Universidad Michoacana de San Nicolás de Hidalgo

Facultad de Ingeniería Eléctrica

“Diseño y construcción de un sistema de control bola y placa basado en microcontrolador”

TESIS

Que para obtener el Título de

INGENIERO EN ELECTRÓNICA

Presenta

Jorge Luis García Ramírez

Asesor

Dr. José Juan Rincón Pasaye

Agradecimientos

A DIOS POR PONERME EN EL CAMINO CORRECTO, POR DARME FORTALEZA Y ENTENDIMIENTO PARA CONCLUIR ESTA ETAPA DE SUPERACIÓN.

A MIS PADRES:

DAMIÁN GARCÍA MARTÍNEZ.

DOROTEA RAMÍREZ LUQUIN.

POR EL AMOR Y APOYO INCONDICIONAL QUE ME HAN BRINDADO DURANTE TODOS LOS DÍAS DE MI VIDA, QUE A PESAR DE LAS ADVERSIDADES CONTRA LAS QUE HEMOS LUCHADO, SIEMPRE HAN DADO PRIORIDAD A LA FORMACIÓN ACADÉMICA DE SUS HIJOS. PAPÁ, MAMÁ HOY HEMOS CONCLUIDO ESTA CARRERA, MUCHAS GRACIAS.

A MIS HERMANOS QUE AUNQUE VIVIMOS ALGUNOS MOMENTOS DIFÍCILES, GRACIAS AL AMOR Y A LOS VALORES QUE NOS INCULCARON NUESTROS PADRES HEMOS PODIDO SALIR ADELANTE. UN AGRADECIMIENTO MUY ESPECIAL A MARY, QUE SIEMPRE SE HA PREOCUPADO Y NOS HA BRINDADO SU APOYADO DURANTE TODO EL TRAYECTO ACADEMICO.

A MI ASESOR, EL DR. JOSÉ JUAN RINCÓN PASAYE POR SER UNA GRAN PERSONA Y UN EXCELENTE ASESOR, POR TODO EL APOYO BRINDADO EN EL DESARROLLO DE ESTE PROYECTO. UN AGRADECIMIENTO AL ING.

FELIX JIMENEZ PEREZ POR COMPARTIR SU CONOCIMIENTO PARA LA ELABORACION DEL PROYECTO.

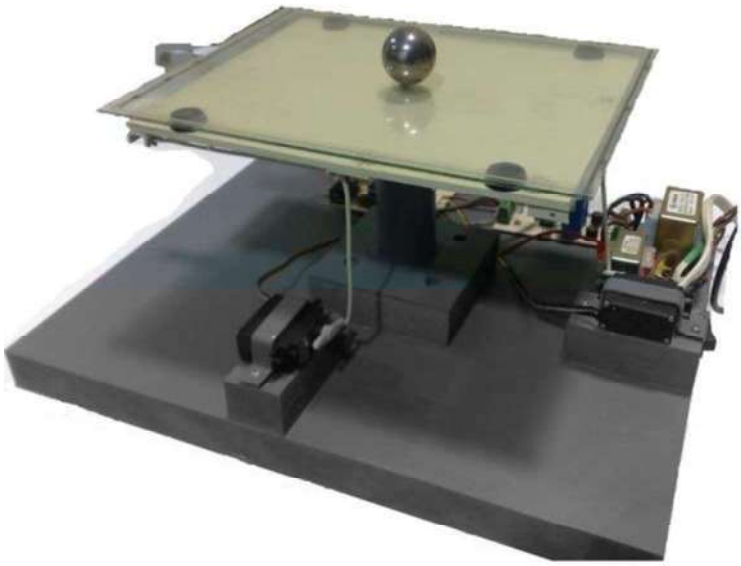
Dedicada a mis padres Damián García y Dorotea Ramírez.

Resumen

En la industria existen sistemas estables e inestables a los que se debe aplicar un esquema de control para estabilizar. Aunque la mayoría de los sistemas son estables existen muchos otros inestables.

En el presente trabajo se describe el diseño y construcción de un prototipo físico de control de una bola sobre una placa, el cual es un sistema por naturaleza inestable y que ha sido construido con fines didácticos, para realizar experimentos de control estabilizante.

Se describe el montaje del prototipo así como su instrumentación electrónica, incluyendo el acondicionamiento de sensores y actuadores, se desarrolla la programación del algoritmo de control Proporcional Integral Derivativo en un microcontrolador.



Lista de figuras

Contenido

CAPITULO 1

INTRODUCCIÓN

1.1 Introducción

Los sistemas de control en la actualidad son de vital importancia, están presentes en todas partes, los podemos encontrar en lugares tan cercanos y cotidianos como en el calentador de agua en nuestra casa, cuando el agua esta fría la *válvula* permite el paso del gas para que se convierta en calor, cuando ya está caliente la misma válvula bloquea el paso del gas, manteniendo así una temperatura adecuada del agua. En los elevadores de los edificios se utiliza el control para regular la velocidad, aceleración y posición. En la industria el control es utilizado para mover grandes equipos con una precisión que de otra manera seria casi imposible. En los automóviles, los robots y hasta en las aeronaves. [Referencia 1]

En el párrafo anterior se han mencionado sistemas estables como el control de temperatura del agua e inestables como el control de un elevador o una aeronave.

Cuando se controla un sistema estable el objetivo principal del control es mantener el punto de operación deseado aún en presencia de perturbaciones, de manera que si no hubiera controlador el sistema simplemente se va a un punto de operación no deseado pero estable, por ejemplo la temperatura del agua de uso doméstico podría ser más baja que la temperatura deseada.

En un sistema inestable el control es un elemento crítico, ya que sin éste el sistema opera de acuerdo a su naturaleza inestable, lo cual puede provocar

efectos catastróficos en el proceso para el cual está destinado, por ejemplo la caída de un elevador o de una aeronave**,** solo por citar algunos ejemplos.

En el presente proyecto se diseña y construye un sistema de control inestable consistente en un prototipo de bola y placa con la instrumentación adecuada para que los usuarios, que en este caso son estudiantes, trabajen con este tipo de sistemas, con el objetivo de familiarizarse con las dificultades en su modelado y control.

1.2 Antecedentes

En la Facultad de Ingeniería Eléctrica de la Universidad Michoacana de San Nicolás de Hidalgo se han desarrollado prototipos didácticos para la enseñanza del control automático. Un único sistema inestable, el *ball and beam* y, algunos sistemas estables como el control de motores de corriente directa y de nivel de líquido, este ultimo mostrado en la figura 1.1.

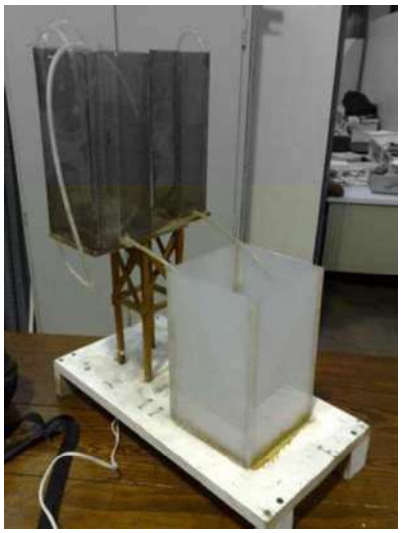


Fig. 1.1 Sistema de nivel de líquido.

El *ball and* beam es el único prototipo en el ámbito local, sin embargo a nivel nacional e internacional existen varios sistemas inestables de propósito didáctico.

Existen también prototipos didácticos comerciales de este tipo comercializados por empresas como Microchip, Feedback, National Instruments y Quanser, entre otras.

En la figura 1.2 se muestran dos prototipos inestables didácticos comerciales para la enseñanza del control. Del lado izquierdo tenemos un péndulo invertido rotativo, el *Entrenador de Ingeniería Quanser para NI ELVIS (QNET)* con un precio aproximado de MX$ 80,095 [Referencia 2] y, en la parte derecha un sistema bola y placa funcionando con sensores ópticos, su nombre es *Rotary Control Challenge, 2DOF Ball Balancer Experiment* [Referencia 3].

Fig. 1.2 Prototipos didácticos comerciales.

1.3 Objetivo

En el presente trabajo se tiene como objetivo general el diseño y construcción de un sistema bola y placa basado en microcontrolador, así como su esquema de control para estabilizarlo.

Para lograr el objetivo general se cumplirán los objetivos particulares como se muestran enseguida:

- Diseñar y construir un prototipo físico “bola y placa” basado en una pantalla sensible al tacto y dos servomotores.
- Construir los acondicionadores de señal y los actuadores.
- Obtener el modelo matemático del prototipo.
- Elaborar el software para el control del sistema usando el microcontrolador pic18f4550.
- El sistema debe proporcionar efectos visibles de estabilidad o inestabilidad.

1.4 Justificación

El prototipo propuesto proporciona efectos visibles al experimentador donde se pueden apreciar los efectos de inestabilidad y estabilización mediante un control retroalimentado.

Una vez construido el sistema inestable, permitirá realizar experimentos que de lo contrario sólo se podrían hacer en simulación.

Solo se ha desarrollado en la facultad el *ball and beam,* para la enseñanza del control de sistemas inestables y, los prototipos disponibles comercialmente tienen un costo muy elevado.

Con la construcción de un prototipo se economizará y además generará la experiencia y satisfacción de auto equipamiento.

1.5 Metodología

La metodología utilizada para el desarrollo del presente proyecto será dependiendo de la etapa del trabajo, se usarán diversas metodologías por ejemplo:

- Revisión bibliográfica acerca del control retroalimentado y manual del microcontrolador PIC18F4550 y PIC18F819.

- Diseño del prototipo, experimentación y medición de las variables de interés.
- Técnicas de diseño de circuitos para acondicionamiento de señales.
- Diseño de actuadores para servomotores.
- Diseño de controladores PID.
- Prueba y error en el prototipo real para realizar ajustes.

1.6 Descripción de los capítulos

En el capítulo 1 se encuentra una breve descripción del trabajo, antecedentes locales, nacionales e internacionales, objetivos, justificación y metodología que se utilizará en la elaboración del proyecto.

En el capítulo 2 el lector puede encontrar información sobre el diseño y la construcción de nuestro prototipo físico. Se describe el modo de operación de los elementos utilizados, diseño y características de los actuadores así como la especificación de los sensores y el controlador.

En el capítulo 3 se encuentran los diagramas de bloques, una breve explicación del protocolo de comunicación Inter-Integrate Circuit o I2C, también se describe el hardware de control de cada servomotor.

En el capítulo 4 se menciona el modelo matemático, así como las características del control en lazo abierto y lazo cerrado, el control PID aplicado, se describe el desarrollo del Software final del sistema, utilizando un PIC18F4550 con comunicación I2C, así como las pruebas realizadas y los resultados.

El capítulo 5 trata de las pruebas realizadas en sensor, servos, en lazo abierto y lazo cerrado con la sintonización del controlador PID ajustando las ganancias K_p, K_i y K_D, así como el periodo de muestreo y las perturbaciones.

En el Capítulo 6 se presentan las conclusiones finales del proyecto y los posibles trabajos futuros.

CAPITULO 2

DESCRIPCIÓN DEL PROTOTIPO

En presente capítulo se describe el proceso que se sigue para el diseño y la construcción del prototipo físico de bola y placa. Se describe cada uno de los componentes que se utilizan en el prototipo, se explica el modo de operación de la pantalla touchscreen (sensor), y los servomotores (actuadores), también se mencionan algunas de las características de los microcontroladores utilizados para la adquisición de datos y el control, así como las pruebas en lazo abierto.

Se denomina prototipo a un sistema implementado por primera vez que servirá de base para construir otros de la misma clase. [Referencia 4]

2.1 Proceso de diseño

El proceso seguido en el diseño de las partes mecánicas del prototipo se hace de manera empírica, dada la falta de experiencia en la teoría o el manejo de mecanismos o acoplamientos mecánicos y se puede resumir en los siguientes pasos:

1. Se propone una arquitectura general adecuada en dimensiones a la pantalla y los servomotores disponibles.
2. Se proponen esquemas de acoplamiento mecánico para cada eje y para el apoyo central.
3. Se montan los esquemas y se prueban, al encontrar problemas se descarta el esquema y se regresa al paso 2.

Por la falta de experiencia se tuvo que iterar varias veces entre los pasos 2 y 3 hasta encontrar los esquemas adecuados de acoplamientos y apoyo central. En la figura 2.1 se muestra en forma esquemática una vista lateral del prototipo, como quedará el sistema una vez ensambladas todas las piezas.

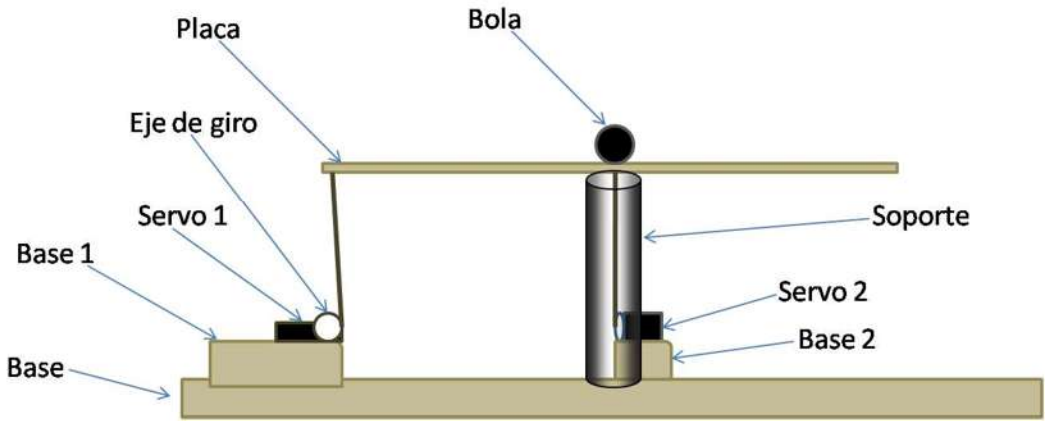


Fig. 2.1 Prototipo a construir.

2.1.1 Dimensiones del prototipo y materiales utilizados

Se eligió como base del prototipo el material denominado MDF (tablero de fibra de de densidad media) debido a su facilidad de corte y estabilidad de forma respecto al tiempo, además de la facilidad de dar un acabado final en barniz o pintura. En la figura 2.2 se ilustra la forma y dimensiones de la base de MDF. Las dimensiones que se escogieron, con el objetivo que el instrumento didáctico sea de fácil transportación pero a la vez resistente, la base es muy importante para lograr ambos objetivos.

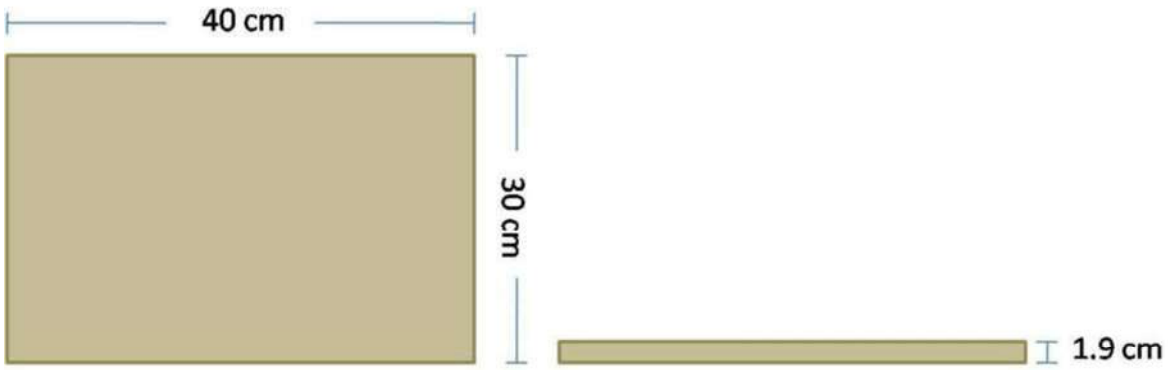


Fig. 2.2 Base del prototipo.

La placa (pantalla touchscreen) que utilizaremos mide 206.0mm x 271.0mm, es con la que se cuenta para el prototipo final, se muestra en la figura 2.3

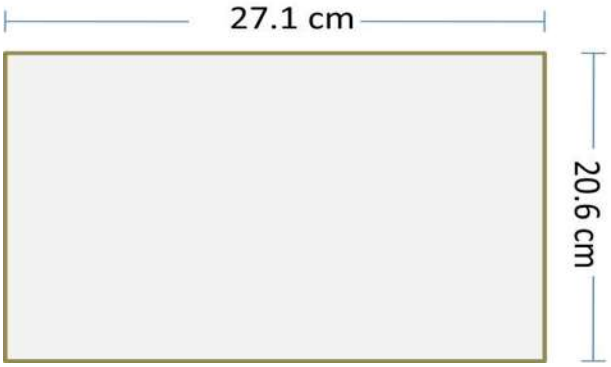


Fig. 2.3 Placa resistiva.

Los acoplamientos mecánicos fueron las partes que más trabajo y tiempo costaron, decidiendo finalmente lo siguiente:

Acoplo metálico para cada eje: Se utilizó un alambre metálico al cual se le dio la forma mostrada en la figura 2.4 y que permite acoplar el eje del servomotor correspondiente con un extremo de la base de la placa.

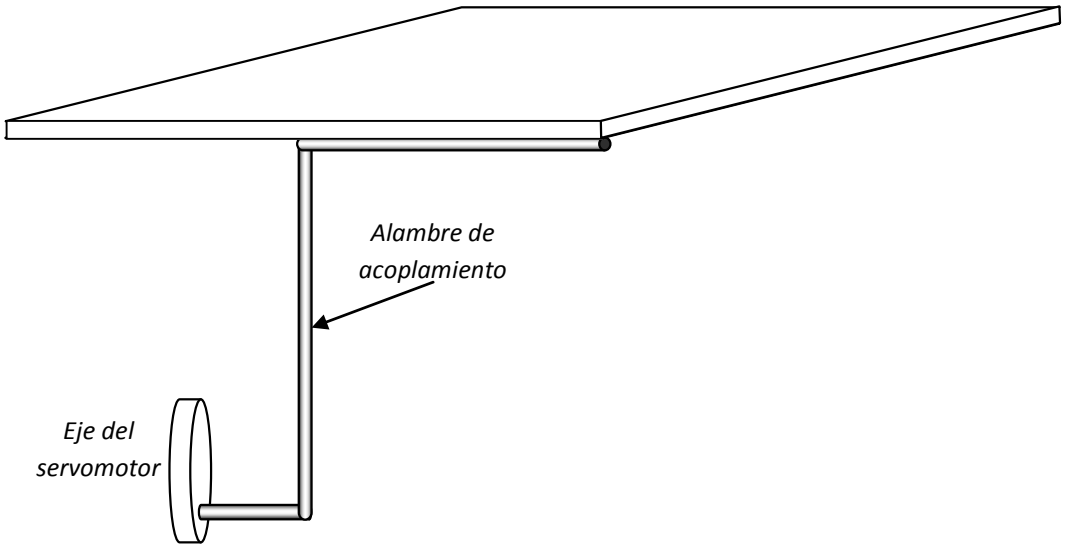


Figura 2.4 Acoplamiento entre servomotor y placa.

Apoyo central de la placa: Esta fue una de las piezas más difíciles de visualizar desde el principio. Al final se implementó con una bola de desodorante roll-on que puede girar en todas direcciones, pero que mantiene su centro geométrico en la misma posición. La bola está montada en una base cilíndrica como se muestra en la figura 2.5.

Figura 2.5 Bola central de apoyo de la placa.

2.2 Funcionamiento

El objetivo del sistema de control es mantener la bola en su posición de reposo en el centro de la placa, de manera que cualquier fuerza que trate de moverla de dicha posición se considera una perturbación.

Por ejemplo, si hay una perturbación en el sistema que tienda a desplazar la bola a la derecha, como se muestra en la figura 2.6, donde se muestra el efecto de una perturbación en dirección del eje x, con lo cual la bola se mueve en ese eje, mientras que sobre el eje *y* no hubo ninguna variación o recorrido, entonces el servo 1 deberá ejercer la acción de control necesaria hasta regresarla a su posición de reposo (posición central de la placa) como se aprecia en la figura 2.7.

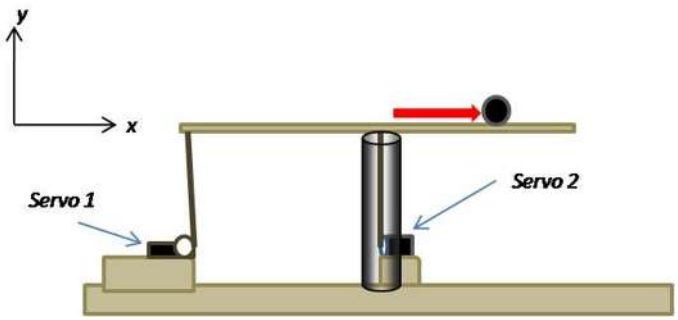


Fig. 2.6 Perturbación del sistema en el eje “x”.

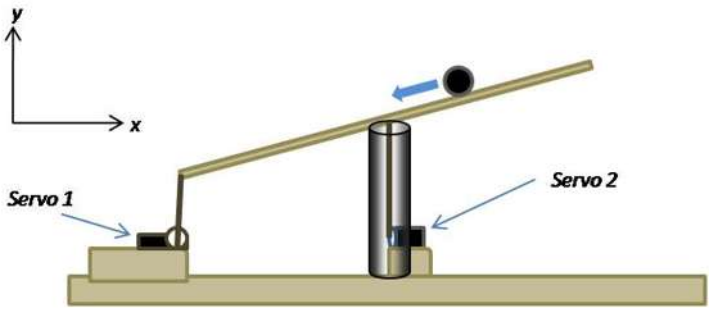


Fig. 2.7 Acción de control servo 1 para corregir perturbación en el eje “x”.

En forma similar, cuando ocurre una perturbación en dirección del eje y, la bola se moverá sobre éste eje *y*, el servo 2 será el encargado de estabilizarla, como se ilustra en la figura 2.8 y 2.9 respectivamente.

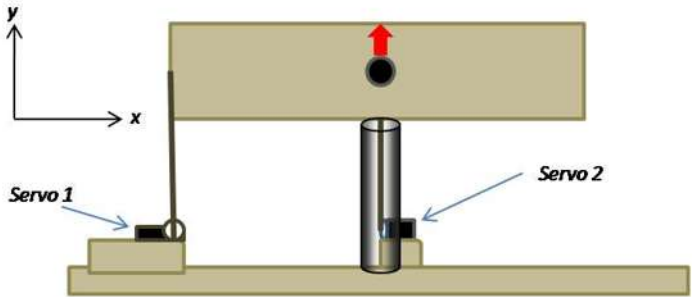


Fig. 2.8 Bola desplazándose en eje "*y*" debido a una perturbación.

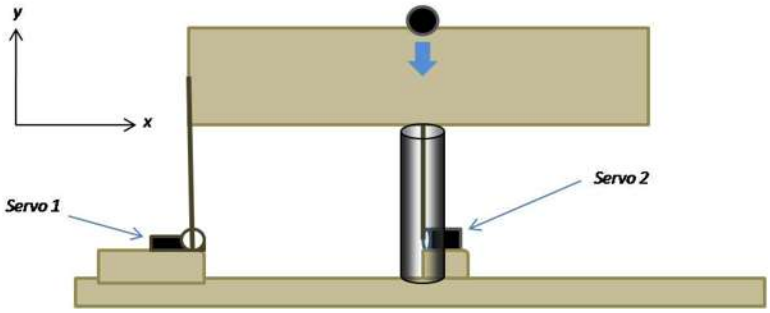


Fig. 2.9 Acción de control servo 2 para corregir perturbación en el eje "y".

Toda perturbación tendrá una componente en *x* y una componente en *y*, de manera que ambos servomotores se activarán para corregir los efectos de una perturbación.

En las figuras 2.10a, 2.10b y 2.10c se puede ver cómo quedó construido finalmente el prototipo.

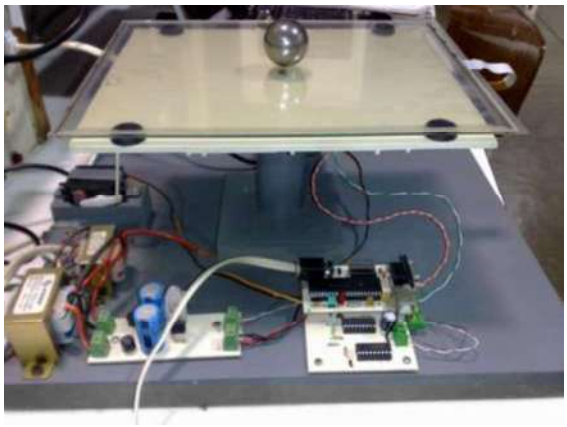


Fig. 2.10a Vista frontal.

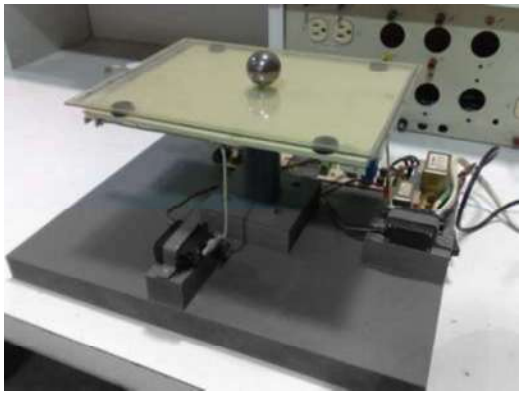


Fig. 2.10b Perspectiva 1.

Fig. 2.10c Perspectiva 2.

2.3 La pantalla touchscreen

Comercialmente se encuentran a la venta pantallas touchscreen basadas en diferentes tecnologías tales como:

- Capacitivas.
- De onda acústica superficial (también llamadas *SAW*, del inglés *Surface Acoustic Wave*).
- Iinfrarrojos.
- Galga extensiométrica.
- Imagen óptica.

- Tecnología de señal dispersa.
- Reconocimiento de pulso acústico.
- Resistivas.

Siendo esta última, la que se muestra en la figura 2.11, la que se utiliza en el presente proyecto.

Una pantalla touchscreen es una pantalla que mediante el toque directo sobre su superficie permite el intercambio de datos. El contacto se puede hacer con un lápiz, un dedo o cualquier otro objeto capaz de ejercer una presión moderada sobre su superficie.

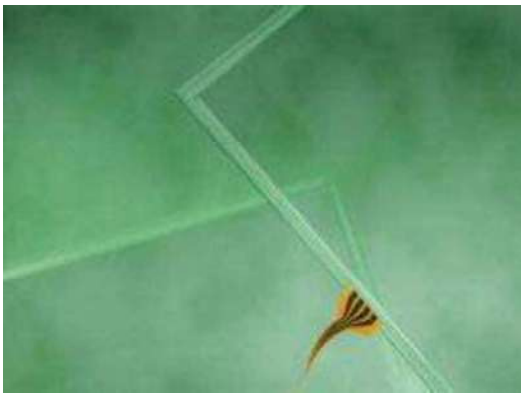


Fig. 2.11 Pantalla touchscreen.

Las pantallas touchscreen resistivas son las más utilizadas por su durabilidad aun en presencia de polvo y agua, además de que su precio es relativamente bajo y su principio de operación es sencillo, lo cual simplifica los circuitos de acondicionamiento. Las razones por las que se eligió la resistiva se muestran a continuación.

- Activación mediante cualquier objeto (en nuestro caso la bola).
- Respuesta de toque rápida, precisa y exacta.
- Superficie duradera, ya que el prototipo será utilizado por mucho tiempo y diversos usuarios.

- Disponible en varios tamaños 5.7"/6.0", 6.4", 8.4" y 10.4".
- Costo relativamente bajo. [Referencia 5]

En la figura 2.12 se muestra como está construida internamente la pantalla touchscreen resistiva.

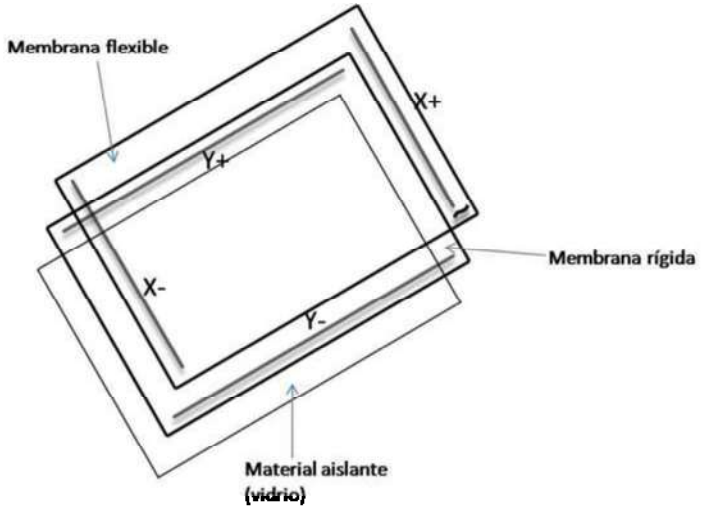


Fig. 2.12 Construcción de la pantalla resistiva.

El material aislante, o vidrio, sirve como base de la pantalla. Entre la membrana flexible y la rígida se encuentra un material resistivo transparente, de tal manera que la resistencia eléctrica varía dependiendo del punto en el que la membrana flexible hace contacto con la membrana rígida. Se puede ver como si cada membrana fuera en si una resistencia distribuida uniformemente, una en el eje "*x*" y otra en el eje "*y*". La figura 2.13 nos sirve de ilustración, en esta se tiene la pantalla como un arreglo de resistencias.

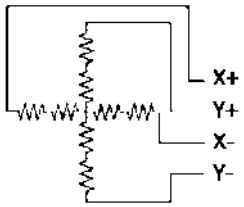


Fig. 2.13 Pantalla como un arreglo de resistencias.

Obtención de la posición de un objeto. Enseguida se explica cómo se toman las pruebas de la pantalla, para esto nos servirá de apoyo la figura 2.14 y 2.15.

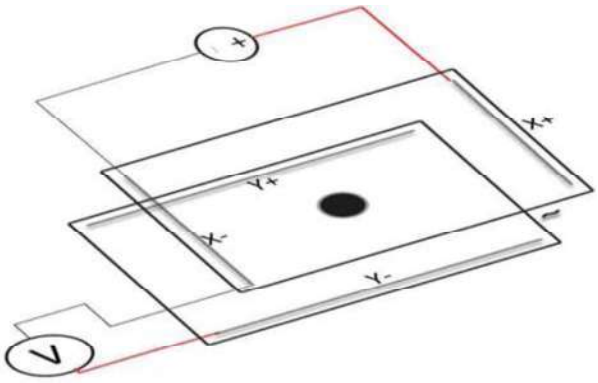


Fig. 2.14 Coordenada eje "*x*".

En la figura 2.14, al hacer circular una corriente por la membrana del eje x, en el momento que hacen contacto sobre la membrana flexible estas dos se juntan, entonces su resistencia disminuye y podemos medir la diferencia de potencial en el otro extremo, o eje *y*, el valor de voltaje lo podemos recoger conectando el positivo del medidor a *y*+ ó *y*-, el valor resultante es el mismo; nótese en la figura 2.14 que el común se debe conectar al negativo de la fuente con la que se alimenta el otro extremo. De esta manera tenemos el valor resistivo correspondiente a la coordenada *x* del punto presionado.

Para encontrar la coordenada *y* se hace de manera similar, figura 2.15, ahora alimentado el eje *y* para medir en el eje *x.* Con esto tenemos las coordenadas (*x, y)* del punto presionado. [Referencia 6]

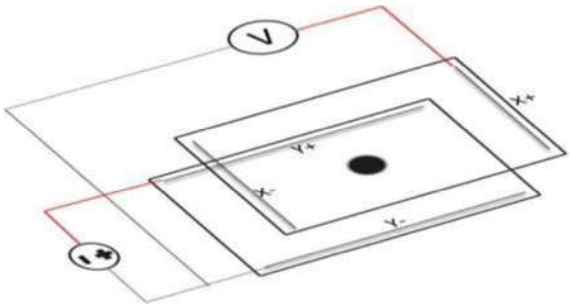


Fig. 2.15 Coordenada eje "*y*".

La pantalla utilizada en nuestro prototipo es la pantalla NJR-1210-IN-W4R del fabricante NJYTOUCH, se muestra en la figura 2.16, la cual tiene las siguientes dimensiones:

- Dimensiones externas: 206.0mm x 271.0mm.
- Dimensiones de área transparente: 192.0mm x 254.0mm.
- Dimensiones de área activa: 186.0mm x 248.0mm.

Fig. 2.16 Pantalla utilizada.

En el lado derecho de la imagen se puede apreciar el conector de cuatro terminales, para adquirir los resultados.

Ya que se obtienen las coordenadas de la bola en la placa. Se procede a capturar los valores de la posición de la bola en diferentes puntos, con el propósito de caracterizar la respuesta de la pantalla, mismos que se muestran en el capítulo 5 como pruebas realizadas.

2.4 Los servomotores

Algunos sistemas que poseen cilindros hidráulicos también pueden ser servocontrolados. Así que los servomotores no necesariamente tienen que ser motores giratorios, es decir, también hay servos no rotativos. En la figura 2.17 se presentan algunos tipos de servos tanto lineales como giratorios.

Fig. 2.17 Diferentes tipos de servos.

En el presente trabajo, se le denomina “servo” a un servomotor.

El servomotor, conocido como servo es un motor eléctrico que puede ser controlado tanto en la posición del rotor como la velocidad. En este caso se utiliza un servomotor controlado en posición del rotor; al enviar una señal codificada a la entrada podemos variar la posición angular del rotor, de tal manera que si queremos que la posición del engranaje no cambie se puede hacer del mismo modo, con una señal especifica en la entrada. Los servos cuentan con un “encoder” que es el que indica la posición.

El servo seleccionado para este proyec.to es el HS-322HD de Hitec, el cual presenta las siguientes caracteristicas.

- Opera en un rango de 4.5 a 6 volts, se controla utilizando una señal PWM.
- Par de 3kg/cm.
- Respuesta lineal.
- Consume poca energia.
- Bajo costo.

El servo alto torque para su tamaño, el motor es pequeño, pero por su sistema de control y juego de engranes hace posible proporcionar un par de hasta 3Kg por cm.

El par máximo necesario para mover el sistema de bola y placa ocurrirá cuando la bola esté en el extremo de la placa ver figura 2.18.

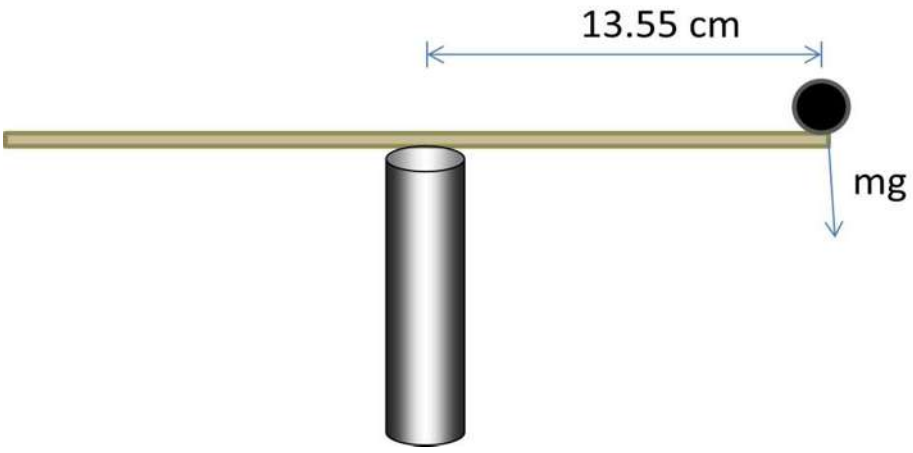


Fig. 2.18 Par máximo.

Por lo tanto el par máximo esta dado por la ecuación 2.4.1.

$$\text{Par máximo} = 13.55 * mg \qquad (2.4.1)$$

Donde mg es la masa, en kilogramos, de la bola utilizada.

Este servo cuenta con **tres terminales**: alimentación (rojo), GND (negro) y control (el cable amarillo). En la figura 2.19 se presenta el servo motor modelo HS-322HD.

Fig. 2.19 Servo HS-322HD.

El rango de operación del servo con el que se trabaja va desde 0 hasta 180 grados. En la figura 2.20 se muestran las características y periodos de operación de la señal pulsante de control y los correspondientes grados de rotación para nuestro servo, el Hitec 322HD. La posición angular del servomotor se controla con el ancho de pulso de la señal de control como se muestra en la figura 2.20. La señal de control puede variar su ancho de pulso desde un mínimo de 0.5 mseg. hasta un máximo de 2.5 mseg. provocando que la posición angular correspondiente varíe desde 180° hasta 0°, pasando por el punto medio a 90° para un ancho de pulso de 1.5 mseg.

La señal enviada al servo debe repetirse (actualizarse), cada 20 mseg para “refrescar” su posición, ya que si no se refresca, el servo regresa a la posición de reposo.

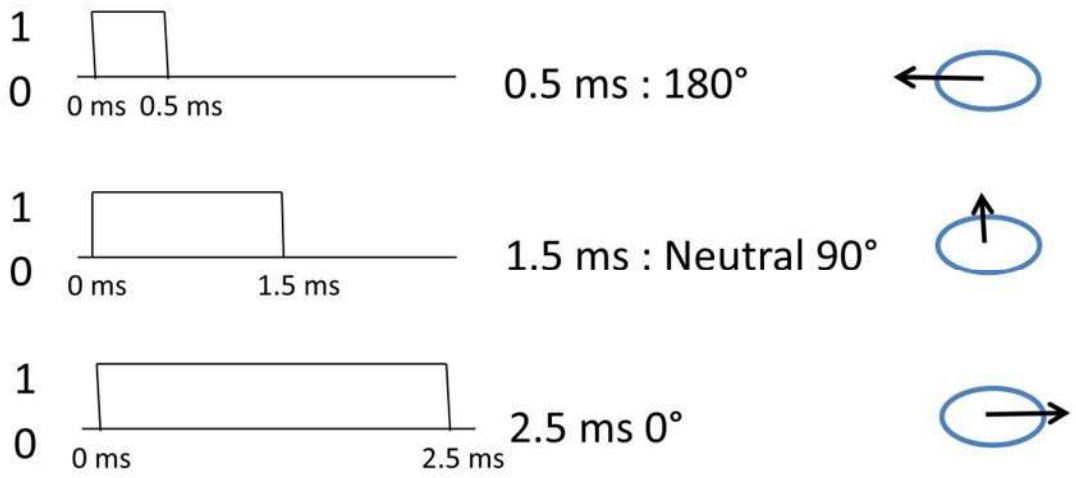


Fig. 2.20 Ancho de pulso necesario para diferente ángulo de giro.

Ahora bien, si se desea una posición que esté entre alguno de estos valores antes mencionados, se puede hacer con facilidad, ya que la respuesta del servo es lineal y cumple la relación de la ecuación 2.4.2.

$$\text{Ancho de pulso} = (0.5+(180-\Theta)/90)\ \text{mseg} \qquad (2.4.2)$$

Por ejemplo para hacerlo girar a 135° se tendrá que generar una señal con duración en alto de 2 ms. Con una duración en alto de 1 ms el servo va a girar a una posición de 45° y ahí se mantendrá hasta que reciba una señal con una duración diferente.

2.5 El microcontrolador PIC18F4550

Un microcontrolador es una microcomputadora dentro de un chip (figura 2.21), está compuesto de tres unidades fundamentales de una computadora: CPU, Memoria y Unidades de E/S.

Utilizamos el microcontrolador PIC18F4550 de Microchip por ser este el estudiado actualmente en los cursos curriculares de microcontroladores, además de su fácil manejo y mínimo de circuitos externos para su funcionamiento, en la Facultad de Ingeniería Eléctrica de la Universidad Michoacana de San Nicolás de Hidalgo, en donde se empleará el prototipo.

Fig. 2.21 PIC18F4550 en su empaquetado tipo DIP de 40 pines.

El PIC18F4550 es el microcontrolador principal del prototipo construido como se muestra en el diagrama de bloques de la figura 2.22.

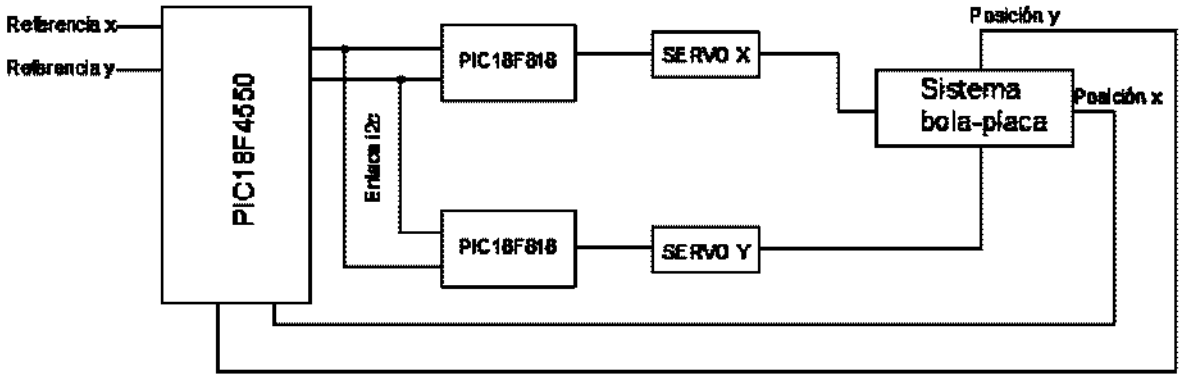


Fig. 2.22 Diagrama bloques.

En nuestro sistema el pic tiene las siguientes funciones:

- Establecer la referencia (*x,y*), por medio de los pines RA2 y RA3 respectivamente. Se habilita el convertidor analógico-digital y, con un voltaje de 0 a 5 volts, conectando un potenciómetro se variar el dato.
- Recibir los datos de la posición (x, y) de la bola en la pantalla (sensor), conmutando los pines entre entradas y salidas, alimenta el eje "*y*" y mide en el eje "*x*", luego alimenta el eje "*x*" y mide el eje "*y*".
- Utilizando la comunicación i2c en modo maestro-esclavo, el PIC18F4550 es programado con el software de control para enviarle los datos a dos esclavos. Siguiendo el algoritmo del controlador PID.

A continuación se describen las características del microcontrolador PIC18F4550, mostrado en la figura 2.23.

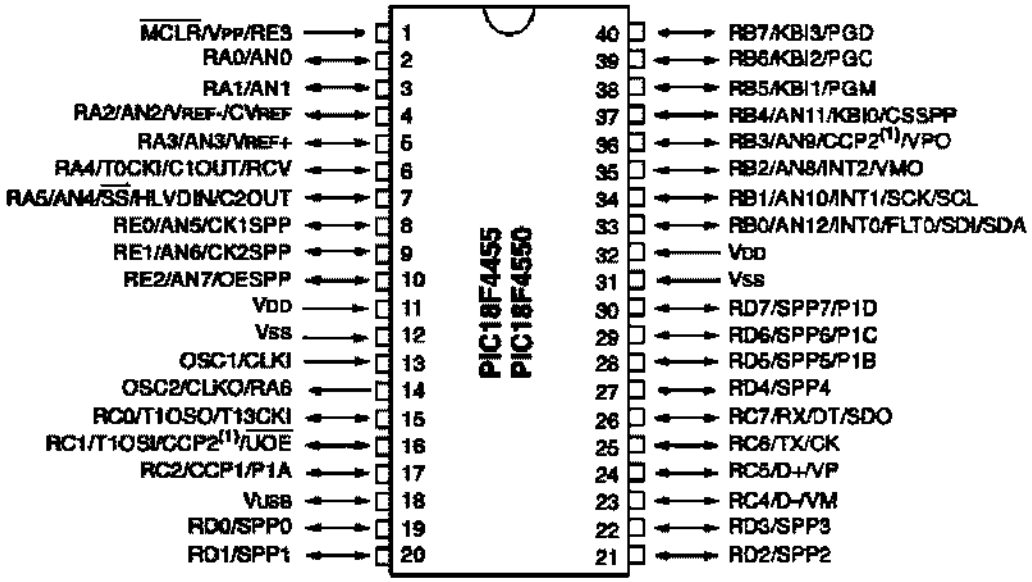


Fig. 2.23 Configuración de pines.

En la figura 2.24 se ilustran las características del PIC18F4550.

Features	PIC18F2455	PIC18F2550	PIC18F4455	PIC18F4550
Operating Frequency	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz
Program Memory (Bytes)	24576	32768	24576	32768
Program Memory (Instructions)	12288	16384	12288	16384
Data Memory (Bytes)	2048	2048	2048	2048
Data EEPROM Memory (Bytes)	256	256	256	256
Interrupt Sources	19	19	20	20
I/O Ports	Ports A, B, C, (E)	Ports A, B, C, (E)	Ports A, B, C, D, E	Ports A, B, C, D, E
Timers	4	4	4	4
Capture/Compare/PWM Modules	2	2	1	1
Enhanced Capture/ Compare/PWM Modules	0	0	1	1
Serial Communications	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART
Universal Serial Bus (USB) Module	1	1	1	1
Streaming Parallel Port (SPP)	No	No	Yes	Yes
10-Bit Analog-to-Digital Module	10 Input Channels	10 Input Channels	13 Input Channels	13 Input Channels
Comparators	2	2	2	2
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT
Programmable Low-Voltage Detect	Yes	Yes	Yes	Yes
Programmable Brown-out Reset	Yes	Yes	Yes	Yes
Instruction Set	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled
Packages	28-pin PDIP 28-pin SOIC	28-pin PDIP 28-pin SOIC	40-pin PDIP 44-pin QFN 44-pin TQFP	40-pin PDIP 44-pin QFN 44-pin TQFP

Fig. 2.24 Caracteristicas del PIC18F4550.

Este microcontrolador dispone de varias unidades funcionales, se nombran a continuacion algunas:

- 5 Puertos de entrada/salida que suman un total de 35 líneas de E/S
- Convertidor analógico-digital de 10 bits de resolución con 13 canales multiplexados.
- Señal de reloj de conversión configurable.
- Tiempo de adquisición programable.

- Posibilidad de establecer el rango de tensiones de conversión mediante tensiones de referencia externas.
- 4 temporizadores para generar las salidas pwm (que en este caso no se utilizan, debido a que se trabajara con frecuencias más bajas).

[Referencia 7]

De estos recursos del microcontrolador se utilizan los siguientes:

Pin 2: RA0 para la adquisición de la coordenada y.

Pin 3: RA1 para la adquisición de la coordenada x.

Pin 4: RA2 para la referencia x.

Pin 5: RA3 para la referencia y.

Pin 11: Alimentación 5V.

Pin 12: GND.

Pin 21: Entrada/salida digital 0/5V.

Pin 22: Entrada/salida digital 0/5V.

Pin 27: Entrada/salida digital 0/5V.

Pin 28: Entrada/salida digital 0/5V.

2.6 El microcontrolador PIC16F818

En la figura 2.22 se muestra la utilización de dos microcontroladores PIC16F818 en el prototipo construido. La función de estos microcontroladores es generar la señal tipo PWM de control para cada servomotor. Se opto por esta alternativa ya que la frecuencia de la señal estándar PWM que genera el PIC18F4550 es de mayor frecuencia que la requerida por los servomotores. De esta manera, cada

PIC18F818, está dedicado a controlar un servomotor con el valor de pulso recibido desde el microcontrolador maestro en el bus I2C. Ver figura 2.25.

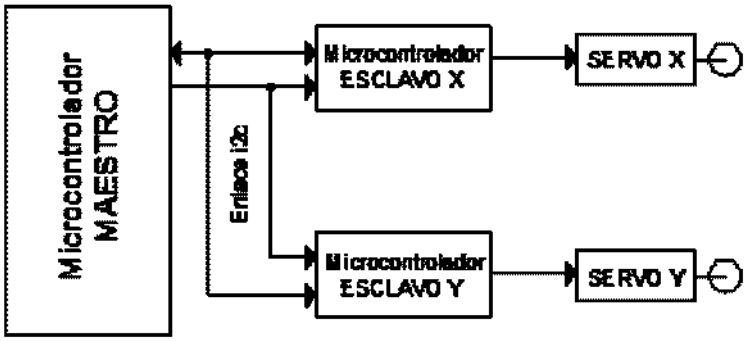


Fig. 2.25 Diagrama bloques maestro-esclavos.

En la figura 2.26 se muestra la configuración de pines del PIC16F818 en su encapsulado tipo DIP.

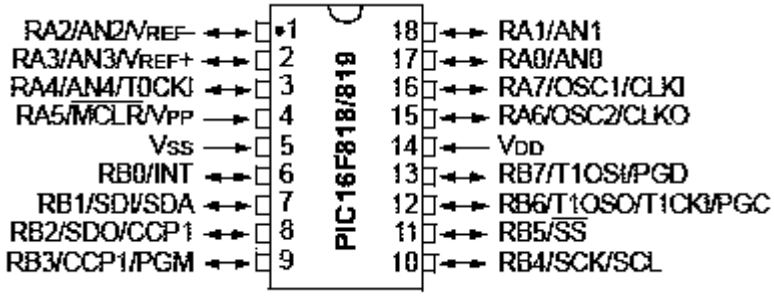


Fig. 2.26 PIC16F819 en su empaquetado DIP de 18 pines.

CAPITULO 3

INSTRUMENTACIÓN DEL PROTOTIPO

En el presente capitulo se describe la instrumentación utilizada en el prototipo, es decir, la medición de la posición (*x,y)* de la bola sobre la placa y la generación de la señal de control para cada servo. Se presenta también el diagrama de bloques para el sistema de control en lazo cerrado y, una breve descripción de la comunicación i2c utilizada en la parte de control de los servomotores.

3.1 El sensor de posición

Un microcontrolador maestro es el que inicia la transferencia de datos, genera la señal de reloj y finaliza la transferencia de datos. A continuación se describe el procedimiento para la adquisición de la posición de la bola, de manera práctica mediante los pines del microcontrolador maestro.

Como ahora se sabe la pantalla touchscreen funciona como un arreglo de resistencias, ver figura 3.0.

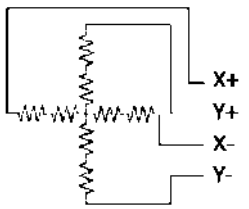


Fig. 3.0 Pantalla touchscreen como un arreglo de resistencias.

La posición (coordenadas *x, y*) se adquiere con el PIC18F4550 utilizando la configuración de la figura 3.1.

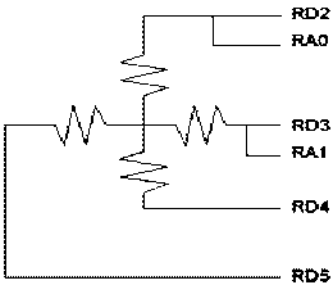


Fig. 3.1 Configuración pantalla-pines micro.

El método utilizado para obtener la posición de la bola requiere que se encuentren multiplexados RD2 con RA0 y, RD3 con RA1 para estar conmutando como entradas y salidas, dependiendo de la coordenada que se desee obtener. Por ejemplo, para obtener la coordenada del eje *y* (ver figura 3.2) se realiza de la siguiente manera:

- Con RD3 y RD5 se alimentan con +5V y tierra los extremos de la placa en la dirección "*y*".
- Se desconectan RD2 y RD4 (configurándolas como entradas logicas).
- Se activa el canal RA0 del convertidor analógico/digital, para recoger el voltaje analógico en el punto de contacto resistivo.

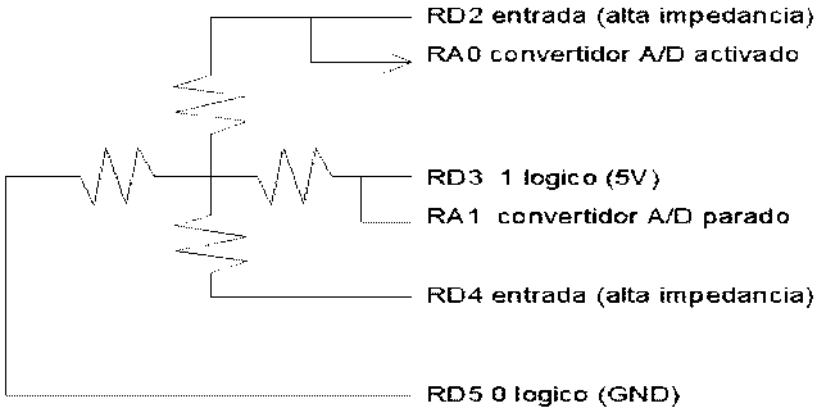


Fig. 3.2 Configuración para obtener coordenada "*y*".

En forma similar para obtener la coordenada *x,* se configuran las señales como se ilustra en la figura 3.3, se alimenta entre RD2 y RD4 y se activa el canal RA1 del convertidor analógico/digital. RD3 y RD5 como entradas con alta impedancia.

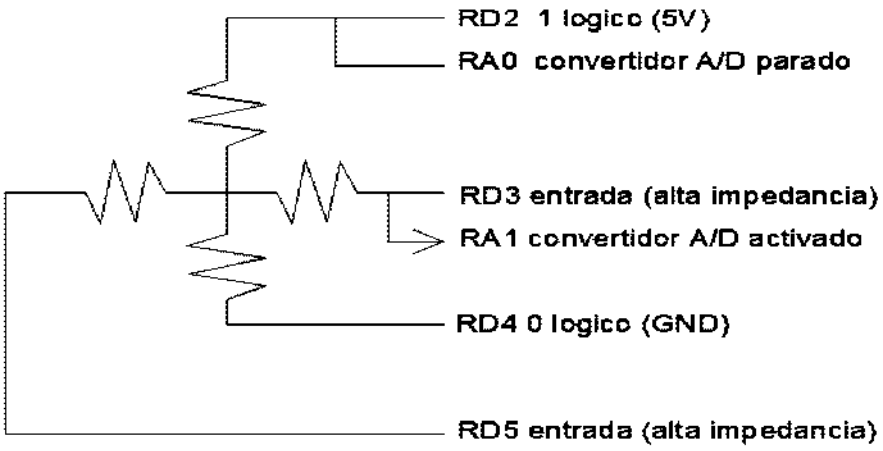


Fig. 3.3 Configuración para obtener coordenada "*x".*

La conexión de la pantalla con el pic maestro se observa en la figura 3.4.

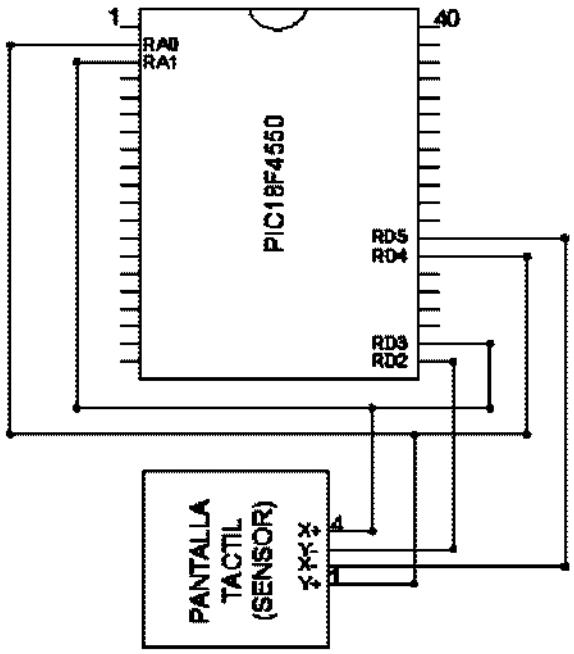


Fig. 3.4 Conexión sensor-microcontrolador.

3.2 El Actuador de los servomotores

Para controlar cada uno de los servos, se utiliza un microcontrolador, el cual se programa de manera previa, este microcontrolador da la oportunidad de controlar los servos utilizando el puerto I2C y asignar una dirección a cada uno de los servos.

El microcontrolador PIC16F818 en modo esclavo recibe los datos del microcontrolador maestro PIC18F4550.

Los pines del PIC16F818 son configurados de acuerdo a lo siguiente:

- Pines 1,2 y 3 asignan la dirección que tendrá el esclavo.
- Pin 4 se conecta a una resistencia de 22 KΩ.
- Pin 5 es GND.
- El microcontrolador PIC16F818 puede generar la señal PWM con una entrada analógica (esta no se utiliza en el presente proyecto) o bien, por medio de la comunicación I2C. Pin 6 puesto a 0 volts es para que el microcontrolador trabaje con señal analógica (no se utiliza), pin 6 puesto a 5 volts elige que el microcontrolador PIC16F818 trabaje en modo I2C.
- Pin 7 recibe los datos enviados por el maestro en modo I2C.
- Pin 8 y 9 seleccionan la frecuencia con la que opera el microcontrolador, en este caso 100 Hz.
- Pin 10 es la señal de reloj.
- Pin 11 salida de la señal tipo pwm enviada a los actuadores (servos).

La conexión del microcontrolador PIC16F818 es la mostrada en la figura 3.5. Donde las resistencias conectadas a la línea de reloj SCL y a la línea de datos SDA son de 2.2 KΩ. La resistencia conectada a MCLR es de 22 KΩ.

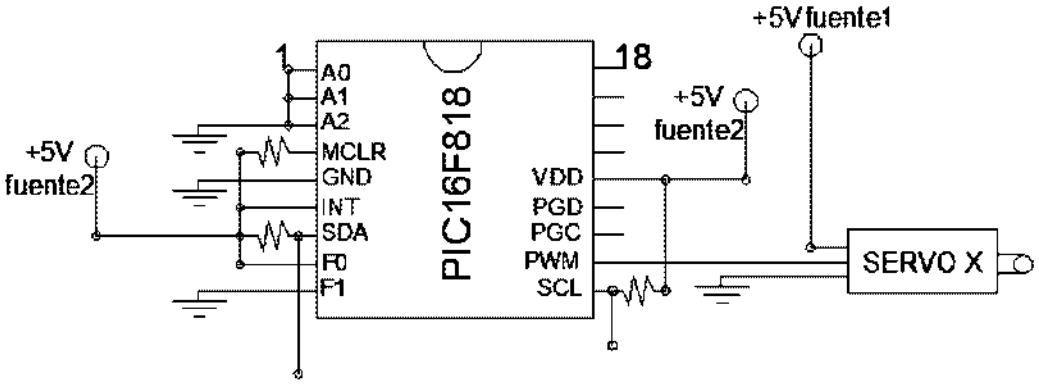


Fig. 3.5 Conexión del microcontrolador PIC16F818

En la figura 3.6 se muestra el diagrama de bloques del sistema de control para los servomotores basado en un microcontrolador maestro y dos esclavos dedicados a generar la señal tipo PWM, el control se realiza utilizando el bus I2C.

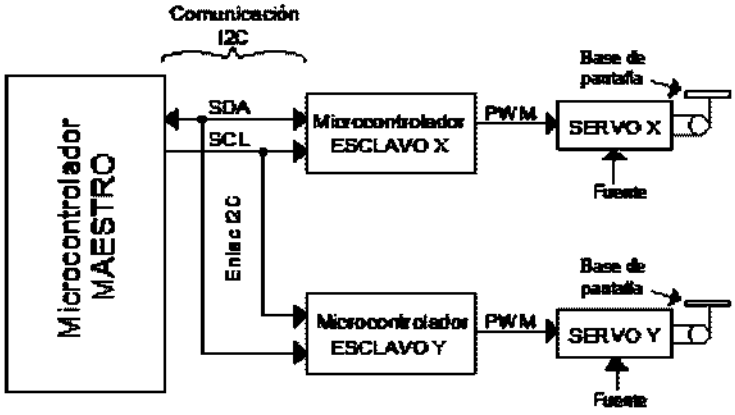


Fig. 3.6 Diagrama de bloques maestro-esclavos.

Cada vez que el microcontrolador maestro tiene un dato listo lo direcciona al esclavo correspondiente a través del bus I2C.

Los esclavos "*x*" o "*y*" reciben el dato de ancho de pulso del maestro y generan las señales tipo PWM de cada servo respectivamente, de acuerdo al dato de ancho de pulso recibido.

3.3 Protocolo de comunicación I2C

El protocolo de comunicación serie Inter-Integrate Circuit o i2c fue desarrollado por la empresa Philips. Este utiliza solo dos líneas para la transferencia de información, una de las líneas es la que transfiere datos, de manera bidireccional llamada SDA y; la de reloj para la sincronización de la información, esta línea es conocida como SCL, es unidireccional y es generada siempre por el maestro.

Este tipo de conexión aunque no puede alcanzar la velocidad de la conexión en paralelo, es muy usada debido a que el hardware a desarrollar es mucho más sencillo. Es multidireccional, es decir, podemos tener varios maestros y pueden estar enviando o recibiendo información a/o desde varios esclavos. En nuestro caso usaremos solo un maestro (ver figura 3.6) que envía información al esclavo "*x*" y al esclavo "*y*", [Referencia 8] de acuerdo a la siguiente secuencia:

- Maestro direcciona a esclavo "x".
- Maestro envía el dato a esclavo "x".
- Maestro termina la transferencia.

Con el protocolo I2C el microcontrolador maestro le envía datos continuamente a los dos microcontroladores esclavos.

NOTA: El software de los microcontroladores auxiliares fue facilitado por el Ing. Félix Jiménez Pérez, profesor en la Facultad de Ingeniería Eléctrica.

3.3.1 Resistencias Pull-Up

Las dos líneas (SDA y SCL) son de tipo “open drain” por lo que requieren ser conectadas a una resistencia conectada al voltaje de alimentacion +Vdd como se ilustra en la figura 3.7.

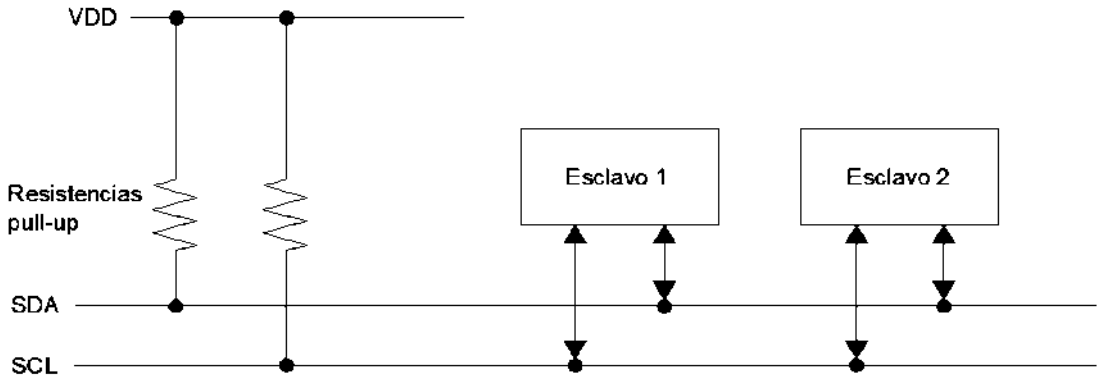


Fig. 3.7 Conexión de dos esclavos en I2C.

3.3.2 Transferencia de un dato

Ahora bien, el dato que se envía por la línea SDA debe ir acompañado de un pulso de reloj en la línea SCL como se muestra en el diagrama de tiempo de la figura 3.8.

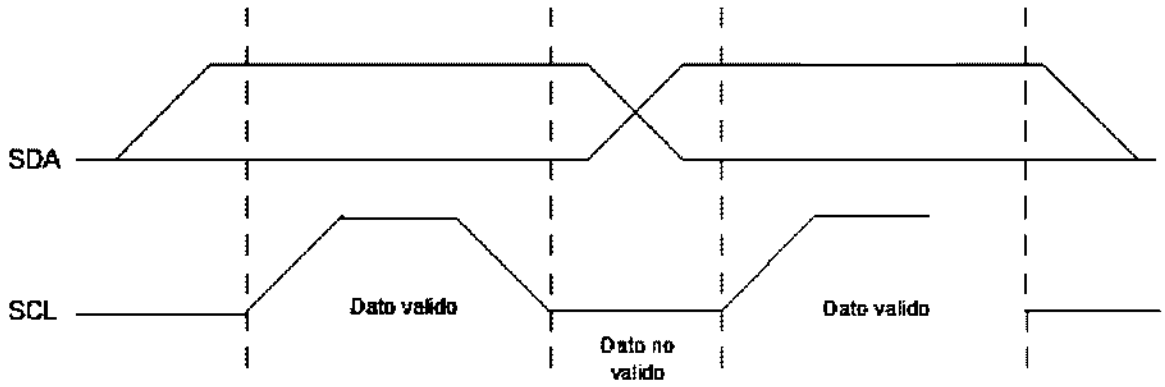


Fig. 3.8 Transferencia de un bit.

Basándonos en la figura 3.8 el dato será válido solo cuando el nivel lógico en la línea SDA sea estable mientras la línea SCL está en 1 lógico, la línea SDA solo puede cambiar de estado cuando la línea SCL sea 0 lógico.

En la figura 3.9 se ilustra la condición de inicio y paro de la transferencia de datos para el protocolo de comunicación i2c.

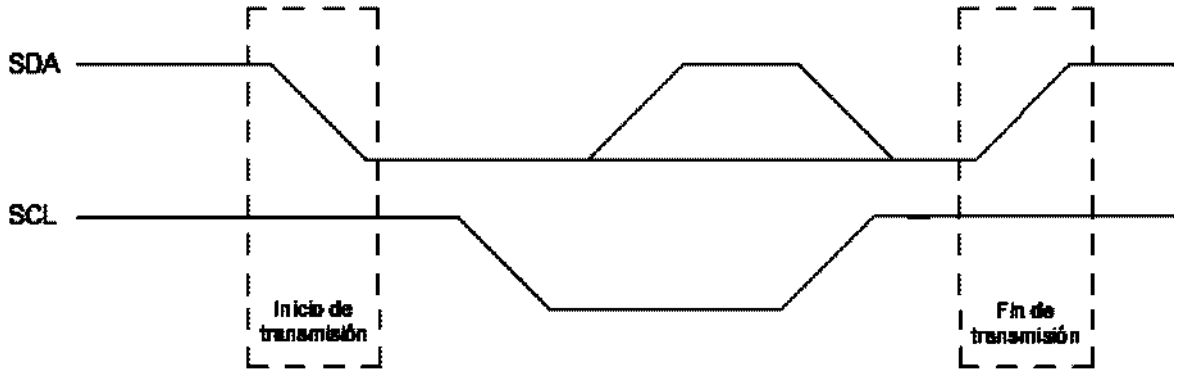


Fig. 3.9 Condición de inicio y fin.

Nótese que para iniciar la transmisión SCL está en alto mientras que SDA está en flanco de bajada, para finalizar la transmisión SDA está en flanco de subida y SCL debe estar en alto (1 lógico).

3.3.3 La condición de reconocimiento ACK

Los paquetes de datos enviados por la linea SDA son de 8 bits seguidos de un bit de reconocimiento o ACK. Comenzando por el bit más significativo. Como se muestra en la figura 3.10 durante el bit de ACK el esclavo pone en 0 lógico la linea SDA mientras el maestro pone en 1 lógico la linea SDA.

Si el esclavo por alguna razón no recibió el dato enviado por el maestro, no genera el bit de ACK, es decir no pone en 0 la línea SDA mientras el maestro generó el pulso de reloj numero 9.

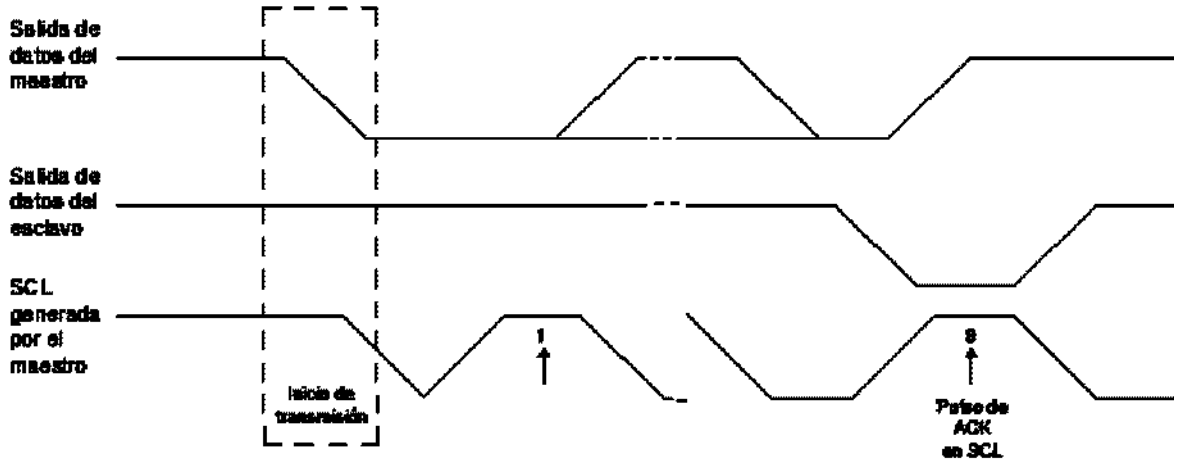


Fig. 3.10 ACK en la transmisión.

3.3.4 Formato del dato

Para ejemplificar el formato que lleva cada byte de información ver figura 3.11.

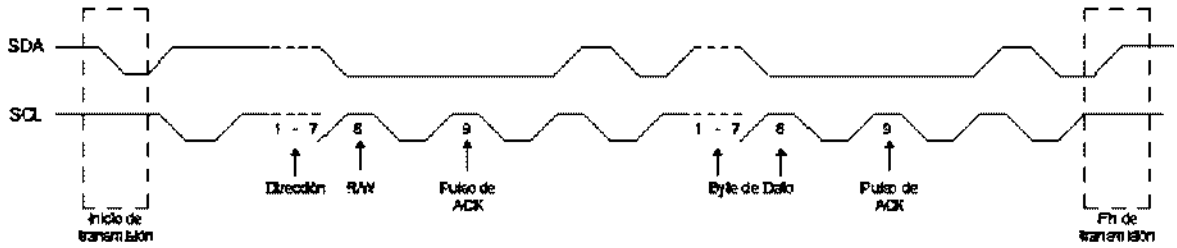


Fig. 3.11 Formato del dato.

Cuando se deja de enviar datos es necesario generar la condición fin. Si quiere enviar otro dato después de finalizada la transmisión es necesario generar la condición inicio.

La forma en la que nosotros comunicamos al microcontrolador maestro (PIC18F4550) con los microcontroladores esclavos (PIC´s16F819) es la mostrada en la figura 3.12 y lleva el siguiente orden:

- El maestro genera la condición de inicio de transmisión.
- Por la línea SDA envía la dirección, que previamente se le asigna, al esclavo *x.*
- Se le indica que la operación a realizar será escritura de datos (R/W=0).
- El maestro espera a que esclavo *x* responda con un ACK de reconocimiento de dirección, de esta manera el maestro sabe que el esclavo *x* está listo para recibir información.
- Se le envía el dato de 16 bits al esclavo *x*.
- El maestro espera hasta recibir un ACK generado por el esclavo *x,* es decir, cuando el esclavo genera el ACK es porque le ha llegado el dato correctamente.

Y se repite el mismo procedimiento para el esclavo *y.*

Fig. 3.12 Maestro envía datos a dos esclavos.

3.4 Diagrama esquemático del sistema electrónico

El diagrama completo del sistema es mostrado en la figura 3.13.

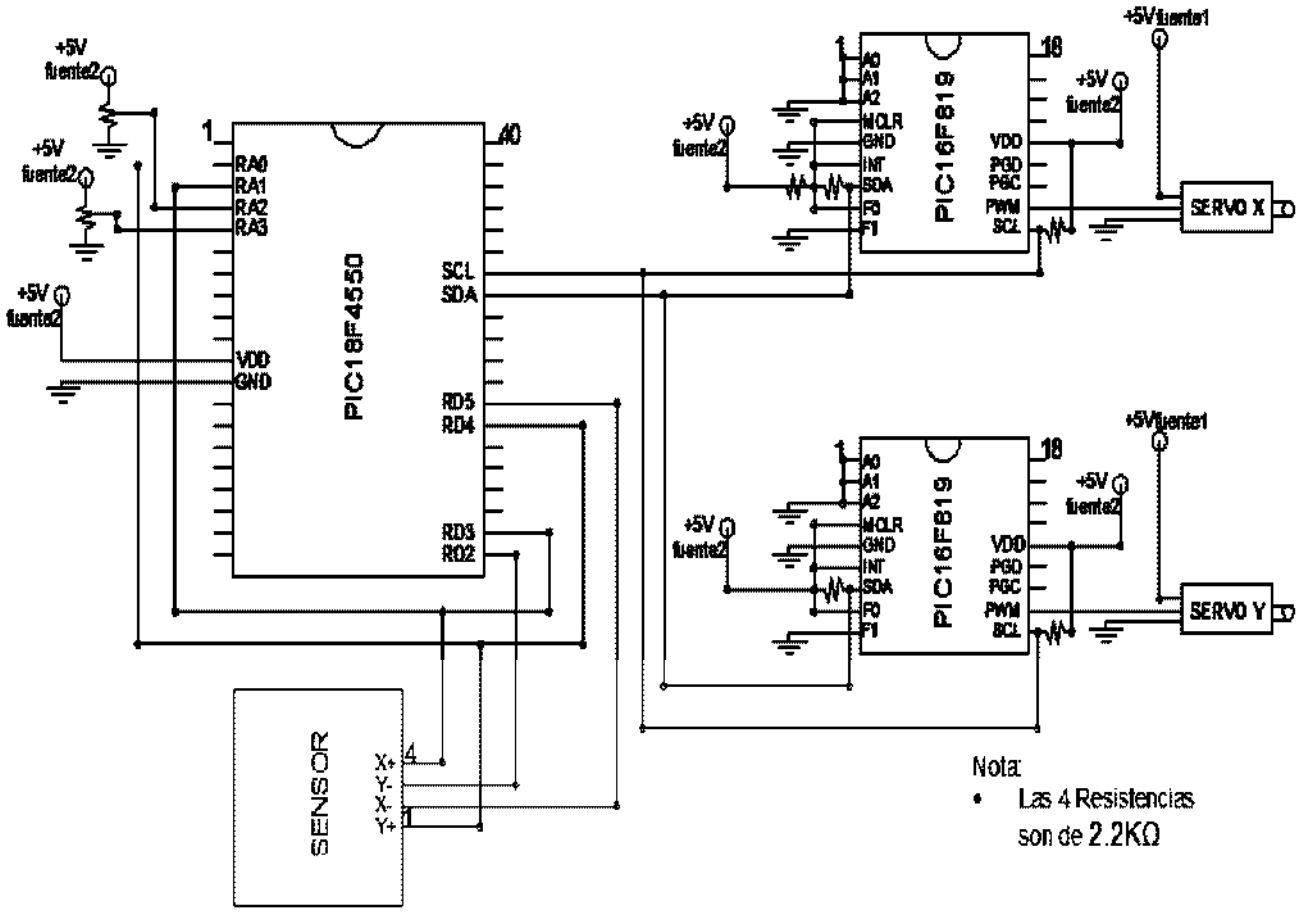


Fig. 3.13 Diagrama esquemático del sistema.

Se diseñaron 2 fuentes de voltaje, la “fuente1” alimenta solo a los 2 servomotores. La “fuente2” alimenta los potenciómetros, con los cuales se coloca la referencia para la bola; los 3 microcontroladores y al sensor.

Las fuentes son independientes, con la finalidad de no introducir ruido en las señales de control, ruido que se pude generar por los servomotores.

CAPITULO 4

MODELADO Y CONTROL DEL SISTEMA

En el presente capítulo se presenta el modelo matemático que gobierna el sistema de bola y placa. Se da una breve descripción del control PID (proporcional-integral-derivativo) se describe el desarrollo del software del control PID, la aplicación en lazo cerrado al sistema inestable bola y placa. Se muestran los algoritmos de los programas y los valores de las ganancias del controlador que se utilizaron para estabilizarlo y la manera en la que se utiliza la comunicación I2C.

4.1 Modelado Matemático

Para el modelo se considera que el movimiento en cada eje no afecta el movimiento en el otro eje, por lo tanto, se puede modelar por separado.

En la figura 4.1 se muestra esquemáticamente uno de los extremos de la placa, para ilustrar el modelo matemático del movimiento en uno de los ejes.

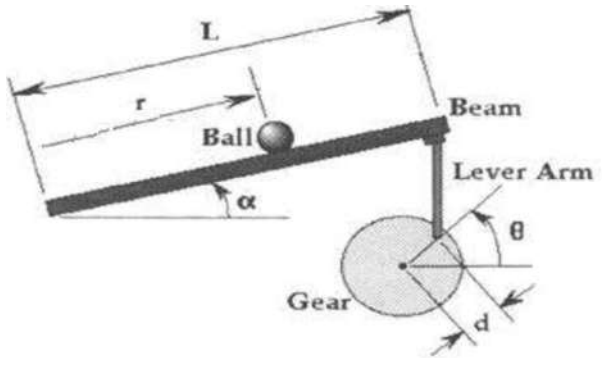


Fig. 4.1 Diagrama esquemático de uno de los extremos de la placa.

El modelo que describe el comportamiento del sistema en cada eje es de tipo no lineal y en forma simplificada está dado por la ecuación 4.1.

$$r'' = -k \sin \alpha \qquad (4.1)$$

Donde:

$$r'' = \frac{d^2 \mathrm{r}}{d^2 \mathrm{t}}$$

$$\alpha = \frac{\mathrm{d}}{\mathrm{L}}\ \Theta$$

La obtención de este modelo matemático se explica en [Referencia 9].

El modelo no lineal se puede simplificar considerando que α es pequeño y entonces se convierte en lineal, el cual se puede expresar por una función de transferencia que relaciona la entrada (ángulo θ controlado por el servomotor) con la salida (posición r de la bola) está dada por la ecuación 4.2.

$$\frac{R(s)}{\theta(s)} = -\frac{mgd}{L\left(\frac{J}{R^2} + m\right)} \frac{1}{s^2} \qquad (4.2)$$

Donde: J es el momento de inercia del eje del motor

m es la masa de la bola

g es la aceleración de la gravedad

L, r, d son las dimensiones físicas mostradas en la figura 2.9.

Como puede verse, el modelo lineal del sistema tiene la forma de un doble integrador, el cual se puede estabilizar mediante una acción de control proporcional-derivativa [Referencia 9].

4.2 El control

El control PID es un sistema de lazo cerrado, también conocido como control retroalimentado, que monitorea la señal de salida o variable controlada (posición *X* ó *Y*) y está continuamente comparándola con el valor de la señal de referencia obtenida, en nuestro caso de un potenciómetro externo, de tal manera que se puede corregir cualquier perturbación agregada al sistema. En la figura 4.2 se presenta el diagrama de bloques del sistema de control retroalimentado.

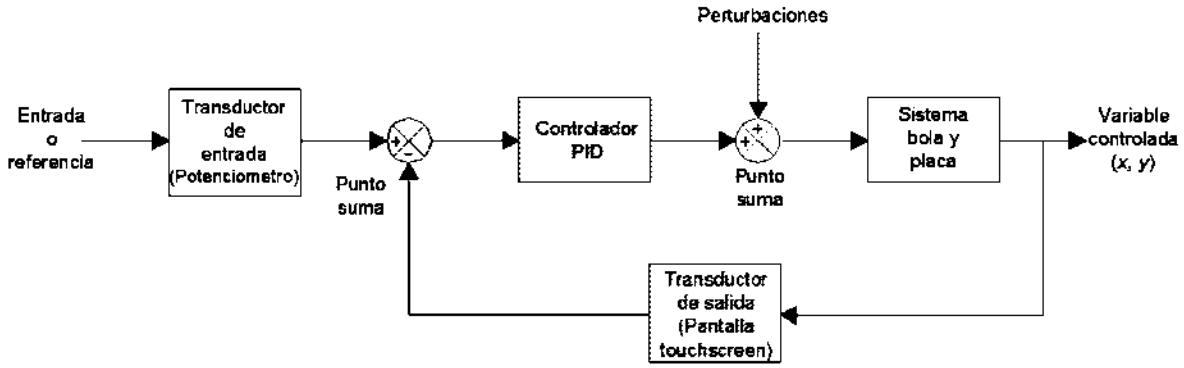


Fig. 4.2 Esquema lazo cerrado.

El controlador PID genera la señal de control de cada uno de los servomotores, la cual es el ancho de pulso de la señal pwm que establecerá su posición angular de acuerdo a la ecuación 4.3.

$$.\theta(f) = k_P\, e(t) + \, k_I \int_0^t e(t)dt + \, k_D \frac{de}{dt} \qquad (4.3)$$

Donde:

$e(t) = error\ actuante$

$e(t) = salida - referencia$

$k_P,\ k_I\ y\ k_D\ \ son\ las\ ganancias\ del\ controlador$

Las ganancias del controlador se deben elegir adecuadamente para lograr el comportamiento deseado del sistema en lazo cerrado.

4.3 Implementación del controlador PID

Como el controlador se implementa en un programa dentro del microcontrolador principal se requiere una versión discreta de la ecuación 4.3. La cual es como sigue. Ver ecuaciones 4.4 a 4.6.

Acción Proporcional $$P(k) = k_P\, e(k) \quad (4.4)$$

Acción Integral $$I(k) = k_I\, e(k) + I(k-1) \quad (4.5)$$

Acción Derivativa $$D(k) = k_D\, (e(k) - e(k-1)) \quad (4.6)$$

Donde:

$$e\,(k) = Salida\,(k) - Referencia\,(k)$$

$$k = Instante\ de\ muestreo\ actual$$

$$k-1 = Instante\ de\ muestreo\ anterior$$

Con las ecuaciones 4.4, 4.5 y 4.6 se genera la señal de control $\theta(k)$ para cada servo, esto da como resultado la ecuación 4.7.

$$\theta(k) = P(k) + I(k) + D(k) \quad (4.7)$$

4.4 Anti wind-up

Para disminuir el efecto indeseable de la saturación en el actuador, se incluye una protección en el programa para detener la acumulación del integrador (ver figura 4.5). Cuando el actuador está saturado es decir, *I(k)* solo se actualiza en el programa si el actuador no está saturado, como se muestra en el diagrama de flujo de la figura 4.3.

4.5 Software desarrollado

En la figura 4.3 se describe la lógica general del programa implementado. En esta figura:

- *Px* es la acción proporcional en el eje *x.*
- *Ix* es la acción integral en el eje *x.*
- *Dx* es la acción derivativa en el eje *x.*
- *Accion_x* es la acción total del controlador PID en el eje *x.*
- *Lee_x* es la posición de la bola en la pantalla.
- *Ref_x* es la referencia x, establecida por el potenciómetro en el pin RA2 del microcontrolador PIC18F4550.
- *Error_x* es la diferencia *lee_x* - *ref_x.*

El código completo se anexa en el apéndice A.

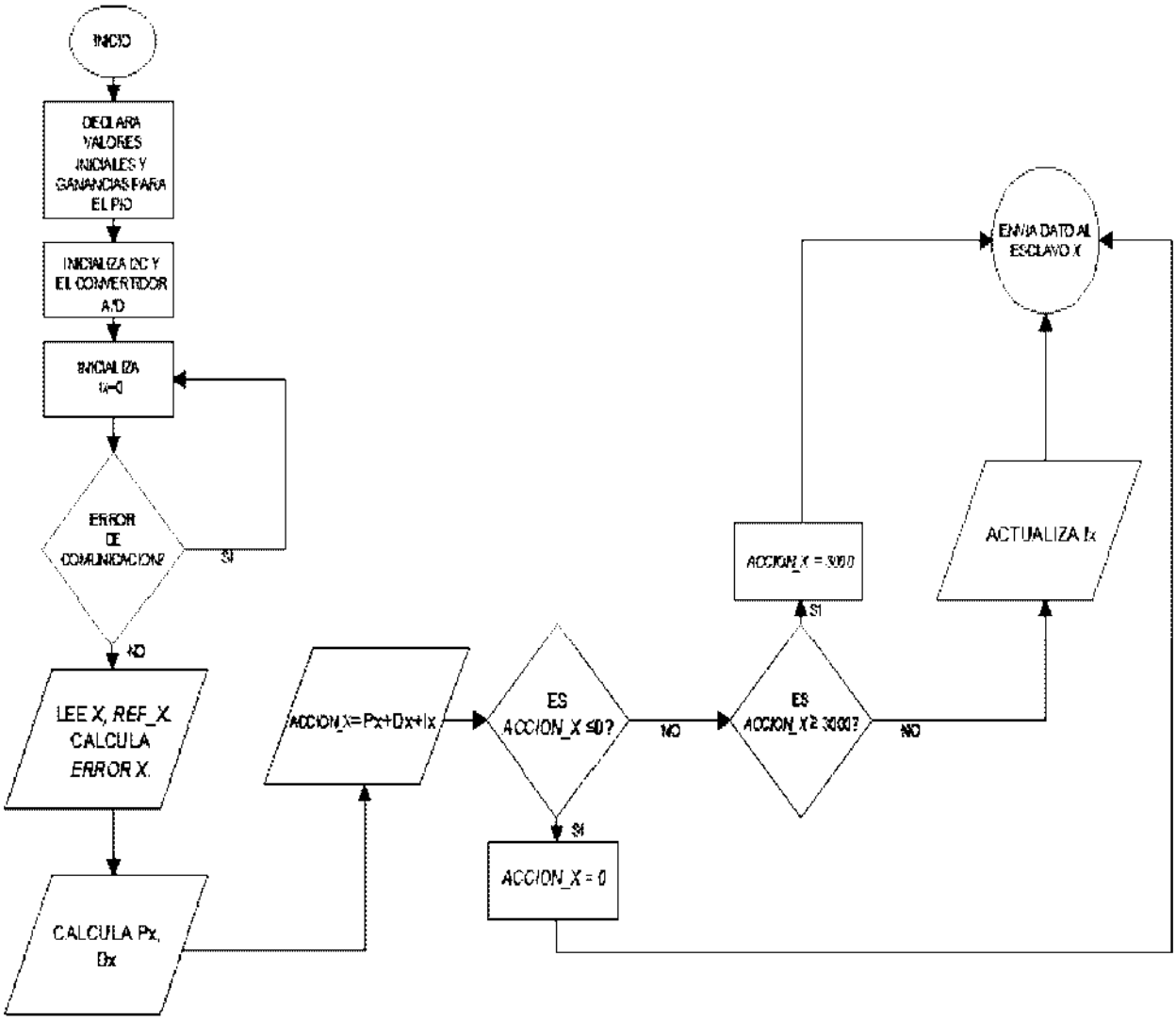


Fig. 4.3 Diagrama flujo para eje "*x*".

De manera similar se realiza la actualización de variables para la coordenada *"y"* y su control.

CAPITULO 5

PRUEBAS REALIZADAS

Este capítulo es dedicado a las pruebas que se realizaron a lo largo del proyecto, Pruebas a la pantalla touchscreen, a los servomotores y algunos programas diseñados para ver su correcto funcionamiento, así como obtener algunos valores necesarios para finalmente elaborar el software de control PID del sistema de bola y placa basado en microcontrolador, descrito en la sección 4.4.

5.1 Caracterización de los sensores

Las primeras pruebas realizadas se hicieron a la pantalla con la que al principio se contaba, una pantalla de 12cm x 10cm.

El eje de las *x* es de 12 cm y lo alimentamos con una fuente de 5 Volts, midiendo con el multimetro como se explica en la figura 2.14 del capítulo 2. Se obtuvo la grafica de la figura 5.0.

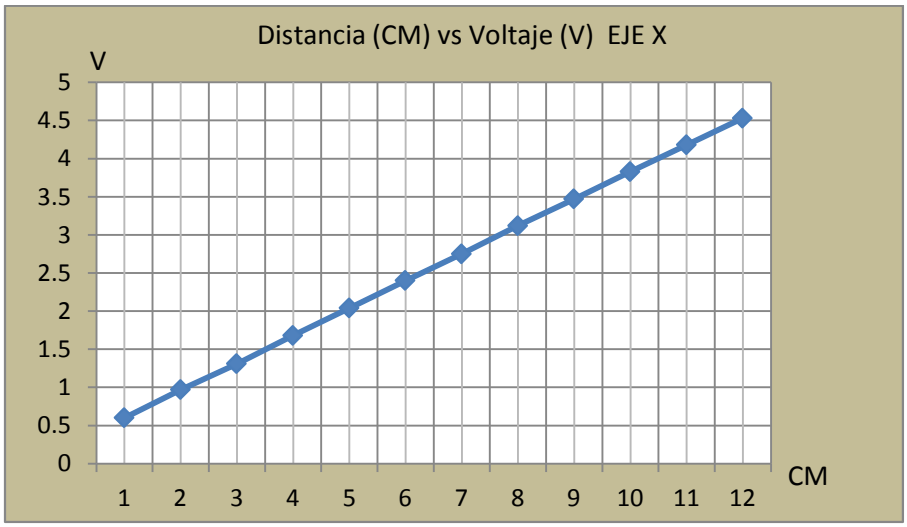


Fig. 5.0 Voltaje vs posición en "*X*".

Para el eje *y* que mide 10 cm, también alimentado con 5 Volts y midiendo con el voltímetro en el eje *x* como lo explica en la sección 2.3, lo referente a la figura 2.15 los puntos graficados son los que se muestran en la figura 5.1.

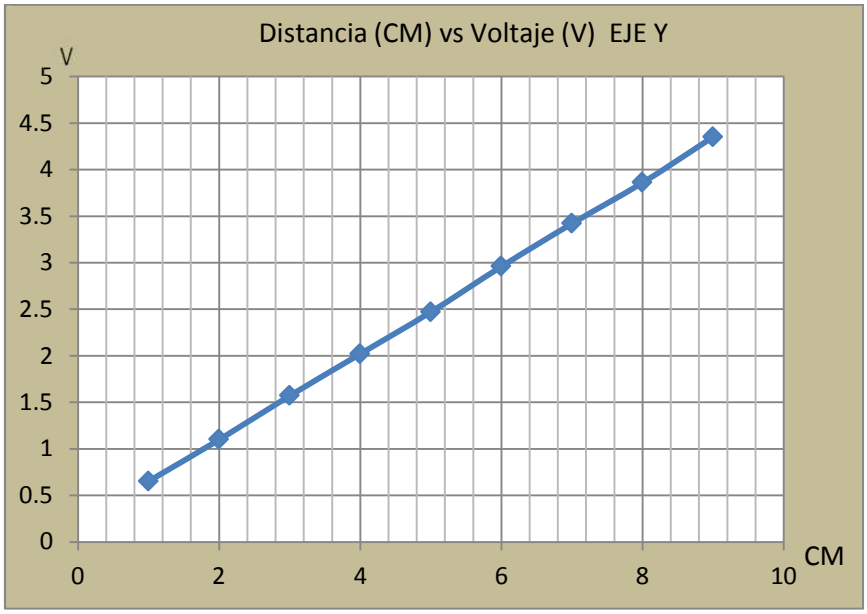


Fig. 5.1 Voltaje vs posición en "Y".

Finalmente se cambió la pantalla por una de mayor tamaño, a la nueva pantalla solo se le realizaron algunas pruebas para comprobar que efectivamente su funcionamiento es el mismo.

Es imposible obtener un valor de voltaje muy cerca de los extremos, ya que el área activa de la pantalla es menor que el área de la sección tranparente, se tiene un área activa 18.6cm x 24.8cm.

De las pruebas anteriores se puede ver que la pantalla es perfectamente lineal y que en el centro de la pantalla el voltaje medido es de 2.5V aproximadamente.

5.2 Control de giro de los servomotores

Se explica el software diseñado para probar la operación de los servomotores. Lo que se pretende es que los servomotores giren en sentido contrario de las manecillas del reloj, desde 10 hasta 170 grados con incrementos de 5°, al llegar a 170° comienza a disminuir, girando en sentido de las manecillas del reloj, con intervalos de 5 grados hasta llegar a 10°. Nótese en la figura 2.20 que el servo gira entre 0 y 180 grados, sin embargo en esta prueba se evita llegar a forzarlo y es por eso que en este programa se opera a 10 grados antes del límite.

Procedimiento:

De la sección 2.4 tenemos:

$$0.5\text{ms}=0° \quad y \quad 2.5\text{ms}=180° \qquad (5.0)$$

Aplicando en la ecuación 4.1 la regla de 3 resulta:

$$1° = 11.111\text{x}10^{-6}\ \text{seg} \qquad (5.1)$$

1TCYx es un ciclo de la frecuencia de reloj del microcontrolador.

$$1\text{TCYx} = \frac{1}{12\text{X}10^{6}\ \text{Hz}} = 83.33\text{X}10^{-9}\ \text{seg} \qquad (5.2)$$

Es decir, para tener un movimiento de 1° ser requiere 133.33 TCY

A continuación, en la figura 5.2 se muestra el algoritmo del programa, el código se encuentra en el apéndice B.

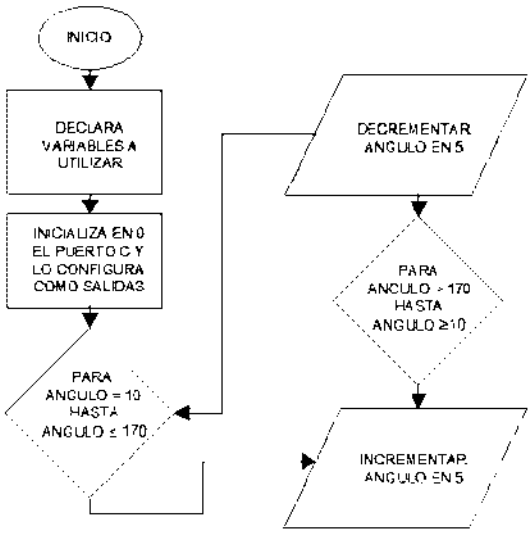


Fig. 5.2 Algoritmo para el giro del servo.

Al experimentar con el software anterior se obtiene el valor del dato enviado a cada servomotor, en donde la pantalla está perfectamente horizontal. Utilizando el mplab en modo simulación el valor para el servo *x* con dirección 0x7E es 800. Para el servo *y* con dirección 0x70 es 1500.

Es importante encontrar el valor del ancho de pulso que se le debe enviar a cada uno de los microcontroladores auxiliares, para que la base de la pantalla esté bien alineada (perfectamente horizontal).

El valor de este dato es diferente para cada uno de los ejes, así el dato enviado al servo 0x70 es diferente al dato enviado al servo 0x7E, ya que la base de la

pantalla no es cuadrada y los ejes de giro de cada motor no son iguales. Este dato es el que enviará el maestro a los esclavos cuando la diferencia entre la referencia y la posición de la bola sea 0. Si el prototipo se opera sobre una superficie como una mesa o un banco que no esté en desnivel, el valor para 0x7E está entre 600 y 850, mientras que para 0x70 es de entre 1300 a 1600.

Si la bola siempre tiende a irse hacia algún extremo de la pantalla, se debe a que el prototipo está montado sobre una superficie que está en desnivel, para compensar este desnivel se procede a ajustar los valores enviados a los servos esclavos, hay que recalibrar los valores enviados a los servos, para encontrar el punto en donde la pantalla se encuentra perfectamente horizontal. Los datos deben estar en los intervalos mencionados en el párrafo anterior.

5.3 Pruebas de sintonización del controlador PID

Con el controlador PID que se había diseñado el valor de actuación en *x* y/o en *y* se salía de los limites que el actuador puede manejar. Fue necesario agregarle un filtro anti-wind-up para limitar la acción integral.

La sintonización del sistema se logró a prueba y error. Obteniendo los siguientes valores. KpX=1.0, KdX=10, Kix=0.00001, KpY=0.8, KdY=5 y KiY=0.000001. Con estas constantes el tiempo en estabilizarse es de 8 segundos aproximadamente, oscila un poco pero al final se estabiliza.

5.4 Regulación en el centro

- El sistema siempre corrige las perturbaciones agregadas, cuando movemos la bola hacia un lugar distinto la accion del controlador PID la lleva al punto de referencia establecido por los potenciometros externos.
- Al presionar dos puntos, la pantalla censa el punto que se presione de una manera más firme. Al poner el dedo de la mano en un extremo de la

pantalla, aun cuando la bola esté en el centro, el sensor le da prioridad al dato que se presiona con mayor firmeza.

- Si se coloca una bola que no sea perfectamente redonda, mientras pese lo suficiente para que la membrana flexible de la pantalla haga contacto con la membrana rígida, el sistema trata de estabilizarlo siempre. Se realizó una prueba utilizando un huevo de gallina.

5.3.3 Regulacion al variar la referencia

Los valores en los que la pantalla detecta el punto que ha sido presionado son:

- Para el limite del eje *x* se tiene que el dato enviado al servomotor será desde 170 hasta 800.
- Para el limite del eje *y* se tiene que el dato enviado al servomotor es de 115 hasta 800.

Por esta razon si en los potenciometros se coloca una referencia de entre 0 y 5v, este dato es acondicionado los limites adecuados en el programa principal. Ver apendice A.

CAPITULO 6

CONCLUSIONES

6.1 Conclusiones

Al momento de montar el prototipo físico es donde surgen los problemas, principalmente en la parte mecánica, pues no se tiene una formación académica en esta área. Aunque se tuvieron que hacer varias pruebas se logró construir el prototipo de manera satisfactoria.

Los sistemas inestables son de gran importancia en la actualidad, en la vida laboral de un ingeniero en electrónica y áreas a fines, esa es la razón por la que es necesario tener un sistema de control inestable para realizar sus prácticas en el laboratorio.

Se investigó la tecnología utilizada en las pantallas touchscreen resistivas y su modo de operación, se implementó un mecanismo lógico de lectura para esta pantalla de de 4 conductores. Se investigó el funcionamiento de los servomotores, la manera de controlarlos con la señal pwm y la comunicación I2C.

Con el proyecto que se ha concluido se realizan algunas prácticas que antes se hacían solo en simulación, ahora las pruebas se hacen en un prototipo físico. Podemos ver que tan estable o inestable es nuestro sistema y comprobar los efectos del controlador PID aplicado.

Con los valores del PID es la mejor respuesta que se pudo lograr con esta pantalla ya que consideramos que debido a la poca fricción entre la bola y la pantalla, la bola se desplaza con una velocidad mayor a la velocidad de los actuadores y de procesamiento del microcontrolador.

El prototipo corrige las perturbaciones agregadas al sistema y proporciona efectos visibles de estabilidad o inestabilidad. Se cumple satisfactoriamente el propósito del proyecto ya que el sistema es controlable.

6.2 Trabajos futuros

Nuestro prototipo servirá de base para construir físicamente los siguientes sistemas inestables de propósito didáctico bola y placa, para el laboratorio de la Facultad de Ingeniería Eléctrica.

Con todo el trabajo realizado en la construcción, bien documentado, el prototipo servirá como posibles tesis futuras, trabajos como seguimiento de trayectorias predefinidas de la bola sobre la pantalla o, implementar otro tipo de control más eficiente.

Referencias

[Referencia 1]

Norman S. Nise, Sistemas de control para ingeniería, Universidad Politécnica del Estado de California, CECSA, 2006.

[Referencia 2]

National Instruments, Entrenadores de Ingeniería Quanser (QNET) para NI ELVIS, 23 de julio de 2010, http://sine.ni.com/nips/cds/view/p/lang/es/nid/203545.

[Referencia 3]

Quanser, Rotary Control Challenge 2x SRV02 + 2DBB = 2DOF Ball Balancer Experiment 23 de julio de 2010, http://www.quanser.com/english/downloads/products/Rotary/2DBB_PIS_032508.pdf .

[Referencia 4]

Diccionario de la lengua española. Página principal. 31 de julio de 2010. http://www.wordreference.com/definicion/prototipo.

[Referencia 5]

Tyco electronics, Elo TouchSystems/Español de Latinoamérica, 15 de junio del 2010, www.elotouch.com.ar/Productos/Touchscreens/CuatroHilos/default.asp.

[Referencia 6]

Sergio R. Caprile, Nota de aplicación CAN-015 “Lectura de TouchScreen resistivo” Cika electrónica S.R.L. www.cika.com/soporte/AppNotes/CAN-015_TouchScreen.pdf.

[Referencia 7]

pic18F2455/2550/4455/4550 Datasheet 28/40/44-PinHigh Performance Enhanced Flash, USB Microcontrollers whit nanoWatt Technology.2007 Microchip Technology Inc.

[Referencia 8]

Falta referencia de donde sacamos las copias de i2c

[Referencia 9]

Wen Yu “No linear PD Regulation for Ball and B. Com. Systems” International Journal of Electrical Engineering Education vol 46 No 1 January 2009.

APENDICE A

El código del programa principal.

```
#include <p18F4550.h>
#include <delays.h>
#pragma config PLLDIV=5, CPUDIV=OSC1_PLL2, USBDIV=2     //para el de 8 PLLDIV=2
#pragma config FOSC=HSPLL_HS, FCMEN=OFF, IESO=OFF, PWRT=OFF
#pragma config BOR=OFF, VREGEN=ON, WDT=OFF, MCLRE=ON
#pragma config LVP=OFF
//#define salida LATBbits.LATB2
unsigned char I2C_Tx(unsigned char Direccion, unsigned int Dato);
void Init_I2C(void);
void Init_ADC(void);
unsigned int leer_x(void);
unsigned int leer_y(void);
unsigned int Ref_X(void);
unsigned int Ref_Y(void);

void main(void)
{
    unsigned int ServoX, ServoY, Pulso_X, Pulso_Y, Dato;
    float Error_X, Error_ant_X, Error_Y, Error_ant_Y;
    float KpX,KdX,KiX,KpY,KdY,KiY,PX,DX,IX, PY,DY,IY;
    float Accion_X, Accion_Y, Pos_X,Pos_Y, ref_x, ref_y;
  KpX=1.0;
  KdX=10;
  KiX=0.00001;
  KpY=0.8;
  KdY=5;
  KiY=0.000001;
  Error_ant_X=0; //error inicial
  Error_X=0;
  IX=0;
  Error_ant_Y=0; //error inicial
  Error_Y=0;
  IY=0;

  //Inicializa el modulo I2C
  Init_I2C();
  Pulso_X=0;
  Pulso_Y=0;
  Init_ADC();
   while(Pulso_X + Pulso_Y == 0)

    {
        Pos_X=leer_x();
//      Pos_X = 779;
        ref_x=(Ref_X()/1023.0)*630.0+170.0;
//      ref_x=479;
        Error_ant_X=Error_X;
        Error_X =ref_x-Pos_X;
        PX = KpX*Error_X;
        DX = KdX*(Error_X-Error_ant_X);
```

```
//Antiwindup  para X
        Accion_X=PX+IX+DX;
        Accion_X=Accion_X+610;
        if(Accion_X<0){
            Accion_X=0;
        }
        if(Accion_X>3000){
            Accion_X=3000;
        }
        if((Accion_X>0)&&(Accion_X<3000)){
            IX = IX-(KiX*Error_X);
        }


        Pulso_X = I2C_Tx(0x7E, Accion_X);

        Pos_Y=leer_y();
//      Pos_X = 779;
        ref_y=(Ref_Y()/1023.0)*630.0+170.0;
//      ref_y=458;
        Error_ant_Y=Error_Y;
        Error_Y =ref_y-Pos_Y;
        PY = -KpY*Error_Y;
        DY = -KdY*(Error_Y-Error_ant_Y);
//Antiwindup  para Y
        Accion_Y=PY+IY+DY;
        Accion_Y=Accion_Y+1330;
        if(Accion_Y<0){
            Accion_Y=0;
        }
        if(Accion_Y>3000){
            Accion_Y=3000;
        }
        if((Accion_Y>0)&&(Accion_Y<3000)){
            IY = IY-(KiY*Error_Y);
        }


        Pulso_Y = I2C_Tx(0x70, (int)Accion_Y);




    }
    Pulso_X=0;
    Pulso_Y=0;

}
/***funcion para la transmision I2C***/
unsigned char I2C_Tx(unsigned char Direccion, unsigned int Dato)
{
unsigned char DatoByte;

        //Bit de inicio
        SSPCON2bits.SEN = 1;
```

```
//Espera a que finalice la condicion de START
while(PIR1bits.SSPIF == 0);
//Limpiar bandera
PIR1bits.SSPIF = 0;

//Envia la direccion del modulo esclavo a controlar
SSPBUF = Direccion;

//Espera a que finalice la transmision
//y revisa si llego un ACK del esclavo
while(PIR1bits.SSPIF == 0);
// Limpiar bandera
PIR1bits.SSPIF = 0;

//Espera a que responda el esclavo
Delay10TCYx(100);
//Error, no respondio el esclavo con el ACK de direccion
if(SSPCON2bits.ACKSTAT == 1)
    return 1;

//Envia el Dato MSB al modulo
DatoByte = Dato >> 8;
SSPBUF = DatoByte;

//Espera a que finalice la transmision y revisa si llego
//un ACK de dato MSB del esclavo
while(PIR1bits.SSPIF == 0);
//Limpiar bandera
PIR1bits.SSPIF = 0;

//Espera a que responda el esclavo con ACK
// de segundo dato de 8 bits
Delay10TCYx(100);
//Error, no respondio el esclavo con ACK
if(SSPCON2bits.ACKSTAT == 1)
    return 2;

// Envia el Dato LSB al modulo
DatoByte = Dato & 0xFF;
SSPBUF = DatoByte;

//Espera a que finalice la transmision y revisa
//si llego el ACK del esclavo
while(PIR1bits.SSPIF == 0);
//Limpiar bandera
PIR1bits.SSPIF = 0;

//Espera a que responda el esclavo
Delay10TCYx(100);
//Error, no respondio esclavo con ACK de dato
if(SSPCON2bits.ACKSTAT == 1)
    return 3;

//Generacion del bit de STOP
SSPCON2bits.PEN = 1;
```

```
        //Espera a que finalice la condicion de STOP
        while(PIR1bits.SSPIF == 0);
        //Limpiar bandera
        PIR1bits.SSPIF = 0;

        Delay10TCYx(100);

        //se resivieron todos los ACK,
        //TODO BIEN
        return 0;
}


/****Funcion para la inicializacion de perifericos del microcontrolador****/
void Init_I2C(void)
{

    // Configura los pines data y clk del i2c como digitales
    ADCON1 = 0b00000101;

    //Puerto B como entradas (digitales) debido a I2C
    // (aunque son salidas de drenaje abierto)
    TRISBbits.TRISB0 = 1;       //SDA
    TRISBbits.TRISB1 = 1;       //SCL

    //Se habilitan las resistencias Pull-up internas
    INTCON2bits.RBPU = 0;

    //Operacion a 100Khz
    //SMBus desabilitado
    SSPSTAT = 0x80;
    //Habilitacion del I2C como master
    SSPCON1 = 0x28;
//  SSPCON2 = 0x00;
    //Velocidad del puerto master
    //100 Khz, con Fosc = 48Mhz
    SSPADD = 0x63;
}

unsigned int leer_x (void)
{
    unsigned int valor=0;
    ADCON0=0x05;        //convertidor entrada RA1
    TRISAbits.TRISA0=1;
    TRISAbits.TRISA1=1;
//configurar entradas

    TRISDbits.TRISD3=1;
    TRISDbits.TRISD5=1;
```

```
//configuracion salidas
    TRISDbits.TRISD2=0;
    TRISDbits.TRISD4=0;
//alimentacion activada
    LATDbits.LATD2=1;
    LATDbits.LATD4=0;

    Delay100TCYx(1);
    ADCON0bits.GO_DONE = 1;


    while(ADCON0bits.GO_DONE == 1){};
    valor=ADRESH;

    valor=valor<<8;
    valor|=ADRESL;
//  Vx=(valor*5.0)/1023.0;

    return(valor);
}

unsigned int leer_y (void)
{
    unsigned int valor=0;
    ADCON0=0x01;    //conviertidor entrada RA0
    TRISAbits.TRISA0=1;
    TRISAbits.TRISA1=1;

//configura salidas
    TRISDbits.TRISD3=0;
    TRISDbits.TRISD5=0;

//configura entradas
    TRISDbits.TRISD2=1;
    TRISDbits.TRISD4=1;

//alimentacion activada
    LATDbits.LATD3=1;
    LATDbits.LATD5=0;

    Delay100TCYx(1);
    ADCON0bits.GO_DONE = 1;

    while(ADCON0bits.GO_DONE == 1){};

    valor=ADRESH;
    valor=valor<<8;
    valor|=ADRESL;
//  Vy=(valor*5.0)/1023.0;

    return valor;
}
```

```
unsigned int Ref_X (void)
{
    unsigned int valor=0;
    ADCON0=0x09;    //conviertidor entrada RA2

    TRISAbits.TRISA2=1;

    Delay100TCYx(1);
    ADCON0bits.GO_DONE = 1;

    while(ADCON0bits.GO_DONE == 1){};

    valor=ADRESH;
    valor=valor<<8;
    valor|=ADRESL;
//  Vy=(valor*5.0)/1023.0;

    return valor;
}

unsigned int Ref_Y (void)
{
    unsigned int valor=0;
    ADCON0=0x0D;    //conviertidor entrada RA3

    TRISAbits.TRISA3=1;

    Delay100TCYx(1);
    ADCON0bits.GO_DONE = 1;

    while(ADCON0bits.GO_DONE == 1){};
     valor=ADRESH;
     valor=valor<<8;
     valor|=ADRESL;
//   Vy=(valor*5.0)/1023.0;

     return valor;
}

void Init_ADC(void)
{
    ADCON1=0b00001011; //configurando la entrada,cuatro canales analogicos para el convertidor A/D
    ADCON0=0b00000001; //RA0 canal convertidor, proceso de conversion parado pero activado
    ADCON2=0b10111110;
}
```

APENDICE B

Diagrama para probar el servo.

```
#include <p18F4550.h>
#include <delays.h>
#pragma config PLLDIV=5, CPUDIV=OSC1_PLL2, USBDIV=2 //para el de 8 PLLDIV=2,
#pragma config FOSC=HSPLL_HS, FCMEN=OFF, IESO=OFF, PWRT=OFF
#pragma config BOR=OFF, VREGEN=ON, WDT=OFF, MCLRE=ON
#pragma config LVP=OFF

void main(void)
{
    unsigned long int TH,TL,N,M,Angulo1,i;  //Declarando variables
    PORTC=0x00;          //inicializando en 0 el puerto C
    TRISC=0x00;          //configurando como salidas
    PORTA=0x00;          //inicializando en 0 el puesto A
    CMCON=0x07;          //Deshabilita el comparador analogico
    ADCON1=0b00001111;    //todas digitales
    TRISA=0xFF;          //salidas
    Angulo1=10;          //iniccializa angulo en 10


    while(1)
    {
//Incrementa angulo desde 10 hasta 170 de 5 en 5
        for (Angulo1==10;Angulo1<=170;Angulo1=Angulo1+5)
        {
//Tiempo en alto para el pulso
            TH=5+(.1111*Angulo1);
            N=TH*1200;
            N=N/1000;
            LATC=0xFF;
                Delay1KTCYx(N);
//Tiempo en bajo
            TL=200-TH;
            M=TL*1200;
            M=M/1000;
            LATC=0x00;
                Delay1KTCYx(M);

//Retardo de 1 seg
                    Delay10KTCYx(200);
                    Delay10KTCYx(200);
                    Delay10KTCYx(200);
                    Delay10KTCYx(200);
                    Delay10KTCYx(200);
                    Delay10KTCYx(200);

        }
```

```
        for (Angulo1>170;Angulo1>=10;Angulo1=Angulo1-5)
        {
            TH=5+(.1111*Angulo1);
            N=TH*1200;
            N=N/1000;
            LATC=0xFF;
                Delay1KTCYx(N);
            TL=200-TH;
            M=TL*1200;
            M=M/1000;
            LATC=0x00;
                Delay1KTCYx(M);
                    Delay10KTCYx(200);
                    Delay10KTCYx(200);
                    Delay10KTCYx(200);
                    Delay10KTCYx(200);
                    Delay10KTCYx(200);
                    Delay10KTCYx(200);
        }

    }
}
```

APENDICE C

Programa para encontrar el punto horizontal de la pantalla.

```
void main(void)
{
    unsigned int ServoX, ServoY, Error, Dato;

    //Inicializa el modulo I2C
    Init_I2C();
    Error=0;

    while(Error == 0)

    {
        //Envia dato de 16bits de 0 a 3072

            Error = I2C_Tx(0x70, 1330); //eje Y
            Error = I2C_Tx(0x7E, 610);  //eje X
            Delay10KTCYx(100);


    }
    Error=0;

}
```