

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
**UNIVERSIDAD
CATÓLICA**
DEL PERÚ

**DISEÑO DE UN SISTEMA DE RECONSTRUCCIÓN DE VOLÚMENES DE
TRONCOS DE MADERA POR ULTRASONIDO PARA LA OPTIMIZACIÓN
DEL PROCESO DE CORTE EN ASERRADEROS**

Tesis para optar el Título de Ingeniero Electrónico, que presenta el
bachiller:

ROYER ROJAS MALÁSQUEZ

ASESOR: Ing. Willy Eduardo Carrera Soria

Lima, 9 de Junio de 2014

RESUMEN

El presente trabajo tiene como propósito contribuir a la solución de la ineficiencia del proceso de corte llevado a cabo en los aserraderos nacionales. Dicha ineficiencia es ocasionada mayormente debido a la selección de un esquema de corte inapropiado, lo cual se traduce en pérdidas de materia prima que alcanzan incluso el 50%. En tal sentido, se hace necesario un proceso que permita reducir las pérdidas de materia prima, de tal manera que además de mejorar la rentabilidad de las empresas madereras, esto evite que un gran número de árboles sean talados de forma innecesaria, contribuyendo al mismo tiempo a reducir la deforestación de los bosques madereros y a la conservación del ecosistema del planeta.

Un proceso completo de optimización de corte de troncos de madera estaría conformado por tres etapas: diseño de un sistema que permita reproducir el volumen de los troncos de madera, desarrollo de un algoritmo computacional para hallar el esquema de corte más apropiado, y finalmente la integración de los dos ítems anteriores. La presente tesis se avoca únicamente a la primera de las tres etapas citadas anteriormente. A este sistema se le denominará en adelante: sistema de reconstrucción de volúmenes.

El sistema de reconstrucción de volúmenes, debido a su naturaleza, estaría conformado por tres partes: parte mecánica, parte hardware y parte software. El diseño de la parte mecánica escapa a los objetivos de la presente tesis, razón por la cual no será tratada en el presente trabajo, y se limitará sólo a una breve descripción de la misma. A cambio de esto, se prestará mayor atención en el diseño de la parte hardware y la parte software.

El presente trabajo se encuentra estructurado de la siguiente forma:

Capítulo 1:

El proceso de corte de madera en los aserraderos nacionales y sus problemáticas.

Capítulo 2:

Estado del Arte.

Capítulo 3:

Diseño de los módulos componentes del sistema de reconstrucción de volúmenes.

Capítulo 4:

Simulación de los módulos componentes del sistema de reconstrucción de volúmenes.

INDICE

CAPÍTULO 1: EL PROCESO DE CORTE DE MADERA EN LOS ASERRADEROS NACIONALES Y SUS PROBLEMÁTICAS	1
1.1 Variables internas.....	2
1.2 Variables externas.....	4
1.3 Proceso de corte de rollizos en los aserraderos nacionales.....	5
1.4 Problemas que se presentan en el corte de rollizos.....	5
1.5 Declaración del marco problemático.....	7
CAPÍTULO 2: ESTADO DEL ARTE	8
2.1 Estado del arte.....	9
2.1.1 Presentación del asunto de estudio.....	9
2.1.2 Estado de la investigación.....	9
a. Reconstrucción de volúmenes mediante el uso de luz infrarroja.....	9
b. Reconstrucción de volúmenes mediante el uso de cámaras.....	11
c. Reconstrucción de volúmenes mediante el uso de señales ultrasonido.....	13
2.1.3 Síntesis sobre el asunto de estudio.....	14
2.2 Modelo teórico.....	15
2.2.1. El sensor ultrasonido SRF04.....	17
a. Descripción.....	17
b. Funcionamiento.....	17
c. Funciones de los pines del sensor SRF04.....	18
d. Características técnicas.....	19
2.2.2. Regulación de la velocidad.....	20
Encoder óptico.....	21
2.2.3. Programación en un entorno de programación visual.....	22
CAPÍTULO 3: DISEÑO DE LOS MÓDULOS COMPONENTES DEL SISTEMA DE RECONSTRUCCIÓN DE VOLÚMENES	23
3.1 Descripción contextual.....	24
3.2 Diagrama de bloques.....	24
3.3 Explicación del diseño.....	24
3.4 Parte mecánica.....	26
3.5 Parte hardware.....	26
3.5.1 Sensores.....	26
a. Requisitos mínimos.....	26
b. Opciones.....	26
c. Criterios de selección.....	27
d. Características del sensor ultrasonido SRF04.....	28

3.5.2 Control.....	28
a. Microcontrolador.....	29
b. Toma de datos de los sensores.....	30
c. Regulación de la velocidad de giro del motor.....	31
3.5.3 Excitador.....	32
a. Giro del motor en doble sentido.....	34
b. Funcionamiento conjunto del bloque de control y el excitador.....	35
3.5.4 Interface de comunicación.....	36
a. Puerto serie RS232.....	36
b. Niveles lógicos RS232.....	37
c. Niveles lógicos TTL (PIC16F84).....	37
d. Circuito integrado MAX232.....	37
e. Conexión puerto RS232 y PIC16F84.....	38
3.6 Parte software.....	39
3.6.1 Control.....	39
a. Diagrama de flujo: Medición de distancias.....	39
b. Diagrama de flujo: Regulación de velocidad del motor.....	40
c. Diagrama de flujo: Cambio de sentido de giro del motor.....	42
3.6.2 PC.....	42
a. Cálculos y definiciones de parámetros.....	42
Parámetros.....	42
Cálculos.....	43
b. Diagrama de flujo principal: Reconstrucción volumétrica del rollizo.....	44
c. Diagramas de flujo de las subrutinas.....	46
Diagrama de flujo de la subrutina: Toma de datos de un anillo.....	46
Diagrama de flujo de la subrutina: Decremento de coordenada z.....	46
Diagrama de flujo de la subrutina: Procesa datos almacenados en tabla.....	47
Diagrama de flujo de la subrutina: Visualiza volumen del rollizo.....	48
d. Algoritmo base para el gráfico volumétrico del rollizo.....	49
3.6.3 Software de visualización.....	52
a. Requisitos mínimos.....	52
b. Opciones.....	52
c. Criterios de selección.....	52
d. Breve descripción del software de visualización elegido.....	53
CAPÍTULO 4: SIMULACIÓN DE LOS MÓDULOS COMPONENTES DEL SISTEMA DE RECONSTRUCCIÓN DE VOLÚMENES.....	55
4.1 Breve explicación del diseño.....	56
4.2 Software de simulación.....	57
4.3 Simulación de la regulación de velocidad de giro del motor DC por PWM.....	58

4.4 Simulación del control de sentido de giro del motor DC.....	67
4.5 Simulación del proceso de medición de distancias.....	74
4.6 Simulación de la comunicación PC - Microcontrolador.....	84
4.7 Simulación del gráfico del rollizo en Visual Basic.....	88
4.8 Análisis de costos.....	92
CONCLUSIONES.....	94
RECOMENDACIONES.....	95
BIBLIOGRAFÍA.....	97



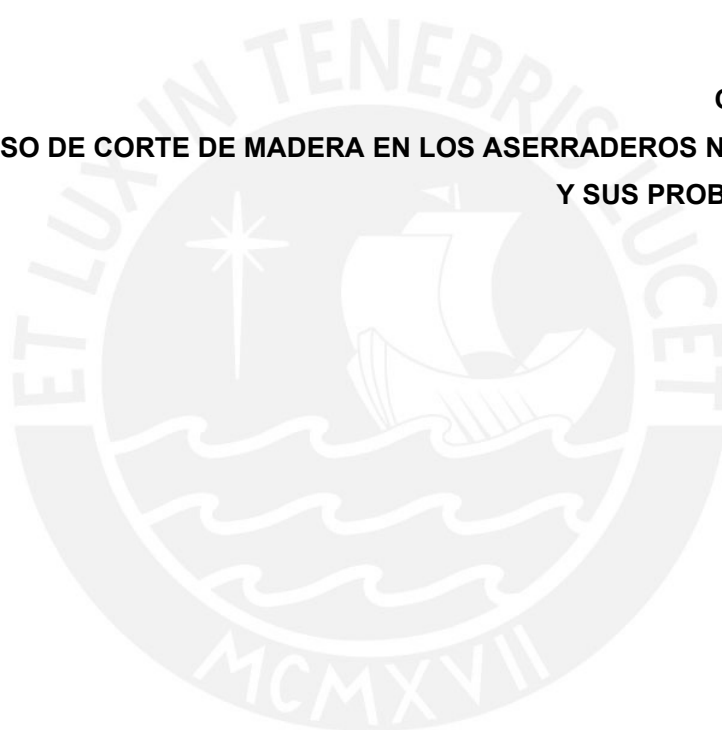
INTRODUCCIÓN

En la actualidad, la eficiencia del proceso de corte en los aserraderos nacionales depende únicamente de la experiencia o sagacidad del jefe de producción u operario, ya que éste, al recibir un tronco de madera, debe hacerse una idea de la forma real del mismo y, a partir de allí, seleccionar el esquema de corte que le parezca más apropiado para cumplir con los requerimientos de los clientes. Como es natural, este método conlleva grandes pérdidas de materia prima para las empresas madereras, las cuales alcanzan incluso el 50%. De aquí se desprende la necesidad de contar con un sistema que permita visualizar la forma real del tronco de madera (o también llamado 'rollizo'), antes de decidir cómo cortarlo. Esto permitiría (posteriormente) aplicar una heurística para seleccionar el esquema de corte con el cual se minimicen las pérdidas de materia prima, mejorando así la rentabilidad de las empresas madereras.

Visto desde otro punto de vista, la optimización del proceso de corte de troncos de madera reduciría considerablemente la tala innecesaria de árboles, contribuyendo de forma indirecta a la conservación del ecosistema en el planeta.

Finalmente, para diseñar el sistema que permita reproducir la forma (o volumen) real de los rollizos (y que en el presente trabajo se denominará: sistema de reconstrucción de volúmenes), se propone el uso de la tecnología ultrasonido, la cual tiene una gran variedad de aplicaciones en diversas áreas, como por ejemplo las ecografías y las ultrasonografías (las cuales emplean ultrasonido para obtener imágenes del interior del cuerpo), entre otras.

CAPÍTULO 1:
EL PROCESO DE CORTE DE MADERA EN LOS ASERRADEROS NACIONALES
Y SUS PROBLEMÁTICAS



1.1 Variables internas

Problemas	Características y causas
El uso de materia prima en los aserraderos ocasiona grandes pérdidas en forma de piezas lampazos y piezas laterales.	Debido a la ausencia de un sistema eficiente que optimice el proceso de corte, las pérdidas pueden alcanzar incluso el 50% del total de material procesado.
Se obtiene gran cantidad de madera dimensionada útil que presenta 'vértices incompletos' o 'cantos muertos', razón por la cual es descartada, produciendo aún más pérdidas.	El personal encargado tiene dificultad para estimar cuáles serían las dimensiones máximas de las tablas que, en cada caso, se pueden obtener de los troncos de madera.
Por lo general, el operador de la sierra tiene dificultad para coordinar la posición del tronco frente a la sierra en relación a las irregularidades y deformaciones que presenta el tronco de madera (o 'rollizo'), ocasionando cortes erróneos y pérdidas.	Para definir los cortes, el operador de la sierra se guía por su propia apreciación visual de la forma del tronco de madera (o 'rollizo').

Para un mejor entendimiento de los problemas concernientes al proceso de corte de troncos de madera en los aserraderos, resulta conveniente revisar los trabajos de Vilaboa [1,2,3,4,5] y Gutierrez [6,7].

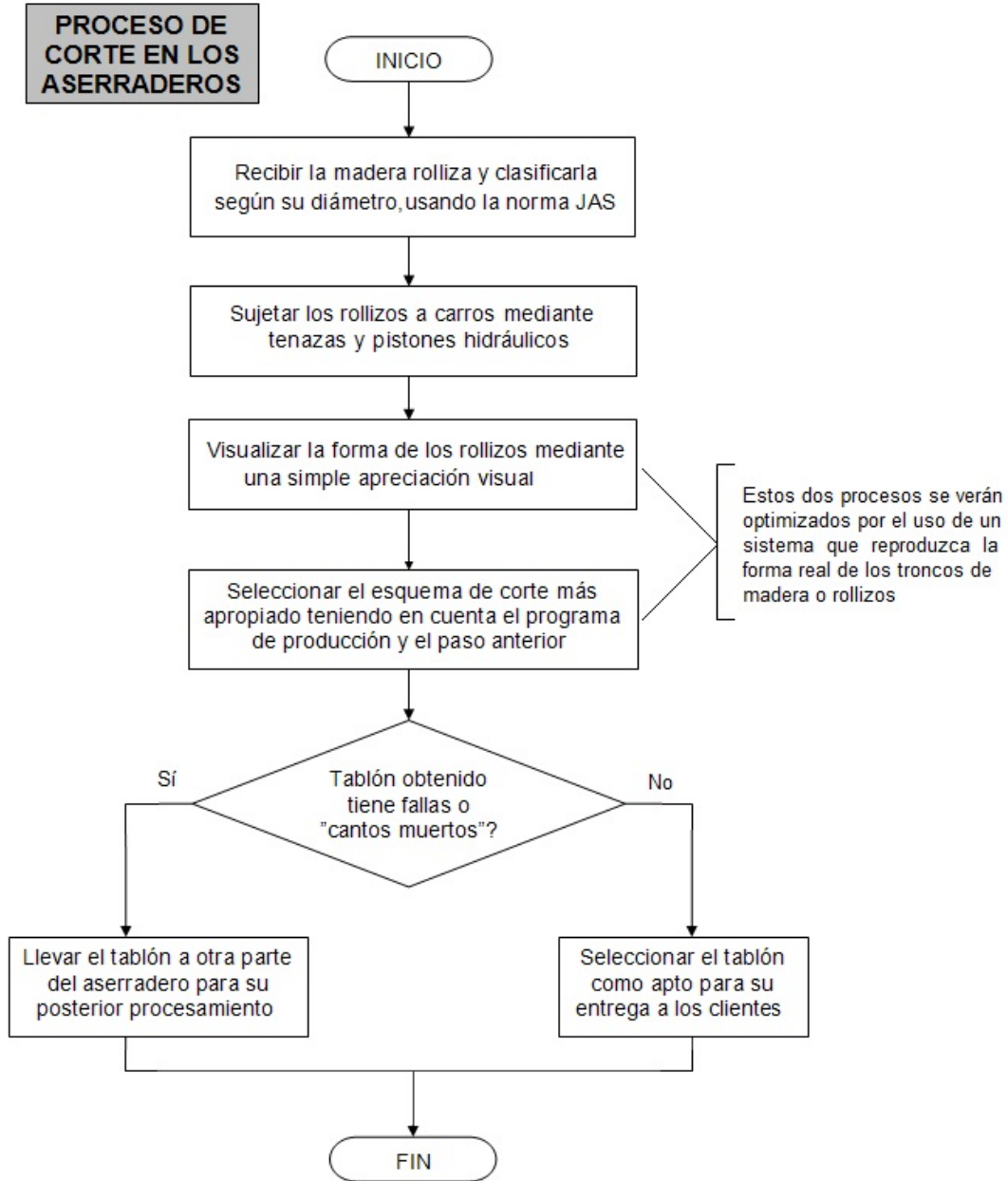


Figura 1.1 Diagrama de flujo del proceso de corte de troncos de madera en los aserraderos [1]

En la Figura 1.1 se muestra el diagrama de flujo del proceso de corte de troncos de madera llevado a cabo en los aserraderos [1,2,3,4,5], donde se evidencia paso a paso las etapas por las que pasa un tronco de madera antes de ser convertido en los tabloncillos que son requeridos por los clientes.

1.2 Variables externas:

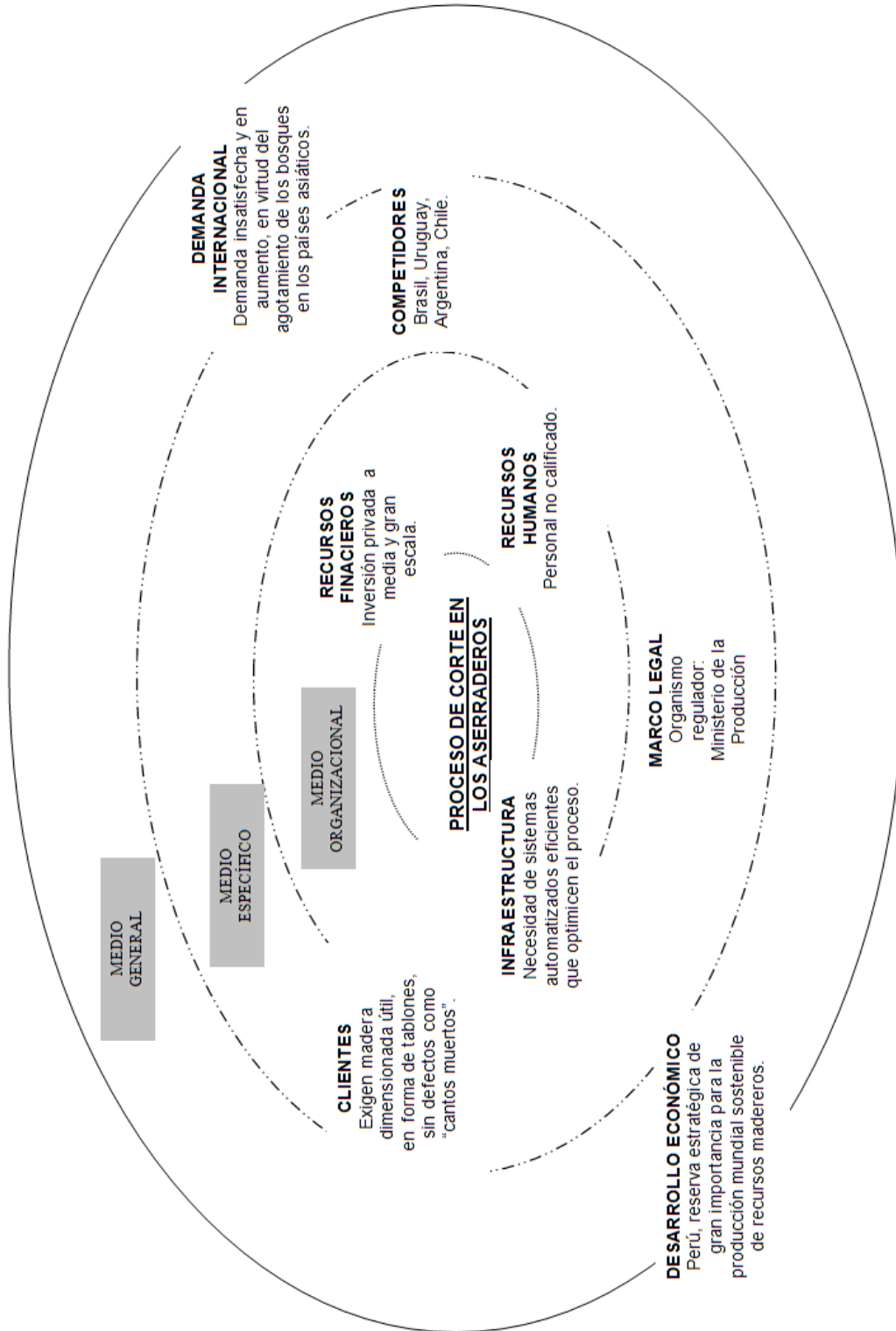


Figura 1.2 Variables externas del proceso de corte en aserraderos

1.3 Proceso de corte de rollizos en los aserraderos nacionales

Cabe señalar que la principal actividad productiva llevada a cabo en los aserraderos nacionales radica en el corte de rollizos (se denomina 'rollizo' a aquel tronco de madera al cual se le ha quitado todas las ramas o partes salientes). Esto permite obtener la madera dimensionada que los clientes solicitan (en forma de tablones). Es por ello que la eficiencia con que se realiza el proceso de corte resulta fundamental en la rentabilidad de las empresas madereras [1,2,3,4,5].

Resulta muy útil, como apoyo a la resolución de problemas prácticos, revisar la literatura existente sobre problemas de corte de materiales. En todos los artículos donde se aborda este tipo de problemas se busca satisfacer pedidos, los que se caracterizan matemáticamente por la necesidad de obtener un conjunto de piezas una determinada cantidad de veces. Para esto, se debe cortar el material que viene en tamaños estándares, y aparentemente es un problema simple. Sin embargo, dado que no hay soluciones exactas que, al ser implementadas en programas computacionales, resuelvan todas las soluciones a los problemas, existe una gran cantidad de procedimientos para resolver las diferentes situaciones, compatibilizando de diferente forma las calidades de las soluciones con los tiempos de respuesta [1].

Como menciona Vilaboa [1], en la actualidad los dos tipos principales de corte utilizados en los aserraderos nacionales son: corte por sierras circulares y corte por carros huincha. El corte por sierras circulares es usado para cortar troncos de madera de un diámetro menor o igual a 30 o 40 cm. Este tipo de corte consiste simplemente en hacer pasar los troncos rápidamente, uno tras otro, a través de las sierras circulares, que giran a gran velocidad. En cambio, el corte por carros huincha es utilizado para cortar troncos de madera de gran diámetro (mayor a 30 o 40 cm). En este caso, los troncos son fijados a carros mediante el uso de tenazas o pistones hidráulicos, luego éstos se pasan a través de las sierras según el esquema de corte apropiado, el cual es definido por el jefe de producción y luego encargado al operador de la sierra.

1.4 Problemas que se presentan en el corte de rollizos

Para describir los problemas, en primer lugar supondremos que los rollizos tienen forma cilíndrica. Luego, del proceso de corte se obtienen tres tipos de piezas: laterales, centrales y lampazos [1], tal como se muestra en la Figura 1.3.

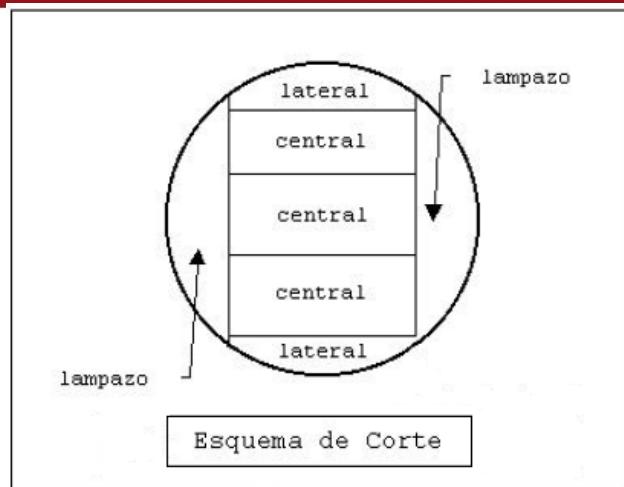


Figura 1.3 Tipos de piezas obtenidas de un proceso de corte [1]

De los tres tipos de piezas, únicamente las piezas centrales son las dimensionadamente útiles. Los lampazos y laterales pasan a otra sección del aserradero para otro tipo de procesamiento. Luego, al aplicar un determinado esquema de corte a un rollizo, es muy probable que se presenten vértices incompletos (o lo que se denomina ‘cantos muertos’) [1].

En la Figura 1.4 se puede apreciar un ‘vértice incompleto’ o ‘canto muerto’, el cual está representado por un pequeño trozo de pieza central de color verde (ver esquina superior derecha de la primera pieza central).

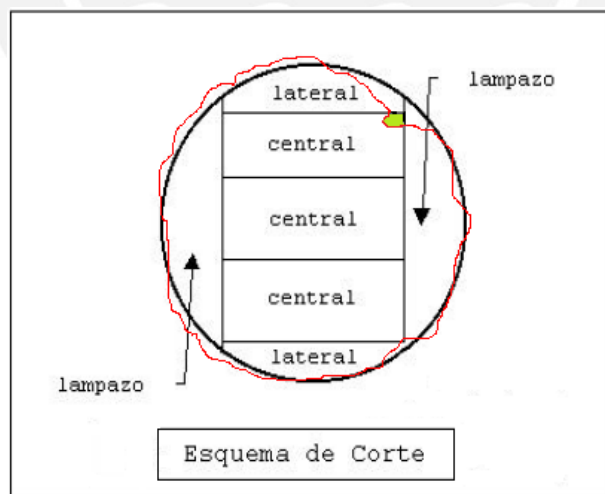


Figura 1.4 Representación de un ‘vértice incompleto’ o ‘canto muerto’ [1]

Así, una pieza central que presenta un canto muerto constituye una pieza fallida, por lo cual es descartada para su comercialización, ocasionando grandes pérdidas. Luego, con el fin de minimizar la producción de piezas con vértices incompletos

(cantos muertos), es frecuente que se exija a cada programa de corte que las escuadrías (medidas de las tablas) a obtener estén comprendidas dentro de la circunferencia correspondiente al menor diámetro de la sección axial del rollizo. Pareciera que lo importante en estos casos es realizar un adecuado procesamiento de la información visual para calcular el diámetro del mayor cilindro (para maximizar el uso de materia prima) que puede insertarse en el interior de cada rollizo sin que sobrepase la masa de la madera (para que no se produzcan cantos muertos) [1], tal como se puede apreciar en la Figura 1.5.

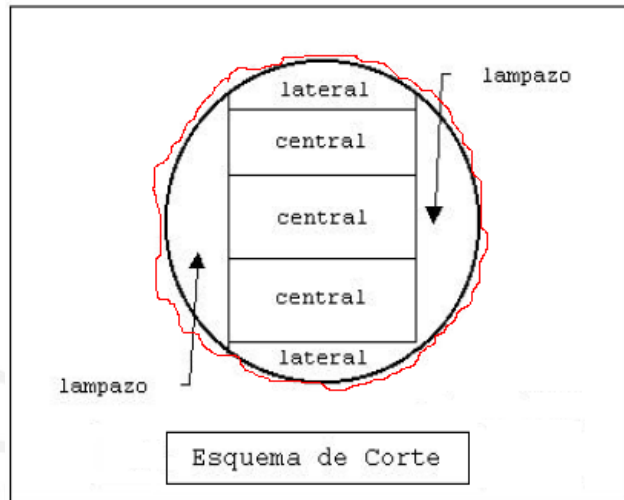


Figura 1.5 Representación del mayor cilindro uniforme dentro de un rollizo [1]

1.5 Declaración del marco problemático

En la actualidad, cerca del 50% de madera se desperdicia durante el proceso de corte en los aserraderos, y esto se debe a que el operador de la sierra al recibir un tronco de madera, determina visualmente su forma y de acuerdo a su experiencia analiza las diversas formas de aserrío. Ello provoca que el proceso de corte de rollizos dependa de la apreciación visual que el operador tenga del rollizo. Los cortes erróneos se originan porque el operador de la sierra (incluso uno experimentado) tiene dificultad para coordinar la posición del tronco frente a las irregularidades y deformaciones que presenta el rollizo. Esta precisión lograda 'a ojo del operador' parece ser bastante mala, pues conlleva grandes pérdidas de materia prima. En otras palabras, la labor de optimización de materia prima en los aserraderos nacionales es realizada por personal que se guía, generalmente, por su experiencia y buen juicio únicamente [1,2,3,4,5].



**CAPÍTULO 2:
ESTADO DEL ARTE**

2.1 Estado del arte

2.1.1 Presentación del asunto de estudio

Actualmente, el avance de la tecnología ha generado una gran presión competitiva sobre las empresas, las cuales para sobrevivir deben focalizarse en las áreas centrales del negocio con agilidad y eficiencia. En tal sentido, cabe destacar que los procesos industriales ocupan el mayor porcentaje de los diseños y aplicaciones de las empresas, donde se observa un creciente interés en el control de las actividades, la supervisión en tiempo real, el registro de los eventos, etc.

Por otro lado, está comprobado que el aumento de la productividad de una empresa está directamente relacionado con la automatización de los procesos, la cual está orientada a hacer un uso eficiente de los equipos, materia prima y sistemas asociados [1,2,3,4,5].

El presente estudio se centra en el proceso de corte llevado a cabo en los aserraderos y el tipo de tecnología que se aplica en los mismos para enfrentar el reto de obtener la mayor cantidad de madera dimensionada útil a partir de troncos de forma irregular.

2.1.2 Estado de la investigación

Como menciona Vilaboa [1], el desarrollo de un sistema que optimice el proceso de corte en los aserraderos consta de tres etapas:

- Desarrollo de un sistema de procesamiento de imágenes que permita determinar lo más exactamente posible la forma del material a cortar.
- Desarrollo de un software de optimización para el diseño y selección de los esquemas de corte.
- Integrar las dos soluciones anteriores en un sistema computacional que optimice los procesos de corte en los aserraderos.

El presente trabajo de tesis se centra en la primera de las tres etapas mencionadas líneas arriba.

a. Reconstrucción de volúmenes mediante el uso de luz infrarroja

Como se verá más adelante, al igual que en el caso de los sensores ultrasonido, la luz infrarroja puede ser usada para reconstruir el volumen de los rollizos. En este método los sensores infrarrojos constituyen la parte fundamental, pues se usan como medidores de distancia entre el sensor y un punto determinado de la superficie del rollizo. A continuación se muestra la descripción del sensor Sharp GP2DXX (uno de los sensores infrarrojos más usados como medidores de distancia), así como una breve explicación de su funcionamiento.

Sharp GP2DXX

Descripción

El sensor GP2DXX de Sharp Corporation (donde 'XX' representan números que especifican el modelo exacto del sensor), es un dispositivo de reflexión por infrarrojos con medidor de distancia proporcional al ángulo de recepción del haz de luz que incide en un sensor lineal integrado [8].

Funcionamiento

De acuerdo a [8], el sensor GP2DXX emite luz infrarroja por medio de un led emisor de luz infrarroja. Esta luz pasa a través de una lente que concentra los rayos de luz formando un único rayo, para así mejorar la directividad del sensor, la luz sigue una línea recta hacia adelante y cuando encuentra un obstáculo reflectante rebota y retorna con cierto ángulo de inclinación dependiendo de la distancia a la que se encontraba el objeto reflectante. La luz que retorna es concentrada por otra lente y así todos los rayos de luz inciden en un único punto del sensor lineal de luz infrarroja que contiene el receptor del dispositivo. Así, dependiendo del ángulo de recepción de la luz, ésta incidirá en un punto u otro del sensor, pudiendo de esta manera obtener un valor lineal y proporcional al ángulo de recepción del haz de luz.

En la Figura 2.1 se observa un diagrama del funcionamiento de este sensor.

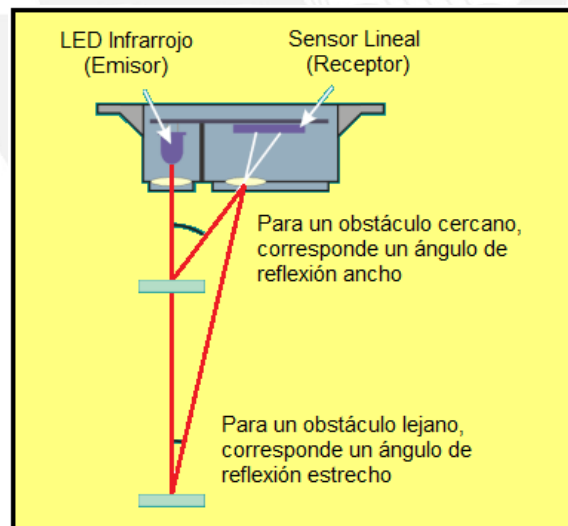


Figura 2.1 Funcionamiento del sensor infrarrojo Sharp GP2DXX [8]

La salida del sensor GP2DXX dependerá del modelo elegido [8]:

- En los modelos analógicos, como el sensor Sharp GP2D12, la salida es un voltaje proporcional a la distancia medida por el sensor.
- En los modelos digitales, como el sensor Sharp GP2D02, la lectura será de 8 bits serie con reloj externo.

- En los modelos booleanos, como el sensor Sharp GP2D05, la salida será de 1 bit y sólo tendremos la medición de una distancia fija.

Uno de los modelos más utilizados es el sensor infrarrojo Sharp GP2D02, el cual utiliza un pin de entrada (Vin) y un pin de salida (Vout) para comunicarse con el procesador principal. Debe tenerse en cuenta que el pin de entrada (Vin) en estado de reposo se encuentra a nivel lógico alto, por lo que se le debe enviar un pulso de nivel lógico bajo con una duración de al menos 70 ms para efectuar la toma de datos, luego de los cuales el pin debe regresar a nivel lógico alto. Transcurridos estos 70 ms, la información de distancia medida (8 bits) ya está lista para ser transmitida por el pin de salida (Vout). Luego, por cada nuevo flanco de bajada en el pin de entrada (Vin), el pin de salida (Vout) transmitirá uno de los 8 bits (en orden, comenzando por el bit más significativo). Finalmente, tener en cuenta que se debe esperar 0.2 ms antes del primero de los ocho flancos de bajada en el pin de entrada (Vin), el envío de los 8 bits debe tomar al menos 1 ms, y luego de leídos estos 8 bits se debe esperar al menos 1.5 ms para el siguiente pulso de 70 ms (nueva toma de datos) [8,9]. En la Figura 2.2 se muestra el diagrama de tiempos del sensor Sharp GP2D02.

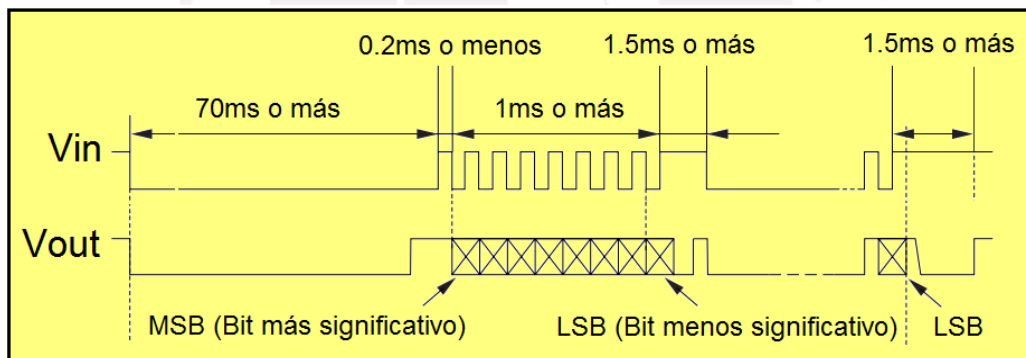


Figura 2.2 Diagrama de tiempos del sensor Sharp GP2D02 [9]

Luego, el proceso de reconstrucción de volúmenes se basará en obtener la mayor cantidad de distancias correspondientes a los diferentes puntos de la superficie del tronco (haciendo uso de los sensores infrarrojo como medidores de distancia). Posteriormente, con el uso de un software visual y un trabajo de programación, esta información sería procesada para reproducir la forma real del tronco de madera o rollizo, obteniendo una imagen tridimensional que pueda ser visualizada en la pantalla de una computadora.

b. Reconstrucción de volúmenes mediante el uso de cámaras

Como se hace referencia en el trabajo de Vilaboa [1], otro de los métodos para detectar la forma de la superficie de troncos de madera (desarrollado por la

Universidad de Concepción, Chile) se apoya en el uso de tres cámaras separadas 120 grados sexagesimales en torno al eje axial del tronco de madera o rollizo.

Las señales que captan las cámaras pasan sucesivamente a una tarjeta digitalizadora, la cual transforma la señal continua en pulsos que pueden ser transmitidos directamente a una computadora. En el mercado existen varias versiones de hardware y software que realizan esto para computadoras compatibles. Finalmente, se tiene en la memoria de una computadora tres matrices de datos que indican, según el valor de cada byte (8 bits), las tonalidades de gris que ‘observan’ las cámaras en cada punto de sus respectivos campos visuales tal como se muestra en la Figura 2.3.

Un módulo del sistema que tiene gran importancia es aquel que modela la forma de los rollizos según lo que indican las matrices entregadas por la tarjeta digitalizadora. Precisamente, con el fin de garantizar que este módulo opere adecuadamente y no entregue resultados erróneos, se decide filmar los rollizos en la forma antes indicada. Para cada rollizo filmado, este módulo del sistema debe detectar las distancias que hay (a lo largo del rollizo) desde el borde del manto hasta un eje axial de referencia. Para ello se debe detectar los puntos donde cambia fuertemente la tonalidad de gris que se produce al pasar del manto (color claro) al fondo (color oscuro). Esto permite detectar la forma de ambas líneas de contorno que ‘observa’ cada cámara [1].

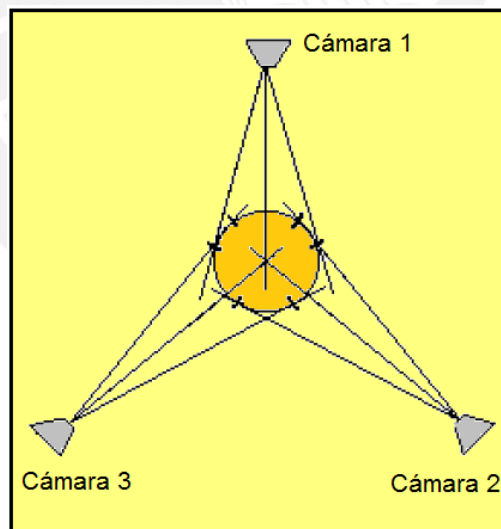


Figura 2.3 Sistema de cámaras para reconstrucción de volumen [1]

Con esta información es posible obtener un modelo matemático de la forma del manto del rollizo. Para facilitar el trabajo matemático se debe modelar el rollizo en coordenadas cilíndricas. Así, una matriz de datos bidimensional indica, para un conjunto de puntos del eje axial y para un conjunto de ángulos, las distancias

existentes desde este eje central de referencia hasta el manto. Esto proporciona una malla discreta sobre la forma del manto del tronco de madera o rollizo.

Una vez obtenida la forma de los rollizos, deben ser aplicadas diversas heurísticas de corte que sean compatibles con la obtención de tablas del tipo, cantidad y medidas que figuran en cada programa de producción, con lo cual se obtiene un alto rendimiento en el proceso de corte [1].

Lamentablemente, uno de los problemas que presenta el uso de cámaras para detectar la forma volumétrica de los troncos de madera radica en que cuando las cámaras son usadas en un ambiente de alta humedad, las lentes de las cámaras suelen empañarse, con lo cual se distorsiona enormemente la reconstrucción volumétrica de los troncos de madera. Una solución a este problema consistiría en limpiar constantemente las lentes de las cámaras. Sin embargo, esto último convierte a la detección de la forma volumétrica de troncos de madera (rollizos) mediante cámaras en un método poco práctico y menos eficiente.

c. Reconstrucción de volúmenes mediante el uso de señales ultrasonido

El presente trabajo de tesis propone un método para obtener la reconstrucción de volúmenes basado en el uso de sensores ultrasonido. Los ultrasonidos son básicamente sonido como el que los seres humanos oímos normalmente, con la diferencia de que tienen una frecuencia mayor que la máxima audible por el ser humano. Así, mientras que el rango de frecuencia audible por el ser humano empieza en 16 Hz y llega a un máximo de aproximadamente 20 KHz, para detectar la forma de los troncos de madera (o rollizos) se hará uso de sonido con una frecuencia de 40 KHz (ultrasonidos) [10,11,12].

El funcionamiento básico de los ultrasonidos como medidores de distancia se muestra en la Figura 2.4. Se tiene un transmisor que emite un pulso de ultrasonido, que rebota sobre un determinado objeto y la reflexión de ese pulso es detectada por un receptor de ultrasonidos [10].

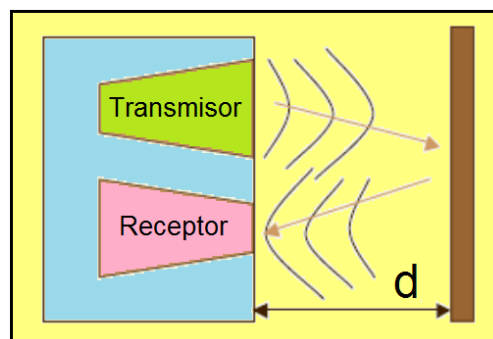


Figura 2.4 Uso de ultrasonido como medidor de distancia [10]

La mayoría de sensores ultrasonido de bajo costo se basan en la emisión de un pulso de ultrasonido cuyo campo de acción es de forma cónica. Luego, en la Figura 2.4, midiendo el tiempo que transcurre entre la emisión del ultrasonido y la recepción del eco del mismo, se puede calcular a distancia a la que se encuentra el objeto que ha producido la reflexión sonora, mediante la fórmula que se aprecia en la Figura 2.5 [10].

$$d = \frac{V \cdot t}{2}$$

Figura 2.5 Fórmula para calcular la distancia a la que se encuentra un objeto, utilizando señales ultrasonido [10].

Donde 'V' es la velocidad del sonido en el aire y 't' es el tiempo transcurrido desde la emisión del pulso de ultrasonido (en el emisor del sensor) hasta que se produce la recepción del eco del pulso ultrasonido (en el receptor del sensor) [10].

Una vez almacenadas las distancias del centro del rollizo a todos los puntos de la superficie del rollizo, mediante un trabajo de programación en un entorno de programación visual a ser definido (como Visual Basic, por ejemplo), dicha información puede ser nuevamente procesada para obtener finalmente una imagen tridimensional de la forma real del rollizo en la pantalla de una computadora.

2.1.3 Síntesis sobre el asunto de estudio

Del recuento de las líneas de investigación y propuestas tecnológicas orientadas a la optimización del proceso de corte de troncos de madera en los aserraderos [1,2,3,4,5,6,7], se puede obtener la síntesis siguiente:

- ✓ Debido a la necesidad de obtener la mayor cantidad de madera dimensionada útil a partir de troncos de forma irregular como resultado del proceso de corte en los aserraderos, este proceso está siendo automatizado.
- ✓ La obtención de una imagen volumétrica que reproduzca fielmente la forma irregular de los troncos de madera juega un papel muy importante en la automatización del proceso de corte en los aserraderos.
- ✓ Otro punto importante, luego de obtener la imagen volumétrica de los troncos de madera (rollizos), es el desarrollo de un software de optimización para el diseño y selección de los esquemas de corte, para lo cual el mejor método se basa en el uso de tecnología computacional que implemente algoritmos matemáticos de optimización.

2.2 Modelo teórico

En conclusión, la optimización del proceso de corte de rollizos en los aserraderos tiene como finalidad obtener diseños de corte suficientemente buenos, de tal manera que se desperdicie la menor cantidad posible de materia prima. Como indica Vilaboa [1], para este fin resulta esencial contar con un sistema que sea capaz de reproducir la forma volumétrica del tronco de madera o rollizo, antes de decidir cómo cortarlo.

Teniendo en cuenta lo expuesto anteriormente, una buena alternativa para el requerimiento de detectar la forma volumétrica de los rollizos consiste en el uso de sensores ultrasonido. En particular, para el presente trabajo, se propone el uso del sensor ultrasonido SRF04, que es un sensor ultrasonido medidor de distancias comercial y bastante fácil de usar.

El sensor SRF04 sería capaz de efectuar la toma de datos correspondientes a los puntos de la superficie del rollizo. Luego, mediante el desarrollo de un programa en un entorno de programación visual como por ejemplo Visual Basic, estos datos podrían ser almacenados en matrices y luego procesados de tal forma que en base a las distancias medidas por el sensor SRF04 (es decir, las distancias del sensor a los puntos de la superficie rolliza) se logre reproducir la forma volumétrica del tronco de madera o rollizo, y ésta sea visualizada en la pantalla de una computadora.

Así, el procedimiento de toma de datos del sensor ultrasonido SRF04 se repetiría hasta obtener la mayor cantidad de datos acerca de cada punto de la superficie rolliza. Esto último se conseguirá montando dos sensores en extremos diametralmente opuestos de una plataforma circular, sobre la cual los sensores se desplazarán a velocidad constante. Luego, el tronco de madera atravesará perpendicularmente esta plataforma circular, desplazándose a lo largo de su eje axial mientras se efectúa la toma de datos por parte de los sensores (operando de forma similar a como funciona un tomógrafo), tal como se aprecia en la Figura 2.6.

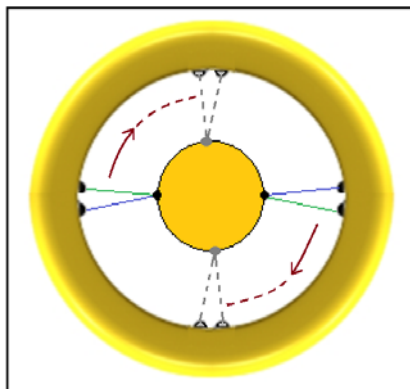


Figura 2.6 Toma de datos de los sensores ultrasonido

En la Figura 2.7 a 2.9 se grafica, de forma muy general, cómo sería la toma de datos de la superficie rolliza para los primeros tres anillos (secciones axiales) detectados por los sensores ultrasonido SRF04. Como se puede intuir gráficamente, incrementando el número de muestras por cada anillo y el número de anillos detectados, el posterior arreglo de todos los anillos en un espacio tridimensional reproducirá la forma volumétrica del tronco de madera o rollizo.

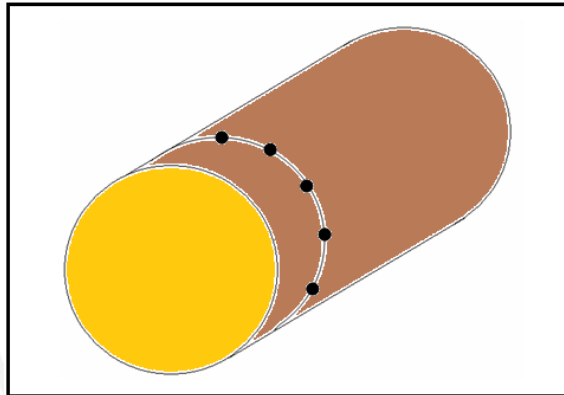


Figura 2.7 Toma de datos del primer anillo del rollizo

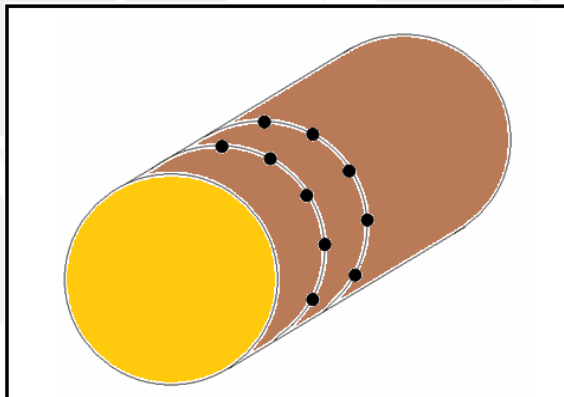


Figura 2.8 Toma de datos del segundo anillo del rollizo

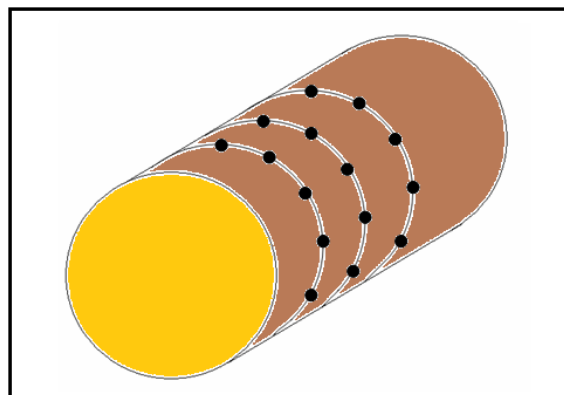


Figura 2.9 Toma de datos del tercer anillo del rollizo

Cabe señalar que es precisamente mediante el uso de un software visual y un trabajo de programación que los anillos serán arreglados (uno a continuación del otro) y presentados en un espacio tridimensional, para posteriormente seleccionar el mejor esquema de corte.

2.2.1 El sensor ultrasonido SRF04

Dado que el sistema de reconstrucción de volúmenes se basará en el uso de señales ultrasonido, a continuación se presenta una breve descripción del sensor ultrasonido a utilizar, el sensor SRF04, el mismo que fue elegido por razones de fiabilidad, prestaciones y costo.

a. Descripción

El SRF04 es un sensor de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentran objetos en un rango de 3 a 300 cm. El sensor funciona por ultrasonidos y contiene toda la electrónica encargada de hacer la medición. En líneas generales, su uso se basa en enviar un pulso de arranque y medir la duración del pulso de retorno. De muy pequeño tamaño, el sensor SRF04 destaca por su bajo consumo, gran precisión y bajo precio [10,11,12,13].

b. Funcionamiento

El sensor SRF04 funciona emitiendo pulsos de ultrasonido inaudibles para el oído humano. Los pulsos emitidos viajan a la velocidad del sonido hasta alcanzar un objeto, entonces el sonido es reflejado y captado de nuevo por el receptor del sensor. Lo que hace el controlador incorporado es emitir una ráfaga de pulsos, y a continuación empieza a contar el tiempo que tarda en llegar el eco. Este tiempo se traduce en un pulso de eco de ancho proporcional a la distancia a la que se encuentra el objeto.

Desde un punto de vista práctico, lo que hay que hacer es enviar una señal de arranque o disparo (de duración mínima de 10 μ s) en el pin 3 del sensor SRF04 (pin de disparo). Luego, leemos el ancho del pulso que proporciona el pin 4 del sensor ultrasonido SRF04 (pin de salida de eco) y medimos su duración (esta duración del pulso de eco será proporcional a la distancia medida, y variará entre 100 μ s y 8 ms).

En caso de que no se produzca ningún eco (porque no se encuentra un objeto), el pulso de eco tendrá una longitud aproximada de 36 ms. Asimismo, tener en cuenta que debe haber un retardo de al menos 10 ms desde que se hace una lectura hasta

que se inicia el siguiente pulso de disparo, con la finalidad de que el circuito se estabilice. En concordancia con lo explicado respecto al funcionamiento, en la Figura 2.10 se muestra el diagrama de tiempos del funcionamiento del sensor SRF04 [10,11,12,13].

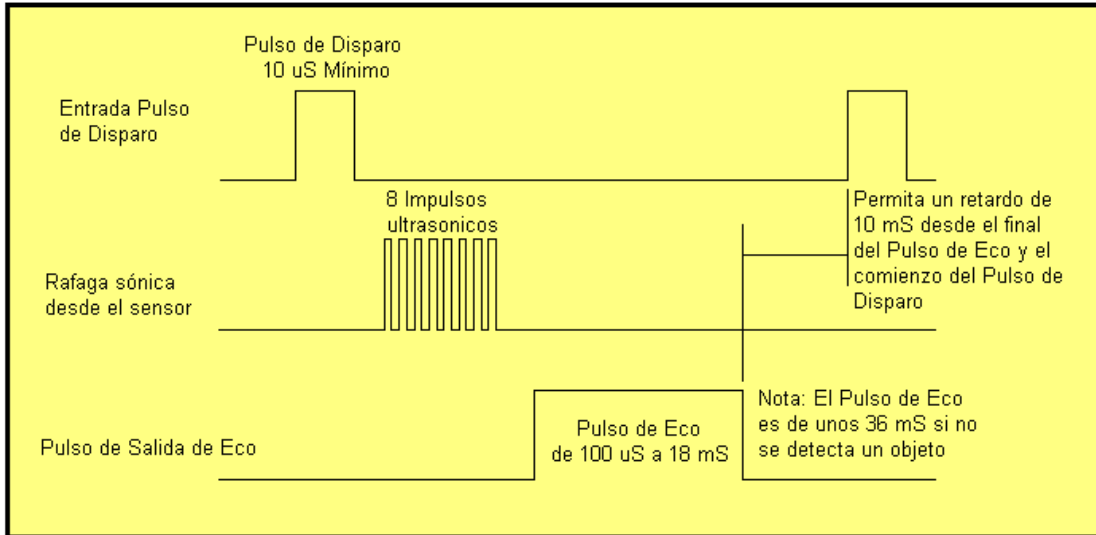


Figura 2.10 Diagrama de tiempos del sensor SRF04 [13]

c. Funciones de los pines del sensor SRF04

Como se puede apreciar en la Figura 2.11, el sensor ultrasonido SRF04 tiene cinco pines: pin de tierra (GND), pin N.C. (Not Connect), pin de disparo, pin de salida de eco y pin de alimentación (+5 Vcc) [10,11,12,13].

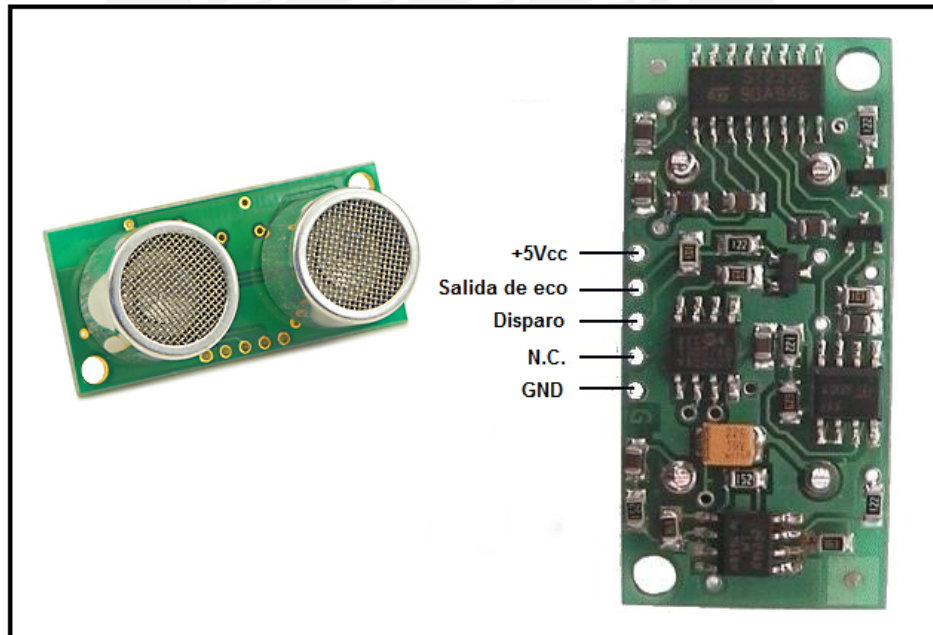


Figura 2.11 Diagrama de pines del sensor SRF04 [11,12]

Asimismo, en la Figura 2.12 se detalla la función de cada uno de los pines del sensor SRF04 [12].

Pin	Función
+5 Vcc	Tensión de alimentación.
Pin de eco	Salida del pulso cuya duración determina el tiempo de recorrido de la señal ultrasonido, proporcional a la distancia medida.
Pin de disparo	Entrada de inicio para una nueva medición. Por este pin se debe aplicar un pulso con una duración mínima de 10 μ s.
N.C.	Línea sin conexión. Se emplea en la fase de fabricación y comprobación del propio módulo. No se debe conectar nada a este pin.
GND	Tierra de alimentación.

Figura 2.12 Función de los pines del sensor SRF04 [12]

d. Características técnicas

En la Figura 2.13 se muestra un cuadro con las principales características técnicas del sensor ultrasonido SRF04 [12].

Parámetro	Valor	Unidad
Dimensiones del circuito	43x20x17	mm
Tensión de alimentación	5	Vcc
Frecuencia de trabajo	40	KHz
Rango máximo de medición de distancia	3	m
Rango mínimo de medición de distancia	3	cm
Duración mínima del pulso de disparo (nivel TTL)	10	μ s
Duración del pulso eco de salida (nivel TTL)	100 a 18000	μ s
Tiempo mínimo de espera entre una medida y el inicio de otra	10	ms

Figura 2.13 Características técnicas del sensor SRF04 [12]

Como una nota adicional, cabe señalar que debido a sus características técnicas, el sensor ultrasonido SRF04 es muy utilizado, por ejemplo, como detector de obstáculos para robots con navegación autónoma. Así, para robots pequeños un solo sensor ultrasonido SRF04 es suficiente para cubrir la parte frontal del robot y así éste detecte si algún obstáculo se presenta en su recorrido. En cambio, para robots de un tamaño mayor pueden ser necesario dos o más sensores. Incluso es una práctica muy común en el diseño de robots el colocar cinco o más sensores detectores de distancia (similares al sensor ultrasonido SRF04) alrededor de todo el robot, de tal modo que éste pueda detectar objetos no sólo por su parte frontal, sino en todas direcciones. Otra práctica muy común en el diseño de robots es montar un sensor medidor de distancia en un servo, el cual mueve el sensor 180 grados sexagesimales mientras se efectúan las mediciones, funcionando como una especie de radar [13].

2.2.2 Regulación de la velocidad

Como se grafica en la Figura 2.14, los dos sensores ultrasonido irían montados en extremos diametralmente opuestos de una plataforma circular, sobre la cual se desplazarán para tomar los datos correspondientes a los puntos que conforman la superficie externa de los rollizos.

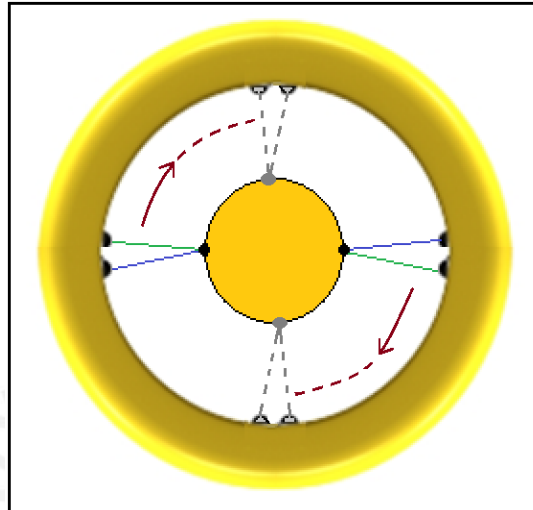


Figura 2.14 Desplazamiento de los sensores ultrasonido (montados sobre la plataforma circular)

En tal sentido, es necesario que la velocidad de desplazamiento de los sensores sobre la plataforma circular sea constante, pues de no ser así la imagen a obtener resultaría distorsionada.

Es por lo anterior que se plantea el uso de un encoder óptico, el cual sensorará la velocidad del motor DC que acciona el desplazamiento de los sensores, y mediante un circuito de realimentación se ajustará el ciclo de trabajo de una señal PWM, de tal manera que ésta modifique la velocidad de giro del eje del motor DC hasta obtener la velocidad constante requerida. El diagrama de bloques del circuito de realimentación que se usará se muestra en la Figura 2.15.

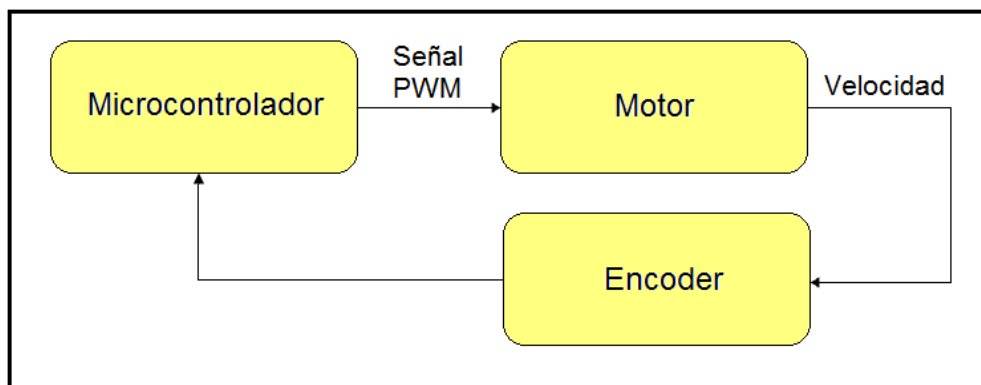


Figura 2.15 Diagrama de bloques del circuito de realimentación para regular la velocidad de giro del eje del motor

Encoder óptico

Como se indica en [14], un encoder es un transductor que transforma el movimiento angular en una serie de pulsos digitales. Estos pulsos generados pueden luego ser utilizados para controlar los desplazamientos de tipo angular.

En el caso de un encoder óptico, la detección del movimiento angular opera en base al principio de exploración fotoeléctrica. El sistema de lectura se basa en la rotación de un disco con ranuras idénticas, igualmente espaciadas, y que se encuentra fijado al eje del motor (por lo cual girará a la misma velocidad angular que dicho eje), tal como se muestra en la Figura 2.15.

El disco será iluminado de modo perpendicular por una fuente de rayos infrarrojos. Las ranuras del disco permitirán o denegarán el paso de la luz (desde el emisor hacia el receptor) de forma alternada. Estos cambios serán detectados por un sensor lumínico (generalmente un fotodiodo o fototransistor) al que en general llamaremos 'receptor', los mismos que luego serán leídos y procesados por un microcontrolador.

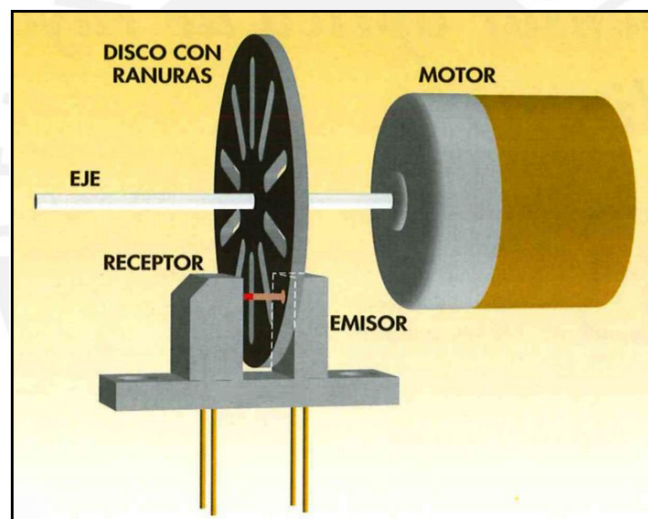


Figura 2.16 Diagrama de funcionamiento de un encoder óptico [32]

En la Figura 2.16 se muestra un bosquejo del funcionamiento de un encoder óptico. Veamos ahora cómo se regula la velocidad del motor en base a la lectura de los pulsos que llegan (o no) al receptor:

- Dado que el disco con las ranuras está fijado al eje del motor, este disco girará con la misma velocidad angular que el eje del motor.
- Habrá una luz emitida constantemente desde el emisor. Dado que el disco se encuentra entre el emisor y el receptor, notamos que la luz únicamente llegará al receptor cuando atravesase las ranuras del disco. De esta forma, en el receptor se leerá una señal cuadrada (con nivel alto cuando la luz pasa a través de las ranuras, y con nivel bajo cuando el disco interrumpe el paso de la luz).

- Si la velocidad angular del eje del motor (y del disco) es muy grande, entonces en el receptor se leerá una onda cuadrada de periodo muy pequeño.
- Si la velocidad angular del eje del motor (y del disco) es muy pequeña, entonces en el receptor se leerá una onda cuadrada de periodo muy grande.
- El microcontrolador se encargará de leer el periodo de la onda cuadrada en el receptor y, teniendo un periodo de onda 'objetivo' (asociado a la velocidad angular deseada), el microcontrolador se encargará de modificar la señal PWM de la siguiente forma:
 - ✓ La señal PWM aumentará la velocidad angular del motor si ésta es menor que la velocidad angular deseada (es decir, si el periodo de la onda cuadrada leída en el receptor es mayor al periodo de onda 'objetivo').
 - ✓ La señal PWM reducirá la velocidad angular del motor si ésta es mayor que la velocidad angular deseada (es decir, si el periodo de la onda cuadrada leída en el receptor es menor al periodo de onda 'objetivo').

Como ya se graficó en la Figura 2.15, esta es la explicación por la cual el sistema para regular la velocidad de giro del eje del motor es un circuito realimentado (en base a la información proporcionada por el encoder).

2.2.3 Desarrollo en un entorno de programación visual

La etapa de desarrollo haciendo uso de un entorno de programación visual (aún por definirse) constituye una parte esencial en el diseño del sistema de reconstrucción de volúmenes, pues es allí donde se procesan los datos obtenidos por los sensores ultrasonido y se obtiene la caracterización volumétrica de los troncos de madera.

Debido al requerimiento de gráficos, entre las opciones del entorno de programación visual a utilizar se consideró:

- **Visual C++**

Visual C++ es un entorno de desarrollo integrado para lenguajes de programación C y C++. Es un software propietario de Microsoft Corporation [15].

- **Visual Basic**

Visual Basic es un entorno de desarrollo integrado propietario de Microsoft Corporation, creado con la intención de simplificar la programación. Además de un editor de código fuente, un depurador y un compilador, incluye un editor para interfaces gráficas, que facilita en gran medida el desarrollo de aplicaciones [16].

- **MatLab**

MatLab es un lenguaje de alto nivel propietario de MathWorks, muy utilizado en el sector industrial, científico y educativo; con aplicaciones en diversos ámbitos, tales como el cálculo numérico, la visualización y la programación [17].

**CAPÍTULO 3:
DISEÑO DE LOS MÓDULOS COMPONENTES DEL SISTEMA DE
RECONSTRUCCIÓN DE VOLÚMENES**



3.1 Descripción contextual

El sistema de reconstrucción de volúmenes propuesto en el presente trabajo de tesis tiene como objetivo la reproducción de la forma volumétrica de troncos de madera. Esto permitiría optimizar el proceso de corte llevado a cabo en los aserraderos mediante la aplicación de diagramas de corte apropiados, reduciendo las pérdidas de materia prima y mejorando la rentabilidad de las empresas madereras.

Visto desde otro punto de vista, el sistema propuesto no sólo ayudaría a mejorar los ratios financieros de las empresas madereras, sino que también (al permitir un uso más eficiente de la materia prima) reduciría la tala innecesaria de árboles en los bosques madereros, contribuyendo a la conservación del ecosistema en el planeta.

3.2 Diagrama de bloques

El diagrama de bloques general del sistema propuesto se presenta en la Figura 3.1.

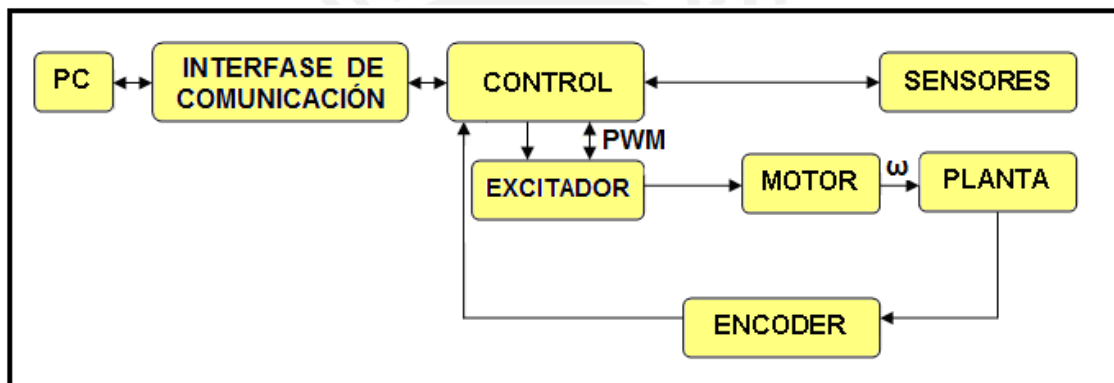


Figura 3.1 Diagrama de bloques del sistema propuesto

3.3 Explicación del diseño

En la Figura 3.2 se muestra una vista exterior del sistema de reconstrucción de volúmenes diseñado en el presente trabajo de tesis.

La plataforma circular que se aprecia en la Figura 3.2, mediante un sistema de engranajes accionados por un motor, será capaz de desplazar a velocidad constante los dos sensores ultrasonido que se encuentran montados en dos extremos diametralmente opuestos de dicha plataforma circular.

Los troncos de madera atravesarán perpendicularmente dicha plataforma circular, también a velocidad constante, moviéndose a lo largo de su eje axial (operando de forma similar a como funciona un tomógrafo). En la Figura 3.2, se puede apreciar que dichos troncos serían, eventualmente, colocados sobre el sistema de rodillos en forma de 'V'. Estos rodillos serían movidos por la acción de uno o dos motores, los cuales proporcionarían a los troncos de madera la traslación a velocidad constante requerida para atravesar perpendicularmente la plataforma circular.

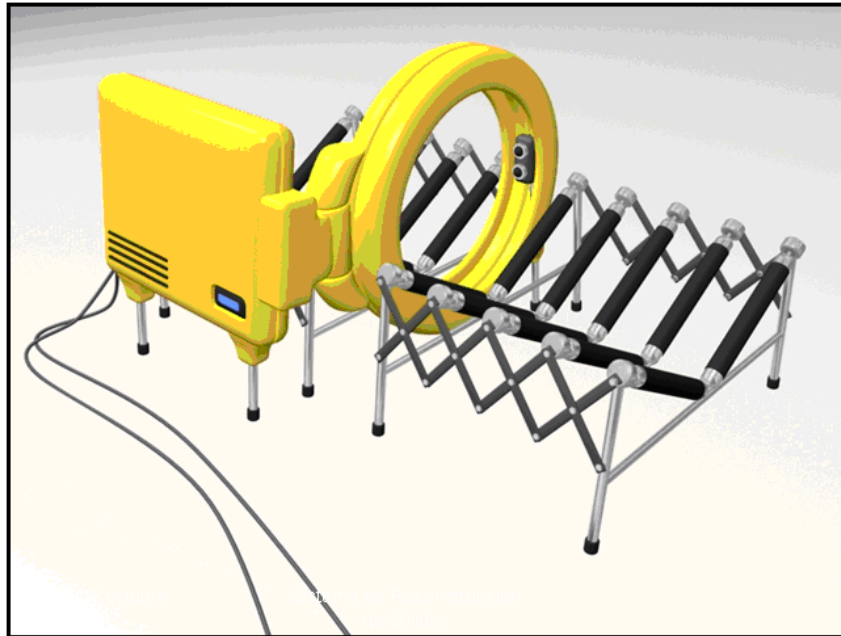


Figura 3.2 Vista exterior del sistema de reconstrucción de volúmenes [18]

El desarrollo del presente trabajo excluirá el diseño del sistema que moviliza los troncos a velocidad constante (sistema de rodillos en forma de 'V' que se aprecia en la Figura 3.2), debido a que los aserraderos ya poseen sistemas de esta naturaleza, provistos de motores de gran potencia capaces de movilizar troncos de gran tamaño y peso.

Como ya se mencionó, a medida que los troncos atraviesan perpendicularmente la plataforma circular a velocidad constante (moviéndose a lo largo de su eje axial), los dos sensores ubicados en extremos diametralmente opuestos de la plataforma circular se desplazarán simultáneamente (también a velocidad constante) sobre dicha plataforma, efectuando la toma de datos.

Resulta importante acotar que según el diseño propuesto, dado que usaremos sólo dos sensores, cada sensor recorrerá únicamente 180 grados sexagesimales sobre la plataforma circular (cada uno tomará un mitad de la información correspondiente a un anillo completo o sección axial del tronco), luego de lo cual regresarán a su posición original, y este proceso de recorrido se repetirá mientras dure la toma de datos.

Luego, los datos obtenidos por los sensores serán procesados por un sistema de procesamiento de datos y, una vez ya procesados, serán enviados a una computadora por medio de una interface de comunicación.

Finalmente, en la computadora los datos recibidos serán procesados nuevamente, de tal modo que por medio de un trabajo de programación (haciendo uso de un lenguaje

visual), estos datos serán usados para reproducir la forma volumétrica de los troncos de madera, la cual será presentada en la pantalla de la computadora.

En base a lo explicado, el diseño del sistema de reconstrucción de volúmenes fue dividido en tres partes:

- **Parte mecánica**

Involucra el diseño del sistema mecánico que proporciona el desplazamiento de los sensores sobre la plataforma circular.

- **Parte hardware**

Involucra el diseño del sistema de control de velocidad y giro de un motor DC, diseño del sistema de toma de datos, diseño de la interface de comunicación, etc.

- **Parte software**

Involucra el diseño de los algoritmos para la visualización gráfica del rollizo y para el procesamiento de los datos en el bloque de control.

3.4 Parte mecánica

La parte mecánica sería la planta propiamente dicha, y estaría conformada principalmente por la plataforma circular antes descrita, la misma que a través de un sistema de engranajes accionados por un motor, proporcionaría el desplazamiento de los sensores sobre dicha plataforma.

Dados los objetivos y el alcance del presente trabajo, la parte mecánica será omitida para poner un mayor énfasis en la parte hardware y software.

3.5 Parte hardware

3.5.1 Sensores

a. Requisitos mínimos

El sensor ultrasonido que vayamos a usar debe cumplir con los siguientes requisitos:

- Debe ser de gran fiabilidad para el proceso de medición de distancias. Es por esto que se debe escoger un sensor de los que, debido a su uso, sea comúnmente denominado 'medidor de distancias'.
- Debe ser de un costo accesible, en caso de requerirse la realización de pruebas.

b. Opciones

En función de los requisitos mínimos, se consideraron dos opciones:

- **Sensor Ultrasonido PING de Parallax:**

Conocido sensor usado como medidor de distancias que posee tres pines (GND, 5V, SIG), así como un emisor y un receptor de ultrasonidos. En la Figura 3.3 se muestra la vista exterior del sensor ultrasonido PING de Parallax [19].

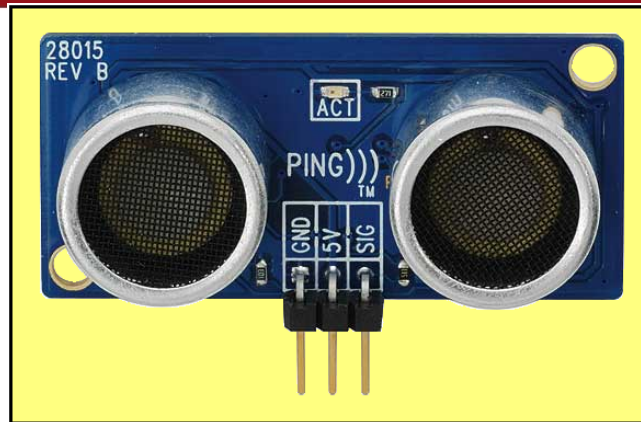


Figura 3.3 Vista exterior del sensor ultrasonido PING de Parallax [19]

- **Sensor Ultrasonido SRF04:**

Conocido sensor usado como medidor de distancias que posee cinco pines (GND, N.C., Disparo, Eco y 5V), así como un emisor y un receptor de ultrasonidos. En la Figura 3.4 se muestra la vista exterior del sensor ultrasonido SRF04 [20].

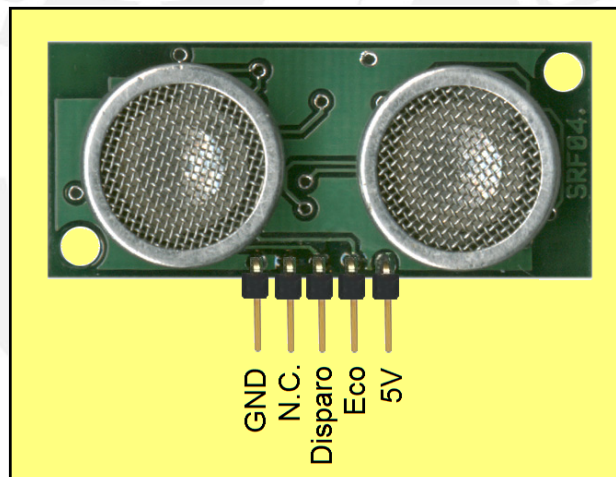


Figura 3.4 Vista exterior del sensor ultrasonido SRF04 [20]

c. Criterios de selección:

Para el desarrollo del presente trabajo de tesis, se eligió el sensor ultrasonido SRF04 en base a los siguientes criterios:

- **Facilidad de uso:** El sensor ultrasonido SRF04 tiene un pin de disparo y otro pin de eco (ver Figura 3.4); a diferencia del sensor ultrasonido PING de Parallax, en el cual el pin SIG (ver Figura 3.3) cumple la función de pin de disparo y también la función de pin de eco. Así, al elegir el sensor ultrasonido SRF04, el hecho de que cada pin cumpla una función distinta facilitará el trabajo de programación.

- **Fiabilidad:** El sensor ultrasonido SRF04 es muy fiable en el proceso de detección de objetos. Para darnos una idea de su fiabilidad, este sensor es capaz de detectar un palo de escoba a una distancia de 3 metros.
- **Costo:** El sensor ultrasonido SRF04 es menos costoso que el sensor PING de Parallax. Además de ello, el sensor SRF04 es mucho más comercial (es más fácil encontrar un sensor ultrasonido SRF04 en el mercado nacional).

d. Características del sensor ultrasonido SRF04

Las principales características del sensor ultrasonido SRF04 son citadas a continuación [10,11,12]:

- ✓ Tensión de alimentación de 5V.
- ✓ Consumo típico de 30 mA.
- ✓ Consumo máximo de 50 mA.
- ✓ Rango de medición de 3 a 300 cm.
- ✓ Capaz de detectar un palo de escoba a 3 metros de distancia.
- ✓ Pulso de disparo mínimo de 10 μ s, con lógica TTL.
- ✓ Pulso de eco de 100 μ s a 8 ms.
- ✓ Dimensiones de 43x20x17 mm.
- ✓ Peso de 10 gramos aproximadamente.

3.5.2 Control

El bloque de control (ver diagrama de bloques en la Figura 3.1) hará uso de un microcontrolador para las siguientes funciones:

- **Toma de datos de los sensores:** El microcontrolador enviará la señal de disparo para que los sensores ultrasonido empiecen a efectuar la toma de datos de forma sincronizada. Del mismo modo, el microcontrolador recibirá la señal de eco enviada por los sensores.
- **Regulación de la velocidad de giro del eje del motor:** El microcontrolador se encargará de regular la velocidad de desplazamiento de los sensores sobre la plataforma circular, la cual (como se mencionó anteriormente) debe ser constante.

Del mismo modo, el microcontrolador se encargará de recibir la señal de cambio de sentido de desplazamiento de los sensores sobre la plataforma circular, y enviará la orden para invertir dicho sentido de desplazamiento (de tal modo que los sensores regresen a su posición inicial sobre la plataforma circular).

- **Comunicación con la PC vía puerto serie:** El microcontrolador, luego de recibir la información enviada por los sensores ultrasonido, la procesará para traducirla a distancia medida y finalmente la re-enviará a la PC por el puerto serie.

a. Microcontrolador

Requisitos mínimos

El microcontrolador que vayamos a usar debe cumplir con los siguientes requisitos mínimos (propios del sistema de reconstrucción de volúmenes diseñado):

- El microcontrolador debe permitir tanto la entrada como la salida de datos, razón por la cual debe contar con un número suficiente de puertos entrada/salida.
- Debido a que el presente trabajo de tesis se encuentra avocado al diseño y no a la implementación, el microcontrolador elegido debe contar con un software que permita la realización de simulaciones de circuitos, de tal manera que se pueda comprobar el correcto funcionamiento del diseño.
- El microcontrolador debe ser de una programación sencilla y fácil de aprender.

Opciones

En función de los requisitos mínimos que debe cumplir el microcontrolador a usar, se consideraron dos opciones:

- **Microcontrolador ATmega8:** Se tuvo en consideración este microcontrolador debido a la familiaridad en el uso del mismo en la Pontificia Universidad Católica del Perú (PUCP). Posee arquitectura CISC.
- **Microcontrolador PIC16F84:** Microcontrolador de uso muy popular por aficionados a la robótica y electrónica. Posee arquitectura RISC.

Criterios de selección:

Para el desarrollo del presente trabajo de tesis, se eligió el microcontrolador PIC16F84 en base a los siguientes criterios:

- **Simulación:** El microcontrolador PIC16F84 cuenta con un software de simulación gratuito (Proteus), bastante versátil y muy fiable para la simulación de circuitos diseñados en base a este microcontrolador.
- **Facilidad de uso:** Conocimiento del presente tesista en la programación de aplicaciones usando el microcontrolador PIC16F84. Además de ello, este microcontrolador cuenta con una vasta información técnica, documentación de proyectos y aplicaciones, foros de discusión, etc.
- **Costo:** El microcontrolador PIC16F84 es menos costoso que el ATmega8.

Características del microcontrolador PIC16F84

Las principales características del microcontrolador PIC16F84 son citadas a continuación: [21,22,23,24,25]:

- ✓ Circuito integrado de 18 pines.
- ✓ Conjunto de instrucciones RISC, muy fácil de memorizar y entender.
- ✓ Dos puertos de entrada/salida (I/O): cinco pines en el puerto A y ocho pines en el puerto B.
- ✓ Memoria Flash de programa (1Kx14).
- ✓ Memoria EEPROM de datos (64x8).
- ✓ Memoria RAM (68x8)
- ✓ Un temporizador/contador de 8 bits

En la Figura 3.5 se muestra el encapsulado de 18 pines PIC16F84.

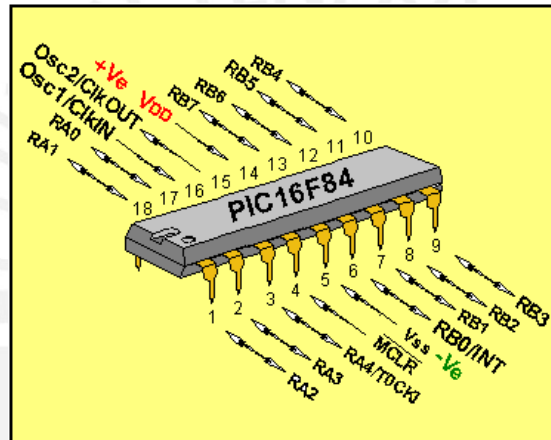


Figura 3.5 Diagrama de pines del microcontrolador PIC16F84 [26]

b. Toma de datos de los sensores

En la Figura 3.6 se muestra un diagrama bastante simplificado de cómo el sensor ultrasonido SRF04 efectuaría la toma de un dato según el diseño del sistema de reconstrucción de volúmenes propuesto en el presente trabajo de tesis. Como allí se puede apreciar, teniendo ya energizado el sensor ultrasonido SRF04 (conectando los pines de alimentación y GND a una fuente de +5 Vcc y a tierra, respectivamente), el microcontrolador PIC16F84 enviará primero un pulso de disparo (de una duración mínima de 10 μ s) que llegará al pin de disparo del sensor ultrasonido SRF04. Este pulso de disparo es la señal para que el sensor efectúe una toma de datos, por lo cual el sensor emitirá una señal de ultrasonido con dirección al objeto a detectar (en este caso, el tronco de madera o rollizo).

Luego, internamente el sensor empezará a contar el tiempo que tarda en llegar el eco producido por el rebote de la señal ultrasonido con la superficie del tronco de madera. Este tiempo se traduce en un pulso de eco (de duración proporcional a la distancia a

la que se encuentra el tronco de madera) que el sensor SRF04 arrojará como salida por el pin de salida de eco. Este pulso de salida de eco será luego recibido por el microcontrolador PIC16F84, el cual procesará la información.

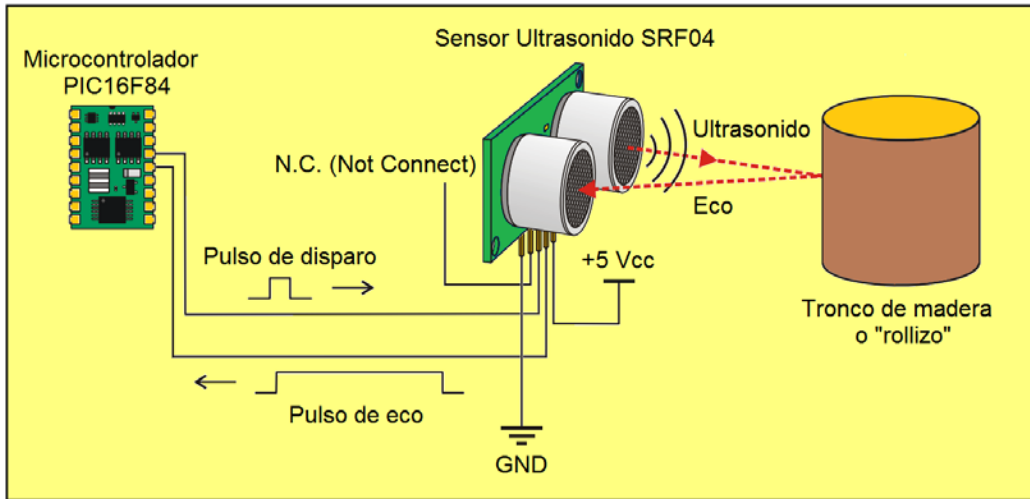


Figura 3.6 Toma de datos del sensor ultrasonido SRF04 [19]

En cuanto al procesamiento que efectuará el microcontrolador PIC16F84 teniendo como entrada el pulso de eco, resulta importante recordar las especificaciones técnicas del sensor ultrasonido SRF04 (ver Figura 2.13), donde se indica que el pulso de eco del sensor varía de 100 μ s a 18 ms, que corresponden proporcionalmente a mediciones de distancias (entre el sensor SRF04 y el tronco de madera) de 3 a 300 centímetros. Así, tenemos que:

$$\text{Rango de pulso de eco} = 18\text{ms} - 100 \mu\text{s} = 18000 \mu\text{s} - 100 \mu\text{s} = 17900 \mu\text{s}$$

$$\text{Rango de distancia medible} = 300 \text{ cm} - 3 \text{ cm} = 297 \text{ cm}$$

Entonces, haciendo la proporción:

$$(\text{Pulso de eco} / \text{Distancia medida por el sensor}) = 17900 \mu\text{s} / 297 \text{ cm} \approx 60 \mu\text{s} / \text{cm}$$

Este cálculo nos indica que por cada centímetro de distancia medida por el sensor ultrasonido SRF04 la duración del pulso de eco se incrementa en 60 μ s. De esta forma, para obtener la distancia medida en centímetros, el microcontrolador PIC16F84 debe contar únicamente el número de periodos de 60 μ s que conforman el pulso de eco, y este número de periodos será el equivalente a la distancia medida en centímetros. Por ejemplo, si el pulso de eco recibido por el microcontrolador tuviera una duración de 420 μ s, entonces la distancia medida por el sensor ultrasonido SRF04 se calcula fácilmente mediante una simple división: $(420 \mu\text{s} / 60 \mu\text{s}) = 7 \text{ cm}$.

c. Regulación de la velocidad de giro del motor

Partiremos del hecho de que la velocidad de giro de un motor DC depende del valor medio de la tensión aplicada en sus extremos. Por ello, para controlar la velocidad

de giro del eje del motor DC que moverá los sensores ultrasonido sobre la plataforma circular, se utilizará un sistema de modulación por ancho de pulso (o PWM: Pulse Width Modulation, en inglés) de una señal TTL, tal como se muestra en la Figura 3.7. Bajo el control de esta señal PWM, el eje del motor DC girará a una velocidad determinada por la media del nivel de la señal cuadrada.

Como se puede observar en la Figura 3.7, la tensión continua media aplicada al motor se controla manteniendo constante la frecuencia de la señal cuadrada, y variando el tiempo durante el cual la señal permanece a nivel alto, es decir, variando el Ciclo de Trabajo o Duty Cycle. De esta manera, si el ciclo de trabajo es del 50% se suministrará al motor una tensión media equivalente a la mitad de la tensión máxima. Del mismo modo, con un ciclo de trabajo del 20% sólo una quinta parte de la tensión máxima será suministrada a la carga. En este caso, como se puede concluir fácilmente, el hecho de disminuir el ciclo de trabajo de 50% a 20% conllevará a una disminución de la velocidad del motor (y viceversa, un incremento del ciclo de trabajo conllevaría a un incremento de la velocidad del motor).

Para un mejor entendimiento, en la Figura 3.7 se grafica cómo variaría la intensidad de una bombilla cuando ésta es alimentada por tensiones con diferentes ciclos de trabajo. Ver que cuanto mayor sea el ciclo de trabajo, mayor será la intensidad con que brillará la bombilla, y viceversa. Siguiendo esta línea de razonamiento, para el sistema propuesto en el presente trabajo, la intensidad de la bombilla representaría la magnitud de la velocidad a la cual girará el eje del motor DC.

En resumen, vemos que controlando el ciclo de trabajo de una señal PWM (a su vez, regulada por el microcontrolador) podremos controlar fácilmente la velocidad de giro del eje del motor DC [21,22,23,24,25].

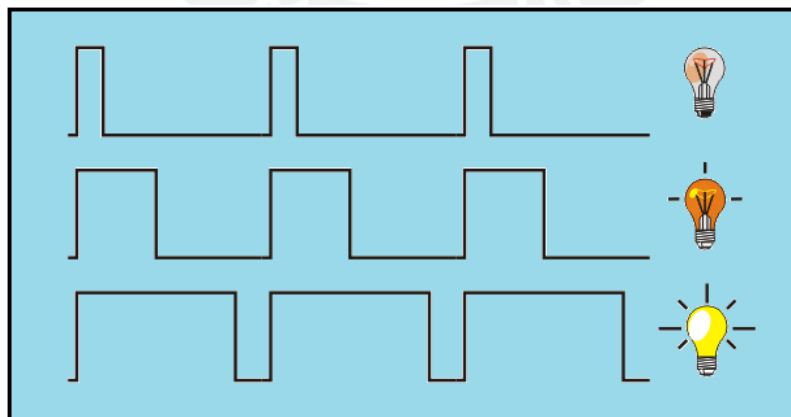


Figura 3.7 Diagrama de modulación por ancho de pulso

3.5.3 Excitador

El bloque excitador (ver diagrama de bloques del sistema, mostrado en la Figura 3.1) es el encargado de controlar el sentido de giro del motor (en acción conjunta con el

bloque de control). En el presente trabajo el excitador estará conformado por el Driver L293B, que es un encapsulado 16 pines, tal como se muestra en la Figura 3.8, donde se muestra además una tabla con el nombre y la función que cumple cada pin [27].

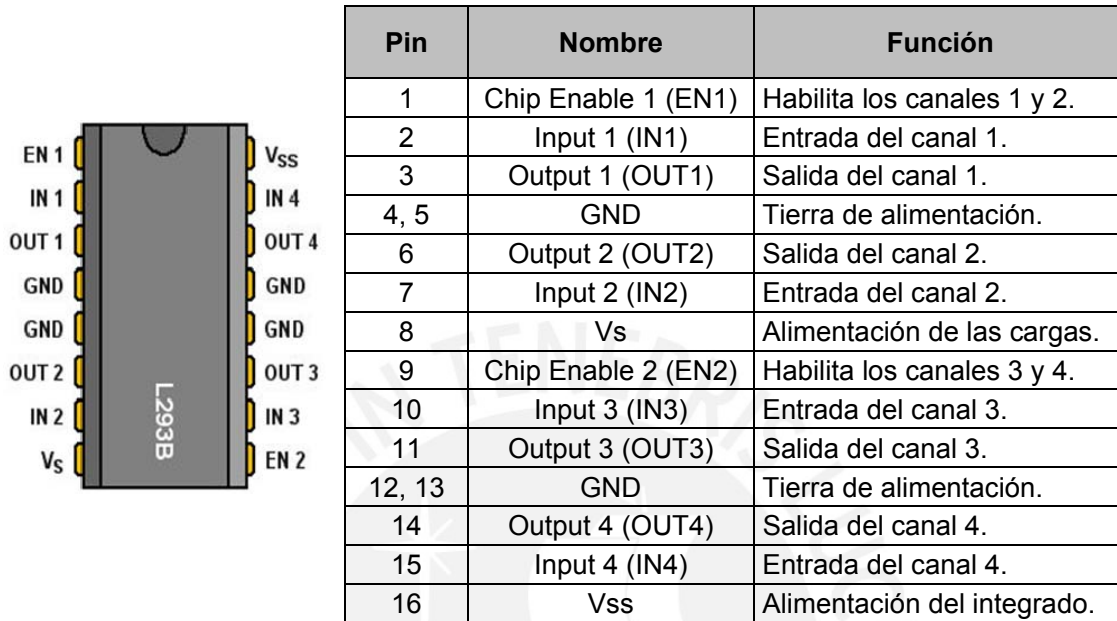


Figura 3.8 Diagrama de pines del encapsulado L293B [27]

El Driver L393B posee cuatro canales capaces de proporcionar una corriente de salida de hasta 1A por canal. Asimismo, cada canal es controlado por señales de entrada compatibles TTL y cada pareja de canales dispone de una señal de habilitación que desconecta la salida de los mismos. Como se puede observar en la Figura 3.8, el Driver L293B dispone de un pin para la alimentación de las cargas que se están controlando (Vs, pin 8), de manera que dicha alimentación es independiente de la lógica de control [27].

En la Figura 3.9 se presenta el diagrama de bloques del Driver L293B. Como se puede ver, la señal de control EN1 (pin 1) activa o desactiva la pareja de canales formada por los drivers 1 y 2. Asimismo, la señal de control EN2 (pin 9) controla la pareja de canales formada por los drivers 3 y 4. Las salidas OUTn se asocian con las correspondientes INn (es decir: IN1→OUT1, IN2→OUT2, IN3→OUT3 e IN4→OUT4). Se muestra también un cuadro resumen del funcionamiento del Driver L293B, en base a los niveles TTL en los pines habilitadores V(ENn) y niveles TTL en los pines de entrada V(INn), los cuales se representan por H (nivel alto TTL), L (nivel bajo TTL) y Z (alta impedancia) [27].

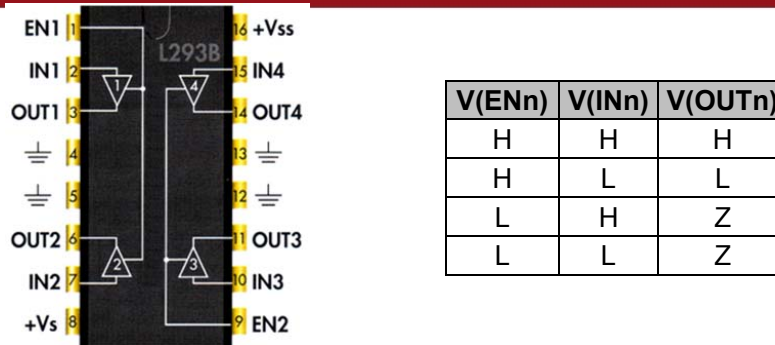


Figura 3.9 Diagrama de bloques del Driver L293B [27,28]

Entre otros datos adicionales acerca del Driver L293B, podemos mencionar que cada uno de los cuatro canales puede soportar hasta una corriente de salida pico de 2 A. Asimismo, el valor máximo de la tensión en los pines de habilitación EN1 y EN2 (pines 1 y 9, respectivamente) es de 7 V.

a. Giro del motor en doble sentido

El circuito mostrado en la Figura 3.10 permitirá controlar el giro del eje del motor DC en doble sentido. Notar que para esto únicamente haremos uso de dos de los cuatro drivers con los que cuenta el integrado L293B, motivo por el cual en la Figura 3.10 aparece la notación ‘1/2 L293B’ [27].

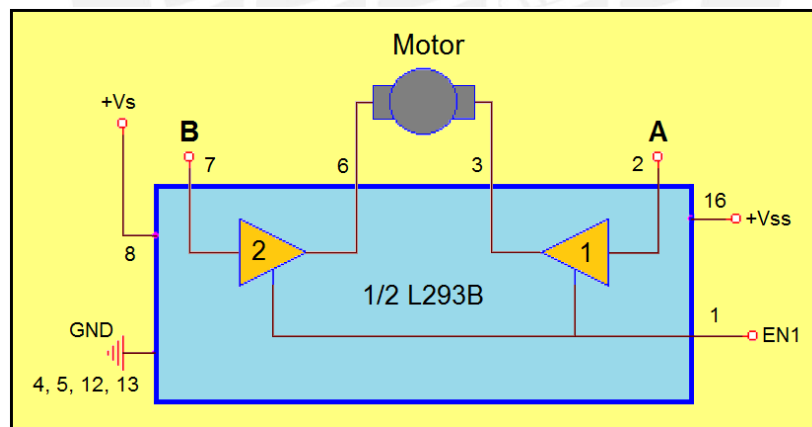


Figura 3.10 Diagrama del circuito de giro del motor DC en doble sentido [27]

En la Figura 3.10, el pin 1 (EN1) es el que habilitará el funcionamiento de los drivers 1 y 2 (de color naranja). Asimismo, los pines 2 y 7 (‘A’ y ‘B’, respectivamente) son los que definirán el sentido de giro del eje del motor, de la siguiente forma:

- Cuando la entrada A (pin 2) esté a nivel bajo y la entrada B (pin 7) a nivel alto, el eje del motor girará en un sentido (digamos, en sentido horario).
- Cuando la entrada A (pin 2) esté a nivel alto y la entrada B (pin 7) a nivel bajo, el eje del motor girará en sentido opuesto al caso anterior (en sentido anti-horario).

En la Figura 3.11 se muestra un cuadro resumen del funcionamiento del circuito, en base a los niveles TTL en el pin habilitador V(EN1) y niveles TTL en los pines de entrada V(A) y V(B), los cuales se representan por H (nivel alto TTL), L (nivel bajo TTL) y X (puede ser H o L) [27].

V(EN1)	V(A)	V(B)	Funcionamiento del motor
H	L	L	Parada rápida del motor.
H	H	H	Parada rápida del motor.
H	L	H	Giro del eje del motor en sentido horario.
H	H	L	Giro del eje del motor en sentido anti-horario.
L	X	X	Motor desconectado.

Figura 3.11 Funcionamiento del circuito de control de giro del motor DC [27]

b. Funcionamiento conjunto del bloque de control y el excitador

Para entender el funcionamiento del control de sentido de giro del eje del motor DC debemos tener en cuenta tres puntos muy importantes:

- La señal PWM emitida por el bloque de control se encargará únicamente de regular la magnitud de la velocidad de giro del eje del motor, independientemente del sentido en el cual se encuentre girando (sentido horario ó anti-horario).
- El bloque de control, por intermedio del bloque excitador, controlará el sentido de giro del motor DC. Tal como se muestra en la Figura 3.12, cuando el bloque de control ponga a nivel bajo el pin 2 (pin 'A') y a nivel alto el pin 7 (pin 'B') del Driver L293B (excitador), el eje del motor DC girará en un determinado sentido (digamos, en sentido horario). Asimismo, cuando el bloque de control ponga a nivel alto el pin 2 (pin 'A') y a nivel bajo el pin 7 (pin 'B'), el eje del motor DC girará en sentido opuesto (en sentido anti-horario).

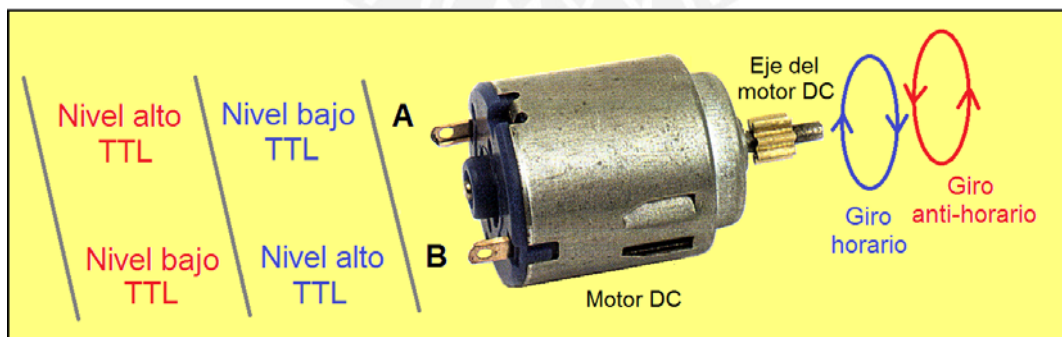


Figura 3.12 Control de giro del eje del motor DC en doble sentido [28]

- Un sensor de fin de carrera será utilizado para enviar la señal de cambio de sentido de giro al bloque de control, para que éste a su vez (por intermedio del excitador) invierta el sentido de giro del eje del motor DC.

3.5.4 Interface de comunicación

a. Puerto serie RS232

En el diseño del sistema de reconstrucción de volúmenes se eligió la comunicación vía puerto serie debido a que es la forma más común y sencilla de comunicar una computadora con cualquier otro dispositivo. El puerto serie de la computadora es compatible con el denominado estándar RS232.

Los puertos serie son accesibles mediante conectores. La norma RS232 establece dos tipos de conectores llamados DB-25 (de 25 terminales) y DB-9 (de 9 terminales), y existen conectores machos como hembras. Cabe señalar que la norma RS232 se estableció para conectar una PC con un módem, por lo que aparecen muchos terminales en los conectores DB-25 que en otro tipo de aplicaciones no se utilizan; por lo cual es mucho más común utilizar el conector tipo DB-9, que será el conector a utilizar en el diseño del sistema propuesto en el presente trabajo de tesis [29].

En la Figura 3.13 se muestra un conector DB-9 hembra, así como la numeración de sus terminales.

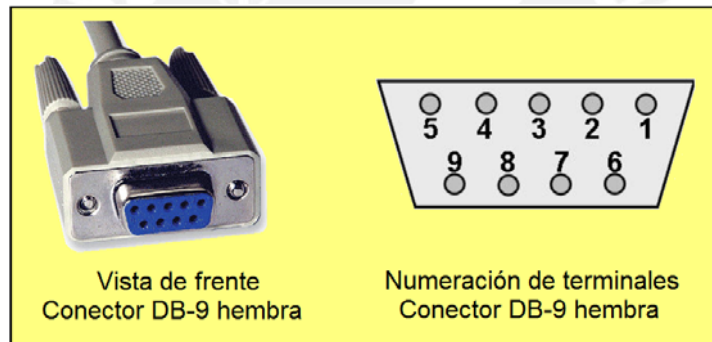


Figura 3.13 Conector hembra DB-9 y numeración de terminales [30,31]

Cada uno de los pines del conector RS232 tiene una función especificada por la norma. En la Figura 3.14 se muestra la función que cumple cada terminal de un conector DB-9.

Terminal DB9	Función
Terminal 1	DCD (Data carrier detect): Detección de portadora de datos.
Terminal 2	RxD (Received data): Recepción de datos.
Terminal 3	TxD (Transmit data): Transmisión de datos.
Terminal 4	DTR (Data terminal ready): Terminal de datos lista.
Terminal 5	GND (Ground): Nivel de tierra.
Terminal 6	DSR (Data set ready): Fijación de datos lista.
Terminal 7	RTS (Request to send): Requerimiento de envío.
Terminal 8	CTS (Clear to send): Borrar para envío.
Terminal 9	RI (Ring indicator): Indicador de llamada.

Figura 3.14 Conector hembra DB-9 y numeración de terminales [30,31]

b. Niveles lógicos RS232

Cabe señalar que la norma RS232 trabaja con lógica negativa, de la siguiente forma:

- Un voltaje entre +3V y +15V representa el nivel lógico '0'.
- Un voltaje entre -3V y -15V representa el nivel lógico '1'.

Estos son los niveles lógicos con los que trabaja el puerto serie DB-9 de la computadora (PC).

c. Niveles lógicos TTL (PIC16F84)

Cabe señalar que TTL trabaja con lógica positiva, de la siguiente forma:

- Un voltaje entre 0V y +0.8V representa el nivel lógico '0'.
- Un voltaje entre +2V y +5V representa el nivel lógico '1'.

Estos son los niveles lógicos con los que trabaja el microcontrolador PIC16F84.

d. Circuito integrado MAX232

Es importante resaltar el hecho de que (como se señaló en los dos párrafos anteriores) el microcontrolador PIC16F84 trabaja con lógica positiva TTL; a diferencia del puerto serie RS232 (DB-9) de la computadora, que trabaja con lógica negativa (norma RS232). Entonces, si lo que deseamos es la comunicación entre el microcontrolador PIC16F84 y la computadora (PC), debe haber un medio capaz de realizar la conversión de niveles lógicos TTL a RS232, y viceversa.

El circuito integrado MAX232 es usado para convertir los voltajes RS232 a voltajes TTL, y viceversa, sin requerir nada más que una fuente de +5 Vcc. Este circuito integrado contiene 2 drivers TTL → RS232 y 2 drivers RS232 → TTL. Asimismo, necesita cuatro condensadores externos de unos pocos microfaradios para generar el voltaje RS232 internamente, tal como se muestra en la Figura 3.15 [32].

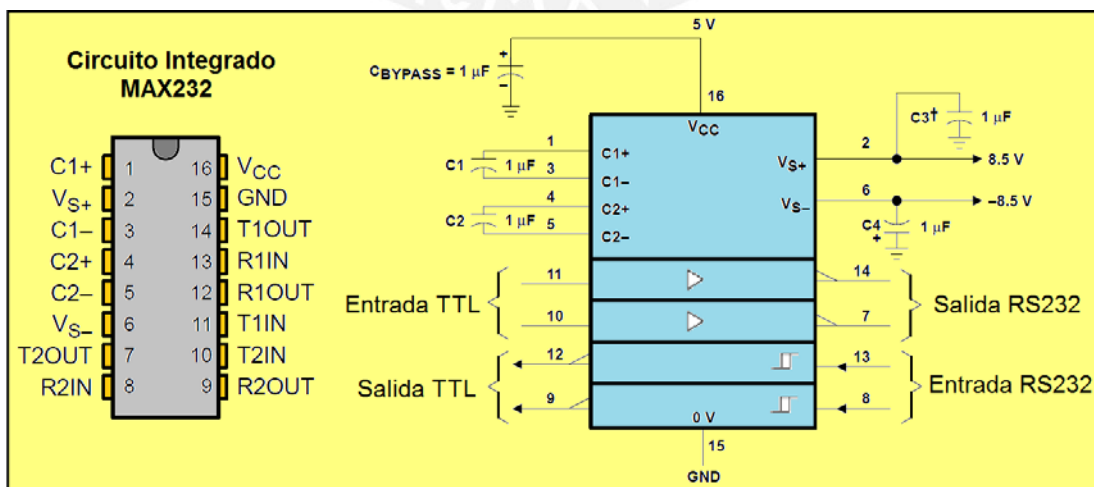


Figura 3.15 Diagrama de pines del circuito integrado MAX232 y su configuración típica [32]

e. Conexión puerto RS232 y PIC16F84

La comunicación entre la computadora (PC) y el microcontrolador PIC16F84 puede llevarse a cabo utilizando como intermediario el circuito integrado MAX232 y únicamente tres terminales del puerto serie DB-9 (puerto con el que cuenta la PC). Estos tres terminales son: pin de transmisión de datos (TxD), pin de recepción de datos (RxD) y pin de tierra (GND).

Como ya se mencionó anteriormente, la necesidad de utilizar el circuito integrado MAX232 como intermediario entre la computadora (PC) y el microcontrolador radica en que los niveles lógicos TTL que salen del microcontrolador no son compatibles con los niveles lógicos RS232 del puerto serie (DB-9) de la computadora (PC). Precisamente el circuito integrado MAX232 se encargará de traducir los niveles lógicos RS232 a niveles TTL, y viceversa. La Figura 3.16 muestra el diagrama esquemático de la conexión de una PC con el microcontrolador PIC16F84.

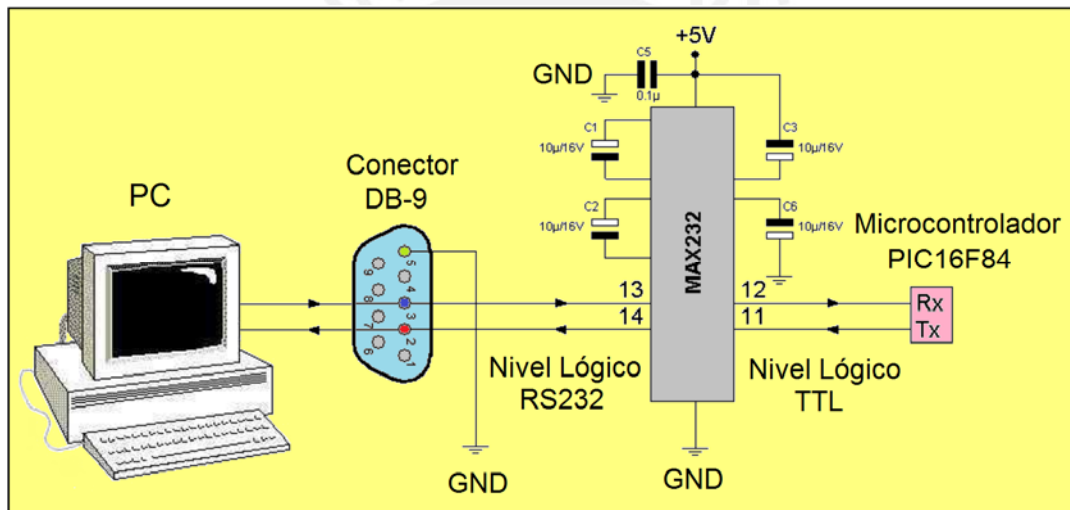


Figura 3.16 Diagrama de conexión entre la PC y el microcontrolador PIC16F84, a través del circuito integrado MAX232 [33,34]

A continuación se explica el diagrama mostrado en la Figura 3.16:

- **Transmisión de datos desde el Microcontrolador PIC16F84 hacia la PC**

Comencemos por la línea de transmisión de datos (Tx) del microcontrolador PIC16F84 (lado derecho de la Figura 3.16). Vemos que esta línea transmite datos con nivel lógico TTL e ingresa al pin 11 del circuito integrado MAX232.

Luego, el circuito integrado MAX232 realiza internamente la conversión de nivel lógico TTL de la data que ingresa por el pin 11, y arroja por el pin 14 la data ya convertida a nivel lógico RS232.

Finalmente, la data que sale del pin 14 del integrado MAX232 (a nivel lógico RS232) ingresa por el terminal 2 del conector DB-9 de la PC (terminal de color

rojo en la Figura 3.16). Recordar que, como se detalló en la Figura 3.14, el terminal 2 del conector DB-9 cumple la función de recepción de datos.

- **Transmisión de datos desde la PC hacia el Microcontrolador PIC16F84**

Comencemos ahora por el terminal 3 del conector DB-9 de la PC (terminal de color azul en la Figura 3.16). Recordar que, como se detalló en la Figura 3.14, el terminal 3 del conector DB-9 cumple la función de transmisión de datos.

Vemos que el terminal 3 del conector DB-9 transmite data con nivel lógico RS232 e ingresa al pin 13 del circuito integrado MAX232.

Luego, el circuito integrado MAX232 realiza internamente la conversión de nivel lógico RS232 de la data que ingresa por el pin 13, y arroja por el pin 12 la data ya convertida a nivel lógico TTL.

Finalmente, la data que sale del pin 12 del integrado MAX232 (a nivel lógico TTL) ingresa por la línea de recepción de datos (Rx) del microcontrolador PIC16F84.

3.6 Parte software:

Con respecto a la parte software del sistema de reconstrucción de volúmenes, ésta corresponde a dos bloques: Control y PC (ver Figura 3.1).

3.6.1 Control

a. Diagrama de flujo: Medición de distancias

El diagrama de flujo mostrado en la Figura 3.17 describe el funcionamiento del sensor ultrasonido SRF04 como medidor de distancias, tal como se explica a continuación paso a paso:

- Primero, el bloque de control envía al sensor ultrasonido SRF04 un pulso de disparo (de una duración mínima de 10 μ s).
- El pulso de disparo será la señal para que el sensor SRF04 efectúe una toma de datos. Así, al recibir el pulso de disparo, el sensor SRF04 emitirá una señal ultrasonido en dirección al objeto cuya distancia al sensor se desea medir.
- La señal ultrasonido rebota en la superficie del objeto y regresa en forma de eco hacia el receptor del sensor ultrasonido SRF04, en donde se espera a la llegada del pulso de eco (este proceso de espera es graficado como un bucle de espera en la Figura 3.17).
- Si la duración del pulso de eco es mayor a 900 μ s, quiere decir que la distancia medida es mayor a 15 cm (la longitud del radio definido para la plataforma circular), y por lo tanto no hay tronco de madera atravesando perpendicularmente la plataforma circular (a lo largo de su eje axial).

- Si la duración del pulso de eco es menor o igual a 900 μ s, entonces se aprovecha el hecho de que 60 μ s de pulso de eco equivalen a 1 cm de distancia medida. Por esta razón se divide la duración del pulso de eco (en μ s) entre 60, a fin de obtener el número de centímetros medidos por el sensor ultrasonido SRF04.

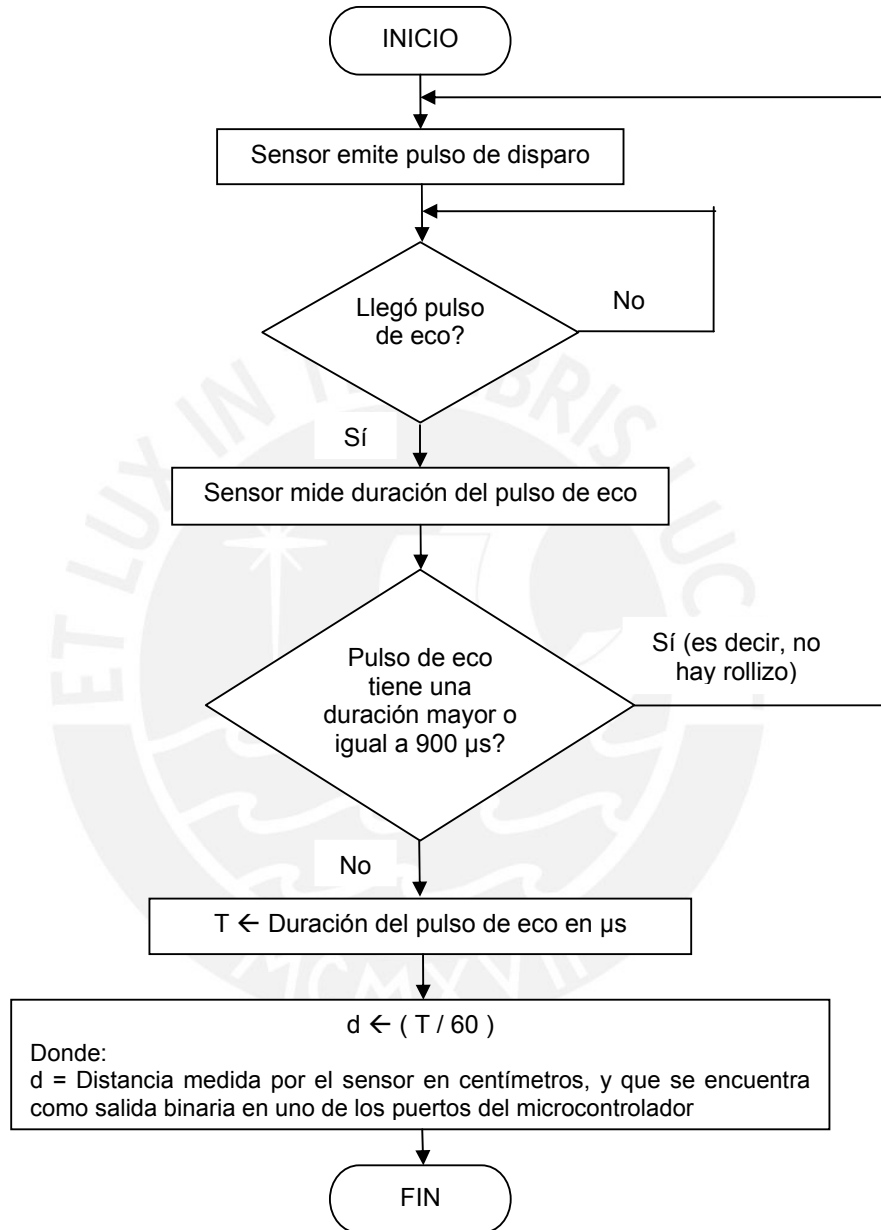


Figura 3.17 Diagrama de flujo: Medición de distancias

b. Diagrama de flujo: Regulación de velocidad del motor

El diagrama de flujo mostrado en la Figura 3.18 describe la regulación de la velocidad del eje del motor DC, tal como se explica a continuación paso a paso:

- Utilizando el encoder, el microcontrolador sensorá constantemente la velocidad ‘actual’ del eje del motor DC.

- Si la velocidad sensada 'actual' es mayor que la requerida, el microcontrolador enviará la orden para disminuir la velocidad (disminuyendo el ciclo de trabajo de la señal PWM).
- Si la velocidad sensada 'actual' es menor que la requerida, el microcontrolador enviará la orden para aumentar la velocidad (aumentando el ciclo de trabajo de la señal PWM).
- Este proceso se repetirá en un bucle infinito, a fin de corregir cualquier desviación de la velocidad requerida en el eje del motor DC.

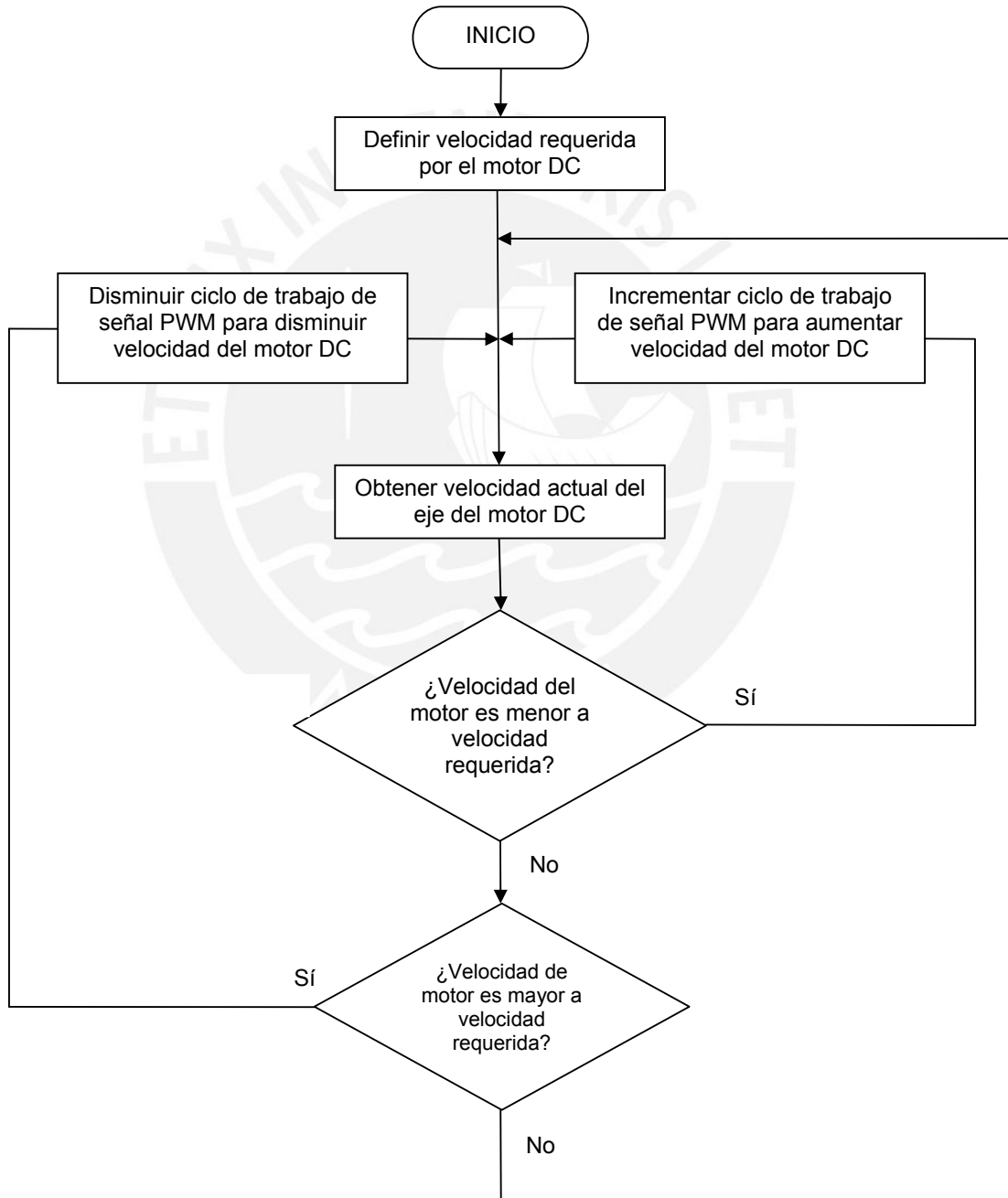


Figura 3.18 Diagrama de flujo: Regulación de velocidad del motor

c. Diagrama de flujo: Cambio de sentido de giro del motor

El diagrama de flujo mostrado en la Figura 3.19 muestra de forma simple cómo el sentido de giro del eje del motor DC será controlado por el estado 'cerrado' o 'abierto' de un interruptor. De esta forma, cuando el interruptor esté cerrado el eje del motor girará en un sentido determinado (digamos, en sentido horario); y cuando el interruptor esté abierto el eje del motor girará en sentido opuesto al caso anterior (es decir, en sentido anti-horario).

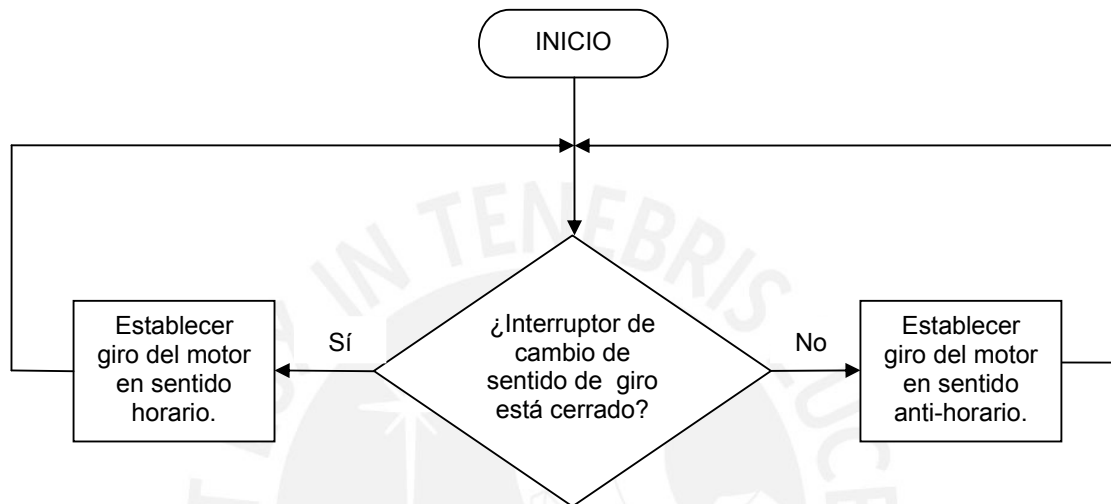


Figura 3.19 Diagrama de flujo: Cambio de sentido de giro del eje del motor

3.6.2 PC

a. Cálculos y definiciones de parámetros

Llegado a este punto, se ha descrito en detalle el diseño y funcionamiento del sistema de reconstrucción de volúmenes. Ahora pasaremos a mostrar la definición de los parámetros y los cálculos efectuados como parte del diseño del sistema.

- **Parámetros:**

Los parámetros que se definieron en el diseño del sistema de reconstrucción de volúmenes son los siguientes:

- ✓ **Dimensiones máximas de los troncos a procesar:**

Diámetro máximo de la sección axial del tronco a procesar: 20cm

Longitud máxima del tronco a procesar: 25cm

- ✓ **Diámetro de la plataforma circular:** 30cm

Es decir, en la Figura 3.2, el radio interno de la plataforma circular es de 15cm

- ✓ **Velocidad de desplazamiento de los sensores sobre la plataforma circular:** 1m/s

- ✓ **Velocidad de traslación de los troncos:** 1cm/s

• **Cálculos:**

✓ **Cálculo de la longitud total de la plataforma circular (L):**

Es importante realizar este cálculo, pues nos permitirá calcular el recorrido de cada sensor sobre la plataforma circular (ver Figura 2.6). Dado que sabemos que el radio interno de la plataforma circular es de 15 cm, procedemos a realizar el cálculo: $L = 2 \cdot \pi \cdot \text{radio} = 2 \cdot \pi \cdot 15\text{cm} = 94.2 \text{ cm} = 0.942 \text{ metros}$

Tener en cuenta que cada sensor recorrerá sólo media vuelta de la plataforma circular, por lo que cada sensor recorrerá en realidad: $0.5 \cdot L = 0.471 \text{ metros}$

✓ **Cálculo del periodo de muestreo de los sensores (T):**

En la Figura 3.20 se puede apreciar el diagrama de tiempos del sensor SRF04. Vemos que el tiempo de toma de una sola muestra (u obtención de una distancia por el sensor) es el tiempo comprendido entre dos pulsos de disparo consecutivos. Como los pulsos de disparo tendrán una duración constante (10 μs), el período de muestreo estará determinado por la duración del pulso de eco (el cual es variable y depende de la longitud medida) y el retardo que hay entre el fin de un pulso de eco y el inicio del siguiente pulso de disparo (que debe tener una duración mínima de 10 ms).

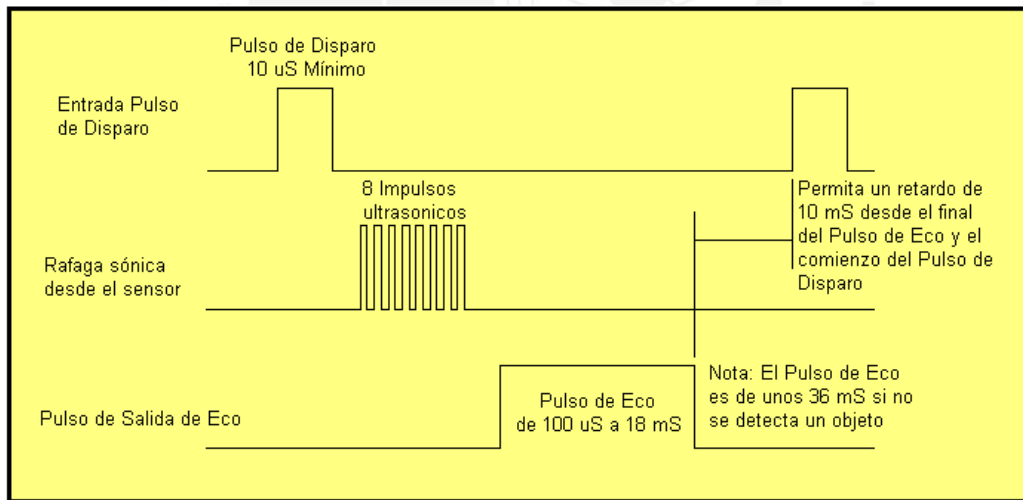


Figura 3.20 Diagrama de tiempos del sensor SRF04 [13]

Sea 'T' el período de muestreo del sensor SRF04: $T = [T_{\text{mín}}; T_{\text{máx}}]$

$T_{\text{máx}} = (T_{\text{disparo}} + T_{\text{eco máx}} + T_{\text{retardo}}) = (10 \mu\text{s} + 18 \text{ ms} + T_{\text{retardo}})$

Dado que $T_{\text{retardo}} \geq 10 \text{ ms}$, re-escribimos la ecuación:

$T_{\text{máx}} \geq (10 \mu\text{s} + 18 \text{ ms} + 10 \text{ ms})$

$T_{\text{máx}} \geq 28.01 \text{ ms}$

Definimos $T_{\text{máx}} = 30 \text{ ms}$, para cuando el sensor mide la máxima distancia.

$$T_{\text{mín}} = (T_{\text{disparo}} + T_{\text{eco mín}} + T_{\text{retardo}}) = (10 \mu\text{s} + 100 \mu\text{s} + T_{\text{retardo}})$$

Dado que $T_{\text{retardo}} \geq 10 \text{ ms}$, re-escribimos la ecuación:

$$T_{\text{mín}} \geq (10 \mu\text{s} + 100 \mu\text{s} + 10 \text{ ms})$$

$$T_{\text{mín}} \geq 10.11 \text{ ms}$$

Definimos $T_{\text{mín}} = 12 \text{ ms}$, para cuando el sensor mide la mínima distancia.

Finalmente, teniendo $T_{\text{mín}}$ y $T_{\text{máx}}$, definimos el rango del periodo de muestreo del sensor ultrasonido: $T = [12\text{ms}; 30\text{ms}]$

✓ **Cálculo del número de muestras por unidad de longitud (N/Long):**

Dado que la velocidad de desplazamiento de los sensores sobre la plataforma circular es de 1m/s y la longitud total de la plataforma circular es de $L = 0.942$ metros, entonces el tiempo 't' que demorará un sensor en recorrer media vuelta de la plataforma circular será:

$$t = (0.5 \cdot L / \text{velocidad}) = (0.5) \cdot (0.942 \text{ m}) / (1 \text{ m/s}) = 0.471 \text{ segundos}$$

Luego:

➤ **Para $T_{\text{máx}} = 30 \text{ ms}$**

En media vuelta ($t = 0.471\text{s}$), el sensor habrá tomado 'N' muestras:

$$N = (\text{Tiempo total de recorrido} / \text{Periodo de muestreo})$$

$$N = (0.471 \text{ s} / 30 \text{ ms}) \approx 16 \text{ muestras}$$

$$N/\text{Long} = N / (0.5 \cdot L) = (16 \text{ muestras}) / (47.1 \text{ cm}) \approx 0.34 \text{ muestras} / \text{cm}$$

➤ **Para $T_{\text{mín}} = 12 \text{ ms}$**

En media vuelta ($t = 0.471\text{s}$), el sensor habrá tomado 'N' muestras:

$$N = (\text{Tiempo total de recorrido} / \text{Periodo de muestreo})$$

$$N = (0.471 \text{ s} / 12 \text{ ms}) \approx 39 \text{ muestras}$$

$$N/\text{Long} = N / (0.5 \cdot L) = (39 \text{ muestras}) / (47.1 \text{ cm}) \approx 0.83 \text{ muestras} / \text{cm}$$

Considerando que $T_{\text{mín}}$ y $T_{\text{máx}}$ representan casos extremos, tener entre 0.34 y 0.83 muestras por cada centímetro (del contorno de una sección axial del tronco de madera) representaría una resolución bastante aceptable al momento de efectuar la reconstrucción volumétrica.

b. Diagrama de flujo principal: Reconstrucción volumétrica del rollizo

En la Figura 3.21 se muestra el diagrama de flujo principal de la reconstrucción volumétrica del rollizo. Dado que los troncos atravesarán la plataforma circular a lo largo de su eje axial (de forma muy similar a como funciona un tomógrafo), se hace el

cambio a sistema de coordenadas cilíndricas, con el eje 'z' como el eje axial de la plataforma circular. Luego, como la velocidad de traslación de los troncos será constante, se puede conocer el valor de 'z' en cualquier momento (partiendo de una referencia). Posteriormente se obtienen las coordenadas cilíndricas (x; y; z) que forman los anillos (para varios valores de 'z' conocidos). Finalmente, se acomodan todos los anillos obtenidos, lo cual reproducirá la forma volumétrica del rollizo.

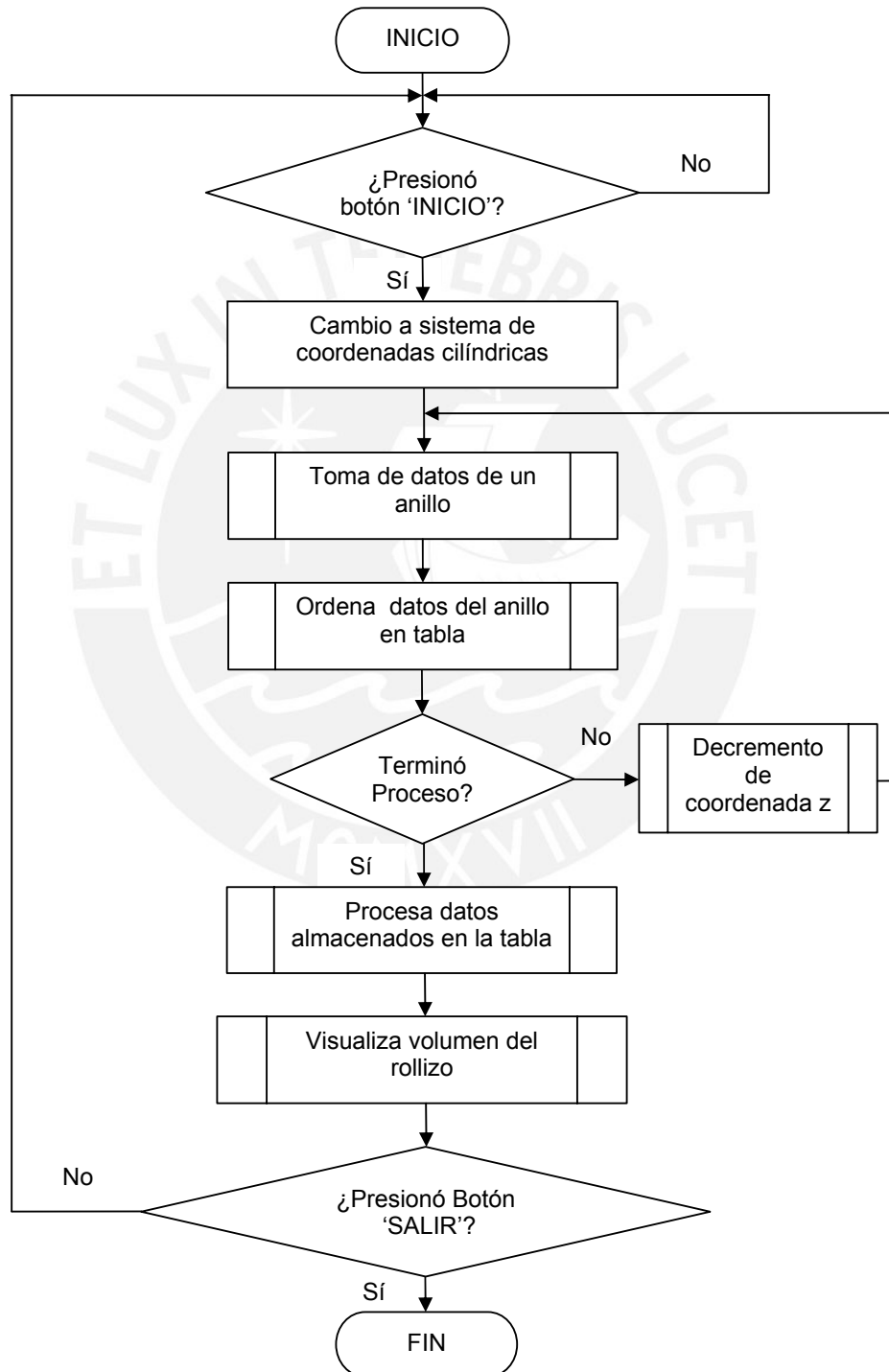


Figura 3.21 Diagrama de flujo principal: Reconstrucción volumétrica del rollizo

c. Diagramas de flujo de las subrutinas

- Diagrama de flujo de la subrutina: Toma de datos de un anillo

En la Figura 3.22 se muestra el diagrama de flujo que explica la toma de datos de un anillo. Para empezar, sabemos que el sensor SRF04 nos dará la distancia 'd' (en cm) desde la posición del sensor hasta un determinado punto de la superficie del rollizo. Luego, como el radio de la plataforma circular es de 15 cm, la distancia del centro de la plataforma circular al punto medido de la superficie del rollizo estará dada por la diferencia: (15-d) cm.

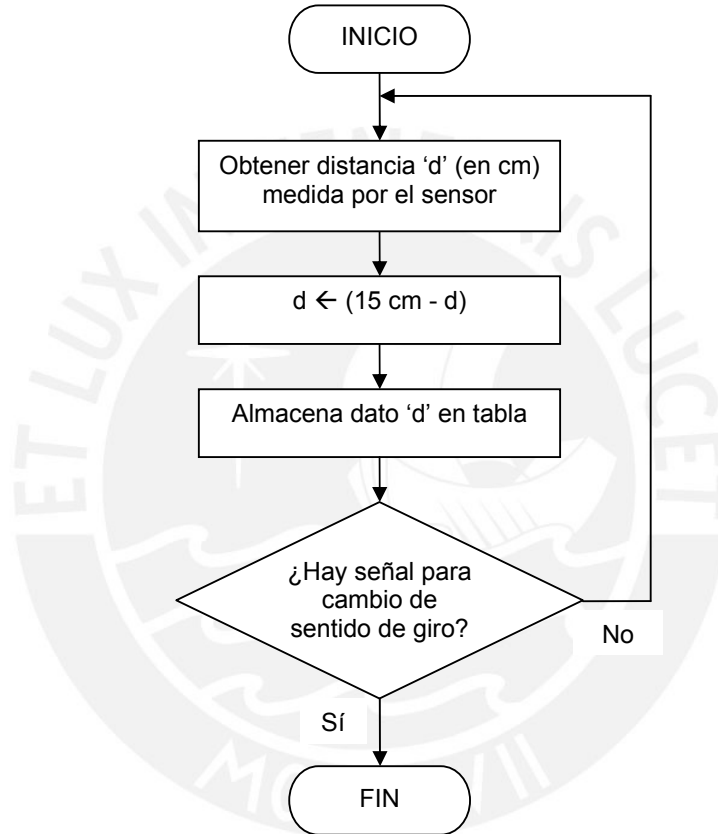


Figura 3.22 Diagrama de flujo: Toma de datos de un anillo

- Diagrama de flujo de la subrutina: Decremento de coordenada z

En la Figura 3.23 se muestra (asumiendo coordenadas cilíndricas) la variación de la coordenada 'z' cada vez que el par de sensores ultrasonido ha efectuado la toma de datos correspondientes a un anillo completo (tener en cuenta que el eje 'z' será el eje axial de la plataforma circular).

Como cada uno de los dos sensores ultrasonido SRF04 (ubicados en extremos diametralmente opuestos de la plataforma circular) se desplaza sólo 180 grados

sexagesimales para tomar la información de un anillo completo, cada sensor recorrerá únicamente la mitad de la plataforma circular (en un tiempo $t = 0.471s$). Dado que se definió como parámetro que la velocidad de traslación de los rollizos será de 1 cm/s, el espacio recorrido por el rollizo (en el eje 'z') luego de la toma de datos de un anillo completo estará dado por: $z = (1 \text{ cm/s}) \cdot (0.471 \text{ s}) = 0.471 \text{ cm}$. Este valor de 0.471 cm representa precisamente la variación de la coordenada 'z' mostrada en el diagrama de flujo de la Figura 3.23.

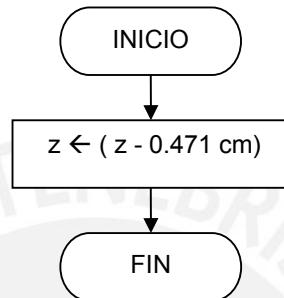


Figura 3.23 Diagrama de flujo: Decremento de coordenada z

- **Diagrama de flujo de la subrutina: Procesa datos almacenados en tabla**

Dado que la última coordenada 'z' será la primera en ser procesada y; de la misma forma, la primera coordenada 'z' será la última en ser procesada, en la tabla de almacenamiento resulta necesario invertir el orden de las coordenadas 'z' asignadas a cada anillo, tal como se describe en el diagrama de flujo de la Figura 3.24.

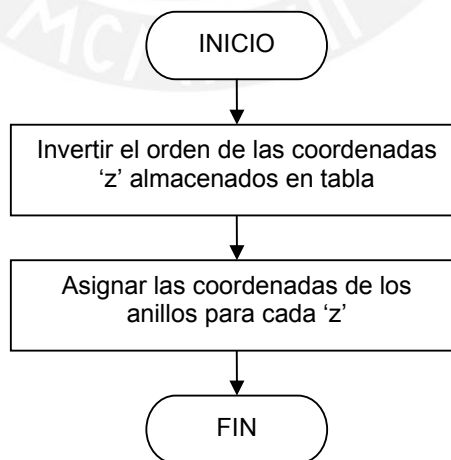


Figura 3.24 Diagrama de flujo: Procesa datos de tabla

- **Diagrama de flujo de la subrutina: Visualiza volumen del rollizo**

En la Figura 3.25 se muestra de forma simplificada el diagrama de flujo de la visualización volumétrica del tronco de madera o rollizo.

El primer paso consiste en leer los datos correspondientes a un solo anillo, los cuales permitirán graficar la poligonal que reproduce su forma. Luego, al haber terminado la toma de datos del anillo, se decrementa la coordenada 'z' y se verifica si hay más anillos por procesar (este proceso de verificación se representa como un bucle en la Figura 3.25).

El proceso descrito en el párrafo anterior se repite hasta que ya no existan más anillos por procesar. Finalmente, se dibuja la totalidad de la forma volumétrica del rollizo como un arreglo del conjunto de anillos obtenidos individualmente (colocándolos uno a continuación del otro en un espacio tridimensional).

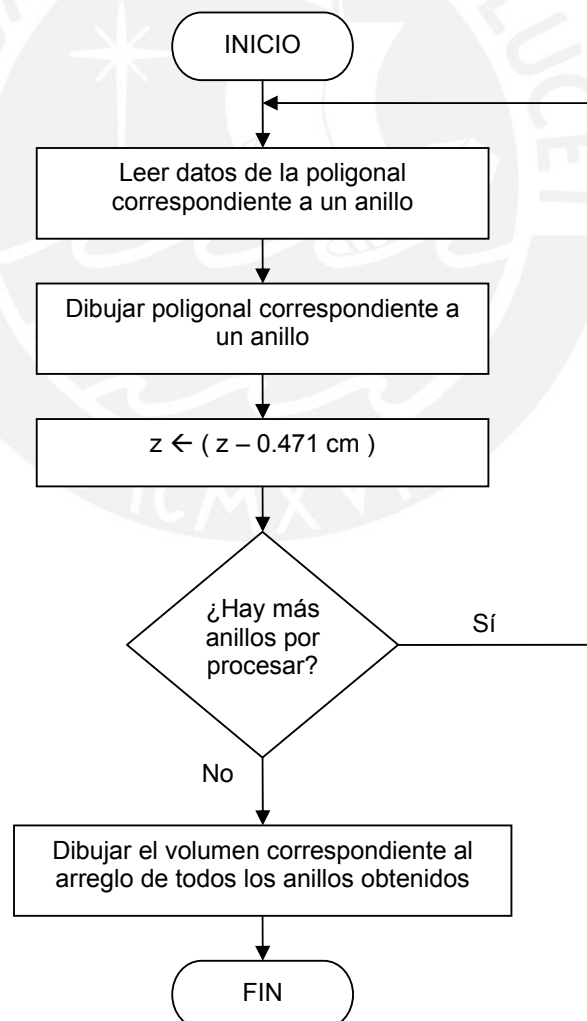


Figura 3.25 Diagrama de flujo: Visualiza volumen del rollizo

d. Algoritmo base para el gráfico volumétrico del rollizo

Para obtener el algoritmo que logrará la representación volumétrica de los troncos de madera (rollizos), partiremos del siguiente razonamiento:

- Si deseamos graficar una línea con una forma determinada (no necesariamente recta), ésta se puede obtener a partir de una sucesión de puntos (cada uno de los cuales es graficado en base a su ubicación en un eje de coordenadas).
- Si lo que deseamos graficar es un plano, éste se puede obtener a partir de una sucesión de rectas paralelas colocadas una a continuación de la otra.
- Si lo que queremos graficar es un cilindro recto de radio 'R', éste se puede obtener a partir de una sucesión de circunferencias de radio 'R' colocadas una a continuación de la otra.

Siguiendo este mismo razonamiento, si lo que deseamos graficar es la superficie de un tronco de madera, podemos suponer que ésta es una superficie cilíndrica (aunque ya no una superficie cilíndrica regular, obviamente). Para este caso en particular, ya no se tratarían de circunferencias colocadas una a continuación de otra, sino de una especie de anillos de forma irregular colocados uno a continuación del otro.

En resumen, para poder reconstruir el volumen del tronco de madera, el proceso consistirá en generar una especie de malla volumétrica tridimensional a partir del conjunto de anillos irregulares obtenidos de forma individual.

Para empezar, lo primero es definir cómo graficar uno de dichos anillos que, colocados uno a continuación de otro, conformarían la superficie rolliza. Si dichos anillos fueran circunferencias de radio R, sería bastante simple obtener el gráfico. Sin embargo, dado que lo que deseamos graficar es una especie de anillo de forma irregular, podemos asumir (sin pérdida de generalidad) que dicho anillo tiene la forma de una elipse.

Ahora, ubiquemos un punto de referencia dentro de la región encerrada por dicha elipse y tracemos los rayos que unen dicho punto de referencia con cada punto de la elipse. Obtendríamos un gráfico similar a lo que se muestra en la Figura 3.26.

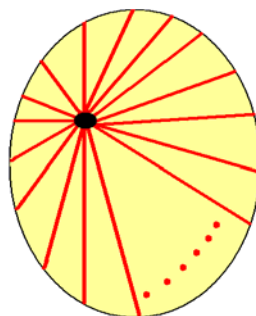


Figura 3.26 Representación de un anillo irregular como una elipse

Si conociéramos a priori las coordenadas del punto de referencia, suponiendo además que el ángulo formado por cualquier par de rayos consecutivos es el mismo y que la longitud de cada uno de los rayos es también conocido, sería posible obtener las coordenadas de los puntos de intersección de cada rayo con la superficie de la elipse (la misma que deseamos reconstruir). Al unir estos puntos de intersección se obtendría una poligonal como la que se muestra de color azul en la Figura 3.27.

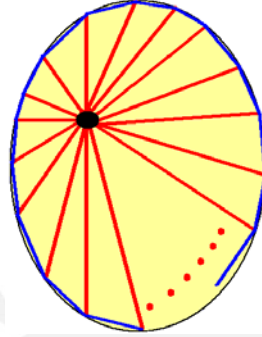


Figura 3.27 Representación de la poligonal para reconstruir una elipse

Como se puede observar en la Figura 3.27, la poligonal resultante (línea azul) tiende a seguir la forma de la elipse. Ahora, si existiera una gran cantidad de rayos, la poligonal prácticamente tendría la forma de la elipse. Más aún, si no se tratara de una elipse, sino de una forma irregular (y si existiese la cantidad de rayos suficientes), el efecto sería el mismo: la poligonal seguiría la forma real del contorno, sección axial o anillo que se desea reconstruir.

Finalmente, colocando estos anillos irregulares uno a continuación del otro en un espacio tridimensional, lograríamos reconstruir la superficie rolliza, tal como se representa en la Figura 3.28 [35,36,37,38].

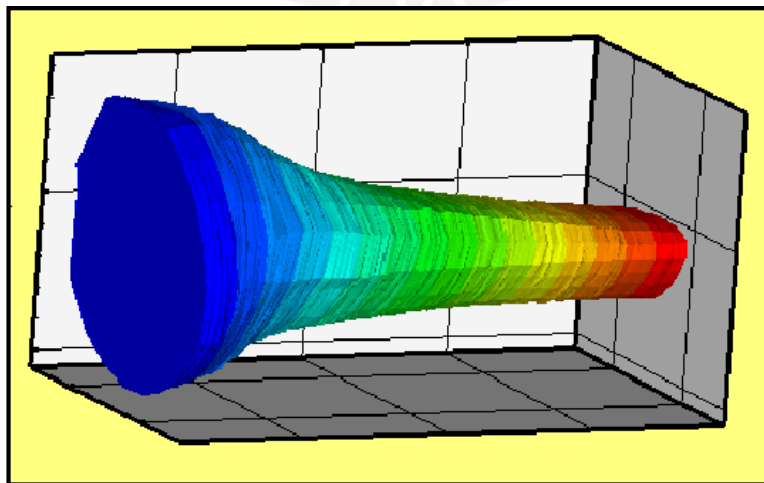


Figura 3.28 Reconstrucción de la superficie rolliza en base a un arreglo de anillos irregulares en un espacio tridimensional [38]

Cabe recordar que en este algoritmo se hicieron tres suposiciones muy importantes:

- El conocimiento a priori la ubicación del punto de referencia desde el cual parten los rayos.
- El conocimiento a priori la longitud de cada uno de los rayos.
- El ángulo de separación entre dos rayos consecutivos es constante.

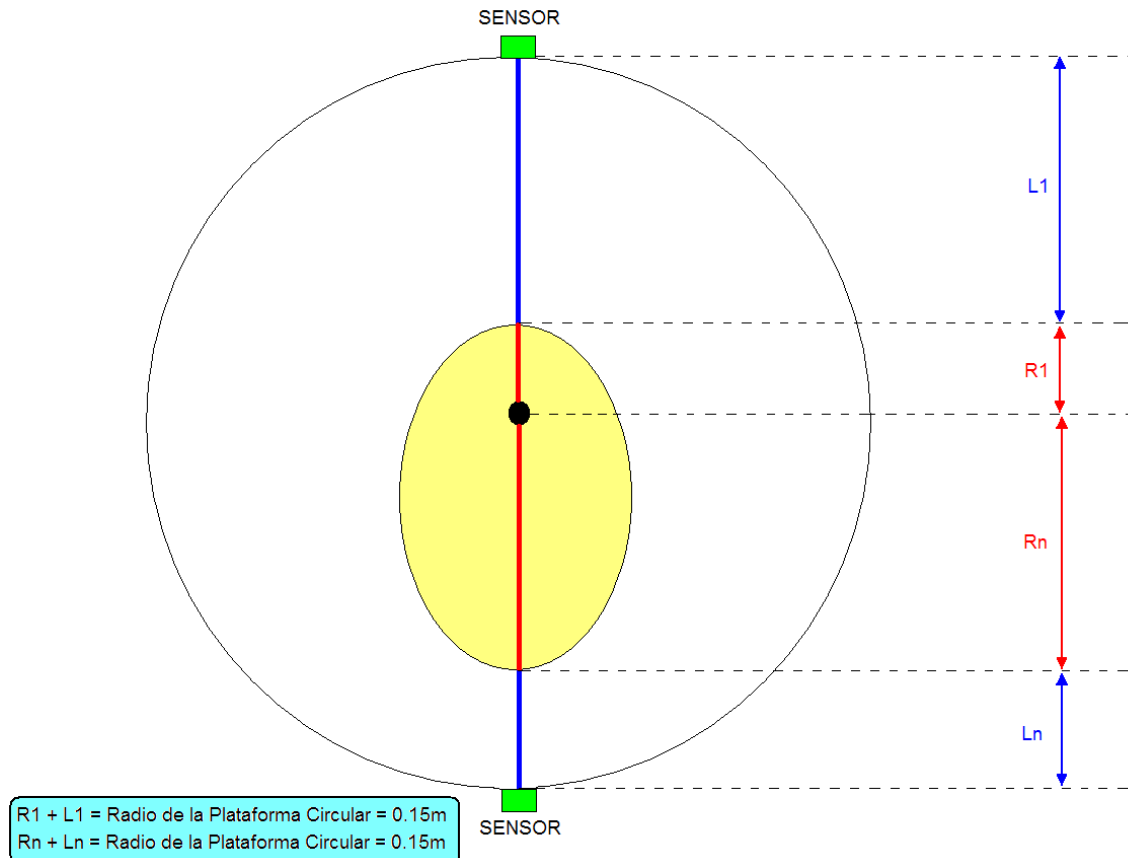


Figura 3.29 Vista frontal del sistema de reconstrucción de volúmenes

La Figura 3.29 representa la vista frontal de la plataforma circular mostrada en la Figura 3.2. En la Figura 3.29 los sensores aparecen de color verde en los extremos superior e inferior; y una sección axial de un tronco de madera aparece de color amarillo. Para facilitar el proceso de reconstrucción de volúmenes del tronco de madera se han tenido en cuenta las siguientes definiciones:

- El punto de referencia coincidirá con el centro de la plataforma circular.
- La longitud de cada uno de los rayos que salen del punto de referencia hacia cada punto de la superficie del tronco de madera (líneas de color rojo) también puede ser calculada. Como sabemos, el sensor mide la distancia hacia cada punto de la superficie rolliza (línea de color azul que sale del sensor) y dado que conocemos a priori la longitud del radio de la plataforma circular (15 cm),

mediante una simple diferencia podemos obtener la longitud de cada rayo que sale del punto de referencia hacia cada punto de un anillo de la superficie rolliza.

De esta forma:

$$R1 = 15 - L1 \quad (\text{donde } R1 \text{ y } L1 \text{ están expresados en cm})$$

Generalizando:

$$Rn = 15 - Ln \quad (\text{donde } Rn \text{ y } Ln \text{ están expresados en cm})$$

- El ángulo de separación entre los rayos (ver Figura 3.27), y que se consideró constante, estará determinado por el período de muestreo de los sensores, el cual es también constante.

Luego, conociendo la ubicación del punto de referencia (centro de la plataforma circular), así como el periodo de muestreo y las longitudes de cada uno de los rayos (calculados mediante el uso de los sensores ultrasonido como medidores de distancia), podremos obtener las coordenadas cilíndricas de los puntos de muestra sobre la superficie rolliza. Finalmente, el volumen del tronco de madera o rollizo podrá ser reconstruido mediante el algoritmo descrito anteriormente.

3.6.3 Software de visualización

a. Requisitos mínimos

El software de visualización a utilizar debe cumplir con los siguientes requisitos mínimos:

- El software debe ser fácil de aprender y usar.
- El software debe facilitar la creación de gráficos y hacer un uso adecuado de los recursos de memoria de la computadora.

b. Opciones

En función de los requisitos mínimos que debe cumplir el software a usar, se consideraron dos opciones:

- **Visual Basic:** Se consideró este software debido a su fácil uso y aprendizaje.
- **MatLab:** Se consideró este software debido a su amplia variedad de aplicaciones y la experiencia en su uso por parte del autor del presente trabajo.

c. Criterios de selección

Para el desarrollo del presente trabajo de tesis se eligió el lenguaje Visual Basic como software de visualización en base a los siguientes criterios:

- **Facilidad de aprendizaje:** El lenguaje Visual Basic es un lenguaje fácil de aprender y permite crear aplicaciones en muy poco tiempo, en comparación con otros lenguajes.
- **Facilita la creación de gráficos:** El lenguaje Visual Basic permite una fácil y rápida creación de gráficos en dos y tres dimensiones. Adicionalmente, a diferencia de MatLab, Visual Basic no hace un uso tan intensivo de memoria de la PC, lo cual facilitará el proceso de desarrollo y reproducción de gráficos.

d. Breve descripción del software de visualización elegido

Visual Basic es uno de los lenguajes más populares a nivel global. La palabra 'visual' hace referencia a la interfaz gráfica del usuario desarrollador. Así, a diferencia de otros lenguajes 'textuales', en lugar de escribir un gran número de líneas de código para implementar un sistema, en Visual Basic se utiliza el ratón para arrastrar bloques predefinidos dentro un formulario.

Una vez lanzado el programa se puede apreciar una ventana similar a la que se muestra en la Figura 3.30.

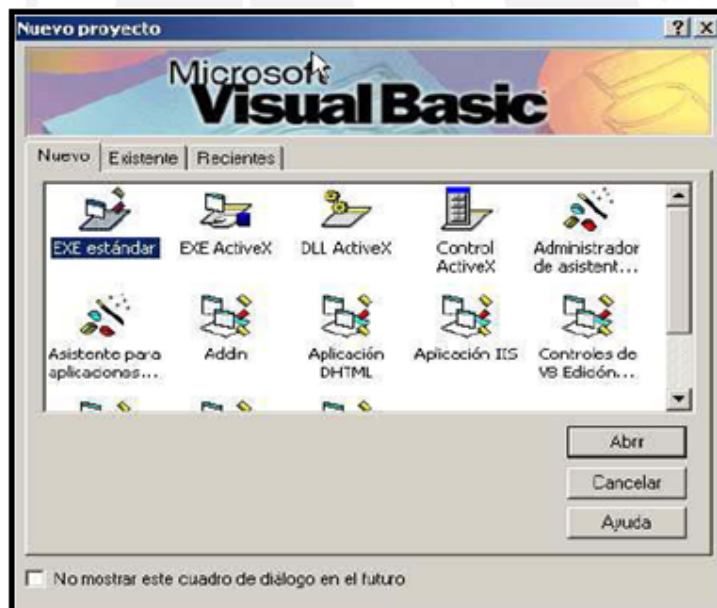


Figura 3.30 Ventana de nuevo proyecto en Visual Basic

En la Figura 3.30 se puede apreciar cómo la ventana posee tres pestañas que hacen referencia al proyecto próximo a abrir: Nuevo, Existente y Reciente. Elegimos la pestaña 'Nuevo', seleccionamos el ícono 'EXE estándar' (el cual se encuentra seleccionado por defecto) y presionamos el botón 'Abrir'. Luego, se nos presentará una ventana como la mostrada en la Figura 3.31.

La Figura 3.31 muestra la interfaz con la que se interactúa al empezar a trabajar en un nuevo proyecto. Ver que la caja de herramientas (lado izquierdo de la Figura 3.31) contiene un conjunto de controles que serán de gran utilidad para el desarrollo de nuestro programa.

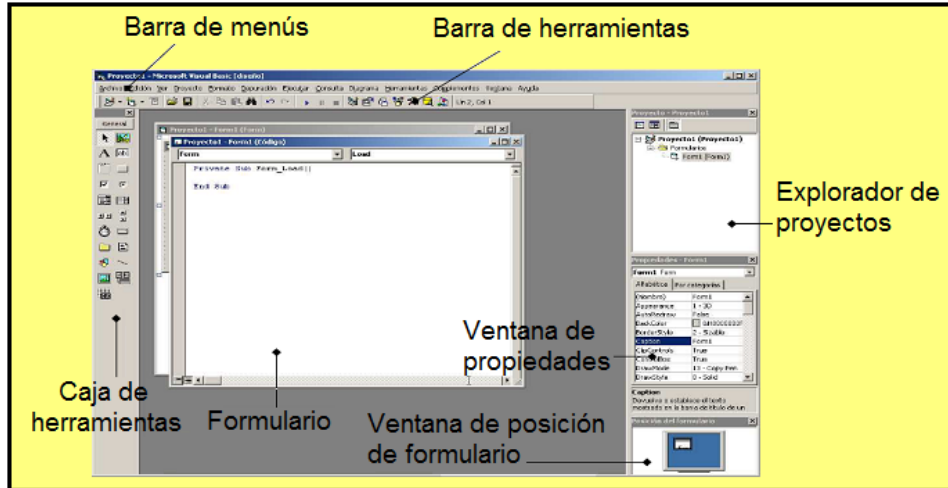


Figura 3.31 Ventana de nuevo formulario en Visual Basic

Asimismo, en la Figura 3.32 se ofrece una breve descripción de la función que cumple cada control en la caja de herramientas del formulario en Visual Basic.



Figura 3.32 Caja de herramientas de Visual Basic

CAPÍTULO 4:
SIMULACIÓN DE LOS MÓDULOS COMPONENTES DEL SISTEMA DE
RECONSTRUCCIÓN DE VOLÚMENES



4.1 Breve explicación del diseño

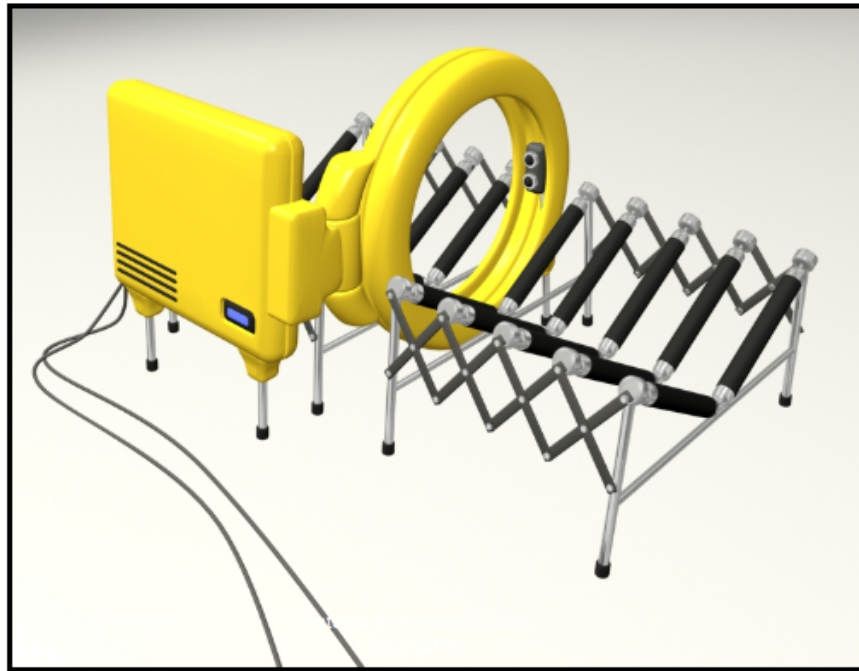


Figura 4.1 Diseño exterior del sistema propuesto [18]

Antes de pasar a mostrar las simulaciones de los módulos que componen el sistema de reproducción de volúmenes, resulta conveniente dar un breve repaso del funcionamiento del sistema:

- El sistema de reconstrucción de volúmenes propuesto en el presente trabajo de tesis consta básicamente de una plataforma circular en la que irán montados dos sensores ultrasonido (ubicados en extremos diametralmente opuestos), los mismos que se desplazarán a velocidad constante sobre la plataforma circular por la acción de un motor DC (ver Figura 4.1).
- Luego, los troncos de madera o 'rollizos' (cuya forma se desea reconstruir) atravesarán perpendicularmente la plataforma circular (moviéndose a lo largo su eje axial, funcionando de forma muy similar a como opera un tomógrafo). Eventualmente, los troncos de madera o rollizos pasarían movidos por el sistema de rodillos en forma de 'V' en la Figura 4.1.
- A medida que los sensores se desplazan sobre la plataforma circular, estos efectuarán la toma de datos correspondiente a las distancias entre dichos sensores y el tronco de madera que está atravesando la plataforma circular.
- Luego, las medidas tomadas por los sensores serán transmitidas a una PC por el puerto serie, para luego ser procesadas por un programa desarrollado en Visual Basic, el cual finalmente se encargará de generar la forma volumétrica del tronco de madera o rollizo y presentarlo en la pantalla de la computadora (PC).

4.2 Software de simulación

El software usado para la presentación de las simulaciones en el presente trabajo de tesis se denomina **Proteus**. Este software se compone de tres módulos básicos:

- **ISIS**: Del inglés: Intelligent Schematic Input System, o módulo para la captura de esquemas.
- **VSM**: Del inglés: Virtual System Modelling, o módulo de simulación.
- **ARES**: Del inglés Advanced Routing Modelling, o módulo para la realización de circuitos impresos.

El módulo ISIS Professional nos permite realizar el dibujo de un circuito para ser posteriormente simulado. Una vez lanzado el programa, la primera ventana que se nos presentará es la que se muestra en la Figura 4.2:

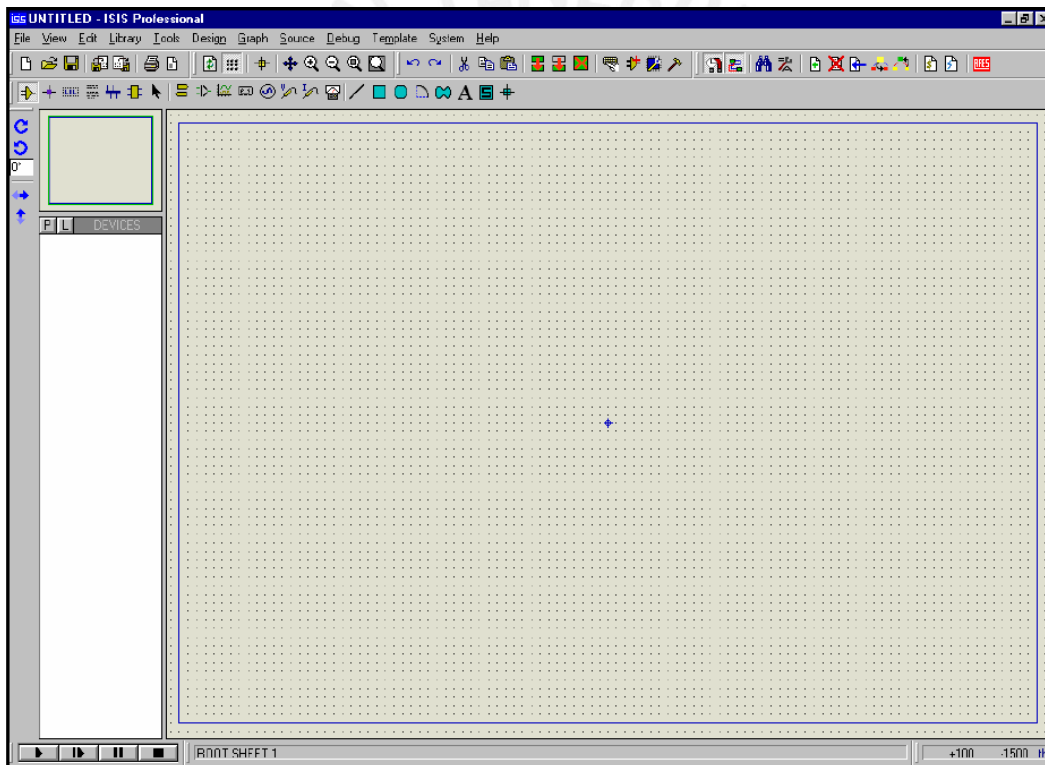



Figura 4.2 Ventana inicial del módulo ISIS Professional

Como se puede apreciar en la Figura 4.2, este software cuenta con muchas barras de herramientas que facilitan tanto el diseño como la simulación de circuitos. Además, este software incluye una extensa colección de librerías con los componentes comúnmente usados.

Utilizando las barras de herramientas y las librerías de componentes, se procede a dibujar el circuito a simular. El uso de ISIS Professional es bastante intuitivo, ya que sólo debemos abrir las librerías, buscar los componentes que deseamos dibujar y

arrastrarlos hacia la zona de trabajo. Luego, con la barra de herramientas, procedemos a efectuar las conexiones y completar lo que falte a nuestro diseño. Así, por ejemplo, obtendremos un circuito como el que se muestra en la Figura 4.3.

Una vez que tenemos el diseño terminado, procedemos a simularlo en la pantalla. Para esto, presionamos el botón ‘Play’ con el símbolo:  que se encuentra en la parte inferior izquierda de la ventana.

Finalmente, es importante señalar que para el caso en el que el circuito involucre el uso de un microcontrolador, Proteus permite editar el archivo fuente, ensamblarlo y simularlo en pantalla para ver como funcionaría el circuito en conjunto. Es precisamente esta última característica del software Proteus la que será de gran utilidad para las simulaciones en el presente trabajo de tesis.

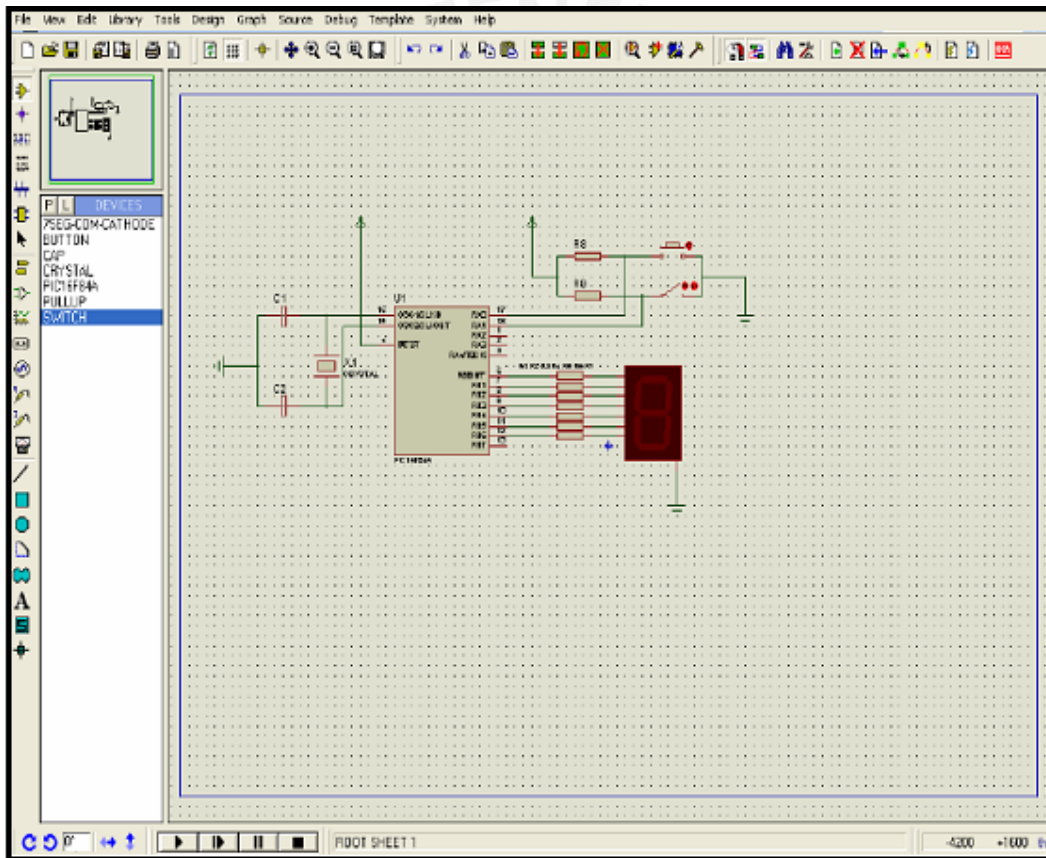


Figura 4.3 Diseño de un circuito en ISIS Professional

4.3 Simulación de la regulación de velocidad de giro del motor DC por PWM

Según el diseño propuesto para el sistema de reconstrucción de volúmenes (ver Figura 4.4, donde se muestra una vista frontal del sistema diseñado), los sensores se desplazarán a velocidad constante sobre la plataforma circular hasta completar un desplazamiento de 180 grados sexagesimales, momento en el cual el sentido de desplazamiento de los sensores se invertirá hasta que hayan girado otros 180 grados

sexagesimales en sentido opuesto (volviendo a la posición inicial), y este proceso se repetirá mientras dure la toma de datos del tronco de madera o rollizo.

En tal sentido, el desplazamiento de los sensores a velocidad constante es un requisito indispensable, pues una velocidad variable afectaría la uniformidad en la toma de datos de los sensores, lo cual conllevaría a obtener una reconstrucción volumétrica distorsionada.

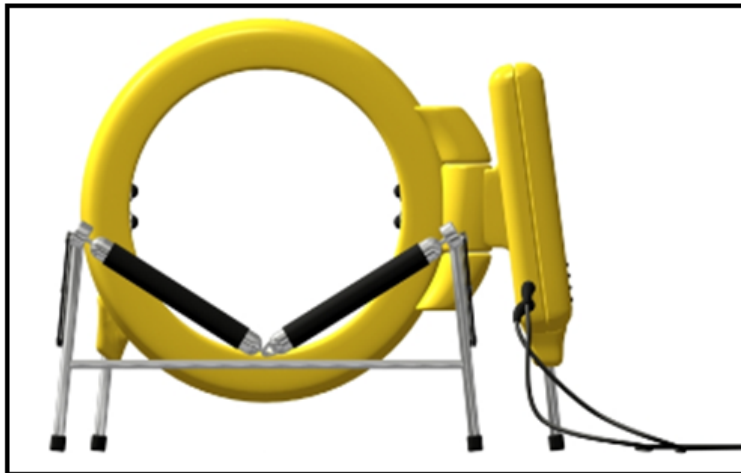


Figura 4.4 Vista de frente del sistema propuesto [18]

Como se indicó en el Capítulo 3, un motor DC será el que accionaría el desplazamiento de los sensores sobre la plataforma circular. Asimismo, la velocidad del motor DC será controlada mediante modulación por ancho de pulso o PWM (del inglés: Pulse Width Modulation).

Para el proceso de simulación, se generará por la línea de salida RB2 del microcontrolador PIC16F84A una onda cuadrada de frecuencia constante e igual a 100Hz (equivalente a un periodo de 10 ms). Esta onda cuadrada tendrá un ciclo de trabajo variable (desde 0% hasta 100%) que dependerá del valor de una entrada digital (RA3...RA0 del PIC16F84A). En la Figura 4.6 (página 61) se muestra el diagrama esquemático del circuito propuesto para regular la velocidad de giro del eje del motor DC por PWM (dibujado en ISIS Professional, uno de los módulos componentes del software Proteus).

Para poder visualizar con mayor facilidad el aumento o disminución del ciclo de trabajo de la onda cuadrada (y sin pérdida de generalidad), la entrada digital variará desde '0000' hasta '1010' (de 0 a 10, en decimal), de tal forma que se presentarán ondas cuadradas con un ciclo de trabajo desde 0% hasta 100% con aumentos progresivos de 10%, tal como se resume en la Figura 4.5.

Entrada digital (RA3...RA0)	Valor decimal de la entrada digital	Ciclo de trabajo (%)	Tiempo de la señal en alta	Tiempo de la señal en baja
'0000'	0	0%	0 ms	10 ms
'0001'	1	10%	1 ms	9 ms
'0010'	2	20%	2 ms	8 ms
'0011'	3	30%	3 ms	7 ms
'0100'	4	40%	4 ms	6 ms
'0101'	5	50%	5 ms	5 ms
'0110'	6	60%	6 ms	4 ms
'0111'	7	70%	7 ms	3 ms
'1000'	8	80%	8 ms	2 ms
'1001'	9	90%	9 ms	1 ms
'1010'	10	100%	10 ms	0 ms
Mayor a '1010'	Mayor a 10	0%	0 ms	10 ms

Figura 4.5 Correlación entrada digital (RA3...RA0) a ciclo de trabajo

La variación del ciclo de trabajo del motor DC en base a una entrada digital (según resumen que se muestra en la Figura 4.5) será analizada gráficamente y en mayor detalle en la Figura 4.6 a Figura 4.11, donde se presentarán y explicarán las simulaciones efectuadas usando el software Proteus.

Para poder tener un mejor entendimiento de los circuitos simulados que se mostrarán en adelante, resulta importante señalar que el software Proteus indica en tiempo real el nivel de voltaje TTL en cualquier nodo del circuito mediante pequeños cuadrados de colores, bajo la siguiente notación:

- ✓ Un cuadrado de color azul sobre un nodo indica nivel bajo TTL o '0' lógico en dicho nodo.
- ✓ Un cuadrado de color rojo sobre un nodo indica nivel alto TTL o '1' lógico en dicho nodo.

Para mayores referencias acerca de las simulaciones de circuitos mediante el uso del software Proteus resulta conveniente revisar el trabajo de Palacios, Remiro y Lucas [21,22,23,24,25,39].

- **Ciclo de trabajo de 0%**

En la Figura 4.6 se muestra el circuito simulado para un ciclo de trabajo de 0% utilizando el software Proteus. Como allí se aprecia, los interruptores conectados a los pines RA3, RA2, RA1, RA0 se encuentran cerrados.

Este estado de los interruptores hará que los pines RA3, RA2, RA1, RA0 del microcontrolador PIC16F84A estén a nivel bajo TTL, formando la entrada digital (RA3...RA0) = '0000'. Esta entrada digital es verificable en el software de simulación, ya que en la Figura 4.6 los pines RA3, RA2, RA1, RA0 tienen asignado un cuadrado de color azul (nivel bajo TTL).

Así, la entrada digital '0000' (0 en decimal) corresponde a un ciclo de trabajo de 0%, tal como se aprecia en la pantalla del osciloscopio, donde el 100% de la onda se encuentra a nivel bajo TTL.

Bajo este escenario, el eje del motor DC no se moverá, pues funcionará como si el motor no estuviera energizado. Ver que, en este caso, el pin RB2 del microcontrolador estará constantemente a nivel bajo TTL (cuadrado de color azul sobre dicho pin), lo cual deshabilitará los drivers del circuito integrado L293B.

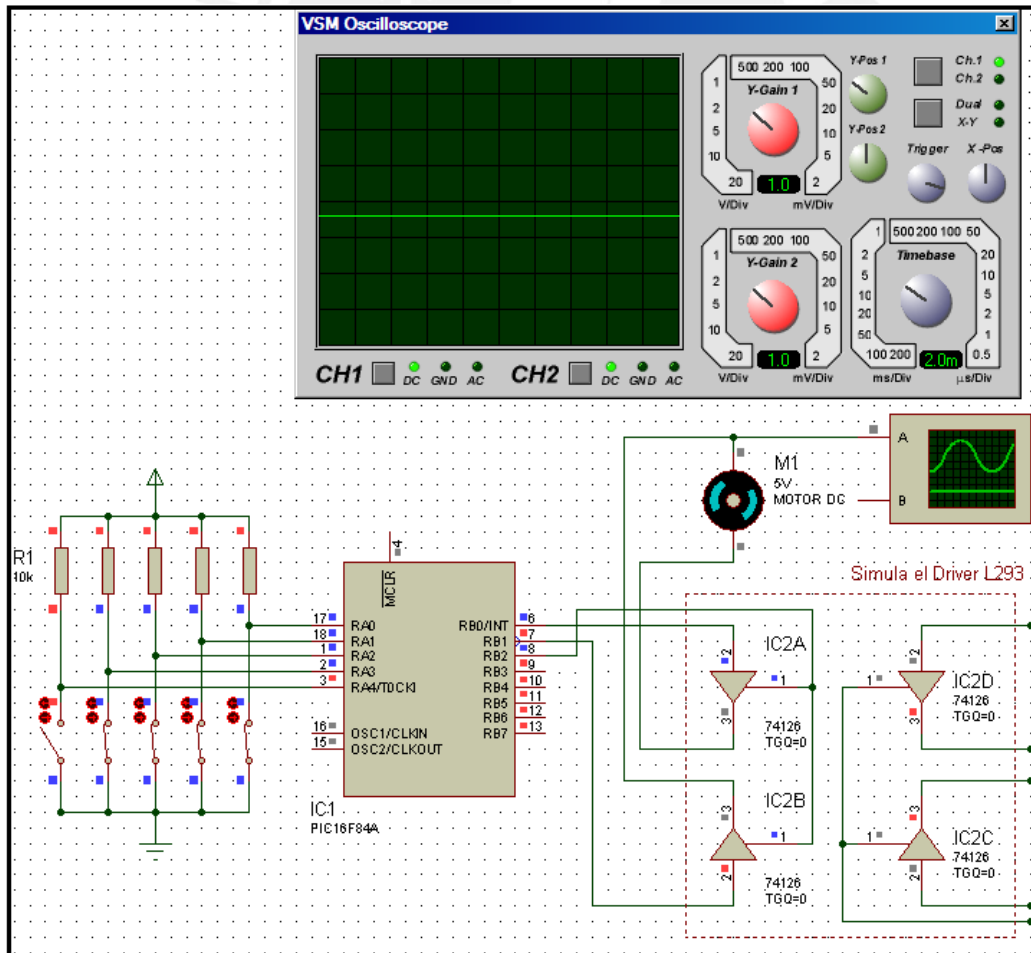


Figura 4.6 Simulación para un ciclo de trabajo de 0% [39]

- **Ciclo de trabajo de 10%**

En la Figura 4.7 se muestra el circuito simulado para un ciclo de trabajo de 10% utilizando el software Proteus. Como allí se aprecia, los interruptores conectados a los pines RA3, RA2, RA1 se encuentran cerrados, y el interruptor conectado al pin RA0 se encuentra abierto.

Este estado de los interruptores hará que los pines RA3, RA2, RA1 estén a nivel bajo TTL, y el pin RA0 a nivel alto TTL, formando la entrada digital (RA3...RA0) = '0001'. Esta entrada digital es verificable en el software de simulación, ya que en la Figura 4.7 los pines RA3, RA2, RA1 tienen asignado un cuadrado de color azul (nivel bajo TTL) y el pin RA0 tiene asignado un cuadrado de color rojo (nivel alto TTL).

Así, la entrada digital '0001' (1 en decimal) corresponde a un ciclo de trabajo de 10%, tal como se aprecia en la pantalla del osciloscopio, donde la fracción de 1/10 del periodo de la onda cuadrada se encuentra a nivel alto y el resto del periodo a nivel bajo. Este incremento del ciclo de trabajo a 10% hará que el motor incremente su velocidad en comparación a cuando el ciclo de trabajo era de 0%.

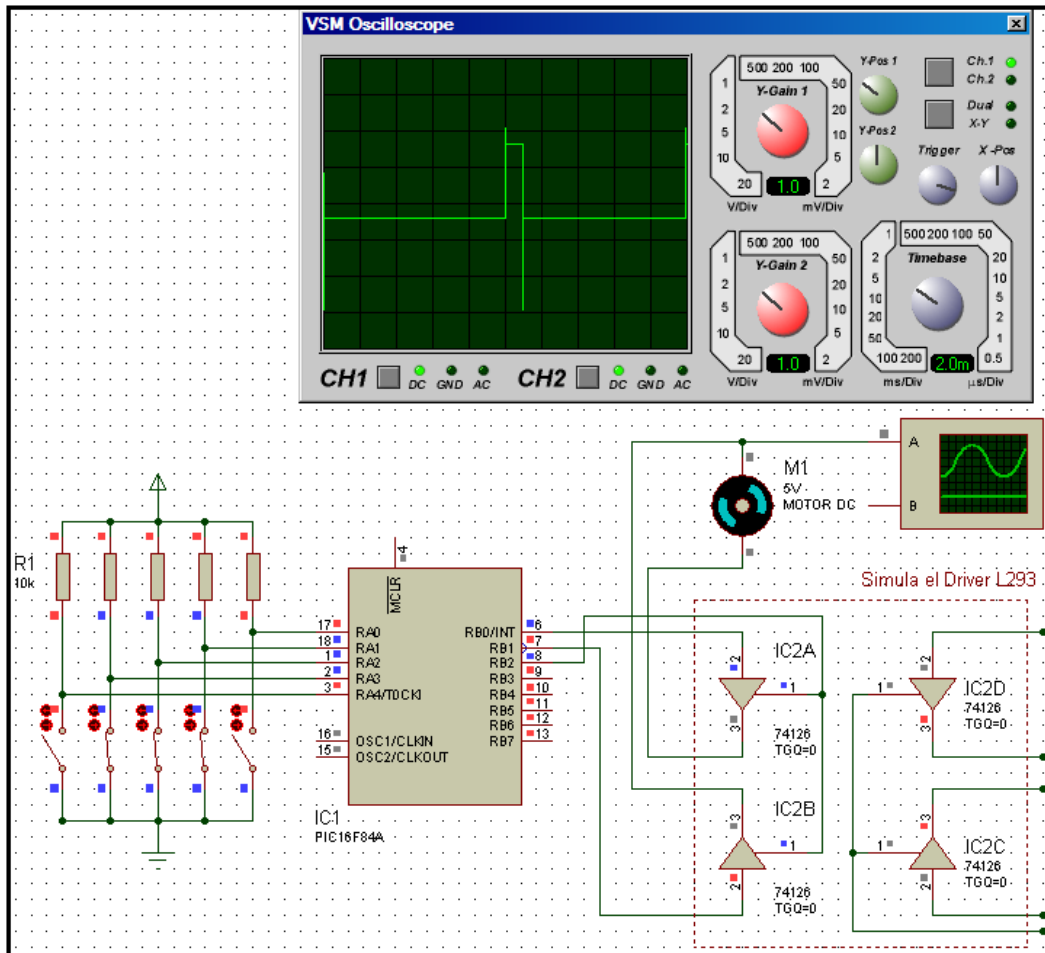


Figura 4.7 Simulación para un ciclo de trabajo de 10% [39]

- **Ciclo de trabajo de 20%**

En la Figura 4.8 se muestra el circuito simulado para un ciclo de trabajo de 20% utilizando el software Proteus. Como allí se aprecia, los interruptores conectados a los pines RA3, RA2, RA0 se encuentran cerrados, y el interruptor conectado al pin RA1 se encuentra abierto.

Este estado de los interruptores hará que los pines RA3, RA2, RA0 estén a nivel bajo TTL, y el pin RA1 a nivel alto TTL, formando la entrada digital (RA3...RA0) = '0010'. Esta entrada digital es verificable en el software de simulación, ya que en la Figura 4.8 los pines RA3, RA2, RA0 tienen asignado un cuadrado de color azul (nivel bajo TTL) y el pin RA1 tiene asignado un cuadrado de color rojo (nivel alto TTL).

Así, la entrada digital '0010' (2 en decimal) corresponde a un ciclo de trabajo de 20%, tal como se aprecia en la pantalla del osciloscopio, donde la fracción de 2/10 del periodo de la onda cuadrada se encuentra a nivel alto y el resto del periodo a nivel bajo. Este incremento del ciclo de trabajo a 20% hará que el motor incremente su velocidad en comparación a cuando el ciclo de trabajo era de 10%.

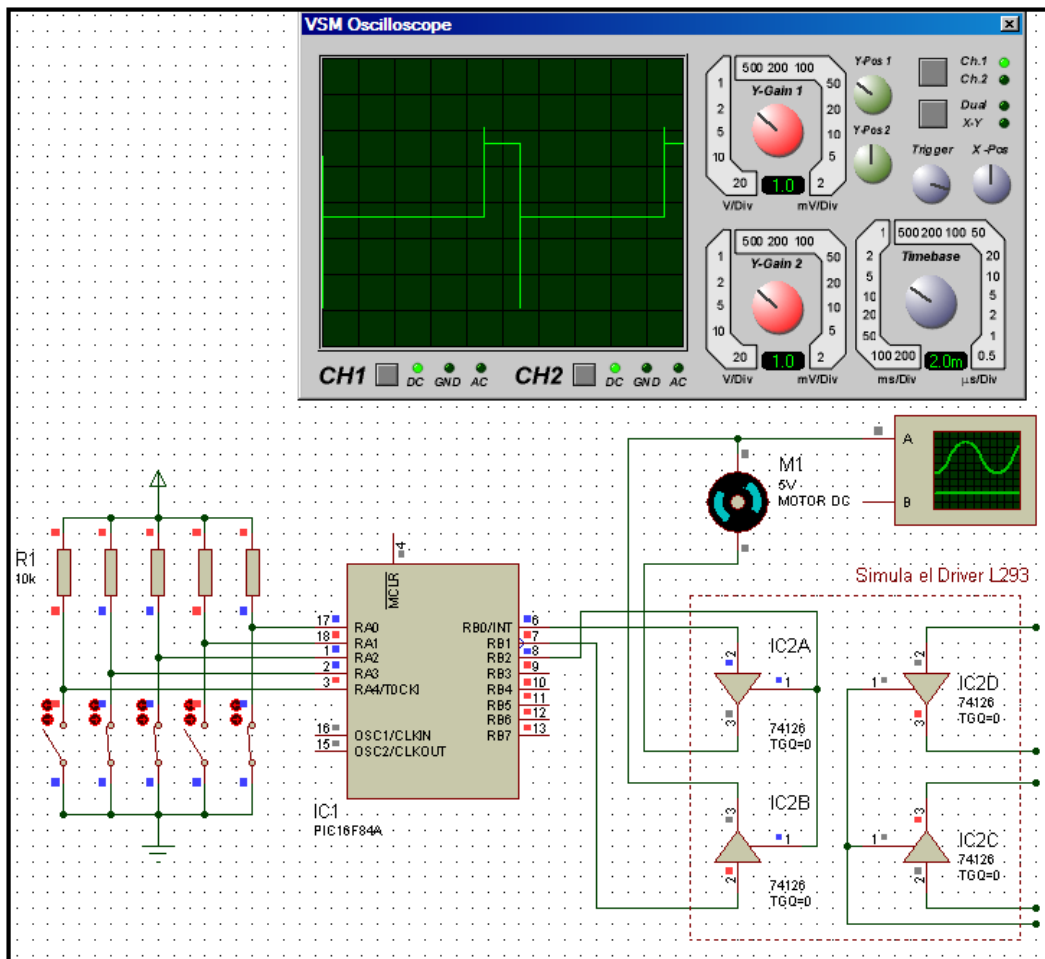


Figura 4.8 Simulación para un ciclo de trabajo de 20% [39]

- **Ciclo de trabajo de 30%**

En la Figura 4.9 se muestra el circuito simulado para un ciclo de trabajo de 30% utilizando el software Proteus. Como allí se aprecia, los interruptores conectados a los pines RA3, RA2 se encuentran cerrados, y los interruptores conectados a los pines RA1, RA0 se encuentran abiertos.

Este estado de los interruptores hará que los pines RA3, RA2 estén a nivel bajo TTL, y los pines RA1, RA0 a nivel alto TTL, formando la entrada digital (RA3...RA0) = '0011'. Esta entrada digital es verificable en el software de simulación, ya que en la Figura 4.9 los pines RA3, RA2 tienen asignado un cuadrado de color azul (nivel bajo TTL) y los pines RA1, RA0 tienen asignado un cuadrado de color rojo (nivel alto TTL).

Así, la entrada digital '0011' (3 en decimal) corresponde a un ciclo de trabajo de 30%, tal como se aprecia en la pantalla del osciloscopio, donde la fracción de 3/10 del periodo de la onda cuadrada se encuentra a nivel alto y el resto del periodo a nivel bajo. Este incremento del ciclo de trabajo a 30% hará que el motor incremente su velocidad en comparación a cuando el ciclo de trabajo era de 20%.

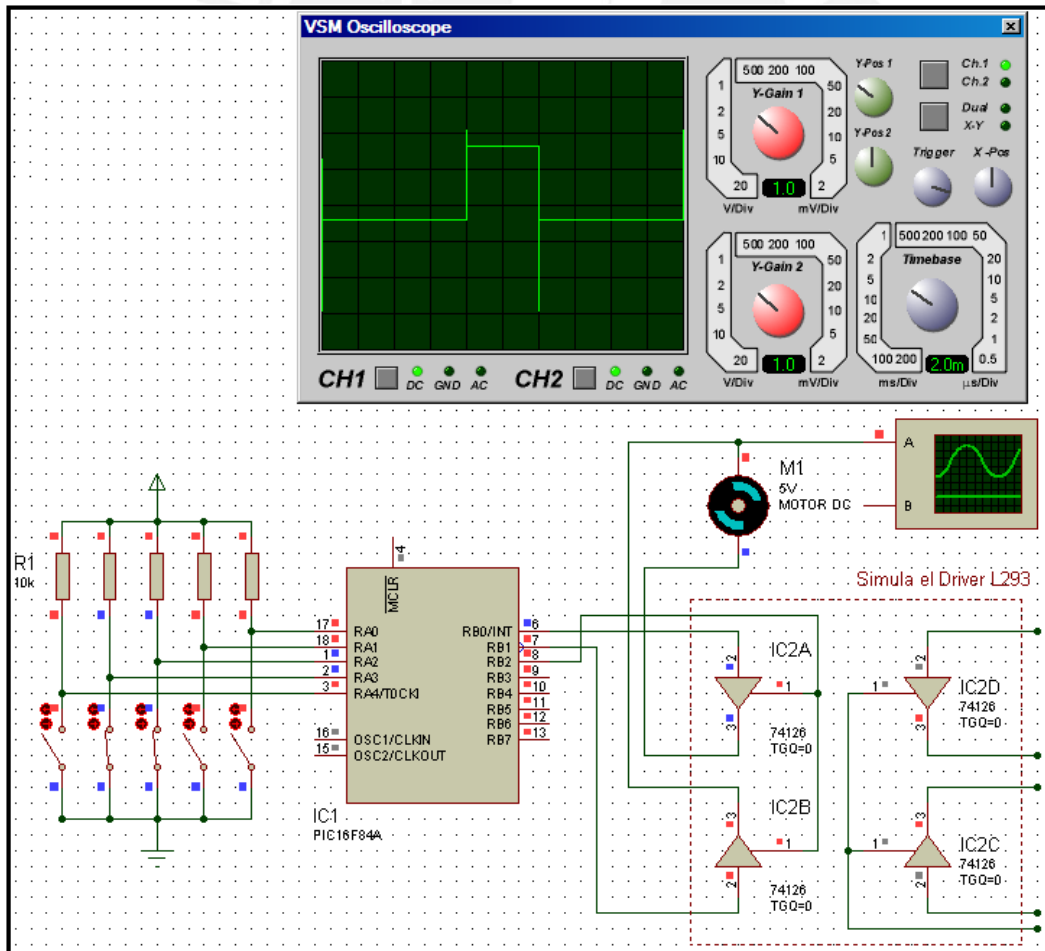


Figura 4.9 Simulación para un ciclo de trabajo de 30% [39]

- **Ciclo de trabajo de 40%**

En la Figura 4.10 se muestra el circuito simulado para un ciclo de trabajo de 40% utilizando el software Proteus. Como allí se aprecia, los interruptores conectados a los pines RA3, RA1, RA0 se encuentran cerrados, y el interruptor conectado al pin RA2 se encuentra abierto.

Este estado de los interruptores hará que los pines RA3, RA1, RA0 estén a nivel bajo TTL, y el pin RA2 a nivel alto TTL, formando la entrada digital (RA3...RA0) = '0100'. Esta entrada digital es verificable en el software de simulación, ya que en la Figura 4.10 los pines RA3, RA1, RA0 tienen asignado un cuadrado de color azul (nivel bajo TTL) y el pin RA2 tiene asignado un cuadrado de color rojo (nivel alto TTL).

Así, la entrada digital '0100' (4 en decimal) corresponde a un ciclo de trabajo de 40%, tal como se aprecia en la pantalla del osciloscopio, donde la fracción de 4/10 del periodo de la onda cuadrada se encuentra a nivel alto y el resto del periodo a nivel bajo. Este incremento del ciclo de trabajo a 40% hará que el motor incremente su velocidad en comparación a cuando el ciclo de trabajo era de 30%.

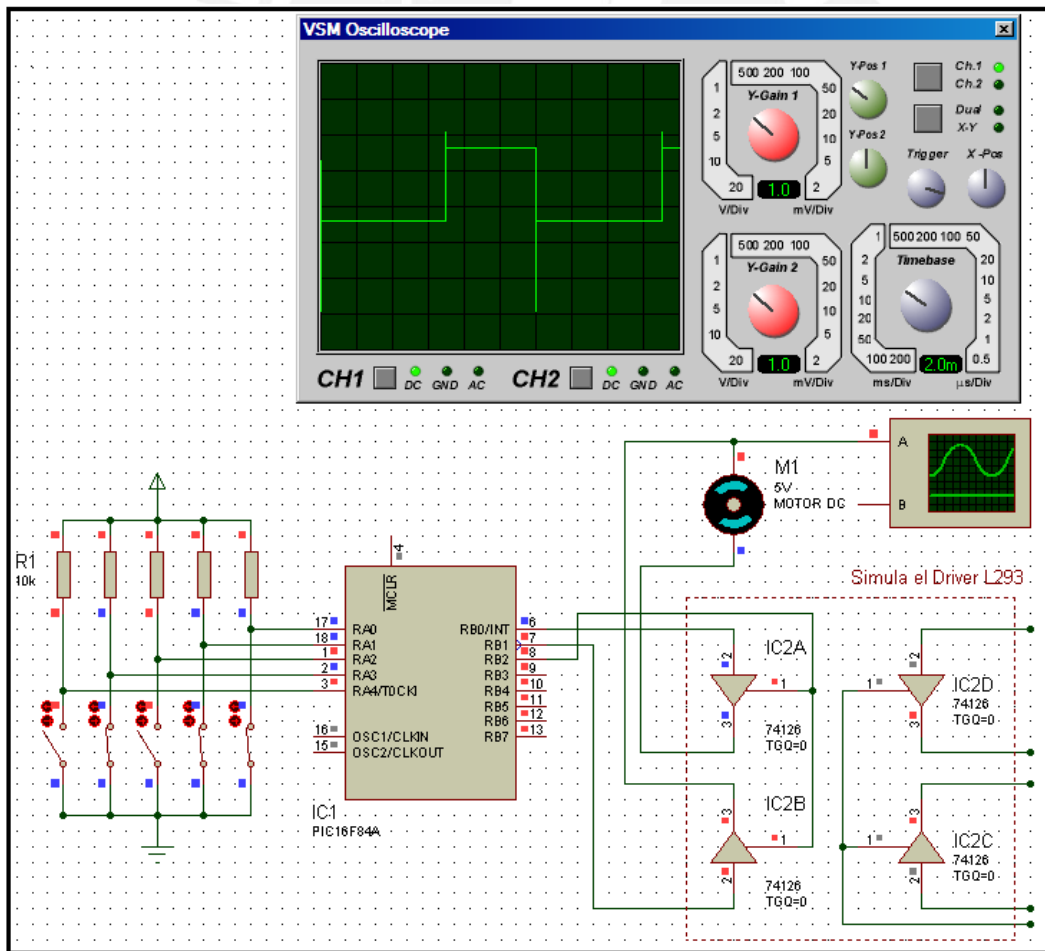


Figura 4.10 Simulación para un ciclo de trabajo de 40% [39]

- **Ciclo de trabajo de 50%**

En la Figura 4.11 se muestra el circuito simulado para un ciclo de trabajo de 50% utilizando el software Proteus. Como allí se aprecia, los interruptores conectados a los pines RA3, RA1 se encuentran cerrados, y los interruptores conectados a los pines RA2, RA0 se encuentran abiertos.

Este estado de los interruptores hará que los pines RA3, RA1 estén a nivel bajo TTL, y los pines RA2, RA0 a nivel alto TTL, formando la entrada digital (RA3...RA0) = '0101'. Esta entrada digital es verificable en el software de simulación, ya que en la Figura 4.11 los pines RA3, RA1 tienen asignado un cuadrado de color azul (nivel bajo TTL) y los pines RA2, RA0 tienen asignado un cuadrado de color rojo (nivel alto TTL).

Así, la entrada digital '0101' (5 en decimal) corresponde a un ciclo de trabajo de 50%, tal como se aprecia en la pantalla del osciloscopio, donde la fracción de 5/10 del periodo de la onda cuadrada se encuentra a nivel alto y el resto del periodo a nivel bajo. Este incremento del ciclo de trabajo a 50% hará que el motor incremente su velocidad en comparación a cuando el ciclo de trabajo era de 40%.

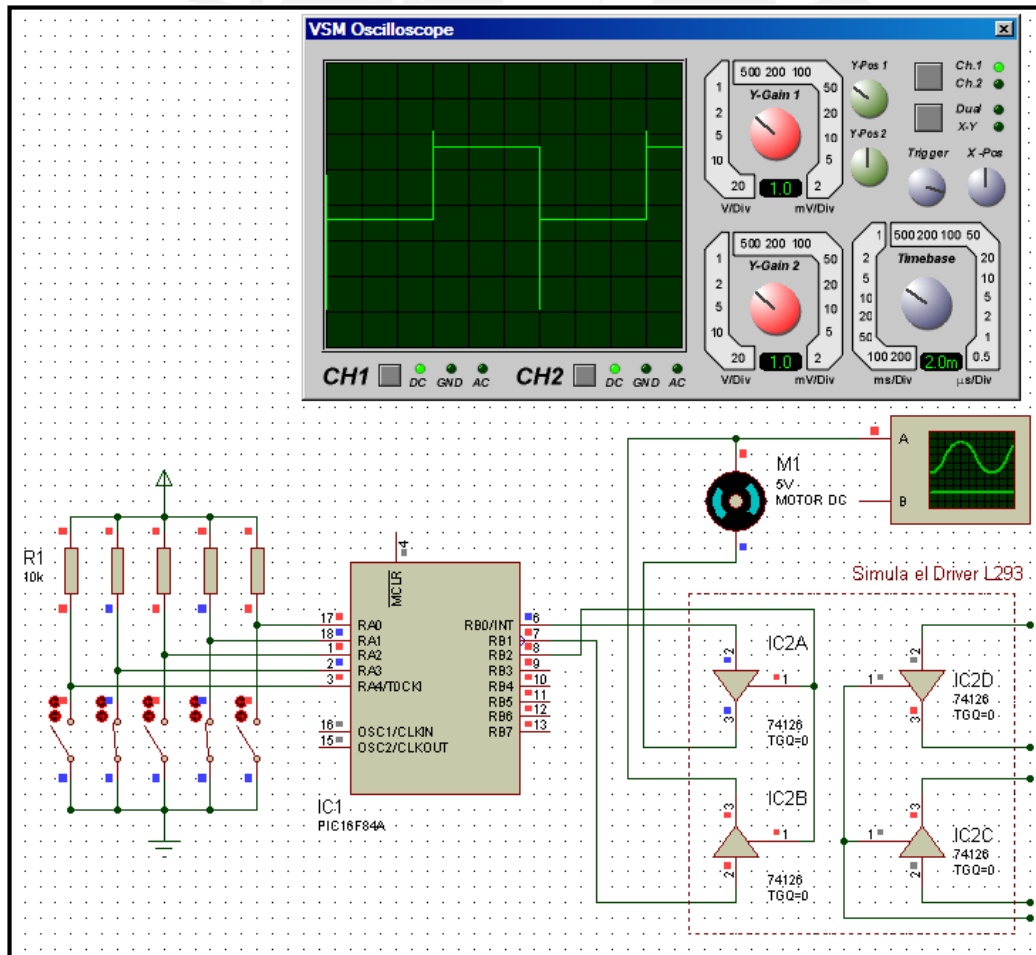


Figura 4.11 Simulación para un ciclo de trabajo de 50% [39]

4.4 Simulación del control de sentido de giro del motor DC

Como se explicó en el Capítulo 3, la inversión del sentido de giro del motor DC implica simplemente un cambio de polaridad en la alimentación del motor mediante el uso del Driver L293B. Como ya se mencionó, para el diseño propuesto, el instante de cambio en el sentido de giro será indicado por un sensor de fin de carrera, el cual enviará la señal al bloque de control para que el sentido de giro sea invierta.

Para el proceso de simulación, y sin pérdida de generalidad, el sensor de fin de carrera ha sido reemplazado por el interruptor conectado al pin RA4 del microcontrolador PIC16F84A. De esta manera, si el interruptor conectado al pin RA4 se encuentra abierto, el eje del motor DC girará en un sentido; y si el interruptor se encuentra cerrado, el eje del motor DC girará en sentido opuesto.

En este punto, resulta importante notar que en las simulaciones mostradas en la Figura 4.6 a 4.11 el interruptor conectado al pin RA4 del PIC16F84A (interruptor que simula el sensor de fin de carrera) se encontraba abierto, lo cual energizaba el motor DC de tal modo que el pin RB0 se encontraba a nivel bajo TTL (cuadrado de color azul sobre dicho pin) y el pin RB1 se encontraba a nivel alto TTL (cuadrado de color rojo sobre dicho pin). Luego, en las simulaciones de la Figura 4.12 a 4.17 se mostrarán las simulaciones de los mismos circuitos, pero con la diferencia de que esta vez el interruptor conectado al pin RA4 del microcontrolador PIC16F84A estará cerrado, lo cual invertirá la polaridad de alimentación del motor DC, por lo que ahora el pin RB0 se encontrará a nivel alto TTL (cuadrado de color rojo sobre dicho pin) y el pin RB1 se encontrará a nivel bajo TTL (cuadrado de azul sobre dicho pin).

Considerando lo expuesto en la sección 3.5.3 del capítulo 3, el resultado teórico que deberíamos obtener al efectuar estas nuevas simulaciones es que el sentido de giro del eje del motor DC se debe invertir. Esta inversión del sentido de giro del eje del motor DC es una consecuencia de la inversión en su polaridad de alimentación, lo cual es fácilmente verificable en el software de simulación Proteus analizando la forma de la onda cuadrada que veremos en el osciloscopio.

Es decir, para verificar la inversión en el sentido de giro del eje del motor DC, en las nuevas simulaciones con el interruptor conectado al pin RA4 cerrado (Figura 4.12 a Figura 4.17), las señales que deberíamos ver en la pantalla del osciloscopio deberían ser similares a las que se mostraron en la Figura 4.6 a 4.11 (respectivamente), excepto que deben estar invertidas, tal como lo comprobaremos.

Nuevamente, para mayores referencias acerca de las simulaciones de circuitos mediante el uso del software Proteus resulta conveniente revisar el trabajo de Palacios, Remiro y Lucas [21,22,23,24,25,39].

- **Ciclo de trabajo de 0%**

En la Figura 4.12 se muestra el circuito simulado para un ciclo de trabajo de 0% utilizando el software Proteus. Como allí se aprecia, los interruptores conectados a los pines RA3, RA2, RA1, RA0 se encuentran cerrados (y a diferencia de la simulación de la Figura 4.6, el interruptor conectado al pin RA4 está cerrado).

Este estado de los interruptores hará que los pines RA3, RA2, RA1, RA0 estén a nivel bajo TTL, formando la entrada digital (RA3...RA0) = '0000'. Esta entrada digital es verificable en el software de simulación, ya que en la Figura 4.12 los pines RA3, RA2, RA1, RA0 tienen asignado un cuadrado de color azul (nivel bajo TTL).

Así, la entrada digital '0000' (0 en decimal) corresponde a un ciclo de trabajo de 0%, tal como se aprecia en la pantalla del osciloscopio, donde el 100% de la onda se encuentra a nivel bajo TTL, de forma similar a lo mostrado en la Figura 4.6. Sin embargo, se debe tener en cuenta que una onda con nivel bajo constante, al ser invertida, gráficamente resulta la misma onda. Al igual que en el caso de la Figura 4.6, con un ciclo de trabajo de 0% el eje del motor DC no se moverá.

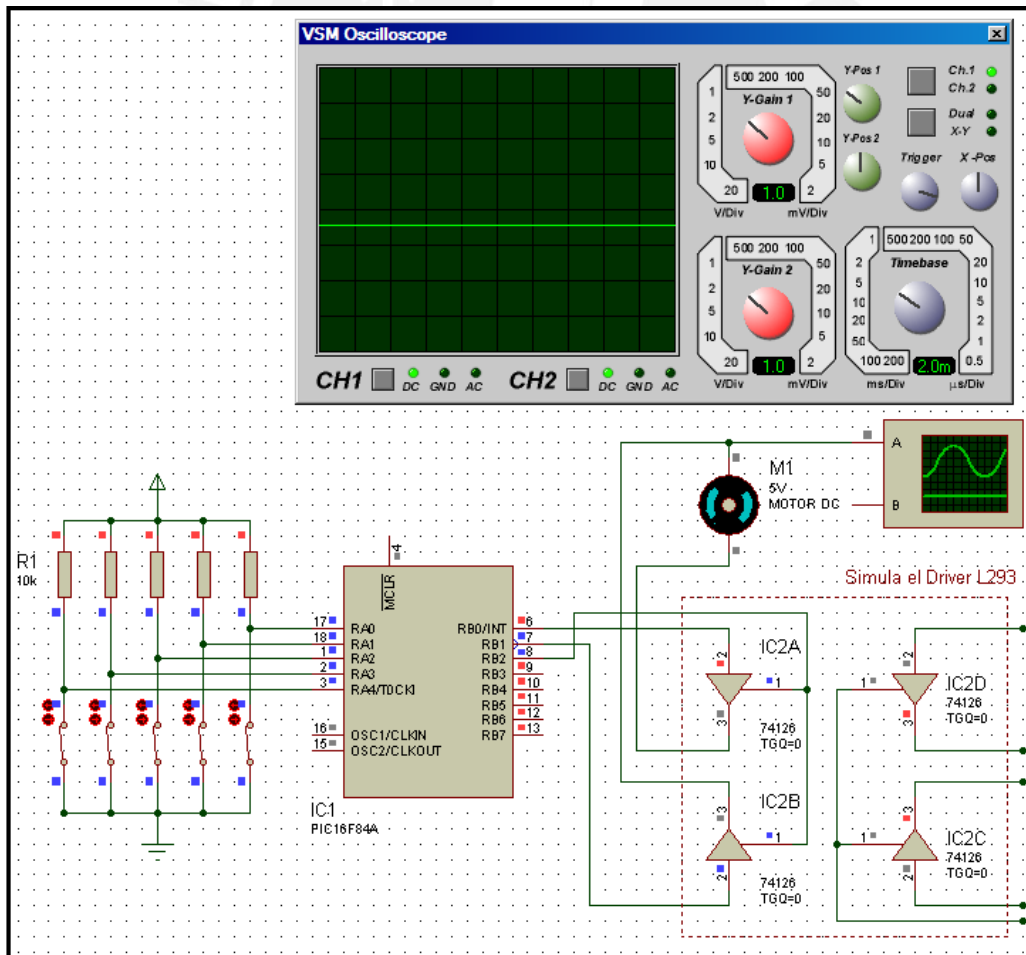


Figura 4.12 Simulación ciclo de trabajo 0% e inversión de giro del motor [39]

- **Ciclo de trabajo de 10%**

En la Figura 4.13 se muestra el circuito simulado para un ciclo de trabajo de 10% utilizando el software Proteus. Como allí se aprecia, los interruptores conectados a los pines RA3, RA2, RA1 se encuentran cerrados, y el interruptor conectado al pin RA0 se encuentra abierto (pero a diferencia de la simulación mostrada la Figura 4.7, el interruptor conectado al pin RA4 está cerrado).

Este estado de los interruptores hará que los pines RA3, RA2, RA1 estén a nivel bajo TTL, y el pin RA0 a nivel alto TTL, formando la entrada digital (RA3...RA0) = '0001'. Esta entrada digital es verificable en el software de simulación, ya que en la Figura 4.13 los pines RA3, RA2, RA1 tienen asignado un cuadrado de color azul (nivel bajo TTL) y el pin RA0 tiene asignado un cuadrado de color rojo (nivel alto TTL). La entrada digital '0001' (1 en decimal) corresponde a un ciclo de trabajo de 10%, y en el osciloscopio se observa la misma señal de la Figura 4.7, pero invertida. Esta señal invertida grafica la inversión del sentido de giro del motor DC. Asimismo, el incremento del ciclo de trabajo a 10% hará que el motor incremente su velocidad en comparación a cuando el ciclo de trabajo era de 0%.

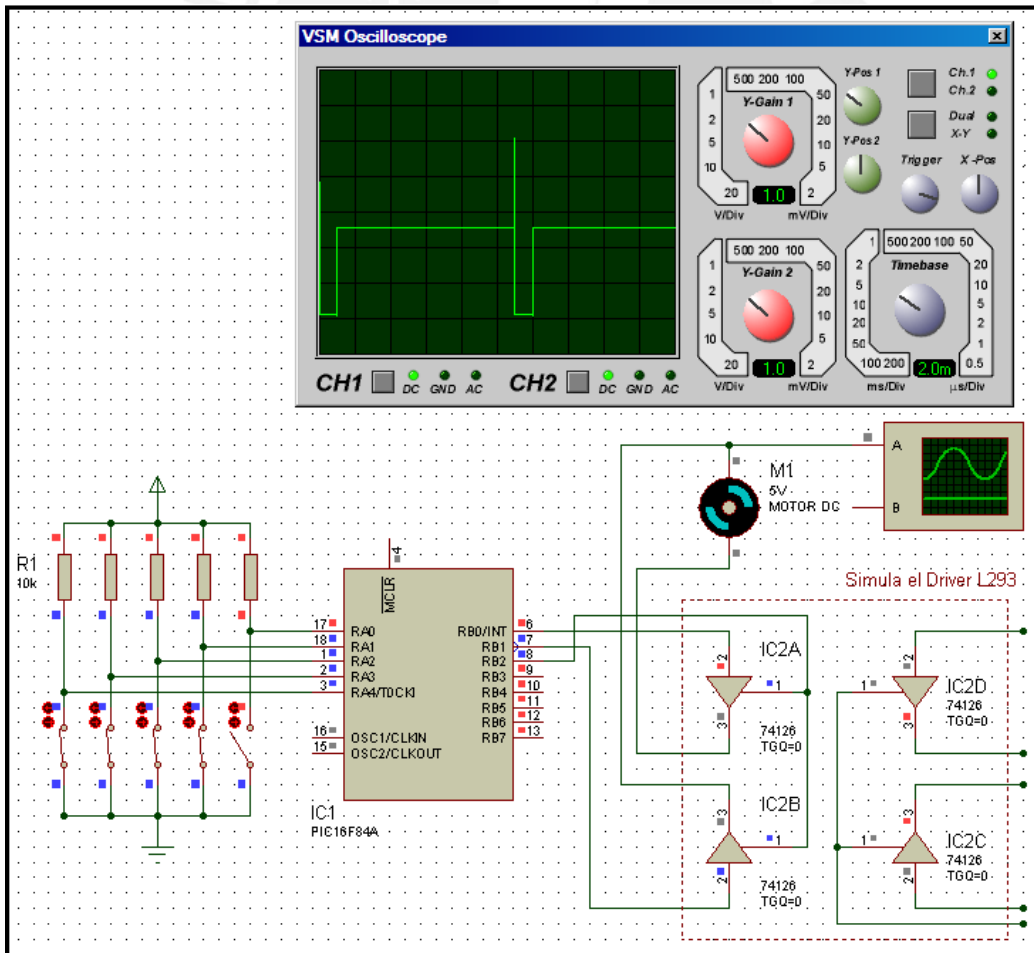


Figura 4.13 Simulación ciclo de trabajo 10% e inversión de giro del motor [39]

- **Ciclo de trabajo de 20%**

En la Figura 4.14 se muestra el circuito simulado para un ciclo de trabajo de 20% utilizando el software Proteus. Como allí se aprecia, los interruptores conectados a los pines RA3, RA2, RA0 se encuentran cerrados, y el interruptor conectado al pin RA1 se encuentra abierto (pero a diferencia de la simulación mostrada la Figura 4.8, el interruptor conectado al pin RA4 está cerrado).

Este estado de los interruptores hará que los pines RA3, RA2, RA0 estén a nivel bajo TTL, y el pin RA1 a nivel alto TTL, formando la entrada digital (RA3...RA0) = '0010'. Esta entrada digital es verificable en el software de simulación, ya que en la Figura 4.14 los pines RA3, RA2, RA0 tienen asignado un cuadrado de color azul (nivel bajo TTL) y el pin RA1 tiene asignado un cuadrado de color rojo (nivel alto TTL). La entrada digital '0010' (2 en decimal) corresponde a un ciclo de trabajo de 20%, y en el osciloscopio se observa la misma señal de la Figura 4.8, pero invertida. Esta señal invertida grafica la inversión del sentido de giro del motor DC. Asimismo, el incremento del ciclo de trabajo a 20% hará que el motor incremente su velocidad en comparación a cuando el ciclo de trabajo era de 10%.

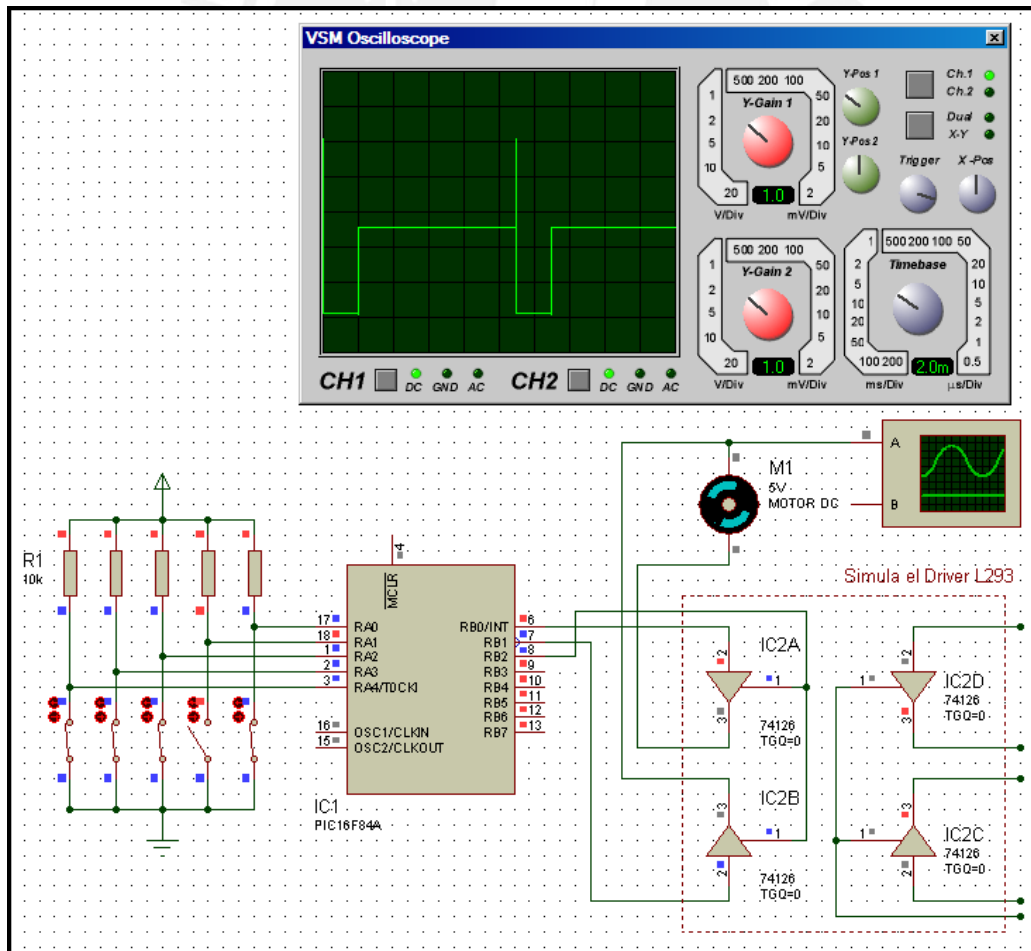


Figura 4.14 Simulación ciclo de trabajo 20% e inversión de giro del motor [39]

- **Ciclo de trabajo de 30%**

En la Figura 4.15 se muestra el circuito simulado para un ciclo de trabajo de 30% utilizando el software Proteus. Como allí se aprecia, los interruptores conectados a los pines RA3, RA2 se encuentran cerrados, y los interruptores conectados a los pines RA1, RA0 se encuentran abiertos (pero a diferencia de la simulación mostrada la Figura 4.9, el interruptor conectado al pin RA4 está cerrado).

Este estado de los interruptores hará que los pines RA3, RA2 estén a nivel bajo TTL, y los pines RA1, RA0 a nivel alto TTL, formando la entrada digital (RA3...RA0) = '0011'. Esta entrada digital es verificable en el software de simulación, ya que en la Figura 4.15 los pines RA3, RA2 tienen asignado un cuadrado de color azul (nivel bajo TTL) y los pines RA1, RA0 un cuadrado de color rojo (nivel alto TTL). La entrada digital '0011' (3 en decimal) corresponde a un ciclo de trabajo de 30%, y en el osciloscopio se observa la misma señal de la Figura 4.9, pero invertida. Esta señal invertida grafica la inversión del sentido de giro del motor DC. El incremento del ciclo de trabajo a 30% hará que el motor incremente su velocidad en comparación a cuando el ciclo de trabajo era de 20%.

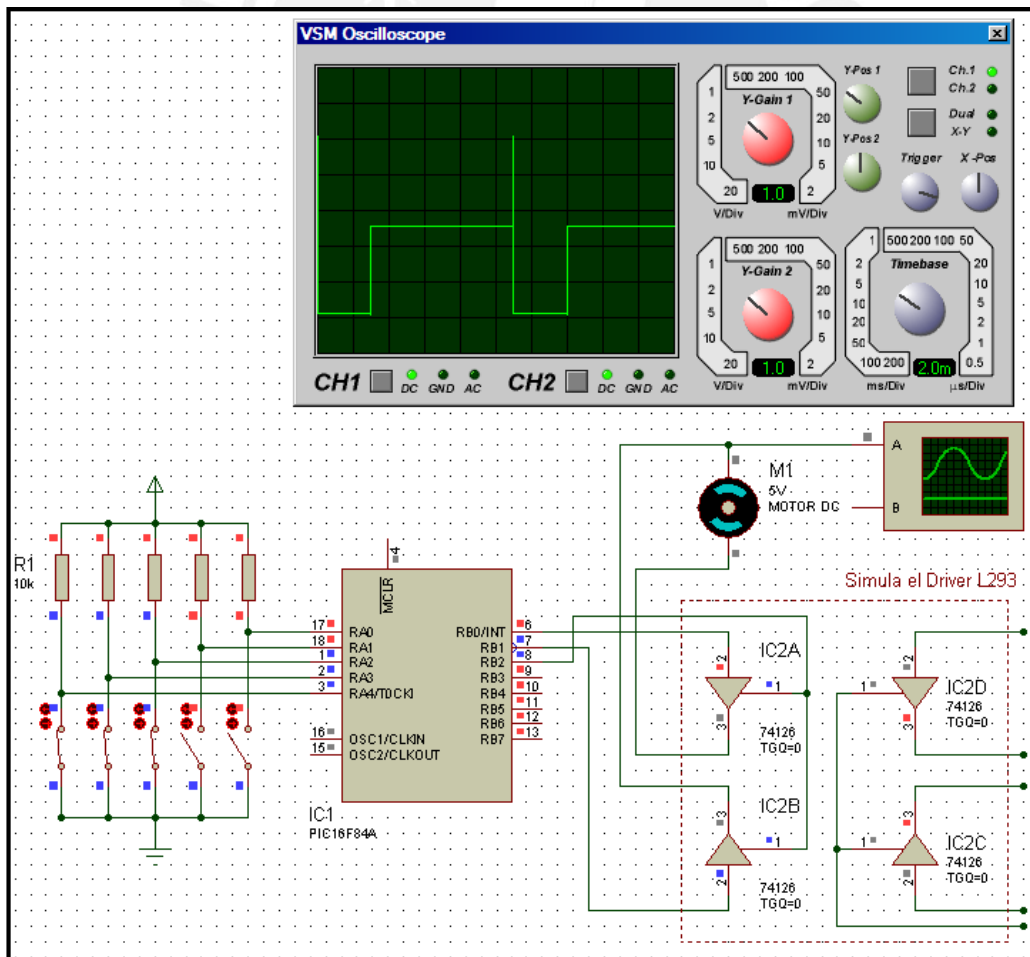


Figura 4.15 Simulación ciclo de trabajo 30% e inversión de giro del motor [39]

- **Ciclo de trabajo de 40%**

En la Figura 4.16 se muestra el circuito simulado para un ciclo de trabajo de 40% utilizando el software Proteus. Como allí se aprecia, los interruptores conectados a los pines RA3, RA1, RA0 se encuentran cerrados, y el interruptor conectado al pin RA2 se encuentra abierto (pero a diferencia de la simulación mostrada la Figura 4.10, el interruptor conectado al pin RA4 está cerrado).

Este estado de los interruptores hará que los pines RA3, RA1, RA0 estén a nivel bajo TTL, y el pin RA2 a nivel alto TTL, formando la entrada digital (RA3...RA0) = '0100'. Esta entrada digital es verificable en el software de simulación, ya que en la Figura 4.16 los pines RA3, RA1, RA0 tienen asignado un cuadrado de color azul (nivel bajo TTL) y el pin RA2 tiene asignado un cuadrado de color rojo (nivel alto TTL). La entrada digital '0100' (4 en decimal) corresponde a un ciclo de trabajo de 40%, y en el osciloscopio se observa la misma señal de la Figura 4.10, pero invertida. Esta señal invertida grafica la inversión del sentido de giro del motor DC. El incremento del ciclo de trabajo a 40% hará que el motor incremente su velocidad en comparación a cuando el ciclo de trabajo era de 30%.

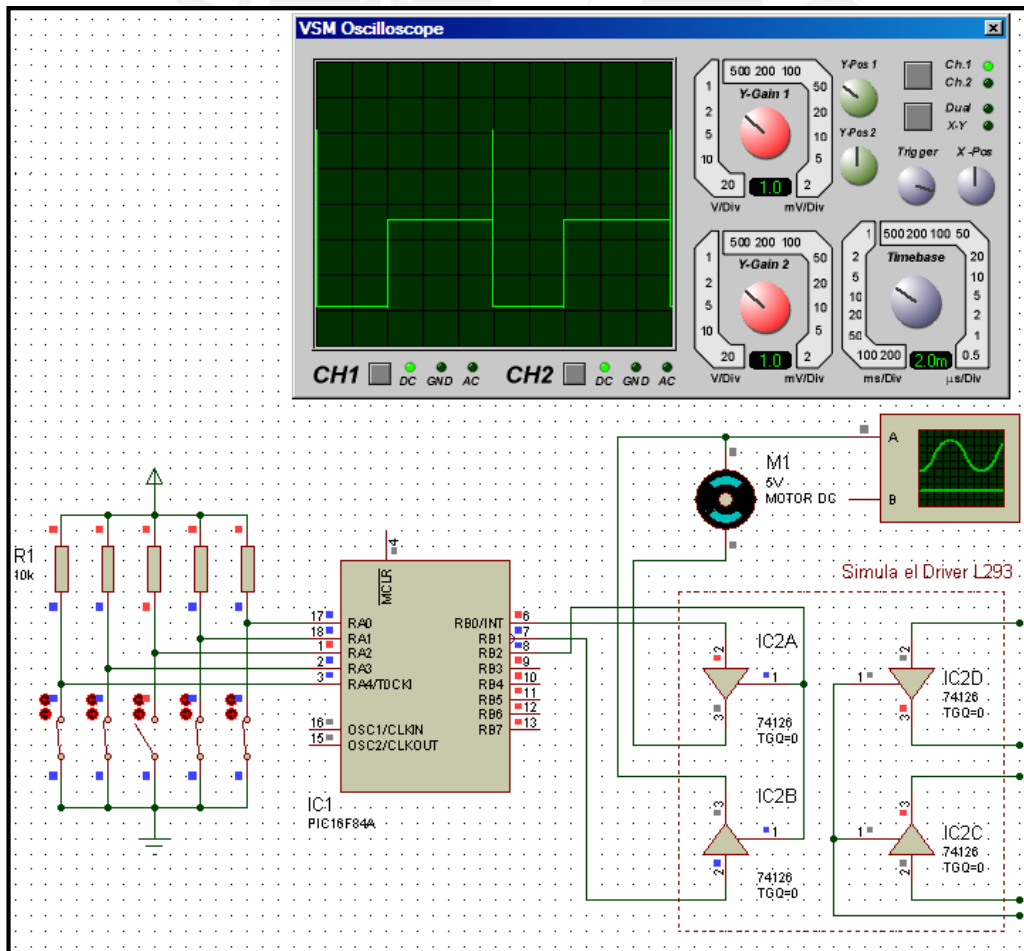


Figura 4.16 Simulación ciclo de trabajo 40% e inversión de giro del motor [39]

- **Ciclo de trabajo de 50%**

En la Figura 4.17 se muestra el circuito simulado para un ciclo de trabajo de 50% utilizando el software Proteus. Como allí se aprecia, los interruptores conectados a los pines RA3, RA1 se encuentran cerrados, y los interruptores conectados a los pines RA2, RA0 se encuentran abiertos (pero a diferencia de la simulación mostrada la Figura 4.11, el interruptor conectado al pin RA4 está cerrado). Este estado de los interruptores hará que los pines RA3, RA1 estén a nivel bajo TTL, y los pines RA2, RA0 a nivel alto TTL, formando la entrada digital (RA3...RA0) = '0101'. Esta entrada digital es verificable en el software de simulación, ya que en la Figura 4.17 los pines RA3, RA1 tienen asignado un cuadrado de color azul (nivel bajo TTL) y los pines RA2, RA0 tienen asignado un cuadrado de color rojo (nivel alto TTL). La entrada digital '0101' (5 en decimal) corresponde a un ciclo de trabajo de 50%, y en el osciloscopio se observa la misma señal de la Figura 4.11, pero invertida. Esta señal invertida grafica la inversión del sentido de giro del motor DC. El incremento del ciclo de trabajo a 50% hará que el motor incremente su velocidad en comparación a cuando el ciclo de trabajo era de 40%.

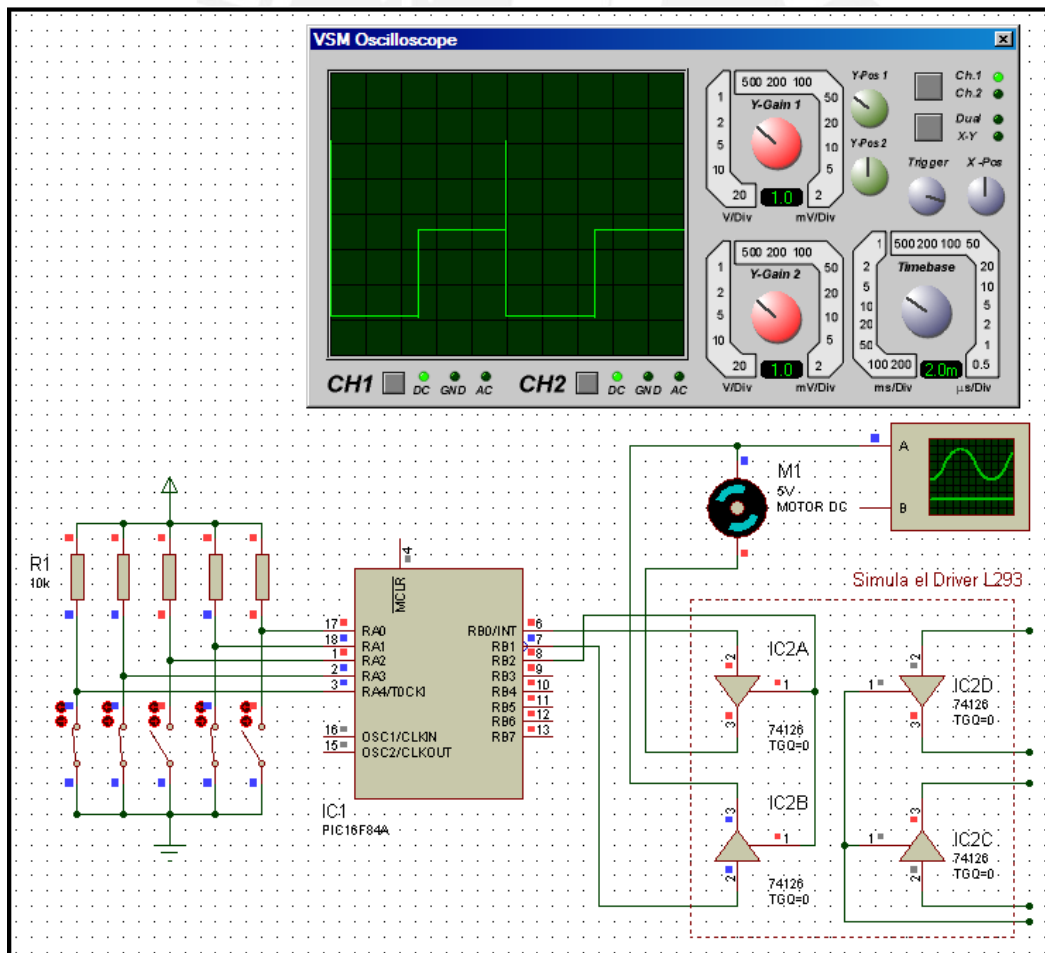


Figura 4.17 Simulación ciclo de trabajo 50% e inversión de giro del motor [39]

4.5 Simulación del proceso de medición de distancias

Únicamente para mostrar la simulación del proceso de medición de distancias (que efectuará el sensor ultrasonido SRF04) en el presente trabajo de tesis se decidió utilizar el entorno AVR Studio 4 y el microcontrolador ATmega8 (en lugar del microcontrolador PIC16F84A). Esta elección se basa en las siguientes razones:

- ✓ AVR Studio 4 permite mostrar de forma más clara el estado de los pines del microcontrolador a medida que se ejecuta paso a paso el programa de medición de distancias. Asimismo, el autor del presente trabajo de tesis tiene experiencia en la depuración de programas haciendo uso de AVR Studio 4.
- ✓ El autor del presente trabajo de tesis tenía ya a su disposición un módulo de entrenamiento que incluye un microcontrolador ATmega8, interruptores, pulsadores, leds, conectores y todo lo necesario para efectuar una simulación del proceso de medición de distancias.

Es decir, la elección de AVR Studio y el microcontrolador ATmega8 (únicamente para efectuar la simulación del proceso de medición de distancias del sensor ultrasonido SRF04) se justifica en términos de facilidad y costos.

En la Figura 4.18 se muestra un diagrama de pines del microcontrolador a utilizar. Como allí se muestra, para la simulación del proceso de medición de distancias, la entrada estará conformada por el pin PD0 del microcontrolador ATmega8, el cual cumplirá las funciones de disparo y eco, aunque no al mismo tiempo, ya que el eco y el disparo no se producen en simultáneo. Asimismo, la salida estará conformada por la salida digital en el puerto B del microcontrolador (PB7...PB0).

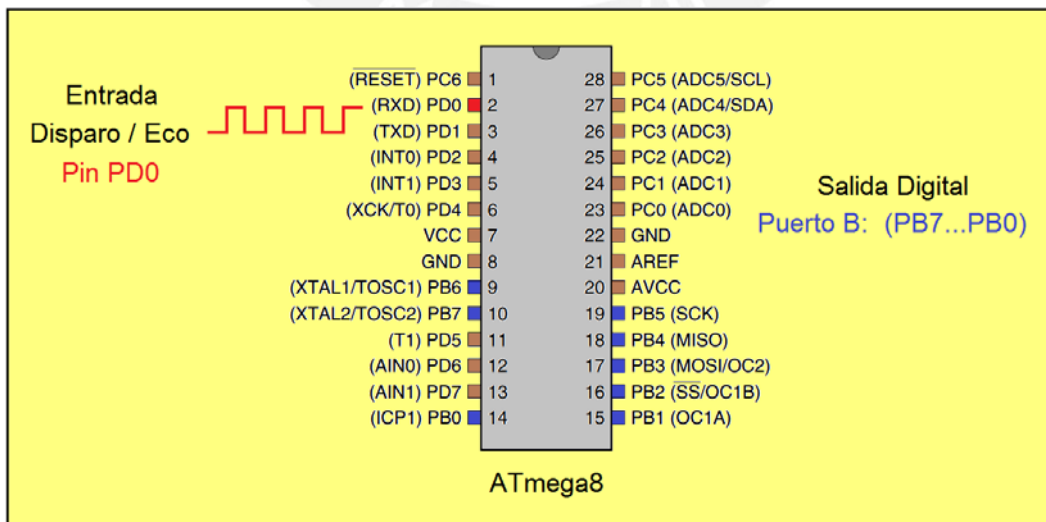


Figura 4.18 Entrada y salida para la simulación de medición de distancias

Para la simulación de la medición de distancias utilizaremos los siguientes puntos, ya expuestos en la sección 3.5.2.b Toma de datos de los sensores (página 30):

- ✓ El pulso de disparo debe tener una duración mínima de 10 μ s. Por ello, se creó la subrutina RETARDO_20 μ s, la cual produce un retardo de 20 μ s, que será el tiempo que durará el pulso de disparo en el pin PD0.
- ✓ Cada 60 μ s de pulso de eco corresponde a 1 cm de distancia medida por el sensor ultrasonido SRF04. Siendo así, el programa en lenguaje ensamblador se encargará de contar cuántos periodos de 60 μ s tiene el pulso de eco, y luego arrojará este número por el puerto B (PB7...PB0), siendo este número la distancia en centímetros medida por el sensor ultrasonido SRF04.

En la Figura 4.19 se muestra el inicio de la simulación del programa de medición de distancias. En dicha figura, como en adelante, la flecha de color amarillo indica el avance de la simulación. Empecemos revisando las siguientes instrucciones:

```
LDI R16, 1
OUT PORTD, R16
```

Estas instrucciones permiten poner a nivel alto el pin de disparo (pin PD0). Es decir, con estas instrucciones forzamos el inicio del pulso disparo.

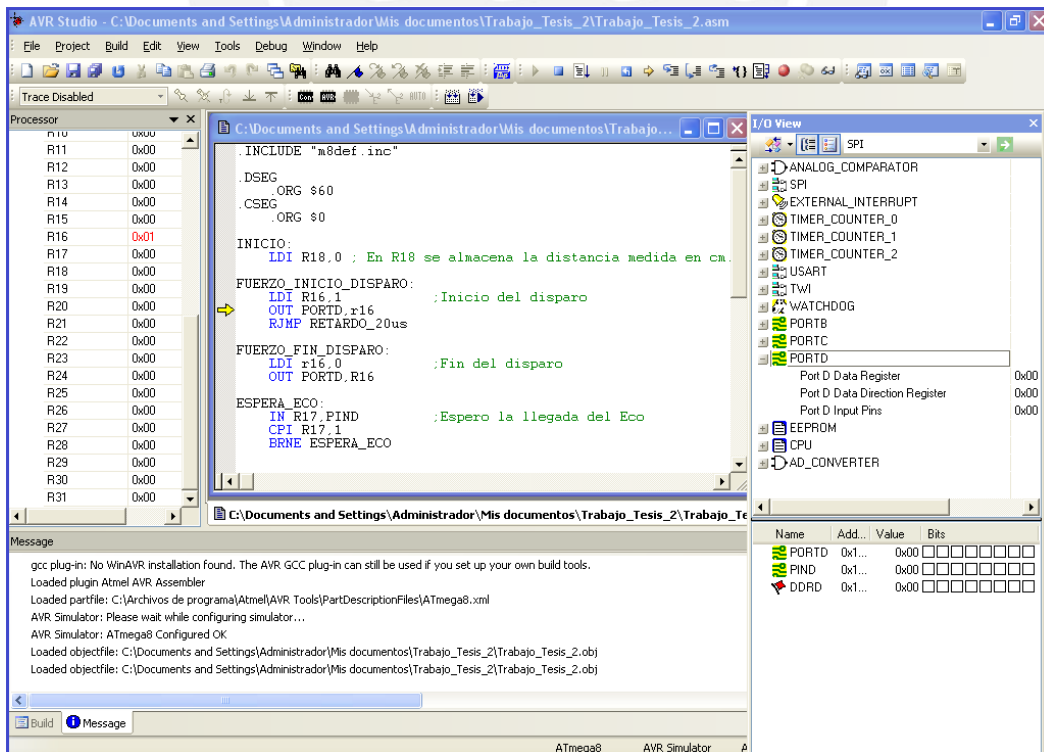


Figura 4.19 Simulación de medición de distancias: Inicio de pulso de disparo

Luego, en la Figura 4.20, la flecha de color amarillo nos indica que el programa ha pasado a ejecutar la siguiente instrucción:

RJMP RETARDO_20µs

Dado que, según las especificaciones del sensor ultrasonido SRF04, el pulso de disparo debe tener una duración mínima de 10 µs, luego de poner el pin PD0 a nivel alto (pulso de disparo), el programa ejecuta la subrutina **RETARDO_20µs**, la cual producirá un retardo de 20 µs, tiempo durante el cual el pulso de disparo (pin PD0) permanecerá a nivel alto.

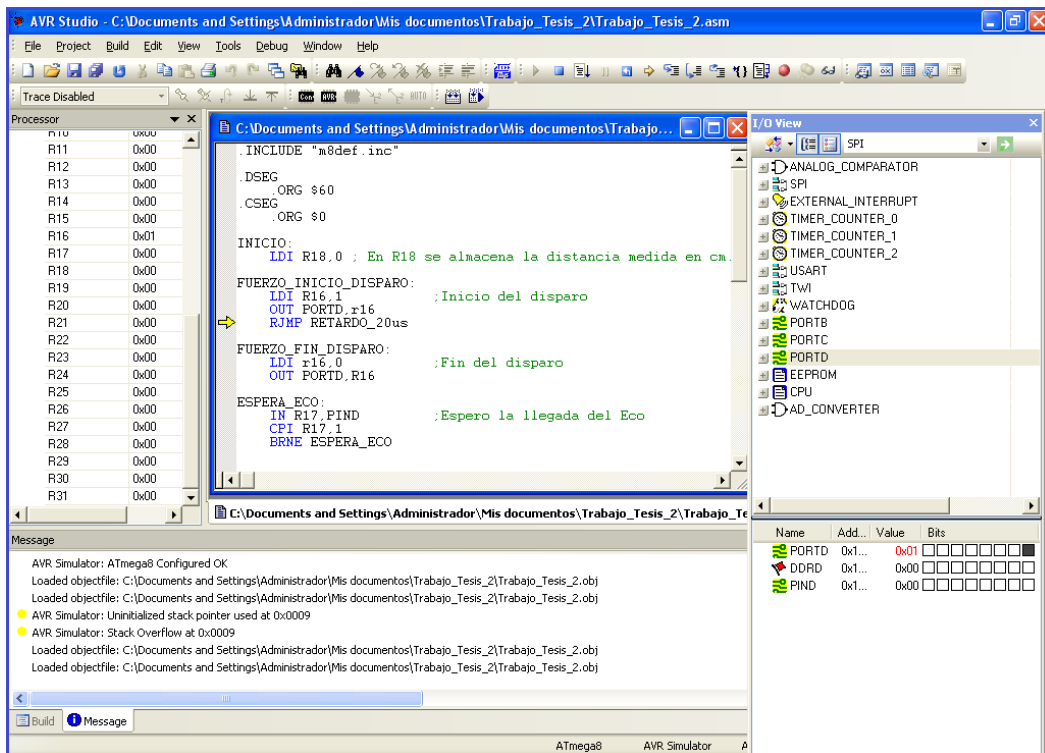


Figura 4.20 Simulación de medición de distancias: Retardo de 20 µs

Cuando el programa salga de la subrutina **RETARDO_20µs**, habrán pasado 20 µs durante los cuales el pulso de disparo (pin PD0) ha estado a nivel alto. Luego de salir de la subrutina **RETARDO_20µs**, tal como se muestra en la Figura 4.21, el programa continúa con la ejecución de las siguientes instrucciones:

```
LDI R16, 0
OUT PORTD, R16
```

Estas instrucciones pondrán a nivel bajo el pin PD0. Es decir, con esto lograremos que el pulso de disparo pase de nivel alto a nivel bajo, y de esta forma el pulso de disparo habrá tenido una duración aproximada de 20 μ s (no se contabilizan los tiempos de transmisión y ejecución de las instrucciones de lenguaje ensamblador en el microcontrolador, por ser tiempos despreciables).

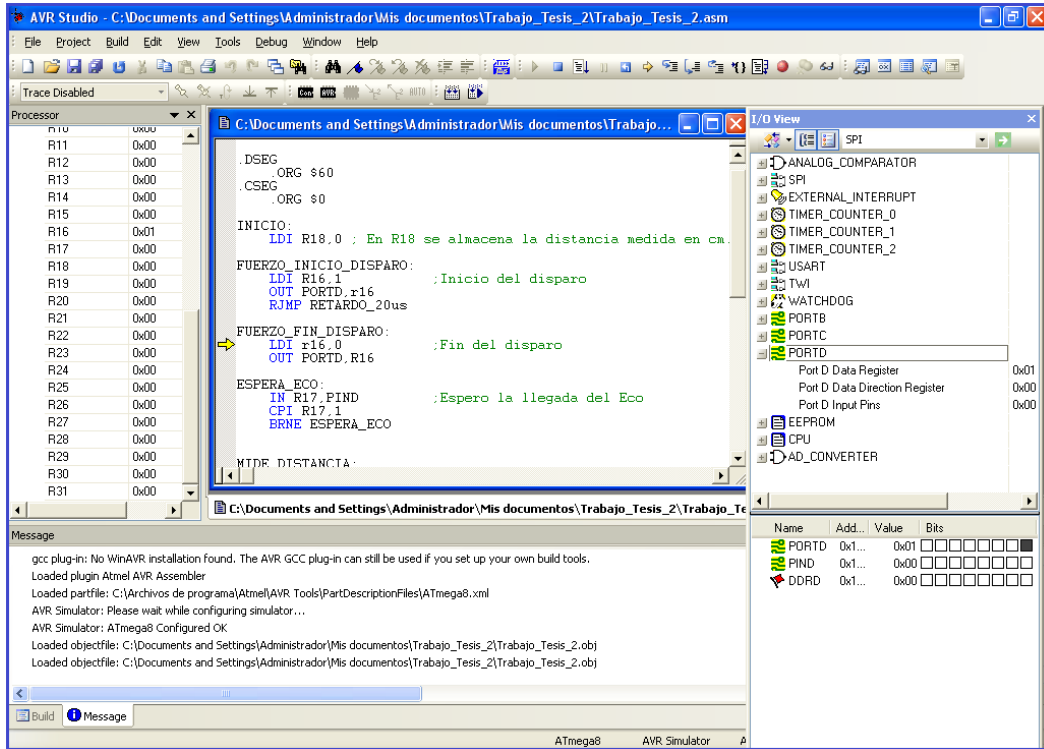


Figura 4.21 Simulación de medición de distancias: Fin de pulso de disparo

Hasta este punto se ha simulado el pulso de disparo analizando el avance de ejecución de las instrucciones en lenguaje ensamblador. Luego, lo que sucede es que el sensor SRF04 emite una ráfaga de ultrasonidos que chocan contra la superficie del tronco y rebotan, regresando en forma de eco hacia el sensor SRF04.

Siguiendo lo expuesto en el párrafo anterior, el programa en lenguaje ensamblador pasa ahora a ejecutar el bucle **ESPERA_ECO**, tal como se muestra en la Figura 4.22.

ESPERA_ECO:

```

IN    R17, PIND
CPI  R17, 1
BRNE ESPERA_ECO
    
```

Estas instrucciones se encargan de leer reiteradamente el nivel de voltaje que aparece en el pin PD0 del microcontrolador, y comparan la lectura obtenida en dicho pin con un '1' lógico.

Si como resultado de la comparación se obtuviera una igualdad, ello indicará que en el pin PD0 ha llegado el pulso de eco. De lo contrario, significará que el pulso de eco aún no ha llegado al pin PD0, por lo cual el bucle se reiniciará. Este proceso se repetirá hasta la llegada del pulso de eco al pin PD0, evento que hará que la ejecución del programa salga del bucle de espera denominado **ESPERA_ECO**.

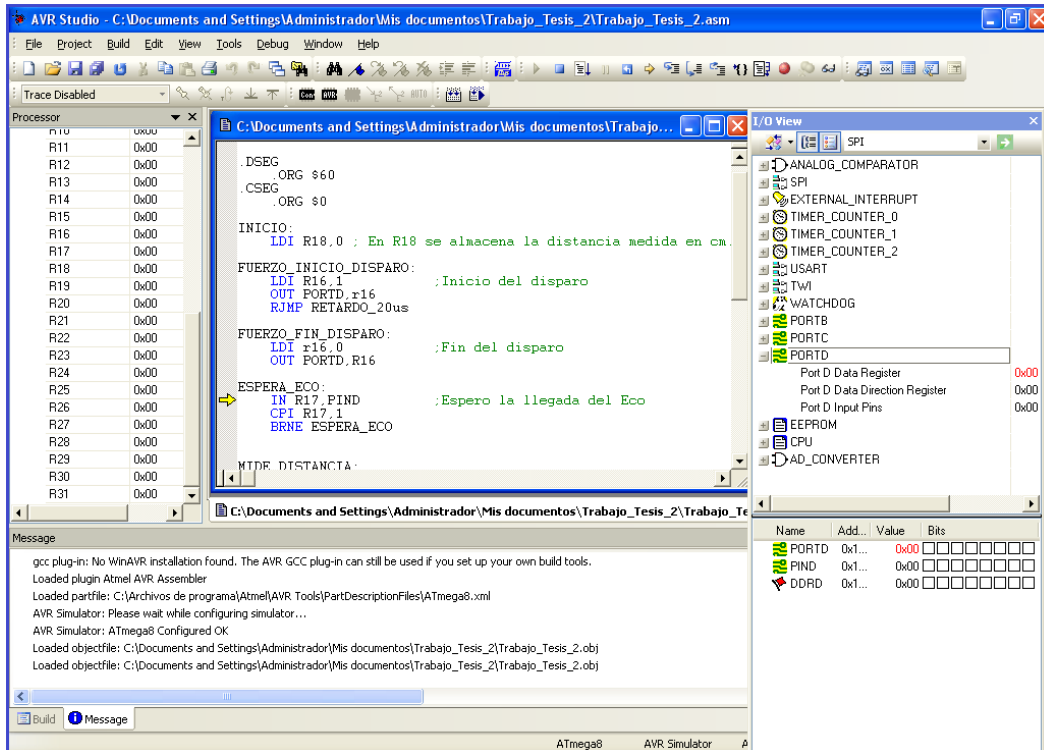


Figura 4.22 Simulación de medición de distancias: Espera llegada de eco

Luego, tal como se muestra en la Figura 4.23, simulamos la llegada del pulso de eco poniendo a '1' el pin PD0. Logramos esto simplemente haciendo clic sobre el primer recuadro de la derecha de PIND (veremos que el recuadro se tornará de color negro, indicando que el pin PD0 pasó a tener el valor de '1' lógico).

Ahora, dado que 60 μ s de duración de pulso de eco equivalen a un centímetro de distancia medida, debemos medir el número de periodos de 60 μ s que están contenidos en el total de duración del pulso de eco; y dicho número será la distancia medida por el sensor SRF04 en centímetros.

Para simular el periodo base de 60 μ s (equivalentes a un centímetro de distancia medida), como se aprecia en la Figura 4.23, el programa llama a la subrutina

RETARDO_60µs, que produce un retardo de 60 µs por cada iteración de la subrutina.

RCALL RETARDO_60us

Como resulta lógico, el programa deberá encargarse de calcular el número de iteraciones que la subrutina **RETARDO_60µs** es ejecutada desde el inicio hasta el término del pulso de eco, ya que dicho número nos indicará la distancia medida por el sensor ultrasonido SRF04 en centímetros.

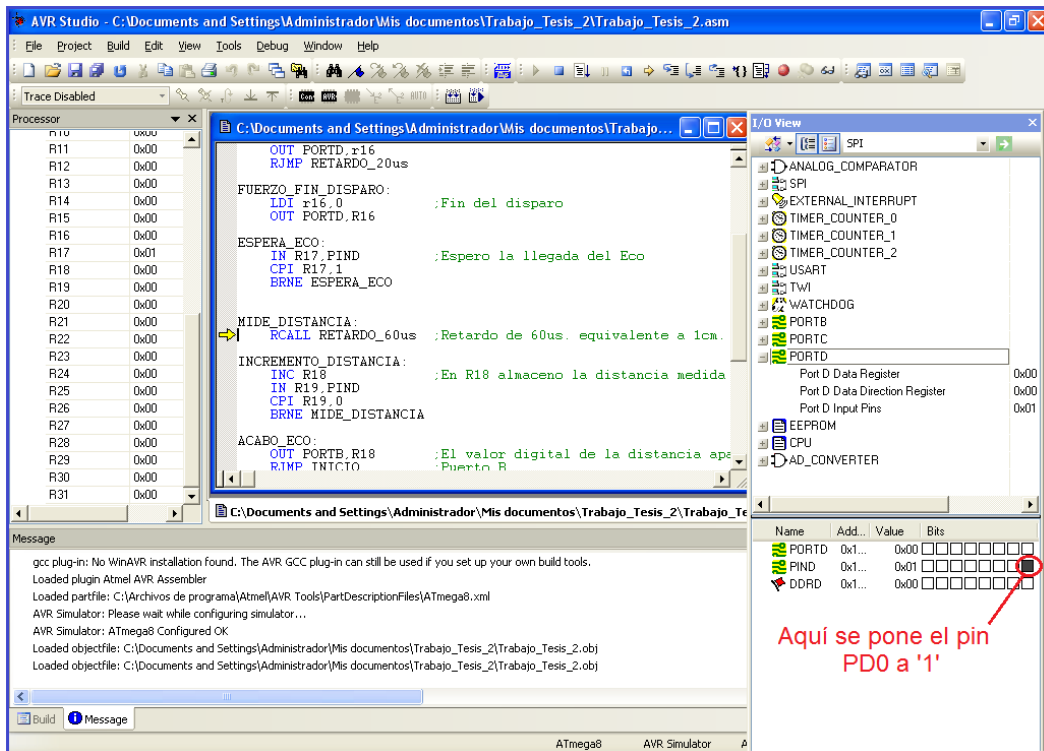


Figura 4.23 Simulación de medición de distancias: Retardo de 60 µs

Para poder contabilizar el número de periodos de 60 µs contenidos en el pulso de eco, en el programa escrito en lenguaje ensamblador se implementó el bucle **INCREMENTO_DISTANCIA** que se detalla a continuación, y que se muestra en la Figura 4.24.

INCREMENTO_DISTANCIA:

```

INC R18
IN R19, PIND
CPI R19, 0
BRNE MIDE_DISTANCIA
    
```

En primer lugar se toma el registro R18, al cual al inicio del programa (ver Figura 4.19) se le asignó el valor inicial de 0 con la instrucción **LDI R18, 0**

Este registro R18 almacenará el número de periodos de 60 μ s contenidos en el pulso de eco. Luego, dado que llegado este punto ya pasó un tiempo base de 60 μ s (pues ya se ejecutó una vez la subrutina **RETARDO_60 μ s**) la primera instrucción (**INC R18**) incrementa el valor del registro R18 en una unidad.

La segunda instrucción (**IN R19, PIND**) coloca en el registro R19 la lectura del puerto D; es decir, coloca en el registro R19 el valor de la lectura en el pin PD0.

A continuación, la tercera instrucción (**CPI R19, 0**) compara la lectura en el pin PD0 con 0, y si como resultado de esta comparación se obtuviera una desigualdad (es decir, que el pin PD0 sigue a nivel alto y el pulso de eco aún no ha terminado) la cuarta instrucción (**BRNE MIDE_DISTANCIA**) envía a ejecutar nuevamente la subrutina **RETARDO_60 μ s** para medir otro periodo adicional de 60 μ s.

Este bucle se repetirá hasta que el pulso de eco haya terminado, evento que hará que la ejecución del programa salga del bucle **INCREMENTO_DISTANCIA**.

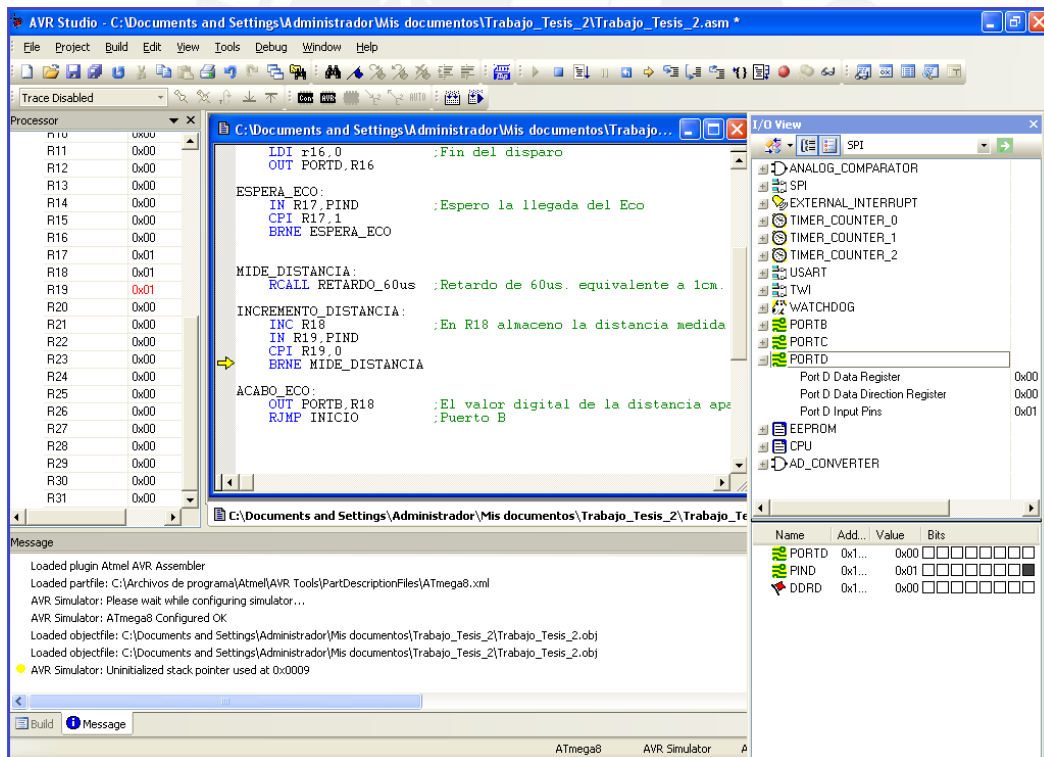


Figura 4.24 Simulación de medición de distancias: Incremento de distancia

Luego, tal como se muestra en la Figura 4.25, simulamos el fin del pulso de eco poniendo a '0' el pin PD0. Logramos esto simplemente haciendo clic nuevamente sobre el primer recuadro de la derecha de PIND (veremos que el recuadro cambiará

de color negro a color blanco, indicando que el pin PD0 pasó a tener el valor de '0' lógico).

Dado que el pulso de eco ha terminado, cuando el programa ejecute la próxima iteración del bucle **INCREMENTO_DISTANCIA** y llegue a la instrucción de comparación de PD0 con el '0' lógico (ver ubicación de la flecha amarilla en la Figura 4.25) se obtendrá una igualdad, por lo que la siguiente instrucción del programa (**BRNE MIDE_DISTANCIA**) ya no enviará la ejecución del programa para un nuevo retardo de 60 μ s (mediante la subrutina **RETARDO_60 μ s**), sino que hará que el programa salga del bucle y pase a la siguiente instrucción.

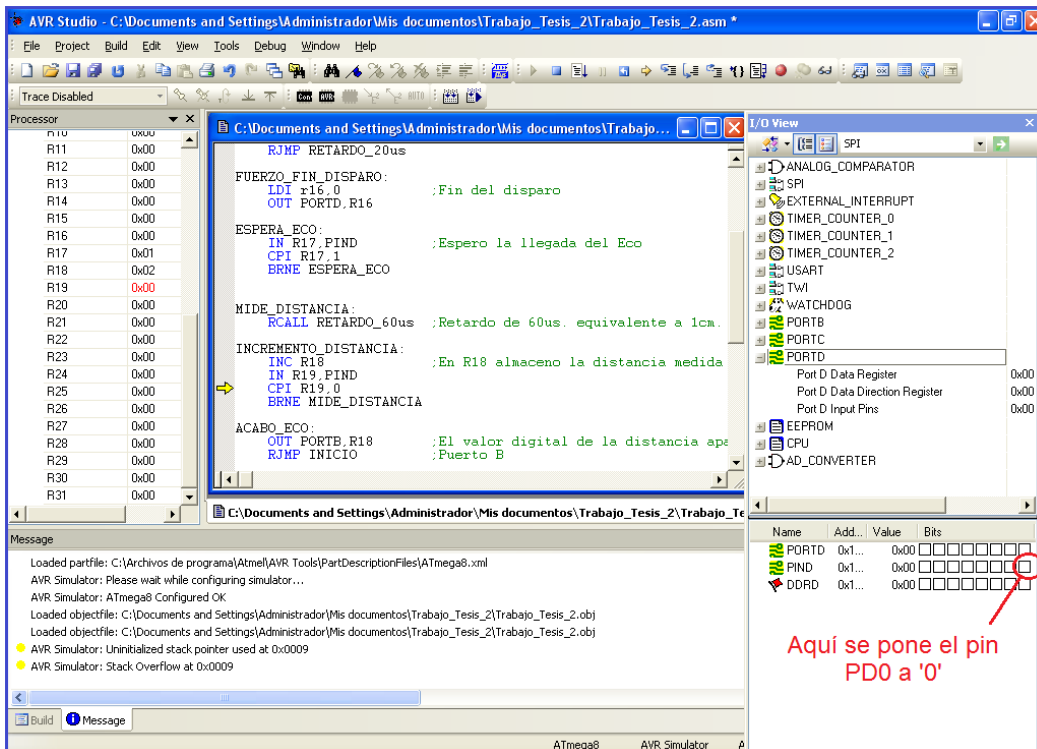


Figura 4.25 Simulación de medición de distancias: Fin de pulso de eco

Llegado este punto, ya se han simulado tanto el pulso de disparo como el pulso de eco en lenguaje ensamblador. Cabe señalar que, para simplificar el proceso de simulación, la ejecución del programa se hizo pasar cuatro veces por la subrutina **RETARDO_60 μ s**. Esto implica que la duración del pulso de eco sería de aproximadamente 240 μ s (no se contabilizan los tiempos de transmisión y ejecución de las instrucciones de lenguaje ensamblador en el microcontrolador, por ser tiempos despreciables).

Como se mencionó anteriormente, el número de periodos de 60 μ s contenidos en el pulso de eco nos da la distancia medida por el sensor en centímetros. Este número de periodos es precisamente el número de veces que el programa haya llamado a la subrutina **RETARDO_60 μ s**, que sabemos es cuatro y que fue almacenado en el registro R18. Esto lo podemos verificar en la Figura 4.26, donde en la parte izquierda vemos los nombres de todos los registros y los valores que almacenan (como se puede notar, el registro R18 aparece con un valor decimal de 0x04).

Siguiendo con el análisis del programa en lenguaje ensamblador, al haber terminado el pulso de eco, la ejecución del programa pasa a las instrucciones que se muestran a continuación.

ACABO_ECO:

```
OUT PORTB, R18
RJMP INICIO
```

Como ya hemos señalado, en el registro R18 se almacena la distancia medida por el sensor, en centímetros. La primera instrucción (**OUT PORTB, R18**) se encarga de mostrar en el puerto B del microcontrolador el contenido del registro R18 como un número binario.

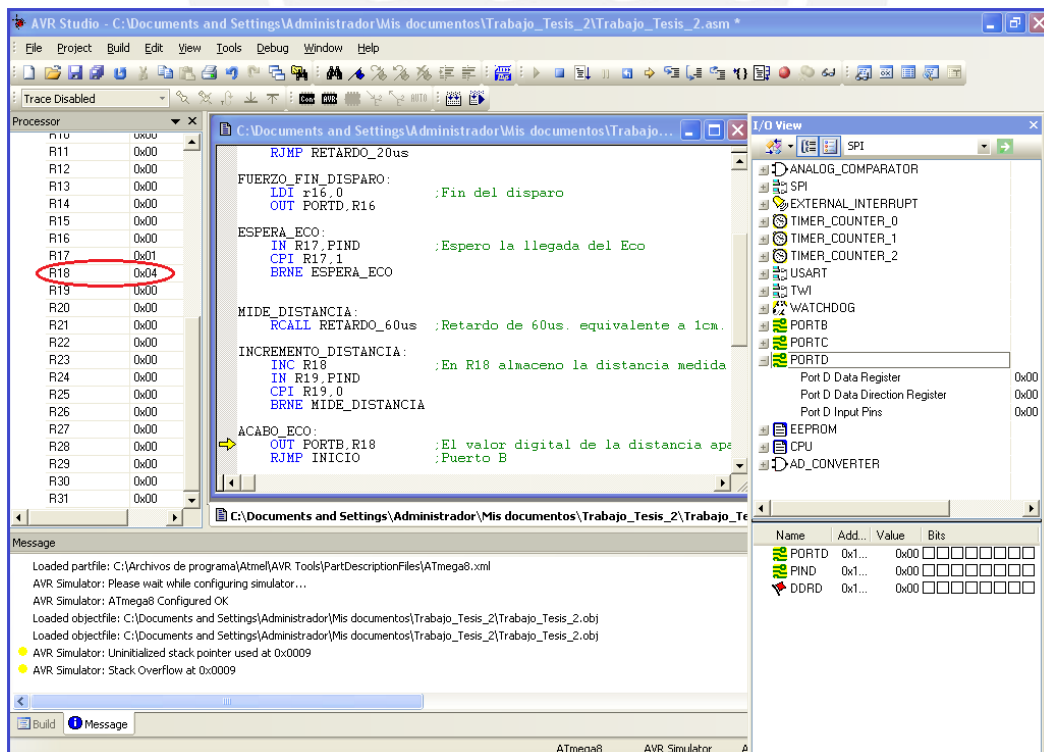


Figura 4.26 Simulación de medición de distancias: Duración pulso de eco

En la parte inferior derecha de la Figura 4.27 se muestra la salida en el puerto B, donde un cuadrado blanco simboliza un '0' y un cuadrado negro simboliza un '1'. De esta forma, la salida en el puerto B expresado en formato binario sería: '00000100', que equivale a '4' en numeración decimal. Es este el valor (en centímetros) medido por el sensor ultrasonido SRF04.

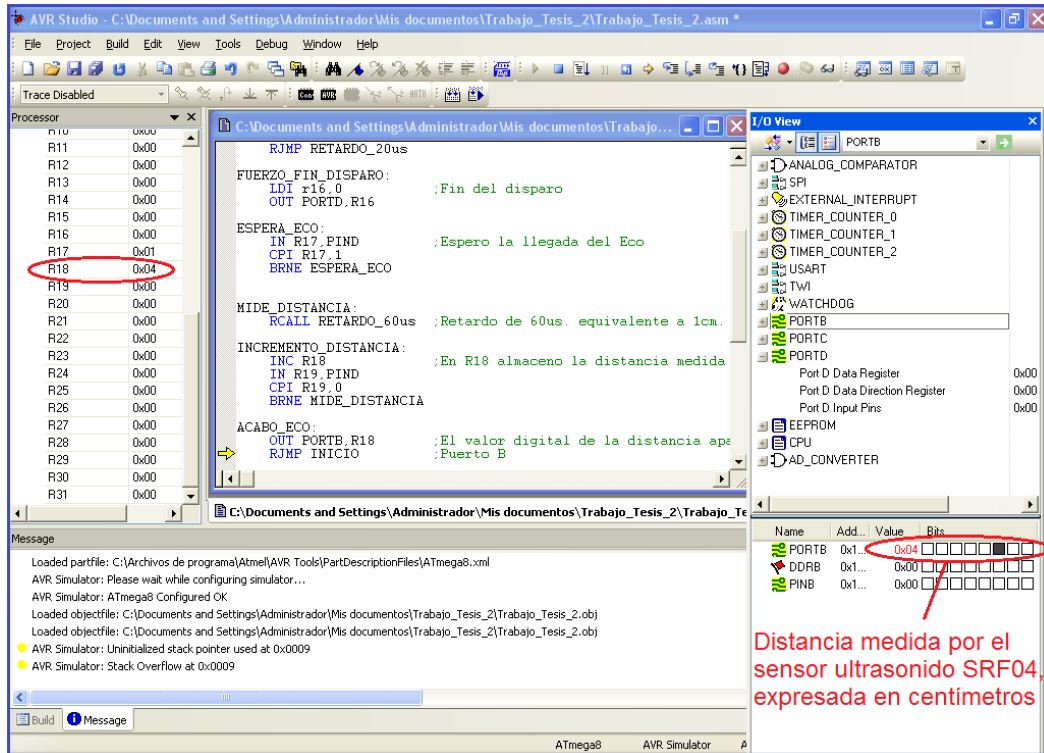


Figura 4.27 Simulación de medición de distancias: Distancia medida en cm

Hasta aquí, utilizando el entorno AVR Studio 4 se ha simulado satisfactoriamente lo siguiente en lo que respecta a la medición de distancias:

- ✓ Pulso de disparo de 20 μ s (debido a que, según las especificaciones técnicas del sensor ultrasonido SRF04, el pulso de disparo debe tener una duración mínima de 10 μ s).
- ✓ Pulso de eco de 240 μ s (equivalentes a una medición de distancia de 4 cm efectuada por el sensor ultrasonido SRF04).
- ✓ Muestra del valor de distancia medida por el sensor (en centímetros) expresado como un número binario por el puerto B del microcontrolador. Para el caso de la simulación efectuada, por el puerto B se mostró la salida digital (PB7...PB0) = '00000100' (o 4 en numeración decimal).

4.6 Simulación de la comunicación PC - Microcontrolador

Una vez obtenida la distancia medida por los sensores ultrasonido como una palabra digital en uno de los puertos del microcontrolador (tal como se vio en el proceso simulación anterior, donde la distancia medida aparece en formato binario por el puerto B), esta información debe ser transmitida del microcontrolador a la PC.

Antes de pasar a mostrar las simulaciones de la comunicación entre la PC y el microcontrolador PIC16F84A, cabe recordar los siguientes puntos, vistos en la sección 3.5.4 Interface de comunicación (página 36 del presente trabajo de tesis):

- ✓ La PC cuenta con un puerto serie DB-9, que trabaja con lógica negativa (norma RS232), de la siguiente forma:
 - Un voltaje entre +3V y +15V representa el nivel lógico '0'.
 - Un voltaje entre -3V y -15V representa el nivel lógico '1'.
- ✓ El microcontrolador PIC16F84A trabaja con niveles TTL (lógica positiva), de la siguiente forma:
 - Un voltaje entre 0V y +0.8V representa el nivel lógico '0'.
 - Un voltaje entre +2V y +5V representa el nivel lógico '1'.
- ✓ Para la comunicación entre PC y microcontrolador PIC16F84A se coloca como intermediario el circuito integrado MAX232, el cual convierte los niveles RS232 (lógica negativa) en niveles TTL (lógica positiva), y viceversa.
- ✓ En la comunicación entre PC y microcontrolador PIC16F84A, la PC utiliza únicamente tres pines de su puerto serie DB-9: pin de transmisión (TxD), pin de recepción (RxD) y pin de tierra (GND).
Del mismo modo, para la comunicación entre PC y microcontrolador PIC16F84A, el microcontrolador PIC16F84A utiliza básicamente dos pines: un pin para la transmisión de datos (Tx) y otro pin para la recepción de datos (Rx). Sin embargo, además de estos dos pines, también hay que considerar otros pines del microcontrolador, como el pin de alimentación y el pin de tierra, además de los pines del puerto donde aparecerá la distancia medida (en centímetros) por el sensor ultrasonido SRF04.

En la Figura 4.28 a 4.30, se muestran las simulaciones de la comunicación entre la PC y el microcontrolador PIC16F84A utilizando el software Proteus. Únicamente con la finalidad de facilitar la realización y comprensión de estas simulaciones, se realizaron las siguientes simplificaciones:

- ✓ Se mostrará una pantalla de fondo negro denominada Virtual Terminal, que simulará el HyperTerminal de la computadora (PC).

- ✓ Se omitirá el circuito integrado MAX232, ya que el elemento Virtual Terminal del simulador es capaz de comunicarse con el microcontrolador PIC16F84A sin necesidad de un intermediario.
- ✓ Para facilitar el proceso de simulación, la distancia medida por el sensor ultrasonido SRF04 (en centímetros), aparecerá como un número binario en sólo cuatro pines del puerto B del microcontrolador PIC16F84A: (PB7...PB4).

En la Figura 4.28 se muestra el primer circuito que simula la comunicación entre la PC y el microcontrolador PIC16F84A. En este circuito debemos notar que el nivel TTL de cada pin (PB7...PB4) es controlado por el estado de un interruptor. Así, cuando el interruptor esté abierto el pin correspondiente tendrá un nivel alto TTL (cuadrado de color rojo sobre dicho pin). Del mismo modo, cuando el interruptor esté cerrado el pin correspondiente tendrá un nivel bajo TTL (cuadrado de color azul sobre dicho pin), pues el interruptor conectará el pin correspondiente a tierra.

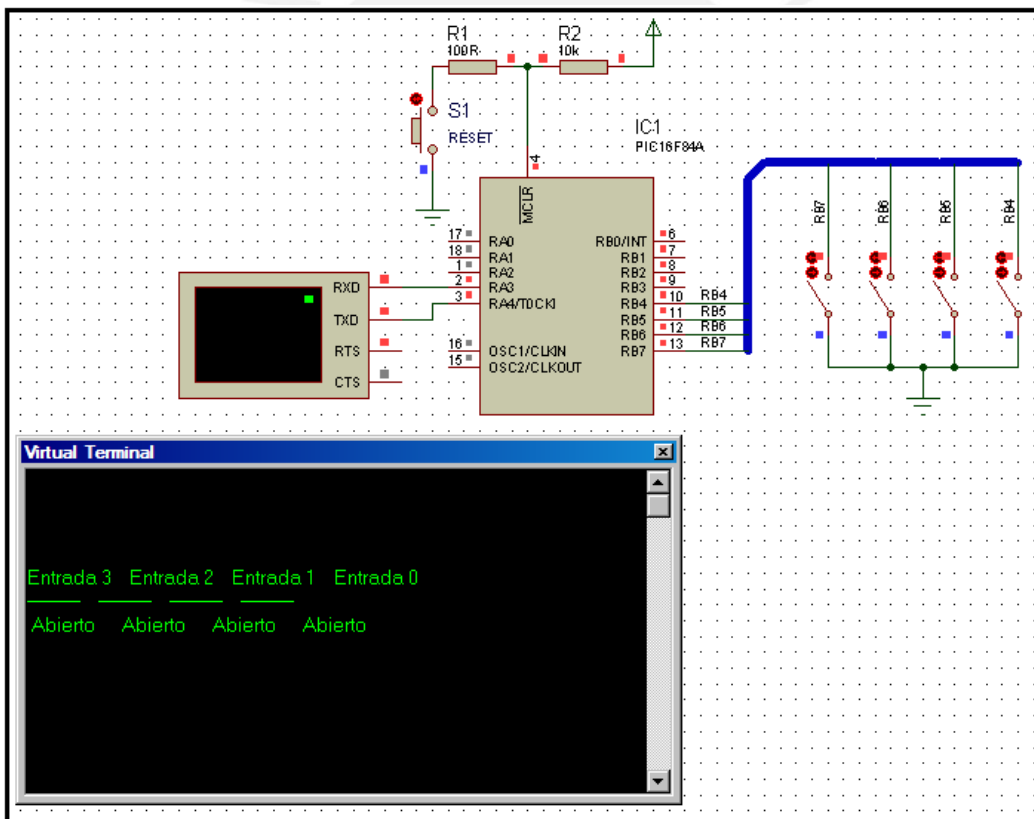


Figura 4.28 Simulación de comunicación PC - Microcontrolador: Transmisión de palabra digital ‘1111’ [39]

Analizando el circuito mostrado en la Figura 4.28, vemos que los cuatro interruptores conectados a los pines (RB7...RB4) del microcontrolador PIC16F84A se encuentran

abiertos, por lo que los cuatros pines tendrán un nivel alto TTL, formando la palabra digital '1111', que simula la medición efectuada por el sensor ultrasonido SRF04.

Esta palabra digital es transmitida por el pin RA3 del microcontrolador PIC16F84A, y esta información ingresa por el terminal RxD del Virtual Terminal. Luego, en el Virtual Terminal se muestra el estado de cada uno de los interruptores conectados a los pines (PB7...PB4): 'Abierto, Abierto, Abierto, Abierto', que equivale a la palabra digital (PB7...PB4) = '1111'.

Aquí cabe señalar que las distancias medidas por el sensor (en forma de palabras digitales) que ingresan al Virtual Terminal serían nuevamente procesadas en la PC mediante un programa desarrollado en Visual Basic, para obtener finalmente la forma volumétrica del tronco de madera o rollo.

En la Figura 4.29 se muestra una segunda simulación de la comunicación entre la PC y el microcontrolador PIC16F84A. Ver que se ha cerrado el interruptor conectado al pin RB4 del microcontrolador, lo cual conectará este pin a tierra, por lo que ahora tendrá un nivel bajo TTL (cuadrado de color azul sobre el pin RB4). Los demás interruptores conectados a los pines RB7, RB6, RB5 siguen abiertos, por lo que estos pines continuarán a un nivel alto TTL (cuadrado de color rojo sobre dichos pines).

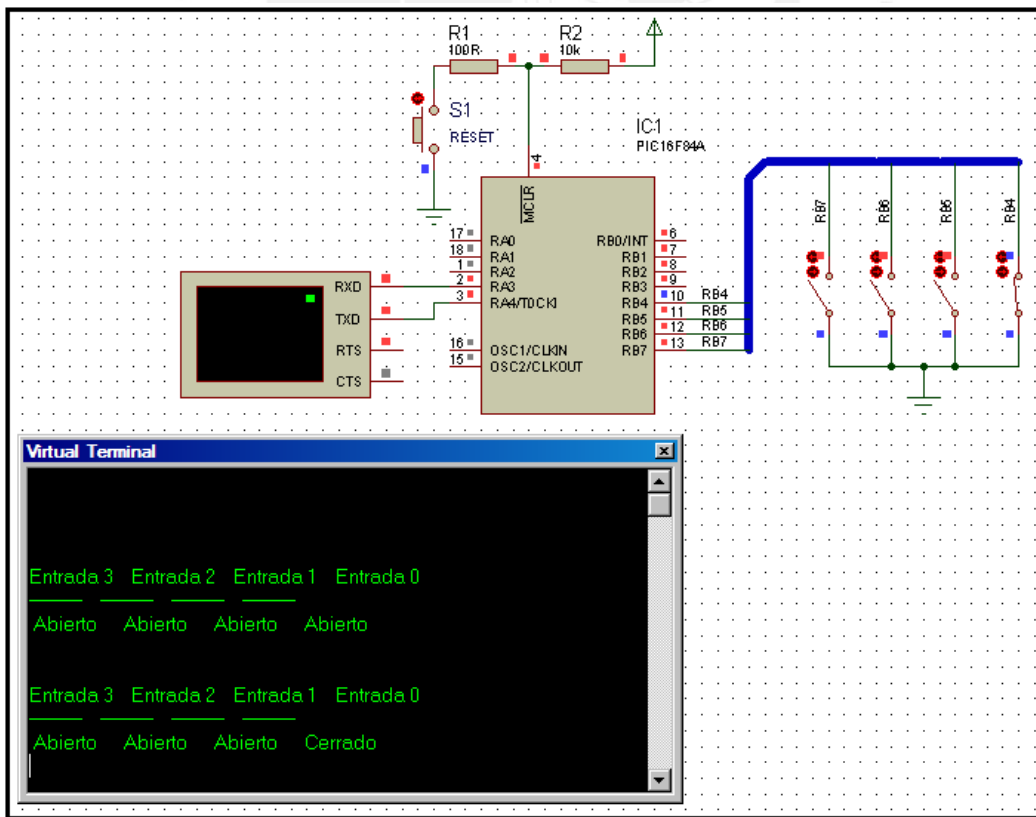


Figura 4.29 Simulación de comunicación PC - Microcontrolador: Transmisión de palabra digital '1110' [39]

Notar cómo en la Figura 4.29, en el Virtual Terminal aparece el estado de los interruptores conectados a los pines (PB7...PB4): 'Abierto, Abierto, Abierto, Cerrado', que equivale a la palabra digital (PB7...PB4) = '1110'.

De esta forma, se puede seguir cambiando el estado de los interruptores conectados a los pines (PB7...PB4), y veremos que este estado se transmitirá al Virtual Terminal, asociando un '1' lógico para un interruptor abierto y un '0' lógico para un interruptor cerrado.

En la Figura 4.30 se muestra una tercera simulación de la comunicación entre la PC y el microcontrolador PIC16F84A. Ver que se han cerrado los interruptores conectados a los pines RB4, RB5 del microcontrolador, lo cual conectará estos pines a tierra, por lo que ahora tendrán un nivel bajo TTL (cuadrado de color azul sobre los pines RB4, RB5). Los otros dos interruptores conectados a los pines RB7, RB6 siguen abiertos, por lo que estos pines continuarán a un nivel alto TTL (cuadrado de color rojo sobre dichos pines). Luego, en el Virtual Terminal aparece el estado de los interruptores conectados a los pines (PB7...PB4): 'Abierto, Abierto, Cerrado, Cerrado', que corresponde a la palabra digital (PB7...PB4) = '1100'.

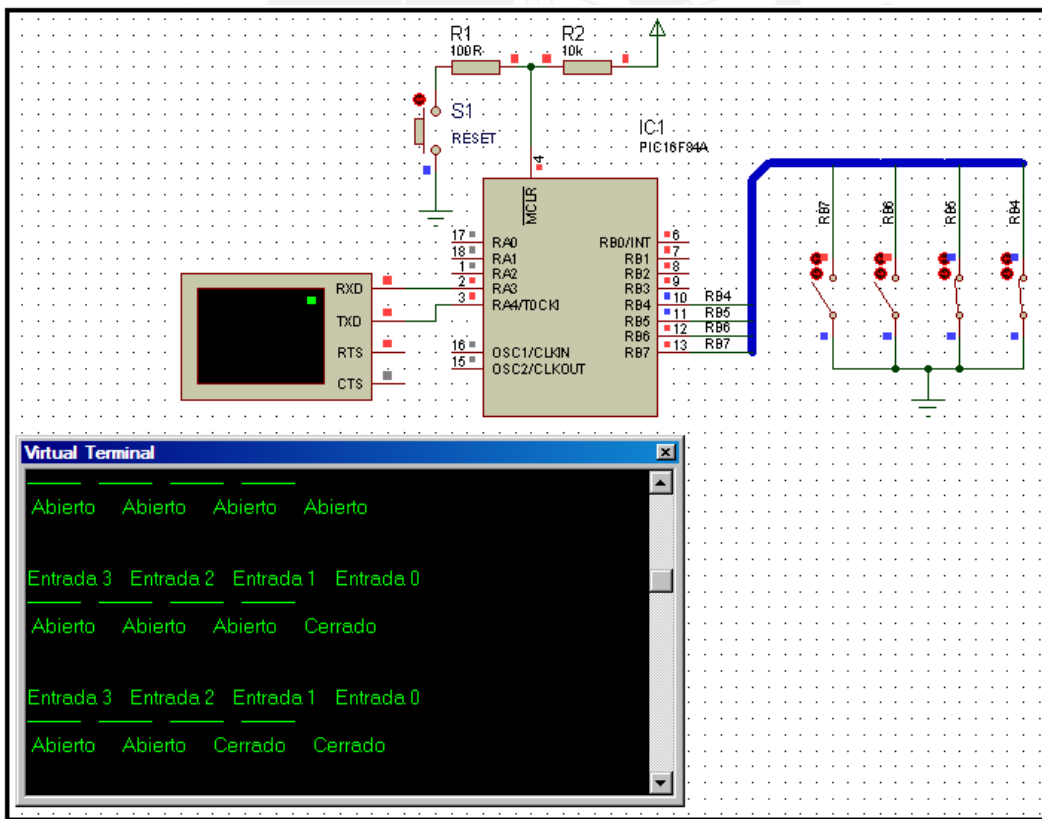


Figura 4.30 Simulación de comunicación PC - Microcontrolador: Transmisión de palabra digital '1100' [39]

Cabe recordar que para mayores referencias acerca de las simulaciones de circuitos mediante el uso del software Proteus resulta conveniente revisar el trabajo de Palacios, Remiro y Lucas [21,22,23,24,25,39].

4.7 Simulación del gráfico del rollizo en Visual Basic

Antes de pasar a mostrar las simulaciones del gráfico del tronco de madera o rollizo en Visual Basic, resulta conveniente recordar el algoritmo expuesto en la sección 3.6.2d Algoritmo base para el gráfico volumétrico del rollizo (página 49 del presente trabajo de tesis), en donde se anotó lo siguiente:

- Si deseamos graficar una línea con una forma determinada (no necesariamente recta), ésta se puede obtener a partir de una sucesión de puntos (cada uno de los cuales es graficado en base a su ubicación en un eje de coordenadas).
- Si lo que deseamos graficar es un plano, éste se puede obtener a partir de una sucesión de rectas paralelas colocadas una a continuación de la otra.
- Si lo que queremos graficar es un cilindro recto de radio 'R', éste se puede obtener a partir de una sucesión de circunferencias de radio 'R' colocadas una a continuación de la otra.

Siguiendo este mismo razonamiento, si lo que deseamos graficar es la superficie de un tronco de madera, podemos suponer que ésta es una superficie cilíndrica (aunque ya no una superficie cilíndrica regular, obviamente). Para este caso en particular, ya no se tratarían de circunferencias colocadas una a continuación de otra, sino de una especie de anillos de forma irregular colocados uno a continuación del otro.

En resumen, para poder reconstruir el volumen del tronco de madera, el proceso consistirá en generar una especie de malla volumétrica tridimensional a partir del conjunto de anillos irregulares obtenidos de forma individual.

El programa en Visual Basic utilizará el algoritmo antes descrito. Así, en la Figura 4.31 se muestra uno de estos anillos, construido con sólo seis muestras (asumiendo que los sensores ultrasonido tomaron datos cada desplazamiento de sesenta grados sexagesimales sobre la plataforma circular). El programa desarrollado en Visual Basic simplemente unirá los puntos de la superficie detectados por los sensores, por lo que con sólo seis muestras el anillo obtenido es un hexágono.

Resulta claro que a medida que se incremente el número de muestras tomadas por los sensores ultrasonido, las poligonales a obtener irán asemejándose a la forma real del anillo irregular o sección axial del tronco de madera.

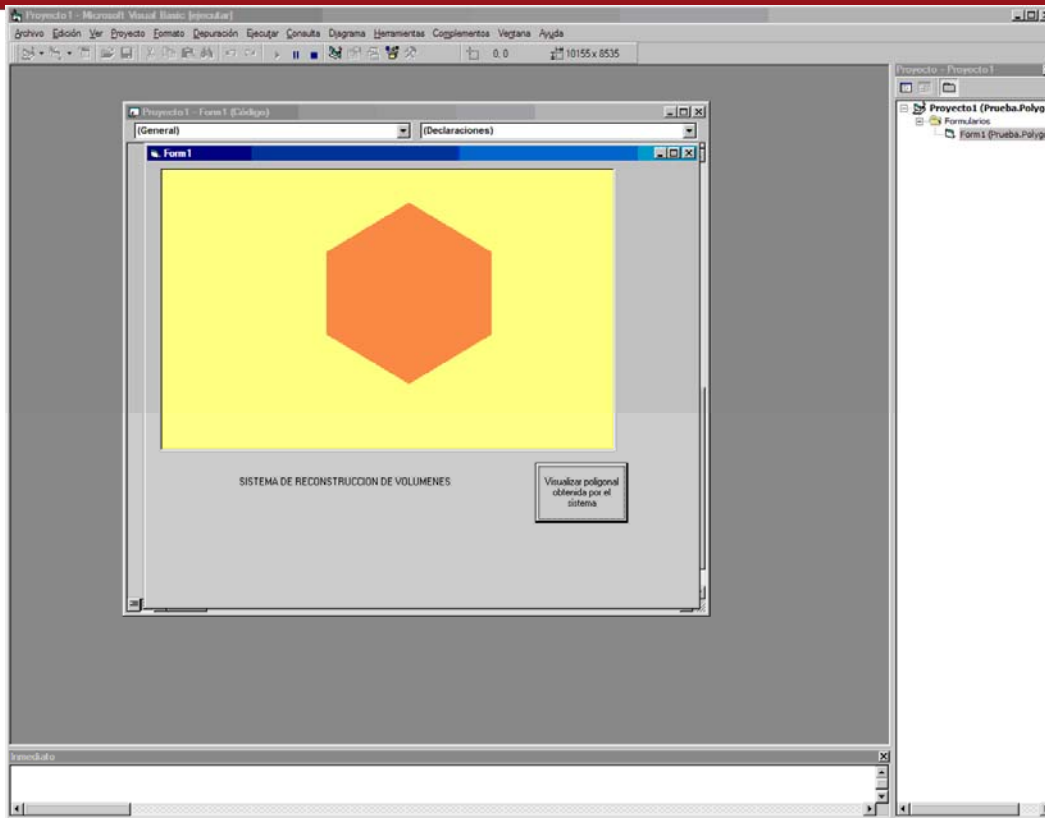


Figura 4.31 Representación de un anillo con una toma de seis muestras

Únicamente para simplificar las siguientes simulaciones de visualización, se decidió asumir que los sensores toman un total de sólo doce muestras por cada anillo. Así, en la Figura 4.32 a 4.35 veremos distintas vistas de una misma superficie cilíndrica en tres dimensiones (superficie obtenida como el resultado de colocar muchos anillos, uno a continuación del otro, en un espacio tridimensional). Asimismo, notar que en la interfaz de usuario se han implementado los siguientes controles:

- **Rotación:** Control que permite efectuar la rotación del sólido en los tres ejes coordenados (x; y; z).
- **Traslado:** Control que permite el desplazamiento del sólido en los tres ejes coordenados (x; y; z).
- **Sombra:** Control que permite variar las tonalidades de sombra del sólido.
- **Zoom:** Control que permite realizar acercamientos o alejamientos para poder visualizar mejor el sólido.
- **Salir:** Control que permite salir de la aplicación

En la Figura 4.32 se muestra una vista frontal de la superficie cilíndrica generada, sin ningún efecto de rotación ni traslación respecto de ningún eje.

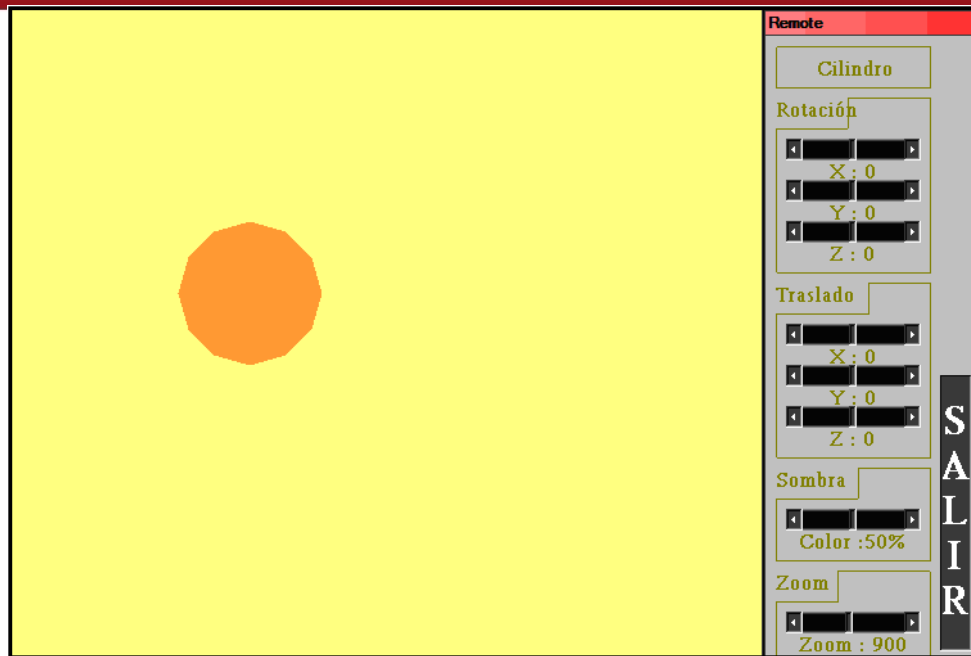


Figura 4.32 Vista frontal de la superficie cilíndrica

En la Figura 4.33 se muestra la imagen tridimensional de la misma superficie cilíndrica mostrada en la Figura 4.32, con la diferencia de que el acercamiento (mediante el control zoom) ha aumentado ligeramente de 900 a 959. Asimismo la superficie cilíndrica tiene un ángulo de rotación de 30 grados sexagesimales respecto al eje 'y', y de -23 grados sexagesimales respecto al eje 'z'.

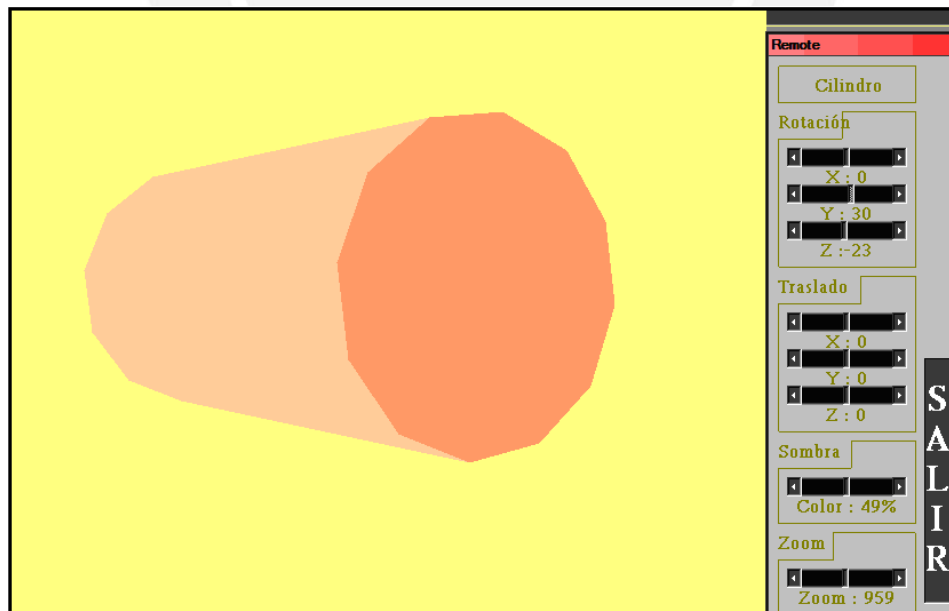


Figura 4.33 Vista de superficie cilíndrica, con rotación de 30° sobre el eje 'y' y de -23° sobre el eje 'z'

En la Figura 4.34 se muestra la imagen tridimensional de la misma superficie, con la diferencia de que el acercamiento (mediante el control zoom) ha regresado al valor de 900. Asimismo la superficie cilíndrica tiene un ángulo de rotación de -47 grados sexagesimales respecto al eje 'y', y de -23 grados sexagesimales respecto al eje 'z'.

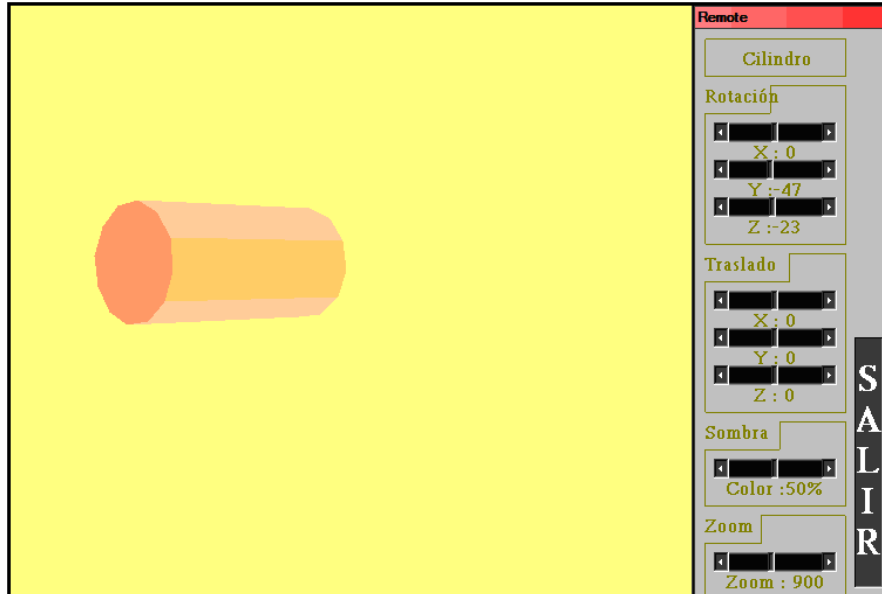


Figura 4.34 Vista de superficie cilíndrica, con rotación de -47° sobre el eje 'y' y de -23° sobre el eje 'z'

Finalmente, en la Figura 4.35 se muestra la imagen tridimensional de la misma superficie cilíndrica que simula el tronco de madera a reproducir, pero esta vez con un zoom de 900, un ángulo de rotación de 40 grados sexagesimales respecto al eje 'x', y cero grados de rotación respecto a los ejes 'y', 'z'.

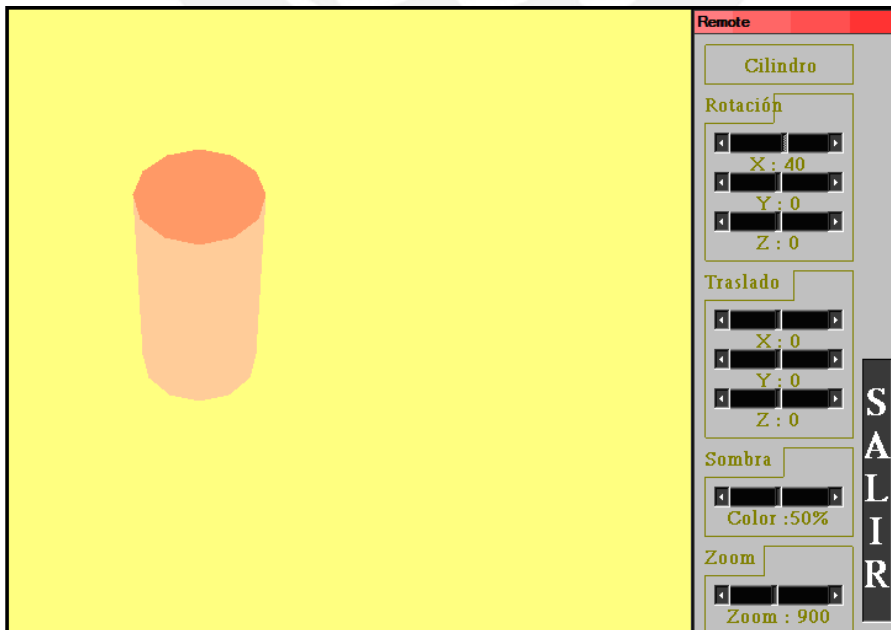


Figura 4.35 Vista de superficie cilíndrica, con rotación de 40° sobre el eje 'x'

Como se pudo observar en las simulaciones de la Figura 4.32 a la Figura 4.35, con sólo doce muestras para reconstruir cada anillo, se obtiene ya una forma cilíndrica del tronco de madera que se desea reproducir. Considerando que según lo expuesto en el cálculo del número de muestras por unidad de longitud (página 44 del presente trabajo de tesis), el número de muestras a tomar por cada anillo variaría entre 16 y 39, podemos afirmar que el sistema de reconstrucción de volúmenes diseñado obtendría imágenes bastante fiables de la forma real del tronco de madera o rollizo.

Para la obtención de los gráficos en Visual Basic, mostrados en la Figura 4.31 a 4.35, resulta de gran utilidad revisar los trabajos de Petroustos [40] y Ceballos [41].

4.8 Análisis de costos

En la Figura 4.36 se muestra el detalle del costo estimado de una eventual implementación del sistema de reconstrucción de volúmenes (diseñado en el presente trabajo de tesis), y para el cual se ha tomado seis meses como el tiempo requerido para la construcción del prototipo.

	Costo en dólares americanos (*)	Costo en nuevos soles
Pago a investigador(es) en las áreas de programación, electrónica y mecánica por servicios prestados por seis meses.	4000	10800.0
Computadora portátil.	1756	4741.2
Costo de licencia por el uso de un entorno de programación visual (costo estimado).	527	1422.9
Sensores ultrasonido.	351	947.7
Motores.	74	199.8
Plataforma circular externa para los sensores ultrasonido.	185	499.5
Plataforma circular interna de desplazamiento para los sensores ultrasonido.	185	499.5
Tarjetas, cables, conectores, adaptadores, fuentes de alimentación, microcontroladores, etc.	370	999.0
Costo por instalación del prototipo en el lugar deseado (transporte, alojamiento, etc.).	370	999.0
Costo de la ingeniería.	5556	15001.2
Total :	US\$ 13374	S/. 36,109.8

(*) Tipo de cambio: (US \$ / Nuevo sol) = 2.7

Figura 4.36 Análisis de costos para implementación de un prototipo del sistema de reconstrucción de volúmenes

Es importante señalar que los costos mostrados en la Figura 4.36 son referenciales y pueden no estar actualizados para la fecha en la cual la parte interesada evalúe la posibilidad de implementar el sistema de reconstrucción de volúmenes.



CONCLUSIONES

1. La implementación del sistema de reconstrucción de volúmenes por ultrasonido, cuyo diseño se detalla en el presente trabajo de tesis, permitiría reproducir de forma fiable la forma volumétrica de troncos de madera. Además del estudio teórico, diseño y simulaciones presentadas en el presente trabajo de tesis, esta afirmación quedó demostrada experimentalmente el año 2007 en el marco del curso Proyecto Electrónico 1 (Especialidad de Ingeniería Electrónica, PUCP), en el que el autor del presente trabajo fue partícipe de la implementación de un prototipo, obteniendo resultados satisfactorios y alineados con los objetivos planteados.
2. Teniendo en consideración el estudio teórico, diseño y simulaciones mostrados en el presente trabajo de tesis, así como la implementación del prototipo construido el año 2007 (mencionado en el párrafo anterior), se concluye que la tecnología ultrasonido constituye una alternativa a tener en cuenta para el proceso de reconstrucción de formas volumétricas. Más aún, los resultados obtenidos abren la posibilidad para probar y/o aplicar la tecnología ultrasonido no sólo para la reconstrucción de volúmenes de troncos de madera, sino también para otros materiales.

RECOMENDACIONES

1. En el diseño planteado en el presente trabajo se usaron solamente dos sensores ultrasonido SRF04. Sin embargo, en una eventual implementación, sería conveniente el uso de un mayor número de estos sensores. Esto mejoraría notablemente la resolución de la imagen obtenida y reduciría la velocidad de procesamiento por cada tronco de madera o rollizo.
2. Complementando el punto anterior, debe tenerse en cuenta que al utilizar un mayor número de sensores, sería necesaria una modificación de la velocidad de toma de datos de los mismos. Asimismo, esto conllevaría también a una modificación de los programas desarrollados en Lenguaje Ensamblador y en Visual Basic.
3. Como se pudo ver en el presente trabajo, usando una palabra digital de tan sólo 4 bits se puede conseguir una regulación bastante aceptable de la velocidad de giro del eje de un motor DC. Más aún, en las simulaciones se mostraron incrementos graduales de sólo 10% en el ciclo de trabajo, lo cual haría un total de 10 incrementos graduales de velocidad (desde 0% hasta 100%). Sin embargo, debe tenerse en cuenta que, en general, si usáramos una palabra binaria de 'n' bits, sería posible obtener un total de 2^n incrementos graduales de ciclo de trabajo.
De esta forma, para un byte ($n=8$), podríamos obtener hasta 256 incrementos graduales del ciclo de trabajo (desde 0% hasta 100%), lo cual se traduciría en 256 incrementos graduales de la velocidad de giro del eje del motor DC. Es decir, utilizando los ocho bits de un byte obtendríamos una regulación de velocidad mucho más precisa.
4. En el presente trabajo, para modelar el sistema de reconstrucción de volúmenes se asumió un diámetro máximo de los troncos de madera igual a 25 cm. Sin embargo, en una eventual implementación del sistema, el diámetro máximo de los troncos de madera (rollizos) sería probablemente mucho mayor y por lo tanto habría que hacer ciertas adaptaciones al diseño.
En primer lugar, como puede parecer obvio, se debería construir una plataforma circular acorde con las dimensiones de los troncos de madera a ser procesados. Luego, se debe modificar la velocidad de desplazamiento de los sensores sobre la nueva plataforma circular (esto mediante la variación del

ciclo de trabajo de una señal PWM, el mismo que puede ser controlado fácilmente por una palabra digital, tal como se explicó en el Capítulo 3).

Por último, debería modificarse también la velocidad de toma de datos de los sensores ultrasonido (dependiendo de la resolución de la forma volumétrica que se desee obtener del tronco de madera), así como los programas en Lenguaje Ensamblador y Visual Basic.



BIBLIOGRAFÍA

- [1] J. Vilaboa. "Oportunidades para la incorporación de software que optimice el corte de rollizos en los aserraderos chilenos", Revista Facultad de Ingeniería, Universidad de Tarapacá, Chile, Num. 4, Enero-Diciembre, 1997.
- [2] J. Vilaboa. "Optimización del rendimiento del material en el corte de piezas", Serie Apuntes de Clases, Departamento de Ingeniería de Sistemas, Universidad de Concepción, Chile, Agosto, 1992.
- [3] J. Vilaboa. "Sistema para optimizar el corte de rollizos de gran diámetro", Actas del Segundo Encuentro Chileno de Computación, Concepción, Chile, Octubre, 1994.
- [4] J. Vilaboa. "Sistema de apoyo a un eficiente corte de rollizos de gran diámetro a través de procesamiento de imágenes", Documento Técnico, Departamento de Ingeniería de Sistemas, Universidad de Concepción, Concepción, Chile, Enero, 1992.
- [5] J. Vilaboa. "El problema del corte de piezas en las industrias", Revista del Colegio de Ingenieros, Consejo Regional de Biobío, Concepción, Chile, Num.5, pp.9-11, Abril, 1992.
- [6] M. Gutierrez. "La madera y su proceso de corte. I Parte", Suplemento de la Revista Chile Forestal, Num.82, Santiago de Chile, Julio, 1982.
- [7] M. Gutierrez. "La madera y su proceso de corte. II Parte", Suplemento de la Revista Chile Forestal, Num.83, Santiago de Chile, Agosto, 1982.
- [8] Sensor reflexivo GP2DXX. X-Robotics: Robótica & μ C PIC [online]. Disponible en: <http://www.x-robotics.com/sensores.htm#gp2d>
- [9] Compact, High Sensitive Distance Measuring Sensor Sharp GP2D2. Datasheet Catalog [online]. Disponible en: <http://pdf.datasheetcatalog.com/datasheet/Sharp/mXvryzu.pdf>
- [10] D. Pérez de Diego. Sensores de distancia por ultrasonidos. PicManía by RedRaven [online]. Disponible en: http://picmania.garcia-cuervo.net/recursos/redpictutorials/sensores/sensores_de_distancias_con_ultrasonidos.pdf

[11] A. Álvarez. Generación y detección por ultrasonido [online]. México: Universidad de las Américas de Puebla. Disponible en:
catarina.udlap.mx/u_dl_a/tales/documentos/lep/alvarez_p_af/capitulo2.pdf

[12] O. Alonso. (2006, Marzo). Ejemplo de sensor medidor de distancias SRF04. Universidad de Alcalá, Departamento de Electrónica: Seminario de Diseño y Construcción de Microrrobots [online]. Disponible en:
<http://www.alcabot.com/alcabot/seminario2006/Trabajos/OscarAlonsoGarrido.pdf>

[13] (2014, Enero). Sensor de distancias por ultrasonidos SRF04. SuperRobótica [online]. Disponible en: <http://www.superrobotica.com/S320110.htm>

[14] (2012, Enero). Cómo hacer un encoder óptico usando partes de un mouse mecánico. Technoreeze [online]. Disponible en:
<http://www.technoreeze.com/2012/01/25/como-hacer-un-encoder-optico-usando-partes-de-un-mouse/>

[15] Visual C++ en Visual Studio 2013. Microsoft Developer Network [online]. Disponible en: <http://msdn.microsoft.com/es-es/library/60k1461a.aspx>

[16] Visual Basic. Microsoft Developer Network [online]. Disponible en:
[http://msdn.microsoft.com/es-es/library/2x7h1hfk\(v=vs.100\).aspx](http://msdn.microsoft.com/es-es/library/2x7h1hfk(v=vs.100).aspx)

[17] MatLab, el lenguaje de cálculo técnico. MathWorks [online]. Disponible en:
<http://www.mathworks.es/products/matlab/>

[18] L. Salazar. Figura realizada para el curso: Proyecto Electrónico 1, Facultad de Arte, Pontificia Universidad Católica del Perú, Lima, 2007.

[19] (2006). Smart Sensors and Applications. Parallax Inc. [online]. Disponible en:
<http://www.parallax.com/sites/default/files/downloads/28029-Smart-Sensors-Text-v1.0.pdf>

[20] Devantech SRF04 Ultrasonic Range Finder. Mark III Robot Store [online]. Disponible en: <http://www.junun.org/MarkIII/Info.jsp?item=23>

- [21] E. Palacios, F. Remiro, L. Lucas. “Microcontrolador PIC16F84: Desarrollo de Proyectos”, Segunda Edición, Editorial Ra-ma, 2006, pp. 77-100.
- [22] E. Palacios, F. Remiro, L. Lucas. “Microcontrolador PIC16F84: Desarrollo de Proyectos”, Segunda Edición, Editorial Ra-ma, 2006, pp. 305-330.
- [23] E. Palacios, F. Remiro, L. Lucas. “Microcontrolador PIC16F84: Desarrollo de Proyectos”, Segunda Edición, Editorial Ra-ma, 2006, pp. 467-480.
- [24] E. Palacios, F. Remiro, L. Lucas. “Microcontrolador PIC16F84: Desarrollo de Proyectos”, Segunda Edición, Editorial Ra-ma, 2006, pp. 515-542.
- [25] (2014). Microcontrolador PIC16F84: Desarrollo de Proyectos [online]. Disponible en: <http://www.pic16f84a.org>
- [26] PIC Programming Course. Talking Electronics [online]. Disponible en: <http://www.talkingelectronics.com/html/PIC-Page01.html>
- [27] SGS-Thomson Microelectronics: L293B Driver Push-Pull de 4 Canales. Todo PIC [online]. Disponible en: http://www.todopic.com.ar/utiles/l293b_driver_en_puente.pdf
- [28] Taller: Gobernar Motores C.C. Cómo hacer tu robot [online]. Disponible en: <http://www.comohacerturobot.com/Taller/taller-MotoresCC.htm>
- [29] Mis primeros programas con PIC16F84A y PIC16F628A. μ Control: Electrónica en general, PICs en particular [online]. Disponible en: <http://www.ucontrol.com.ar/mis-primeros-programas-con-pic16f84a-y-pic16f628a-15/>
- [30] (2012, Octubre). Cable Serial DB9. Diario Redes [online]. Disponible en: <http://bryanclaseredes.blogspot.com/>
- [31] Null Modem Cable DB9 Female. Cablesys [online]. Disponible en: <http://www.cablesys.com/p/593/null-modem-cable-db9-femaledb9-female>
- [32] Texas Instruments: MAX232, MAX232I Dual EIA-232 Drivers/Receivers. Datasheet Catalog [online]. Disponible en: pdf.datasheetcatalog.com/datasheet/texasinstruments/max232.pdf

[33] RS232 Serial Port to use with PIC Microcontrollers. Electronic Circuits Projects [online]. Disponible en:

<http://320volt.com/en/rs232-seri-portun-pic-mikro-denetleyiciler-ile-kullanimi/>

[34] S. W. Harden (2009, Mayo). Simple Case AVR/PC Serial Communication via MAX232. SWHarden [online]. Disponible en:

<http://www.swharden.com/blog/2009-05-14-simple-case-avrpc-serial-communication-via-max232/>

[35] (1999, Octubre). D. Terzopoulos, T. McInerney. T-Snakes. Topology Adaptive Snakes. University of California: Computer Science Department [online]. Disponible en: <http://www.cs.ucla.edu/~dt/papers/mia00/mia00.pdf>

[36] (1993, Noviembre) L.D. Cohen, I. Cohen. Finite-Element Methods for Active Contour Models and Balloons for 2D and 3D Images. Université Paris Dauphine: Centre de Recherche de Mathématiques de la Décision [online]. Disponible en: <https://www.ceremade.dauphine.fr/~cohen/mypapers/IsaacPAMI93.pdf>

[37] (1989, Agosto) L.D. Cohen. On Active Contour Models and Balloons. Université Paris Dauphine: Centre de Recherche de Mathématiques de la Décision [online].

Disponible en:

<https://www.ceremade.dauphine.fr/~cohen/mypapers/cohenCVGIP91.pdf>

[38] E. Coto, O. Rodríguez. "T-Snakes y Triangulación de Delaunay como Método de Generación de Mallados de Estructuras Anatómicas para la Aplicación del Método de Elementos Finitos". Universidad Central de Venezuela. Escuela de Computación: Laboratorio de Computación Gráfica [online]. Disponible en:

<http://ccg.ciens.ucv.ve/~ernesto/papers/CotoCINEMICS04.pdf>

[39] (2014). Microcontrolador PIC16F84: Desarrollo de Proyectos [online]. Disponible en: <http://www.pic16f84a.org/proteus.html>

[40] E. Petroustos. "La Biblia de Visual Basic 2005", Editorial Anaya, 2007, pp. 869-930.

[41] F. Ceballos. "Enciclopedia de Microsoft Visual Basic", Editorial Ra-ma, 2006, pp. 387-432.