. Se tomarán solo los 8 bits más significativos del resultado de la conversión para obtener un resultado estable, mucho menos sensible a las vibraciones de los eslabones del brazo o del efector final. Al usar 8 bits de resolución del convertidor y puesto que

trabajamos en un rango de  $52^\circ$  (los cuales se traducen en niveles de voltaje desde 0V hasta 5V), obtenemos una resolución de:

$$\text{Resolución en grados} = 52^\circ/2^8 = 52^\circ/256 = 0.203^\circ \quad (\text{ec. 19})$$

El manejo del convertidor se realizará mediante interrupciones de conversión completa, es decir que el microcontrolador generará una petición de interrupción específica cada vez que se complete la conversión del voltaje de entrada a un valor proporcional al mismo.

La cuenta del desplazamiento en el eje Z mediante el sensor óptico ranurado se realizará mediante interrupciones externas activadas por flancos de subida producidos al ser obstruida la ranura del sensor por la pestaña acoplada al tornillo sinfín. Cada interrupción externa nos indicará que se ha completado una vuelta del tornillo sinfín y de acuerdo a las señales de control enviadas al circuito excitador del motor se podrá saber el sentido del movimiento para así poder conocer la posición actual en el eje Z.

La comunicación con el software de la computadora se llevará a cabo mediante el USART según lo explicado en el punto 4.2.2. El USART se configura para trabajar a 9600 BAUDS, paridad impar, 8 bits de datos y un bit de parada. Además se le configurará para trabajar mediante interrupciones, generándose solicitudes para estas cada vez que se reciba un dato o cada vez que el registro de salida esté libre. Se debe recalcar que las generaciones de solicitudes de interrupción serán activadas solo cada cierto tiempo y mientras se envíen los datos de las posiciones actuales al software de la computadora. De esta manera se evita una generación excesiva de peticiones de interrupción, lo cual podría interferir con el desarrollo normal del programa.

El control del tiempo para el cálculo del error de las posiciones en ambos ejes y la toma de decisiones para realizar las acciones correctivas se realiza mediante el Temporizador 1.

El Temporizador 1 es un temporizador de 16 bits interno del ATmega8 que cuenta con pre-escaladores de 1, 8, 64, 256 y 1024 que permiten dividir la

frecuencia del oscilador entre el modulo de cada pre-escalador para así poder obtener diferentes bases de tiempo. Además se trabajará con el temporizador configurado en el modo CTC (Clear Timer on Compare match). Se habilita también la generación de peticiones de interrupción al producirse una comparación exitosa. Esto significa que se generará una petición de interrupción cada vez que la cuenta del temporizador llegue al valor almacenado en los registros OCR1AH:OCR1AL. Al combinar la configuración del módulo del pre-escalador con el valor que se almacene en los registros OCR1A se tiene la posibilidad de generar las peticiones de interrupción con una frecuencia configurable.

De esta manera se puede controlar el intervalo de tiempo entre cada paso angular del motor paso a paso.

El programa principal se encarga de realizar las acciones correctivas en caso sea necesarias para cada eje de movimiento. Esto implica el cargar las secuencias de cada paso para el motor paso a paso y enviarlas al circuito excitador, así como generar las señales de control de sentido y el habilitador para el motor de corriente continua. Inicialmente se espera a que se reciba un valor deseado para el eje Y, luego, en función de el error (diferencia entre la posición deseada y la posición actual) se carga el inicio o fin de la tabla donde se tiene almacenada la secuencia a enviar al motor. Luego de eso se procede a verificar si en el eje Z se debe realizar algún movimiento para llegar a la posición deseada y de ser ese el caso, se actúa según el sentido que debe tener el mismo hasta llegar a la posición deseada.

Luego el programa principal se queda en una repetición infinita verificando si debe actuar sobre cada motor y como debe hacerlo, es decir, en que sentido debe mover cada motor.

La Figura 27 muestra el diagrama de flujo del programa principal que se ha descrito en las líneas anteriores.

La decisión de cómo se debe actuar sobre cada motor es tomada en las interrupciones de la siguiente manera:

- Las posiciones deseadas para cada eje son adquiridas mediante transmisión serial empleando el USART del ATmega8. Se produce una solicitud de interrupción cada vez que se recibe un nuevo dato y los datos recibidos tienen que cumplir con una trama asignada para que se les considere como válidos.
- Para el eje Y se está actualizando el valor del resultado de la conversión del nivel de voltaje del potenciómetro constantemente y este resultado se almacena en registros, pero el error solo se calcula en las interrupciones producidas por la comparación exitosa del temporizador con el registro OCR1A. En esta última interrupción se habilita también el movimiento y se determina el sentido del mismo para el motor a pasos. Estos parámetros son almacenados en registros que son usados a modo de banderas, las cuales son verificadas en el programa principal para determinar las acciones a tomar.
- Para el eje Z, los pulsos generados por cada vuelta completada del tornillo sinfín activan una interrupción externa y durante esta interrupción se aumenta o disminuye el valor de un contador, dependiendo de si el motor está avanzando o retrocediendo. El valor de este contador multiplicado por el desplazamiento lineal del efector final por cada vuelta del tornillo sinfín nos informa de la posición actual. En el programa principal se actúa según la posición actual y la posición deseada, moviendo el motor en un sentido o en el otro o apagándolo.
- Una vez se ha completado el movimiento se envían los valores actuales del desplazamiento angular entre los eslabones y el desplazamiento lineal en el eje Z al software de la computadora mediante interrupciones y se reciben las nuevas posiciones deseadas por el USART. Este proceso se repite hasta llegar a la posición final.

De esta manera el programa logra controlar el movimiento de los motores mediante los circuitos excitadores en función de las realimentaciones hasta llegar a la posición deseada y se informa permanentemente del desplazamiento actual de cada eje al software de la computadora.

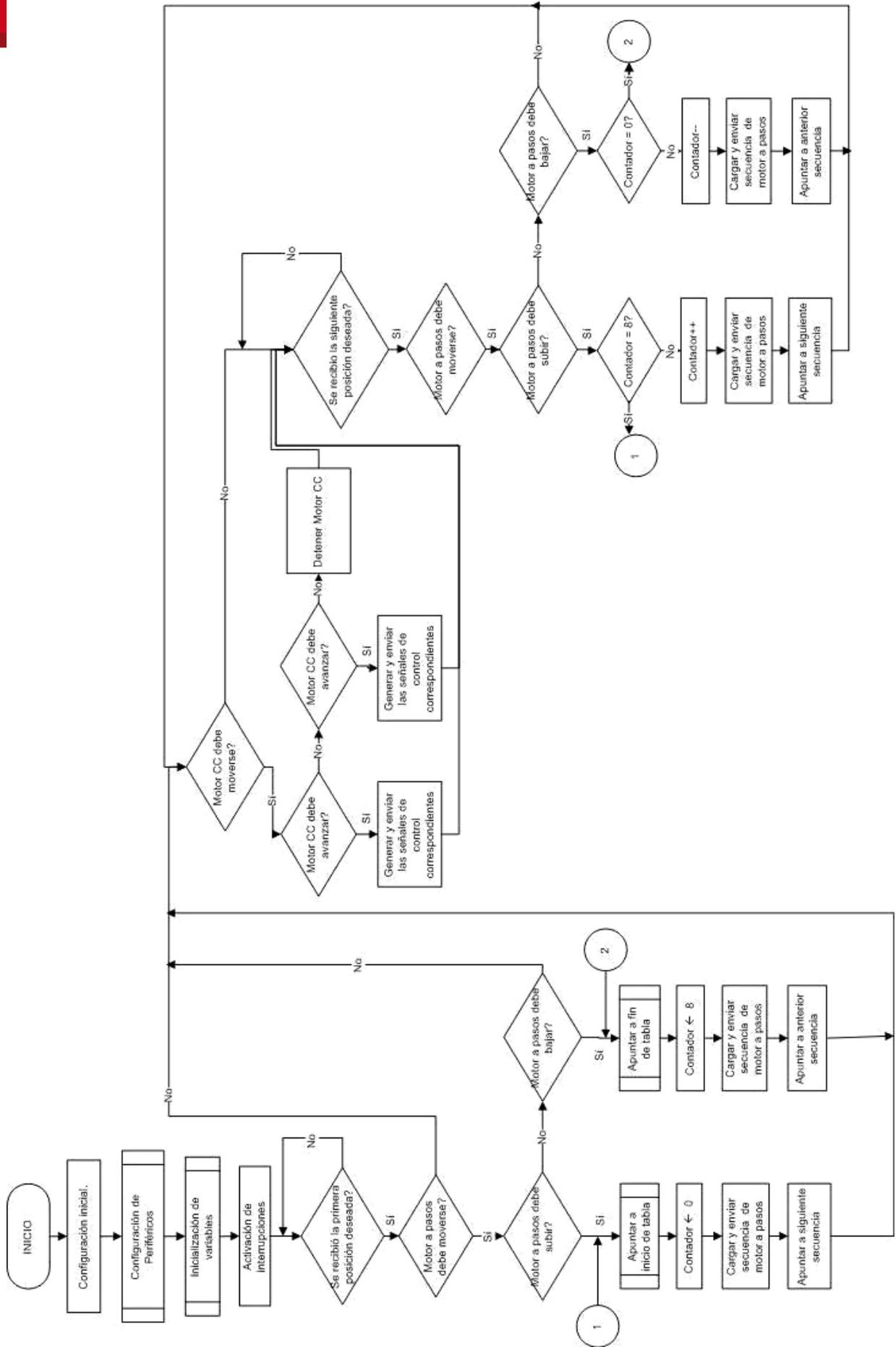


Figura 27. Diagrama de flujo del programa principal del microcontrolador ATmega8.

### 4.3. Diseño e implementación del circuito excitador de cada motor

En caso del sistema robótico presentado, se debe diseñar una interfaz que permita la conexión entre la etapa de control de posición del efector final, en ambos ejes, y los actuadores, de manera que generen la trayectoria deseada evitando tener errores grosos en el trazado deseado, ingresado mediante la interfaz de usuario.

El sistema robótico presentado cuenta con dos actuadores, ambos son motores de corriente continua: un motor de paso a paso del tipo unipolar, encargado de realizar el movimiento en el eje Y mediante la rueda omnidireccional; y un motor de corriente continua de imán permanente, encargado del movimiento en el eje Z mediante el giro del tornillo sinfín.

Al trabajar con dos tipos distintos de motores de corriente continua obliga al diseño específico de circuitos excitadores de cada uno de ellos. El diseño de cada circuito debe incluir no solo los elementos para el correcto manejo de cada motor, sino también protección de los mismos así como aislamiento de la etapa del controlador de posición. El aislamiento es necesario al trabajar con motores ya que estos generan ruido y picos de voltaje, los cuales pueden perjudicar la etapa anterior.

Se proponen para el sistema brazo robot de dos grados de libertad dos circuitos excitadores para motores, uno para el motor de paso a paso unipolar y otro para el motor de corriente continua de imán permanente. Ambos circuitos propuestos siguen un esquema básico similar, el cual se presenta a continuación en la figura 28.

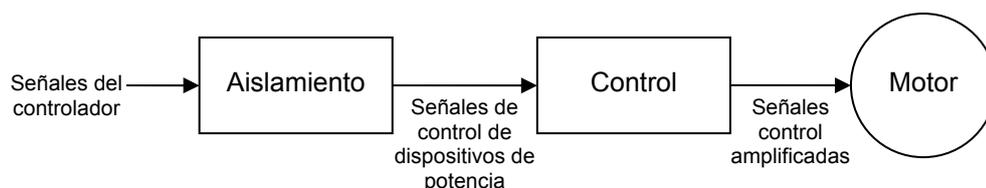


Figura 28. Esquema básico del circuito excitador

#### 4.3.1. Circuito excitador de motor paso a paso unipolar

Para el manejo del motor de paso a paso de tipo unipolar se requieren cuatro señales de control para manejar independientemente cada una de las bobinas. Esta independencia en el manejo de las bobinas brinda la libertad de manejar distintas formas de control sobre el motor paso a paso: fase simple, doble fase y medio paso. Además esta característica nos brinda mayor flexibilidad en el manejo del motor.

Al tener en cuenta que son cuatro las señales de control que vienen del controlador es pertinente aislar cada una de ellas de los dispositivos de potencia que van a manejar cada bobina del motor paso a paso unipolar. Para realizar este trabajo se opta por usar acopladores ópticos, los cuales requieren únicamente un resistor limitador de corriente en la etapa del emisor óptico. La etapa del receptor óptico es la etapa que alimenta a la siguiente etapa del circuito: los dispositivos de potencia.

Los dispositivos de potencia requieren un tipo de alimentación distinta para su funcionamiento (en algunos casos alimentación más elevada que la salida de un microcontrolador), además que trabajan con niveles de corriente y voltaje que los dispositivos de control no manejan. Para el sistema planteado deben manejar las corrientes de cada fase del motor paso a paso.

Además se deben implementar protección para los dispositivos usados, pues al trabajar con cargas inductivas es frecuente que se generen picos de voltaje que puedan malograr el dispositivo.

##### 4.3.1.1. Diseño propuesto

Teniendo en cuenta las consideraciones anteriores, se propone el siguiente circuito excitador de motor paso a paso unipolar para su implementación como parte del sistema del brazo robot de dos grados de libertad, mostrado a continuación en la figura 29. Se busca que el circuito funcione para el motor a utilizar, así como para cualquier motor paso a paso unipolar, de manera que se permita realizar un cambio de motor sin mayor contratiempo. Este diseño está basado en un diseño propuesto por el Ingeniero Javier Chang, en un curso de extensión dictado en el CETAM-PUCP. [13]

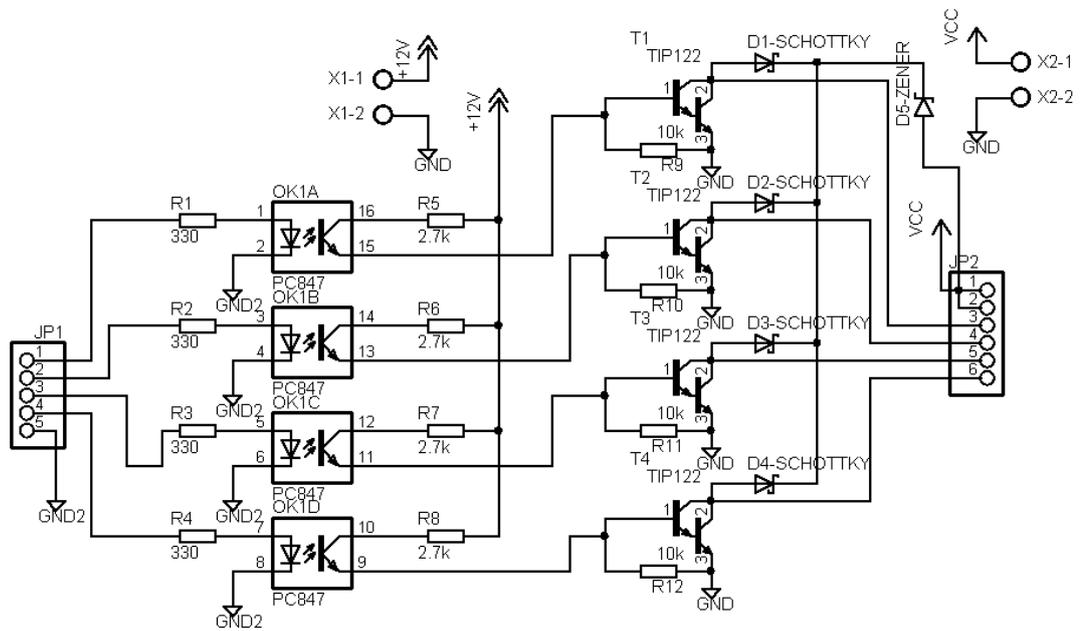


Figura 29. Diagrama esquemático del excitador del motor paso a paso unipolar.

En el diagrama de la figura 29 la entrada del circuito propuesto está en la parte izquierda. Son cuatro entradas digitales, más la referencia de las mismas, provenientes del microcontrolador encargado del control. Estas entradas alimentan a cuatro acopladores ópticos de la serie PC817. Este tipo de acoplador está disponible en el mercado local, además de tener un coste bajo. El circuito cuenta con cuatro resistores limitadores de corriente de valor 330 Ω. Se puede obtener la siguiente ecuación a partir del diseño propuesto:

$$V_{OH} - I_F \times R_F - V_F = 0V \tag{ec. 20}$$

Donde  $V_F$  es el voltaje promedio de conducción del diodo emisor del acoplador óptico, 1.2V según hoja de datos. Esto impondrá que se siga una de las curvas de Corriente de Colector vs. Voltaje Emisor-Colector en la etapa del receptor fotosensible del acoplador. De acuerdo a esta curva se podría exigir una corriente de 12mA sin tener caídas mayores a 1V entre colector-emisor. En esta ecuación  $V_{OH}$  es el voltaje en nivel alto de la salida

de un puerto del microcontrolador, 4.2V (valor mínimo) según hoja de datos del ATmega8. Con estos valores se obtiene, para una resistencia de 330Ω:

$$I_F = 9.09mA$$

El fototransistor del acoplador alimentará la siguiente etapa del excitador: los dispositivos de potencia. Para esta etapa se seleccionaron transistores tipo Darlington debido a su capacidad para manejar un nivel de corriente elevado. Para el diseño del circuito se seleccionaron los transistores TIP122, capaces de manejar corrientes de hasta 5A, valor bastante mayor al usual valor de corriente nominal de motores paso a paso.

El diseño contempla el valor de los resistores de base del transistor TIP122, Para esto se pueden observar las curvas características del TIP122, se usan las de Voltaje de Saturación de Base y de Colector, referidos al Emisor. Se puede notar en la hoja de datos que para valores de corriente en el Colector menores a 1A, se tienen valores de 1.4V y 0.6V respectivamente. Teniendo en cuenta que la alimentación del TIP122 se considera de 12V:

$$V_{CC} - I_B \times R_B - V_{BE} = 0V \quad (\text{ec. 21})$$

Si se toma en cuenta un valor de corriente de 1A para una fase de motor paso a paso y una ganancia de 250 del transistor:

$$I_B = \frac{I_M}{h_{FE}} = 4mA \quad (\text{ec. 22})$$

Al reemplazar los valores y hacer uso de las ecuaciones 21 y 22 obtenemos el siguiente valor para la resistencia de base:

$$R_b = \frac{(12 - 1.4)V}{4mA} = 2.7k\Omega$$

Como se puede observar el valor de resistencia elegido trabaja para corrientes de 1A, sin embargo para trabajo con corrientes menores a este valor el circuito deberá responder de forma satisfactoria igualmente. Se incluye también un resistor entre la Base y el Emisor del transistor como un

elemento de puesta a baja, de manera que asegure el estado de corte cuando el transistor no recibe ninguna señal de control. El valor de resistencia es de  $10k\Omega$ .

Los últimos elementos del circuito, ubicados en el extremo derecho del diagrama esquemático, son los elementos de protección de los dispositivos de potencia, en este caso de los transistores TIP122. Al observar la hoja de datos del TIP122 se puede observar que soportan un máximo voltaje de 100V entre Colector y Emisor. Las cargas inductivas generan picos de voltaje que en muchos casos dañan a los dispositivos de potencia que las manejan. Se incluyen cuatro diodos Schottky de conmutación rápida como parte de la protección de los transistores. Se eligen diodos Schottky pues al ser picos los que se generan, estos son de velocidad elevada. Para el diseño se seleccionó el diodo 1N5819. Este diodo cuenta con una corriente nominal de 1A y un voltaje inverso máximo de 40V.

También se incluye un diodo Zener, el cual tiene una función especial que se describirá a continuación. Al tener bobinas de toma central, donde cada mitad pertenece a una fase del motor paso a paso, al activar una bobina se tiene un incremento en el campo magnético de la bobina opuesta de la otra mitad. El diodo Zener permite que este campo magnético se incremente, lo cual produce un potencial en la otra mitad, limitándolo al valor del diodo Zener más la alimentación del motor. Si se colocase únicamente el diodo Schottky, se activaría cortando la otra mitad del bobinado, lo que provocaría que el incremento en el campo magnético en la mitad de la bobina que desea activarse sea más difícil. [14]

El transistor TIP122 tiene un diodo interno que también funciona como protección del sistema. El diodo interno protege en realidad al transistor que activa la fase opuesta. Al desactivarse el transistor, se genera un pico en el Colector, lo cual genera una caída en el Colector opuesto, por estar unidos mediante la bobina de toma central. Si este último cae debajo del valor de referencia, el diodo interno del TIP122 se activa.

#### 4.3.1.2. Implementación del circuito

De acuerdo al diseño propuesto se llevó a cabo la implementación del mismo. La selección de componentes permitió que todos fueran adquiribles en el mercado local y a un costo relativamente bajo. Una vez implementado el circuito se realizaron mediciones respectivas para contrastar los valores esperados con los reales, además de poder registrar estos valores se puede determinar las pérdidas generadas, en especial en la etapa de potencia, en los transistores TIP122.

En la implementación se usó el motor paso a paso PM25S-048, el cual se empleará en la aplicación del brazo robot, logrando el funcionamiento correcto del mismo. Sin embargo el circuito descrito permite la utilización de otro motor paso a paso. Los resultados de las mediciones se resumen en la tabla 4.

*Tabla 4. Resultados de la implementación del circuito excitador de motor paso a paso unipolar.*

<b>Etapa de acoplamiento óptico</b>	
Corriente de conducción del foto emisor	10.45mA
Voltaje de conducción del foto emisor	1.172V
Voltaje Colector-Emisor en foto receptor	182.1mV
<b>Etapa de transistor TIP122</b>	
Corriente de Base	3.91mA
Voltaje Base-Emisor	1.345V
Corriente de Colector	161.59mA
Voltaje Colector-Emisor	0.671V

En base a estos resultados se pueden obtener dos valores de pérdidas en el circuito. El más relevante es el valor de pérdida en el transistor TIP122, aunque también se tiene el valor de pérdida en el acoplador óptico.

$$P_{DIS} = I_C \times V_{CE} \quad (\text{ec. 23})$$

Al tomar en cuenta los valores medidos en el circuito implementado, se muestran los resultados en la tabla 5.

Tabla 5. Pérdidas en el circuito.

Pérdidas	
Acoplador óptico PC817	0.712mW
Transistor TIP122	108.43mW

#### 4.3.2. Circuito excitador de motor de corriente continua de imán permanente

En el caso de un motor de corriente continua de imán permanente se requiere controlar la corriente que fluye por el bobinado del mismo. En el caso de la aplicación del brazo robot manipulador se controla la dirección mas no la corriente efectiva en el bobinado. Esto debido a la transmisión entre el giro del motor y el desplazamiento lineal del brazo. Sin embargo es conveniente tener en cuenta un diseño que permita un control de velocidad para un control posterior de la misma.

La configuración más utilizada para controlar motores de corriente continua es el circuito Puente H, el cual consiste en un arreglo de transistores que permiten la circulación de corriente en el bobinado del motor en ambos sentidos. El arreglo se presenta a continuación en la figura 30.

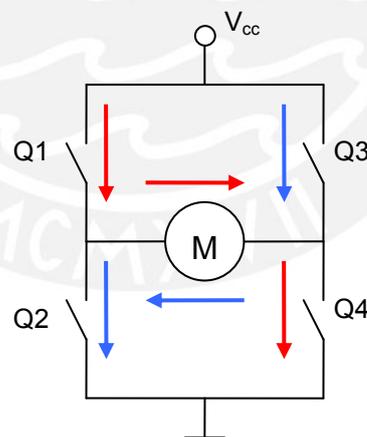


Figura 30. Esquema del Puente H.

Como se puede observar las señales de control que requiere esta configuración son cuatro, una para activar cada transistor. Sin embargo, para facilitar la etapa de control se pueden reducir a únicamente dos señales, debido a que debido en la configuración se requieren activar dos transistores

a la vez cada vez que se requiera activar un sentido, tal como lo muestra la figura.

Al igual que en el caso del excitador del motor paso a paso, se opta por aislar las señales de control de las señales que activarán los dispositivos de potencia e igualmente por emplear acopladores ópticos.

La siguiente etapa del circuito consiste en adaptar la señal de potencia de manera que pueda activar cada transistor necesario para uno de los sentidos de corriente, teniendo en cuenta que no deben activarse dos transistores de la misma columna, pues se generaría un cortocircuito.

También en este caso se deben implementar las protecciones necesarias para los dispositivos de potencia del circuito excitador.

#### **4.3.2.1. Diseño propuesto**

Con las consideraciones anteriormente expuestas, se propone el siguiente circuito excitador de motor de corriente continua de imán permanente para su implementación como parte del sistema del brazo robot de dos grados de libertad, mostrado en la figura 31. Para este caso el circuito diseñado funcionará para cualquier motor de corriente continua que sea alimentado con 24V. Sin embargo, el rango de alimentación puede variarse cambiando el valor de resistencia de un par de resistores. El esquema básico del diseño fue recogido de una página web en línea. [15]

Las entradas del circuito mostrado están a la izquierda del diagrama. Se puede notar que son dos entradas: una para cada sentido de corriente más su referencia. Estas entradas vienen del microcontrolador, aunque por facilidad para la etapa de control, primero se diseñó una interfaz para que el controlador maneje solo una señal para selección y otra para dirección en lugar de señales independientes para el sentido de corriente.

Cada entrada pasa a alimentar a un acoplador óptico de la serie PC817, los cuales como se mencionaron están disponibles en el mercado y son de bajo coste. Para limitar la corriente que pasa por el emisor óptico se cuentan con resistores de valor  $330\Omega$ . Teniendo en cuenta una caída de potencial

promedio de 1.2V para el emisor óptico y que la entrada no vendrá directamente del microcontrolador, si no de una serie de compuertas lógicas de la familia 74HC, la entrada en estado de activo estará de 4.4V aproximadamente.

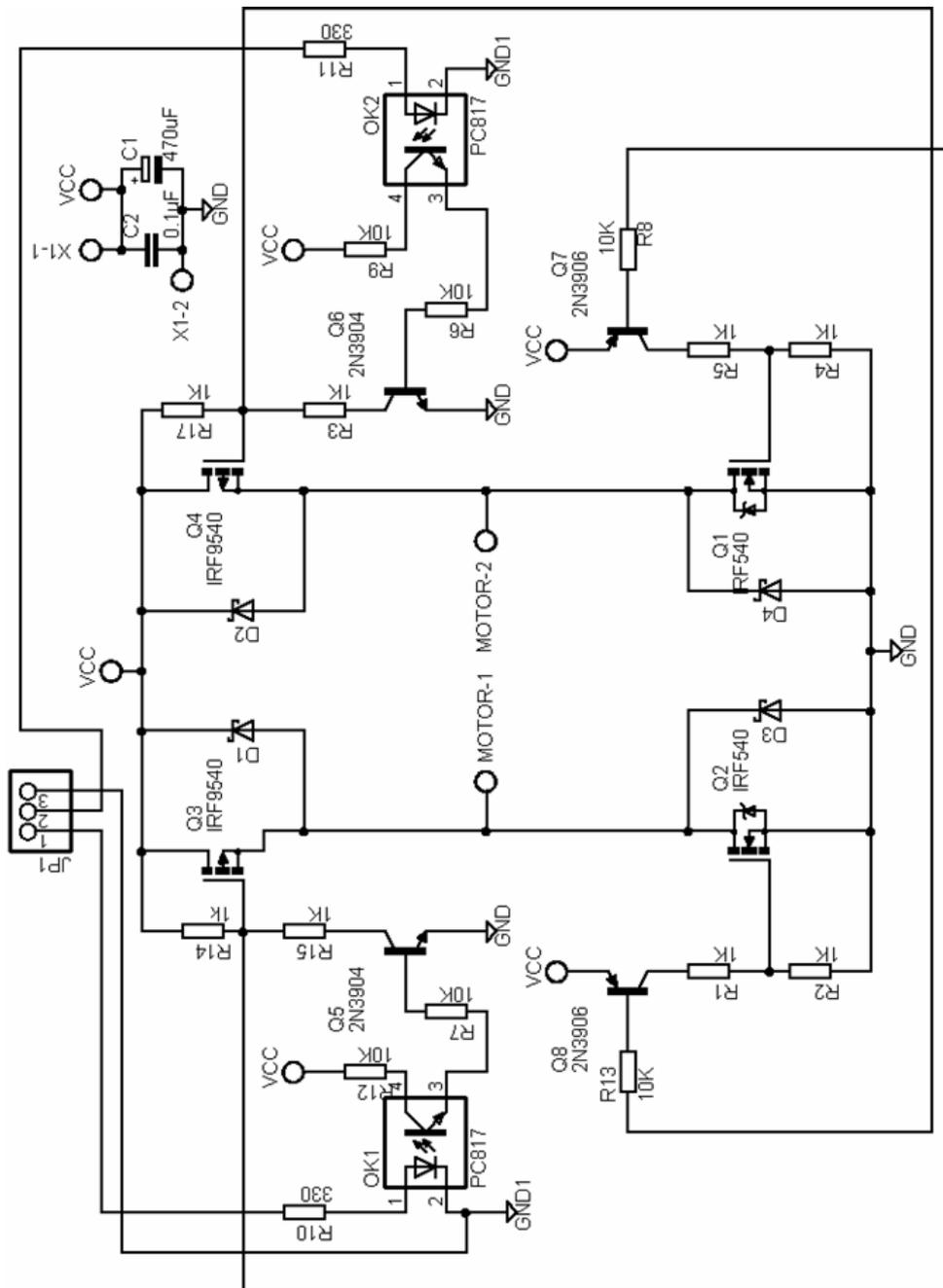


Figura 31. Diagrama esquemático del circuito excitador del motor de corriente continua de imán permanente.

Del diseño mostrado en la figura 31 se obtiene:

$$V_{OH} - I_F \times R_F - V_F = 0V \quad (\text{ec. 24})$$

Al reemplazar los valores mencionados:

$$I_F = 9mA$$

Al trabajar con estos valores, la salida del acoplador óptico queda limitada a un máximo de 10mA sin tener pérdidas mayores a 1V entre Colector y Emisor, además de estar limitada por el máximo de potencia disipada de acuerdo a la curva característica de Corriente de Colector vs. Voltaje Colector-Emisor para una corriente de diodo opto emisor de 10mA.

En la etapa del fototransistor se tienen dos resistores que limitan la corriente que circula en la base de los transistores NPN de código 2N3904 de la siguiente etapa. Si asumimos el estado de saturación podemos obtener una caída de potencial de 0.7V entre Base y Emisor, y agregando la caída de potencial en el receptor óptico de 1V. Se muestran las ecuaciones del sistema para los lazos Base-Emisor y Colector-Emisor del transistor PNP.

$$V_M - V_{BE} - V_{CE-OPTO} - R_B \times I_B = 0 \quad (\text{ec. 25})$$

$$V_M - V_{CE} - I_C \times R_C = 0 \quad (\text{ec. 26})$$

$$I_c \approx 10I_b \quad (\text{ec. 27})$$

Se debe tomar en cuenta que el valor de resistencia del Colector debe estar repartido necesariamente entre dos resistores, pues se requiere un divisor resistivo a la mitad para la etapa posterior que va a alimentar el transistor de tipo PNP. Teniendo este sistema se pueden obtener valores de resistencia y corrientes de polarización en este transistor, que se muestran en la tabla 6. Se toma en cuenta la alimentación del motor en un valor de 24V.

*Tabla 6. Valores calculados para el transistor 2N3904*

Variable	Valor
$R_b$	20k $\Omega$

$R_c$	$2K\Omega$
$I_b$	$1.11mA$
$I_c$	$11.9mA$

Con estos valores calculados obtenemos una ganancia de 10.72 en el transistor. De acuerdo a lo expuesto en el párrafo anterior, se escogen dos resistores de  $1K\Omega$  para el Colector del transistor 2N3904. Lo mismo se tiene en la etapa que activa el otro sentido de corriente: un transistor 2N3904 y sus respectivos resistores de polarización. Al observar el diagrama se puede notar que este transistor (y su imagen que activa el otro sentido) sirve para activar el transistor ubicado en la parte superior del Puente H.

Para cerrar el camino de circulación de corriente se debe activar el transistor complementario. Para realizar este trabajo se emplea un transistor complementario al 2N3904: un transistor tipo PNP 2N3906, que al igual que su complementario es común conocido y barato. Este transistor activará el transistor que cerrará el circuito.

Tomando en cuenta las mismas consideraciones que se tomaron para el transistor 2N3904 en condiciones de saturación, se puede obtener la siguiente ecuación para el lazo Base-Emisor, tomando en cuenta que el transistor 2N3904 está activo y es alimentado con 24V, voltaje de alimentación del motor.

$$V_M - V_{BE} - I_B \times R_B = V_{DIV} \quad (\text{ec. 28})$$

En donde  $V_{DIV}$  denota al resultado del divisor de potencial del transistor NPN, valor calculado a 12V. Al igual que en el caso del transistor 2N3904, se debe tomar en cuenta que el valor de resistencia en el Colector debe ser dividido en dos resistores pues se requiere también de un divisor de voltaje a la mitad para activar la siguiente etapa del circuito. Resolviendo el sistema formado por la ecuación 28, 26 y 27, se pueden obtener los valores deseados, mostrados en la tabla 7.

*Tabla 7. Valores calculados para el transistor 2N3906*

Variable	Valor
$R_b$	10k $\Omega$
$R_c$	2K $\Omega$
$I_b$	1.13mA
$I_c$	11.9mA

Estos valores de resistencia de polarización nos dan una ganancia de 10.53 en el transistor. Como se expuso en el párrafo anterior es necesario dividir el valor de la resistencia de Colector en dos resistores, cada uno de 1k $\Omega$ . Para el otro sentido de corriente se tiene la misma configuración con los mismos valores.

Cabe resaltar que se desprecia el efecto de carga que se tiene entre el divisor resistivo del transistor 2N3904 y la Base del transistor 2N3906. Esto se verá reflejado en una variación de los valores calculados para las corrientes. Sin embargo el efecto no es considerable teniendo en cuenta que el valor de corriente en la base del transistor 2N3906 es diez veces menor al valor de la corriente en el Colector del transistor 2N3904.

La etapa anteriormente descrita que implican los transistores 2N3904 y 2N3906, es una etapa empleada para adaptar las señales de control a señales de niveles adecuados para trabajar con los dispositivos que manejarán la corriente del motor. Los dispositivos seleccionados son transistores tipo MOSFET.

Los transistores MOSFET son dispositivos que requieren un potencial de 10V entre sus terminales de Compuerta y Surtidor para que trabajen en su región activa. Estos dispositivos son activados por voltaje, aunque en realidad se disparan mediante carga, necesaria para cargar sus capacitancias parásitas.

Los dispositivos seleccionados son MOSFET canal N y canal P, ya que para la configuración de Puente H, es más simple manejar ambos tipos de MOSFET y generar los potenciales necesarios para dispararlos. Para esta aplicación se seleccionaron los MOSFET IRF540, canal N, e IRF9540, canal P. Entre las características principales del IRF540 se puede mencionar su

corriente máxima de 33A, su resistencia de encendido de 44mΩ y su voltaje Drenador-Surtidor máximo de 100V. El IRF9540 soporta hasta 23A, su resistencia de encendido es de 0.117Ω y soporta el mismo voltaje Drenador-Surtidor. Ambos dispositivos se encuentran en el mercado local a precio accesible.

Para entender el funcionamiento en conjunto del circuito excitador, se asumirá que una de las entradas está activa. Se muestra en la figura 32 la sección del circuito encargada de activar el MOSFET canal P IRF9540. Si el transistor Q5 está activo, lo que obtenemos en el Colector del mismo es un divisor de potencial, donde el valor aproximado del potencial en R14 es la mitad del VCC, que se fija en 24V. Por tanto el potencial Compuerta-Surtidor del transistor Q3 es -12V, más del necesario para activarlo pero muy lejos del valor límite de -20V, obtenido de la hoja de datos.

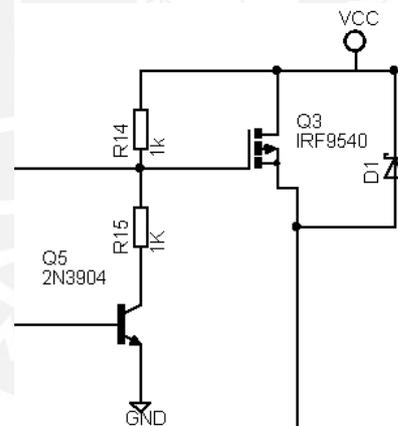


Figura 32. Activación del MOSFET IRF9540.

En la figura 33 se muestra la sección encargada de la activación del MOSFET canal N IRF540, complementario al mostrado en la figura anterior, por tanto se continúa asumiendo que el transistor Q5 está activo. Teniendo esto en cuenta se puede tomar como el potencial del extremo del resistor R8 que no presenta conexión como 12V, lo cual genera que el transistor Q7 esté activo. De este modo obtenemos otro divisor de potencial, obteniendo en R4 un potencial aproximado a la mitad de VCC. Entonces el potencial Compuerta-Surtidor del transistor Q1 se fija en 12V, mayor al necesario para

el estado de conducción y menor al límite de 20V, obtenido de la hoja de datos.

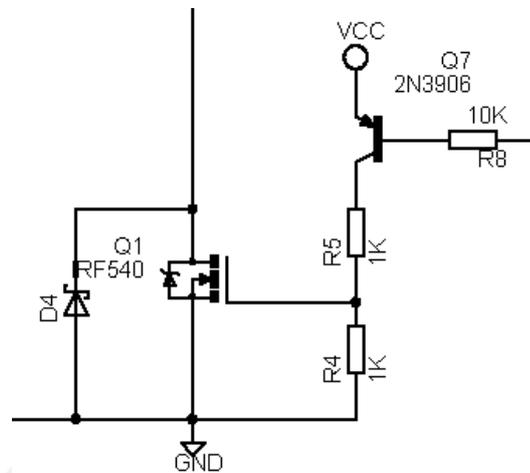


Figura 33. Activación del MOSFET IRF540.

En las dos figuras anteriores (figura 32 y figura 33) también se puede apreciar la protección de los MOSFET contra los picos de voltaje que se puedan generar por la inductancia del motor. Para este trabajo se seleccionaron también diodos Schottky 1N5819, el mismo diodo planteado para la circuito excitador del motor paso a paso unipolar. Estos diodos fijan a que los voltajes en cada Drenador no excedan a la alimentación más el voltaje de conducción del diodo o que no baje a un valor menor al de la referencia menos el voltaje de conducción del diodo.

#### 4.3.2.2 Implementación del circuito

Teniendo como base el diseño propuesto se procedió a la implementación del mismo. Los componentes usados en el diseño son fácilmente adquiribles en el mercado local y a un precio accesible. Una vez implementado el circuito se procede, de manera similar a lo trabajado con el excitador de motor paso a paso unipolar, a la medida de ciertos parámetros para determinar el correcto funcionamiento del circuito, así como de las pérdidas en los elementos activos.

En las mediciones se empleó el motor a usar en el sistema de brazo robot, el motor RS-360SH-09600. A continuación se muestran los resultados de las mediciones en la tabla 8.

*Tabla 8. Resultados de la implementación del circuito excitador de motor de corriente continua de imán permanente.*

<b>Etapa de acoplamiento óptico</b>	
Corriente de conducción del foto emisor	9.24mA
Voltaje de conducción del foto emisor	1.21V
Voltaje Colector-Emisor en foto receptor	76.5mV
<b>Etapa de transistor 2N3904</b>	
Corriente de Base	1.16mA
Voltaje Base-Emisor	0.746V
Corriente de Colector	12.87mA
Voltaje Colector-Emisor	48.2mV
<b>Etapa de transistor 2N3906</b>	
Corriente de Base	-1.06mA
Voltaje Base-Emisor	-0.756V
Corriente de Colector	-11.15mA
Voltaje Colector-Emisor	-80.6mV
<b>Etapa de transistor IRF540</b>	
Voltaje Compuerta-Surtidor	11.96V
Corriente de Drenador	190mA
Voltaje Drenador-Surtidor	5.1mV
<b>Etapa de transistor IRF9540</b>	
Voltaje Compuerta-Surtidor	-11.46V
Corriente de Drenador	190mA
Voltaje Drenador-Surtidor	0.011v

Tomando en cuenta estos datos podemos determinar las pérdidas de potencia en cada elemento activo.

$$P_{DIS} = I \times V \quad (\text{ec. 29})$$

Los resultados de las pérdidas se muestran en la tabla 9.

*Tabla 9. Pérdidas en el circuito.*

Pérdidas	
Acoplador óptico PC817	0.084mW
Transistor 2N3904	1.20mW
Transistor 2N3906	1.66mW
Transistor IRF540	0.0969mW
Transistor IRF9540	2.09mW

#### 4.3.3. Interfaz entre microcontrolador y circuito excitador de motor de corriente continua de imán permanente

El circuito excitador de motor de corriente continua cuenta con dos entradas digitales. Cada una controla la activación de cada sentido de la corriente que circula por el motor. Sin embargo, para simplificar las acciones de control, se optó por diseñar una pequeña interfaz de manera que se controlen dos señales: dirección y habilitador. Esto da la posibilidad a futuro de un control de velocidad, en donde variando la señal del habilitador podría lograrse el control de esta variable.

##### 4.3.3.1. Diseño propuesto

El diseño propuesto consiste en dos compuertas lógicas: dos compuertas buffer tri-estado. Una de las compuertas tiene un habilitador de lógica directa y la otra de lógica inversa. Las compuertas elegidas son: 74HC125 y 74HC126, ambas disponibles en el mercado a precio accesible, además de ser de nivel TTL, compatible con los niveles del microcontrolador.

Para estas compuertas se cuentan con las siguientes características: voltaje de salida en nivel alto de 4.4V mínimo y la corriente máxima a la entrada es de 0.1 $\mu$ A, lo cual es factible con el microcontrolador empleado. Además

permiten exigir una corriente máxima a la salida de 35mA por pin, lo cual es más de lo que se necesita en el circuito.

Se presenta a continuación el diagrama esquemático del circuito de interfaz en la figura 34.

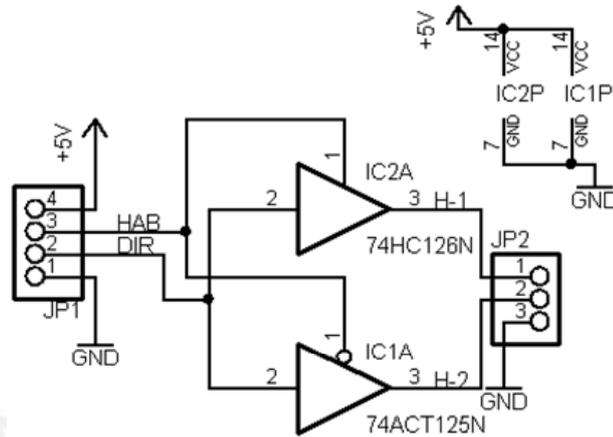


Figura 34. Diagrama esquemático de la interfaz.

#### 4.3.3.2. Implementación de la interfaz

Para la implementación de la interfaz no se encontró percance alguno. El circuito funciona en su totalidad para ambos sentidos de corriente. La comunicación con el microcontrolador permite que las señales de control lleguen de forma exitosa hacia los acopladores ópticos del circuito excitador de motor de corriente continua de imán permanente.

Se muestra en la tabla 10 mediciones tomadas a la tarjeta de interfaz.

Tabla 10. Valores medidos en la tarjeta de interfaz.

Variable	Valor
Voltaje a la salida	3.3 V
Corriente de salida	6.33 mA

#### 4.3.4. Cálculo de disipadores para dispositivos de potencia

Al trabajar con dispositivos de potencia en muchos casos es necesaria la presencia de disipadores térmicos. Esto se debe a que cada dispositivo de potencia cuenta con un área de trabajo de operación segura. Si se exige al

dispositivo trabajar a condiciones de mayor disipación de energía, ocurrirá que saldrá de ésta área, provocando una excesiva elevación de temperatura en la junta de silicona del semiconductor. Bajo esta condición se generan cambios químicos y metalúrgicos, relacionados exponencialmente con la temperatura de junta. [16]

Para realizar los cálculos de los disipadores se tiene en cuenta el siguiente esquema de disipación térmica en los transistores:

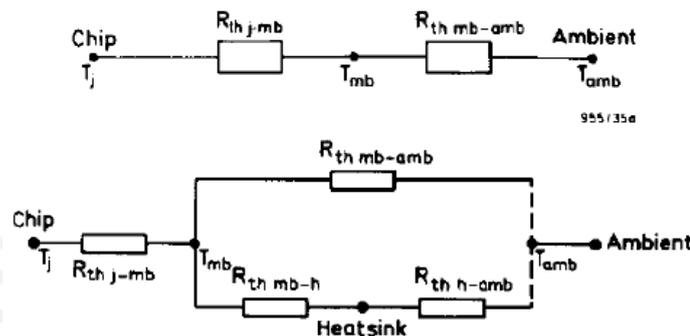


Figura 35. Esquema de disipación térmica. Arriba: sin disipador. Abajo: con disipador. [16]

#### 4.3.4.1. Transistores TIP122

Para el trabajo de los TIP122 se desea una condición de trabajo en la cual la temperatura máxima del encapsulado será de 50 °C. Usando la hoja de datos del transistor TIP122, se obtiene la potencia máxima del gráfico Potencia máxima vs. Temperatura de encapsulado, de donde se obtiene el dato que la máxima potencia a disipar es 50W. También podemos obtener la temperatura máxima de junta en la hoja de datos: 150 °C. Con estos datos podemos obtener una resistencia térmica entre la junta y el encapsulado ( $R_{th j-c}$ ):

$$P_{tot\ max} = \frac{T_{jmax} - T_c}{R_{th\ j-c}} \tag{ec. 30}$$

Se obtiene el siguiente valor:  $R_{thj-c} = 2\ K/W$

Donde las unidades de resistencia térmica están en unidades de grados Kelvin sobre unidad de potencia en Watts.

Se asumen los peores casos en el transistor TIP122: un potencial Colector-Emisor de 4V y una corriente de 1A, como se explicó el usual máximo para un motor paso a paso, tenemos una potencia de 4W. Sin embargo este valor es el doble del máximo permitido para una temperatura ambiente de 25 °C, que es de 2W. Teniendo en cuenta el esquema de la figura 35:

$$P_{tot\ max} = \frac{T_{j\ max} - T_{amb}}{R_{th\ j-c} + R_{th\ c-d} + R_d} \quad (\text{ec. 31})$$

Obteniéndose:

$$R_d = 22\ K/W$$

Donde  $R_d$  es la resistencia térmica del disipador y  $R_{th\ c-d}$  es la resistencia entre el encapsulado y el disipador, de valor 1K/W si se usa silicona termoconductor. Se escoge el camino de disipación mediante el disipador, por tanto la potencia máximo es de 4W, la temperatura máxima de juntura será reducida a 125 °C por seguridad. [17]

Del resultado, la resistencia térmica máxima debe ser de 22K/W, entonces se puede dimensionar un disipador de 5W/K, el cual es un disipador pequeño para empaque TO-220, accesible en el mercado local.

#### 4.3.4.2. Transistores IRF540 e IRF9540

Para ambos transistores MOSFET, se encuentran en sus hojas de datos los valores de resistencias térmicas necesarios para realizar el cálculo térmico. Así se obtienen los datos (teniendo en cuenta la misma notación usada para el transistor TIP122):  $R_{th\ j-c} = 1.15K/W$ ,  $R_{th\ c-d} = 0.5K/W$  y  $R_{th\ j-a} = 62\ K/W$ . Este último valor se desprecia asumiendo que la potencia se disipará mayoritariamente mediante el disipador. Además podemos obtener la temperatura de juntura máxima: 175 °C, sin embargo por seguridad se tomará como 150 °C.

En cuanto a la potencia disipada por los transistores si se tienen valores diferentes. Se trabajará con los datos del IRF9540, puesto presenta el peor

caso de los dos. Usando el dato de  $R_{DS(ON)} = 0.117\Omega$  y teniendo en cuenta una corriente de 2A (en realidad su carga es la del motor de corriente continua de 230mA, pero se incrementa para darle flexibilidad al circuito de trabajar con motores de mayor potencia). La potencia disipada es:

$$P_{DIS} = I_D^2 \times R_{DS(ON)} \quad (\text{ec. 32})$$

Obteniéndose:  $P_{DIS} = 0.468W$

Para lo cual el cálculo térmico se realiza utilizando (ec.31), utilizada en la sección anterior, para lo cual se obtiene:

$$R_d = 265.4 K/W$$

El valor obtenido es bastante elevado. Esto se debe a que no se ha tomado el valor de la potencia máxima permisible en el transistor: 140W en caso del IRF540 y 130W en caso del IRF9540. Por tanto no se tiene la necesidad de un disipador. Sin embargo se opta por ponerle también un disipador de 5K/W, disipador pequeño para el empaque TO-220.

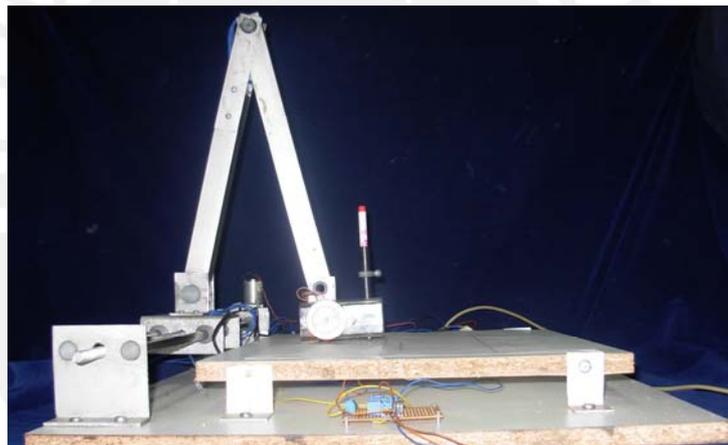
## CAPÍTULO 5

### IMPLEMENTACIÓN DEL SISTEMA Y PRUEBAS

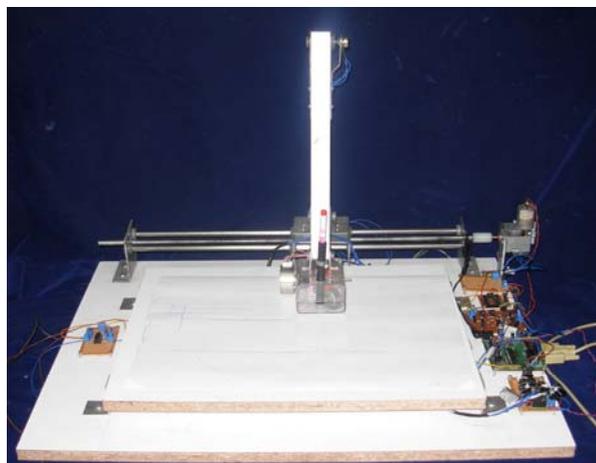
#### 5.1. Implementación del sistema

La implementación del sistema contempla la interconexión de todas las etapas de circuitería y el montaje de los sensores empleados para las realimentaciones.

Las figuras 36 y 37 muestran las fotografías del sistema implementado. Las tarjetas de circuito impreso fueron fijadas en la base de la estructura y el cableado fue realizado de tal manera que permita la movilidad del efector final sin presentar problemas.



*Figura 36. Vista lateral del brazo robot implementado.*



*Figura 37. Vista frontal del brazo robot implementado.*

## 5.2. Pruebas del sistema

Se realizaron las siguientes pruebas al sistema:

- a. Trazado de líneas horizontales. Se considera como tales a las líneas paralelas al eje Z. Con esta prueba se mide la capacidad del sistema de mantener constante la posición en el eje Y mientras se realiza desplazamiento en el otro eje. El principal factor de error que se presenta en esta prueba es la inercia del sistema a este tipo de movimiento. Esto se presenta al iniciar un movimiento en el eje Z, pero este movimiento no es transferido al efector final sino hasta después de presentarse cierto grado de deformación de los eslabones del brazo. Esto genera un error entre la posición final deseada y la real que nos resulta imposible de leer y por ello mismo de corregir.
- b. Trazado de líneas verticales.  
Con esta prueba se busca evaluar la capacidad del sistema para realizar movimiento en el eje Y, manejado por el motor paso a paso mientras el eje Z no presenta movimiento. En este caso el error que se presenta es el error generado por el paso angular del mismo motor paso a paso, que presenta pasos discretos y no permite que se puedan controlar movimientos menores a este. También debe tomarse en cuenta que al manejar el motor en secuencia de medios pasos el torque aplicado al mismo no es constante, lo cual también es una fuente de error al momento de realizar el desplazamiento del efector final.
- c. Trazado de líneas con ángulo de  $45^\circ$ .  
La finalidad de esta prueba es evaluar la capacidad del sistema para realizar movimientos en simultaneo en los ejes Y y Z. Así mismo se busca evaluar el desempeño del mismo al realizar dicha tarea midiendo la posición final respecto a la inicial y así obtener el ángulo del trazado realizado.

## 5.3. Ajuste de parámetros del sistema

Al momento de realizar las pruebas del sistema se encontró como problema principal que la estructura que sostiene el efector final se inclinaba, dejando de apoyar el efector en el plano de trabajo y que la fuerza de fricción entre la

rueda omnidireccional y el plano de trabajo no era la suficiente para desplazar el efector final. Para corregir estos problemas se tomaron las siguientes acciones:

- Se aumentó el peso de la estructura que sostiene al efector final al añadirle pesos en el interior de dicha estructura.
- Se cambió el punto de sujeción de la estructura que sostiene el efector final. Esto se realizó puesto que el brazo no estaba fijo a dicha estructura a la altura del eje de la rueda omnidireccional, lo que ocasionaba un momento de torsión que provocaba la inclinación de dicha estructura.
- Se redujo la fricción entre el pin de la estructura que sostiene el efector final y el brazo, para que en ese eje se tenga mayor libertad de rotación y reducir así la inclinación de la misma.

Otro parámetro que se reguló fue el corrimiento de la señal respecto al nivel de tierra. Esto se realizó, según lo visto en el punto 4.2.1.4, mediante el ajuste del valor del potenciómetro de precisión OFFSET hasta obtener 0VDC a la salida del circuito de acondicionamiento con el brazo totalmente retraído. Así mismo, se reguló la ganancia de este circuito hasta obtener 5VDC en el punto de mayor extensión del brazo mediante el ajuste del valor del potenciómetro de precisión GAIN.

Con respecto a la interfaz gráfica de usuario, se modificó la posición inicial en el eje Y para que el brazo robot opere su área de trabajo. Inicialmente el brazo robot debía de trabajar en todo el plano tamaño A3, pero debido a las limitaciones del brazo implementado y la posición inicial en el eje Y es de 230mm. Este cambio implicó calcular nuevamente los valores iniciales de las variables que definen estado del sistema ya que estos son fijados por el programador manualmente.

También se modificaron las dimensiones del brazo para la simulación gráfica porque estas variaron levemente con respecto a las dimensiones teóricas indicadas en el plano de la estructura.

## 5.4. Resultados

### Prueba de trazado de líneas horizontales:

Se considera como línea horizontal a cualquiera que sea paralela al eje Z. El trazado de líneas horizontales es realizado por el bloque de control de actuadores mediante el accionamiento del motor de corriente continua. Mediante la interfaz de usuario se pide al sistema que realice una línea horizontal de una longitud de 100mm. Esto se realiza 5 veces consecutivas en diferentes lugares del plano de trabajo, para así obtener el error máximo. Se evalúa la respuesta del sistema midiéndose el error en longitud del trazado, la desviación máxima respecto a la línea paralela ideal y el error angular.

Los errores a calcular son los siguientes:

- Error de longitud. Calculado según (ec. 31).
- Desviación máxima. Máxima desviación respecto a la línea paralela ideal.
- Error angular. Calculado según (ec. 32).

$$\text{Error de longitud} = \text{Longitud ideal} - \text{Longitud real} \quad (\text{ec. 31})$$

$$\text{Error angular} = \text{ángulo ideal} - \text{ángulo real} \quad (\text{ec. 32})$$

En la Figura 38 se aprecian los trazados realizados por el sistema en este caso. La línea roja es la paralela ideal al eje Z y los trazados en azul son los realizados por el sistema.

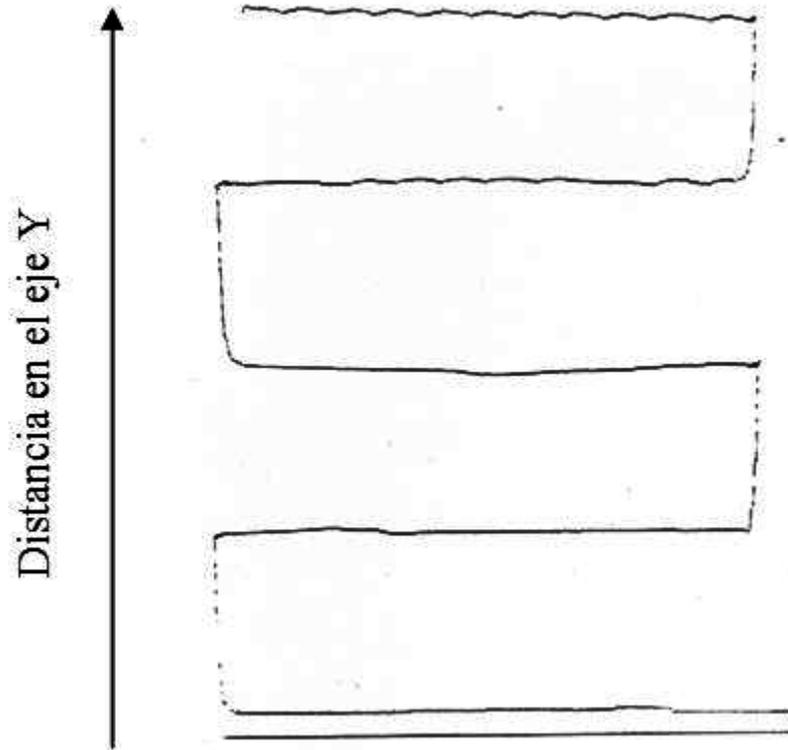


Figura 38. Prueba de trazado de líneas horizontales de 100mm. de longitud.

Se aprecia que la desviación respecto a la paralela ideal al eje Z aumenta conforme la distancia en el eje Y aumenta. Esto se debe a las características mecánicas del sistema, puesto que al aumentar la distancia en el eje Y aumenta también la componente de fuerza que desplaza el efector final aumentando la distancia en el eje Y. Luego el sistema intenta regresar a su posición deseada y debido a estas dos razones la línea presenta cierta deformación respecto a la paralela al eje Z ideal.

Los errores de longitud se presentan principalmente debido a la inercia que debe vencer el sistema al cambiar el sentido de movimiento. Esto genera fuerzas que ocasionan que inicialmente se desplace la base del brazo pero no el extremo donde se ubica el actuador, lo cual se traduce en que el sistema registra un desplazamiento mayor al real.

En la Tabla 11 se aprecian los resultados de las pruebas.

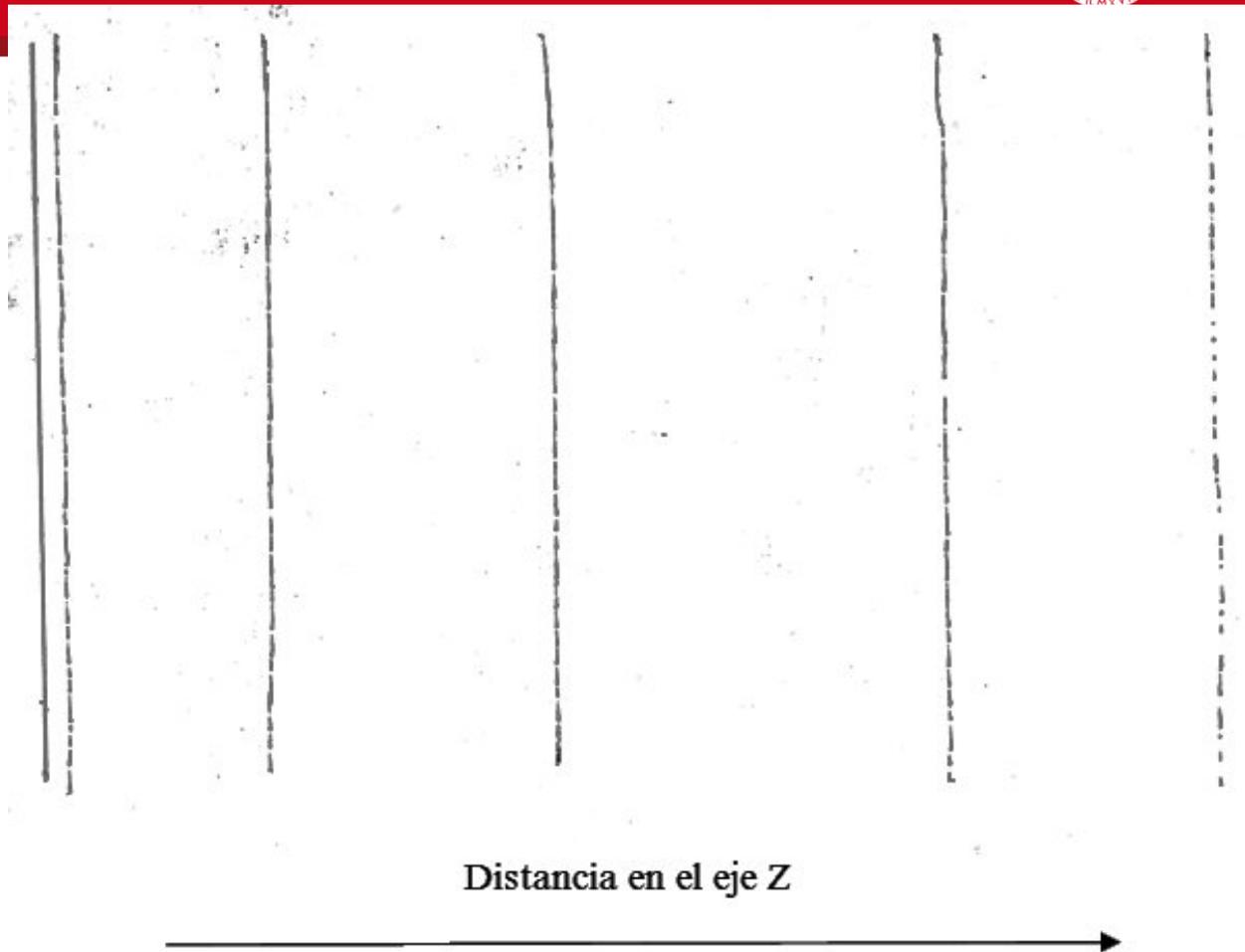
Tabla 11. Resultados de la prueba de trazado de líneas horizontales

Prueba	Error porcentual (%)	Desviación máxima (mm)	Error angular (°)
1	2	1	0
2	6	1.3	0.3
3	7	1.5	0.9
4	8	1	0.4
5	9	2.5	1.5

### Prueba de trazado de líneas verticales:

Se considera como línea vertical a cualquiera que sea paralela al eje Y. El trazado de líneas verticales es realizado por el bloque de control de actuadores mediante el accionamiento del motor a pasos. Mediante la interfaz de usuario se pide al sistema que realice una línea vertical de una longitud de 100mm. Esto se realiza 5 veces consecutivas en diferentes lugares del plano de trabajo, para así obtener el error máximo. Se evalúa la respuesta del sistema midiéndose el error en longitud del trazado, la desviación máxima respecto a la línea paralela ideal y el error angular.

En la Figura 39 se aprecian los trazados realizados por el sistema en este caso. La línea roja es la paralela ideal al eje Y y los trazados en azul son los realizados por el sistema.



*Figura 39. Prueba de trazado de líneas verticales de 100mm de longitud.*

En esta prueba se puede apreciar que la desviación es mucho menor en comparación a la prueba de trazado de líneas horizontales. Esto se da porque la fuerza necesaria para vencer la resistencia al movimiento que ejerce la caja reductora acoplada al tornillo sin fin es considerable. Esto implica que no se perturbará la posición Z fácilmente y por ende se tendrá una desviación menor.

El error de la longitud se debe principalmente a la variación angular de cada paso del motor, la cual es de  $7.5^\circ$  por paso. Considerando que el radio de la rueda omnidireccional es de 20 mm, cada paso da un desplazamiento de 2.6 mm por paso.

Tabla 12. Resultados de la prueba de trazado de líneas verticales

Prueba	Error porcentual (%)	Desviación máxima (mm)	Error angular (°)
1	-2	0.5	0.3
2	1.5	1	0.5
3	2	0.5	0.3
4	-1	0.2	0.1
5	-2	0	0

#### Prueba de trazado de líneas con 45° de inclinación:

La siguiente prueba consiste en activar ambos motores mediante la interfaz gráfica de usuario y controlar el sistema para realizar 5 líneas con una pendiente de 45° de 70.7 mm de longitud (un desplazamiento de 50mm para ambos ejes). Luego de realizar todos los trazos se procede a analizar la respuesta del sistema. Se indica el error de longitud, error en los ejes Y y Z y el error angular.

En la Figura 40 se aprecian los trazados realizados por el sistema en este caso. La línea roja es la línea que une los puntos inicial y final. Los trazados en azul son los realizados por el sistema.

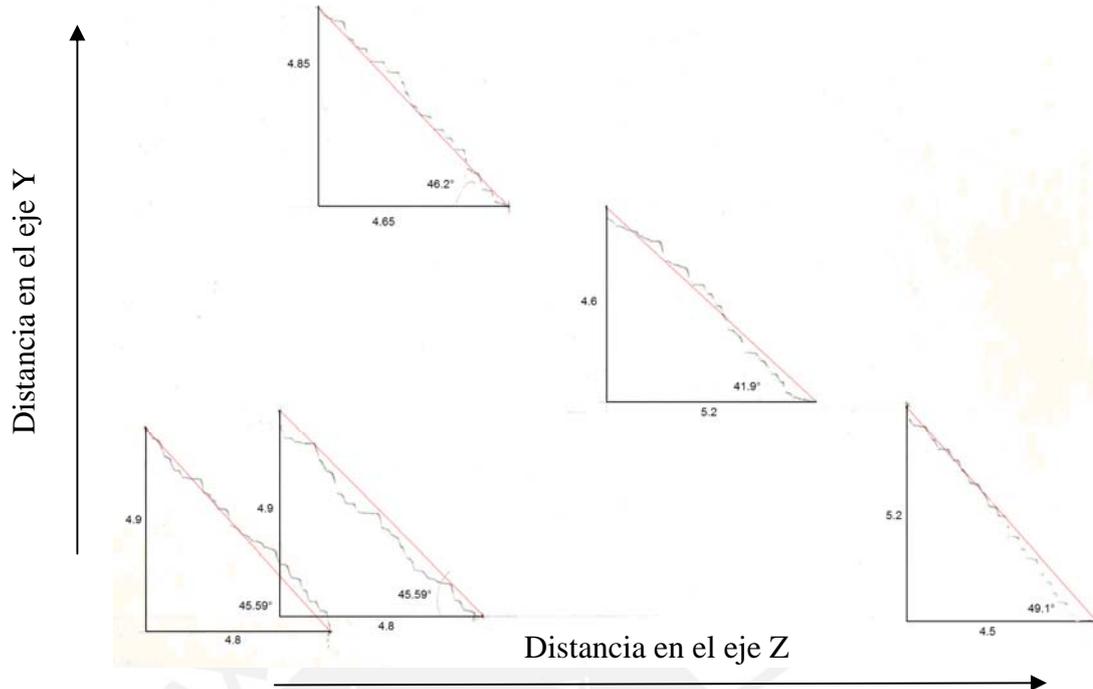


Figura 40. Prueba de trazado de líneas de 45° de inclinación.

En los resultados de la prueba se puede observar que los errores son más evidentes que en las pruebas realizadas anteriormente. Ya que se trata de una superposición de movimientos, los errores de los movimientos individuales también se superponen.

Además, al realizar un movimiento en diagonal los motores se deben de mover de forma sincronizada para que el efector final se sitúe en la posición deseada. En los movimientos individuales, el tiempo que tarda el sistema en responder no afecta a las gráficas, sin embargo en un movimiento diagonal si lo hace. Es por esta razón que en algunos puntos del trazo se visualizan líneas rectas completamente horizontales o verticales.

Tabla 13. Resultados de la prueba de trazado de líneas de 45° de inclinación

Prueba	Error porcentual de longitud (%)	Error porcentual en el eje Y (%)	Error porcentual en el eje Z (%)	Error angular (°)
1	1.74	-4	10	4.1
2	0.81	8	-4	3.1
3	2.96	5	-7	1.2
4	1.99	2	4	0.59
5	1.99	2	4	0.59

## CONCLUSIONES

1. El sistema de control empleado, a pesar de ser sencillo, es efectivo para esta aplicación y cumple con los requerimientos del caso. Esto ha sido demostrado con las pruebas realizadas.
2. La velocidad de desplazamiento en el eje Z está limitada la reducción de velocidad para obtener más torque, efectuada por la caja reductora acoplada al eje del motor y por la relación de conversión entre el número de revoluciones y el desplazamiento lineal.
3. La precisión del desplazamiento en el eje Y está limitada por el paso angular del motor utilizado y la precisión del potenciómetro utilizado para la realimentación del sistema.
4. A partir de los resultados obtenidos se puede decir que el funcionamiento del sistema presenta errores debido a factores mecánicos que no habían sido considerados al momento del diseño. Los circuitos electrónicos y los programas implementados para el presente proyecto funcionan correctamente.

## 5. RECOMENDACIONES

1. Para mejorar el desempeño del sistema se pueden escoger otro tipo de actuadores para el manejo del movimiento de los ejes, los cuales presenten características más recomendables para el trabajo.
2. Se puede mejorar la precisión en el eje Y aplicando técnicas de manejo de motores a pasos más complejas. Una de ellas es la técnica de micro-pasos, teniendo cuidado del requerimiento de torque, puesto que esta técnica reduce el torque incremental entre cada micro-paso. También puede lograrse un mejor desempeño acoplado una caja reductora en el motor encargado de este eje, logrando disminuir el paso angular y aumentando el torque, teniendo en cuenta que las dimensiones de la misma sean las apropiadas.
3. La precisión en el eje Z se puede mejorar mediante el empleo de más interruptores ópticos o de algún otro tipo de sensor rotatorio, teniendo en cuenta siempre la relación de conversión del movimiento rotacional a lineal del tornillo sinfín, pues si se emplea un sensor de amplia resolución la precisión sería imperceptible y por tanto se estaría sobredimensionando el sensor.
4. Es recomendable aumentar el peso de la estructura que contiene al efector final. Esto es debido a que en su estado actual el peso de la estructura no es el suficiente como para generar la fricción necesaria entre la rueda omnidireccional y la superficie de trabajo para el movimiento del brazo. Por esta razón para la etapa de pruebas se tuvo que agregar pesos externos para obtener una respuesta adecuada.

## BIBLIOGRAFÍA

- [1] Ricardo Soto, Rodolfo Pérez, Rodrigo Cuevas.  
ROBÓTICA [en línea]  
<http://www.industria.eda.cl/Academicos/AlexanderBorger/Docts%20Docencia/Seminario%20Aut/trabajos/vespertino%202005/Trabajo%20Robotica/t1.ppt>  
Universidad de Atacama – Colombia
- [2] T. Bojko.  
2004, “Educational Cartesian Robot Based on Linear Drives”, Faculty of Mechanical Engineering and Robotics. Cracow, Poland.
- [3] Ortega M.  
2005, “Diseño mecánico de un brazo manipulador tipo SCARA de 4 GDL”, Pontificia Universidad Católica del Perú.
- [4] DirectIndustry [en línea]  
<http://www.directindustry.es/cat/automatismos-regulacion-robotica/robots-scara-A-617.html>
- [5] Manufactura integrada por computadora: Robótica [en línea]  
<http://www.itchihuahuauii.edu.mx/academico/II/SFP/Robotica04.PDF>  
Instituto Tecnológico de Chihuahua II
- [6] Oscar Reinoso, Alejandro Filgueira, César Fernández, Luis M. Jiménez, Luis Payá  
Control híbrido con reorientación para trazado de figuras en superficies 3D [en línea]  
<http://isa.umh.es/arvc/documentos/articulos/ArticuloControlHibridoJA05.pdf>  
Departamento de Ingeniería de Sistemas Industriales  
Universidad Miguel Hernández de Elche – España
- [7] Chris H. Pappas, William H. Murray  
2002, “Visual C++.Net Manual de referencia”, McGraw Hill

- [8] Sistemas UNI  
2008, “Visual C++”, Universidad Nacional de Ingeniería
- [9] Microsoft Corporation  
2008, “Visual C++” [en línea]  
<http://msdn.microsoft.com/es-es/visualc/default.aspx>
- [10] Robertson Bayer  
2008, “Windows Serial Port Programming” [en línea]  
<http://www.robbayer.com/serial.php>
- [11] Javier Chang Fu  
Motores DC y Motores de Pasos  
Curso de extensión CETAM – PUCP
- [12] Eugene Blanchard  
The Using MOSFET Website – A better H bridge [en línea]  
<http://www.cadvision.com/blanchas/hexfet/>
- [13] Philips Semiconductors  
Power Semiconductor Applications
- [14] Cálculo de disipadores de calor [en línea]  
<http://www.terra.es/personal2/equipos2/disipadores.htm>