

# **ESCUELA POLITÉCNICA NACIONAL**

**ESCUELA DE INGENIERÍA**

**CONTROLADOR UNIVERSAL EN BASE A UN  
MICROPROCESADOR**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
ELECTRONICA Y CONTROL**

**ALEJANDRO NICOLAS VELASCO JARA**

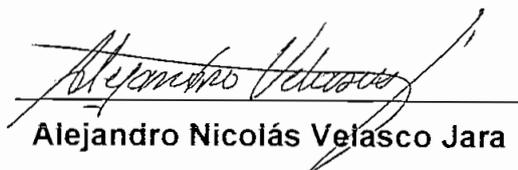
**DIRECTORA: ING. ANA RODAS**

**Quito, Septiembre 2003**

## DECLARACIÓN

Yo, Alejandro Nicolás Velasco Jara, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

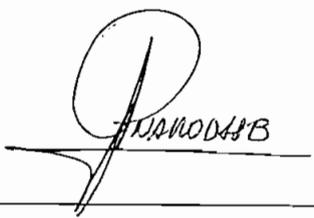
A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



Alejandro Nicolás Velasco Jara

### CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Alejandro Nicolás Velasco Jara, bajo mi supervisión.

A handwritten signature in black ink, appearing to read 'ANARODAS', is written over a horizontal line. The signature is stylized with a large, circular initial 'A'.

**Ing. Ana Rodas**  
**DIRECTORA DE**  
**PROYECTO**

## DEDICATORIA

A mis padres, a mis hermanas, a mi abuelo y a todas las personas extraordinarias que han permitido que culmine mi carrera.

Un especial reconocimiento a la Ing. Ana Rodas por su valiosa guía en el desarrollo de este trabajo

*ALEJANDRO*

# CONTENIDO

<b>PRESENTACION .....</b>	<b>IX</b>
<b>CAPITULO 1. ....</b>	<b>1</b>
<b>GENERALIDADES .....</b>	<b>1</b>
1.1. <b>EL CONTROLADOR DIGITAL .....</b>	<b>3</b>
1.2. <b>TIPOS DE CONTROLADORES DIGITALES .....</b>	<b>6</b>
1.2.1. <b>CONTROLADORES TODO O NADA .....</b>	<b>6</b>
1.2.2. <b>CONTROLADORES DE ACCION PROPORCIONAL .....</b>	<b>7</b>
1.2.3. <b>CONTROLADOR PROPORCIONAL INTEGRAL DERIVATIVO .....</b>	<b>9</b>
1.3. <b>ALGORITMOS DE CONTROL PID .....</b>	<b>12</b>
1.3.1. <b>DIFERENCIAS ENTRE ALGORITMOS .....</b>	<b>13</b>
1.4. <b>ESPECIFICACIONES DE UN CONTROLADOR PID .....</b>	<b>14</b>
1.5. <b>DISEÑO DEL ALGORITMO PID .....</b>	<b>15</b>
1.5.1. <b>ANALISIS EN EL PLANO "S" .....</b>	<b>16</b>
1.5.2. <b>LUGAR GEOMETRICO DE LAS RAICES PARA UNA ACCION PROPORCIONAL .....</b>	<b>17</b>
1.5.3. <b>LUGAR GEOMETRICO DE LAS RAICES PARA UNA ACCION INTEGRAL .....</b>	<b>18</b>
1.5.4. <b>LUGAR GEOMETRICO DE LAS RAICES PARA UNA ACCION DERIVATIVA .....</b>	<b>19</b>
1.5.5. <b>ANALISIS EN EL PLANO Z .....</b>	<b>20</b>
1.6. <b>EL CONTROLADOR UNIVERSAL .....</b>	<b>25</b>
1.6.1. <b>CONTROLADOR UNIVERSAL DISEÑADO .....</b>	<b>25</b>
<b>CAPITULO 2. ....</b>	<b>31</b>
<b>DISEÑO DE HARDWARE .....</b>	<b>31</b>
2.1. <b>TARJETA PRINCIPAL .....</b>	<b>31</b>
2.1.1. <b>EL MICROCONTROLADOR .....</b>	<b>32</b>
2.1.2. <b>MEMORIA EEPROM .....</b>	<b>34</b>
2.1.3. <b>PROGRAMADOR PIC .....</b>	<b>35</b>
2.1.4. <b>CONECTORES .....</b>	<b>36</b>
2.2. <b>PERIFERICOS Y COMPONENTES EXTERNOS .....</b>	<b>41</b>
2.2.1. <b>DESCRIPCION .....</b>	<b>41</b>
<b>DESARROLLO DEL SOFTWARE .....</b>	<b>46</b>
3.1. <b>CARACTERISTICAS DEL LENGUAJE C .....</b>	<b>46</b>
3.1.1. <b>REQUERIMIENTOS DEL COMPILADOR .....</b>	<b>47</b>
3.2. <b>EL MC68HC08GP32 .....</b>	<b>48</b>
3.3. <b>REGISTROS DEL CONTROLADOR DISEÑADO .....</b>	<b>52</b>
3.3.1. <b>REGISTROS DE SETPOINTS .....</b>	<b>52</b>
3.3.2. <b>REGISTRO DE IDENTIFICACION .....</b>	<b>53</b>
3.3.3. <b>REGISTRO DE ENTRADA .....</b>	<b>54</b>
3.3.4. <b>REGISTRO DE CONFIGURACIÓN DE LECTURA .....</b>	<b>54</b>
3.3.5. <b>REGISTROS DE SALIDAS .....</b>	<b>55</b>
3.3.6. <b>REGISTRO DE PARÁMETROS DE COMUNICACIÓN .....</b>	<b>57</b>
3.3.7. <b>REGISTRO DE CONFIGURACION DE SALIDAS .....</b>	<b>58</b>
3.3.8. <b>REGISTROS DE ESCALA Y OFFSET DE LECTURA .....</b>	<b>62</b>
3.3.9. <b>REGISTRO MISCELLANEOUS .....</b>	<b>64</b>
3.3.10. <b>REGISTRO CONTRASTE DISPLAY (LCDDR) .....</b>	<b>65</b>
3.4. <b>SOFTWARE DE OPERACIÓN .....</b>	<b>66</b>
3.4.1. <b>PROGRAMA PRINCIPAL .....</b>	<b>66</b>
3.4.2. <b>SUBROUTINA DEL ALGORITMO DE CONTROL .....</b>	<b>67</b>
3.4.3. <b>SUBROUTINA DE BASE DE TIEMPO PARA EL CONTROL .....</b>	<b>69</b>
3.4.4. <b>SUBROUTINAS DE USO COMUN .....</b>	<b>71</b>
3.4.5. <b>SUBROUTINAS DE COMUNICACIÓN SERIAL I2C .....</b>	<b>73</b>
3.4.6. <b>SUBROUTINAS PARA EL MANEJO DEL DISPLAY LCD .....</b>	<b>85</b>
3.4.7. <b>SUBROUTINAS PARA LA COMUNICACIÓN SERIAL RS232 .....</b>	<b>90</b>

3.4.8.	<i>SUBROUTINAS PARA LA ADQUISICION DE LA VARIABLE MEDIDA</i> .....	94
3.4.9.	<i>SUBROUTINAS PARA LA INTERFAZ CON EL TECLADO</i> .....	96
3.5.	<i>SOFTWARE DE INTERFAZ CON EL USUARIO</i> .....	98
3.6.	<i>SOFTWARE DE INTERFAZ CON EL COMPUTADOR</i> .....	104
3.6.1.	<i>SOFTWARE EN VISUALBASIC</i> .....	108
<b>CAPITULO 4.</b> .....		<b>114</b>
<b>PRUEBAS Y RESULTADOS</b> .....		<b>114</b>
4.1.	<b>CONSIDERACIONES ESPECIALES EN LA PROGRAMACIÓN</b> .....	114
4.2.	<b>PRUEBAS DE LECTURA DE LA VARIABLE MEDIDA</b> .....	115
4.3.	<b>PRUEBAS A LA ENTRADA DEL CONTROLADOR</b> .....	117
4.4.	<b>PRUEBAS DE LA SALIDA DEL CONTROLADOR</b> .....	122
4.4.1.	<i>SALIDA ANALOGICA</i> .....	122
4.4.2.	<i>SALIDA DIGITAL PARA EL RELE DE ESTADO SÓLIDO</i> .....	123
4.5.	<b>PRUEBAS DE CONTROL EN SISTEMAS DE LAZO CERRADO</b> .....	125
4.5.1.	<i>PRUEBAS EN MODULOS DIDACTICOS DEL LABORATORIO</i> .....	125
4.5.2.	<i>SIMULACION DE PLANTAS A TRAVES DEL COMPUTADOR</i> .....	130
4.6.	<b>PRUEBAS DE COMUNICACIÓN SERIAL CON EL COMPUTADOR</b> .....	133
<b>CAPITULO 5.</b> .....		<b>137</b>
<b>CONCLUSIONES Y RECOMENDACIONES</b> .....		<b>137</b>
5.1.	<b>CONCLUSIONES</b> .....	137
5.2.	<b>RECOMENDACIONES</b> .....	139
<b>REFERENCIAS BIBLIOGRAFICAS</b> .....		<b>141</b>
<b>ANEXOS</b> .....		<b>142</b>

## RESUMEN

Cada día es más frecuente la necesidad de controlar plantas industriales y se hace más extensible el desarrollo de los microprocesadores, por lo que se requiere tener la asociación microprocesador – control para los procesos de mayor o menor complejidad.

En nuestro medio, los controles no son muy sofisticados, de ahí el interés de implementar módulos de control que puedan ser configurables para diferentes procesos.

Por esta razón, considerando que el equipamiento en la industria nacional, en la mayoría de los casos, es adquirido en el exterior y existiendo además la posibilidad de desarrollar en nuestro medio alternativas propias basadas en la última tecnología y a costos relativamente bajos, se ha tratado de elaborar un proyecto a base de un microprocesador con tecnología Motorola.

Este trabajo está encaminado al diseño e implementación de un controlador que permita ser configurado para distintos procesos mediante un teclado y una pantalla de visualización amigables con el usuario.

Esta unidad permite seleccionar el tipo de entrada de voltaje DC o corriente DC para la medición del proceso. Estas entradas son completamente escalables a unidades virtuales de ingeniería, con un punto decimal seleccionable, perfecto para uso en temperatura, presión, flujo u otro tipo de procesos.

El control de los procesos se puede lograr mediante estrategias de control on- off o PID. También se lo puede utilizar con propósitos de monitoreo, donde se requiere únicamente la lectura de la variable de proceso o la activación de una alarma como sistema de seguridad.

Entre sus características posee dos salidas digitales que manejan relés de estado sólido (SSR), una salida analógica de voltaje y comunicación serial RS-232.

Está desarrollado un software en VisualBasic que permite configurar ciertos parámetros del equipo mediante un computador y utiliza un protocolo con el cual

se puede adaptar como parte integral de un sistema de control mayor gobernado por una computadora principal.

Este controlador se ha logrado integrar a proyectos de Control de Procesos e Instrumentación diseñados en la Escuela Politécnica Nacional, lo que demuestra que su utilización puede adaptarse fácilmente a diferentes equipos.

Como una aplicación al uso del controlador, se ha simulado una planta ideal en LabView y se ha registrado el comportamiento del controlador a cambios en la señal de referencia.

## PRESENTACION

El presente trabajo tiene como objetivo principal el diseño y la construcción de un controlador universal. Para ello, se ha tomado como referencia los controladores comerciales existentes y que tienen mayor asentamiento en el campo de la instrumentación como son OMEGA, NEWPORT y HONEYWELL.

Utilizando la tecnología Motorola y el programa de desarrollo CodeWarrior de Metrowerks se ha desarrollado el controlador, motivo de este proyecto.

Se presentan cinco capítulos que abarcan las etapas principales para el desarrollo del equipo.

En el Capítulo 1 se encuentra material introductorio a los sistemas de control, se definen los controladores digitales, comerciales, los algoritmos utilizados y se describe el controlador diseñado.

El Capítulo 2 describe el diseño y construcción del hardware. El Capítulo inicia con una descripción de la tarjeta principal utilizada (MINIMAX/908-C) que incluye en la arquitectura al microcontrolador MC68HC08 para luego describir la tarjeta diseñada y sus interconexiones.

En el Capítulo 3 se presenta el software desarrollado para el controlador cuya programación se ha dividido en tres secciones. En la primera sección se describe el funcionamiento interno que tiene el microcontrolador con los diferentes periféricos. A continuación se detalla la interfaz que existe entre el controlador y el usuario. En la tercera sección se describe la comunicación que tiene el equipo con un computador personal.

En el Capítulo 4 se detallan las pruebas realizadas con el controlador diseñado y los resultados obtenidos a partir de ellas. Este capítulo presenta las pruebas de lectura y de entrada de la variable de proceso a ser medida en el aparato

diseñado; las pruebas de las salidas que posee y el comportamiento que tiene el equipo en sistemas de lazo cerrado tanto en módulos didácticos implementados en el laboratorio de Instrumentación como en procesos simulados a través de una computadora.

Para terminar se presenta en el Capítulo 5 las conclusiones y las recomendaciones, consecuencias de la realización de este proyecto.

**CAPITULO 1.**  
**GENERALIDADES**

## CAPITULO 1.

### GENERALIDADES

Con el advenimiento de la era industrial, una de las preocupaciones más importantes en el campo de la ingeniería ha sido la continua búsqueda de técnicas que permitan controlar a las máquinas de una manera automática.

La automatización, que viene a ser un término normalmente empleado como la sustitución de hombres por máquinas en el lugar de trabajo, es un resultado del avance de la tecnología.

Actualmente el control automático es una parte integral de los procesos industriales y de manufactura. Las plantas modernas contienen una gran cantidad de variables de entrada y de salida e implican un control complejo. Sin embargo, la disponibilidad de microprocesadores, computadores, etc. facilitan la operación de los sistemas.

Un sistema es una combinación de componentes que actúan de manera conjunta para cumplir un objetivo determinado. El sistema básico de control requiere de una variable de entrada para tomar una acción mediante los elementos del sistema de control sobre una variable de salida. La Figura 1-1 representa un sistema básico de control.

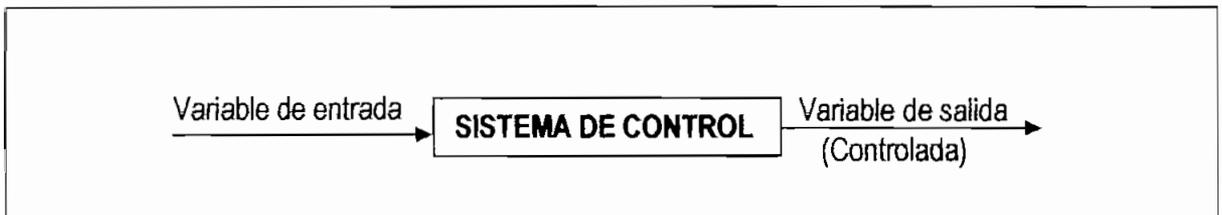
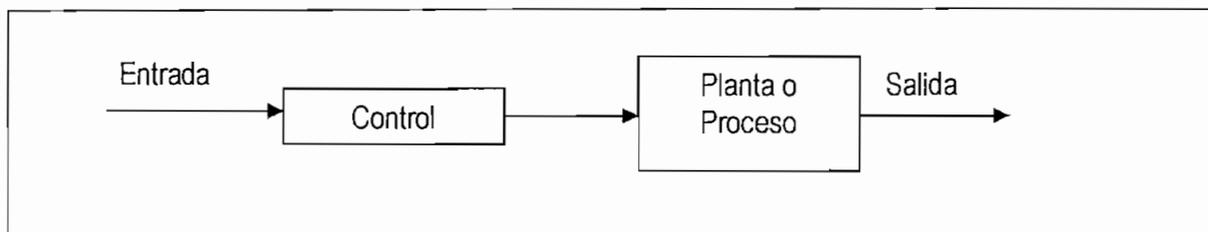


Figura 1-1: Sistema Básico de Control

Por ejemplo, la intensidad de luz en una habitación se puede controlar mediante una perilla. La posición angular de la perilla constituye la entrada, y la intensidad de luz la salida. Los elementos utilizados comprenden el sistema de control.

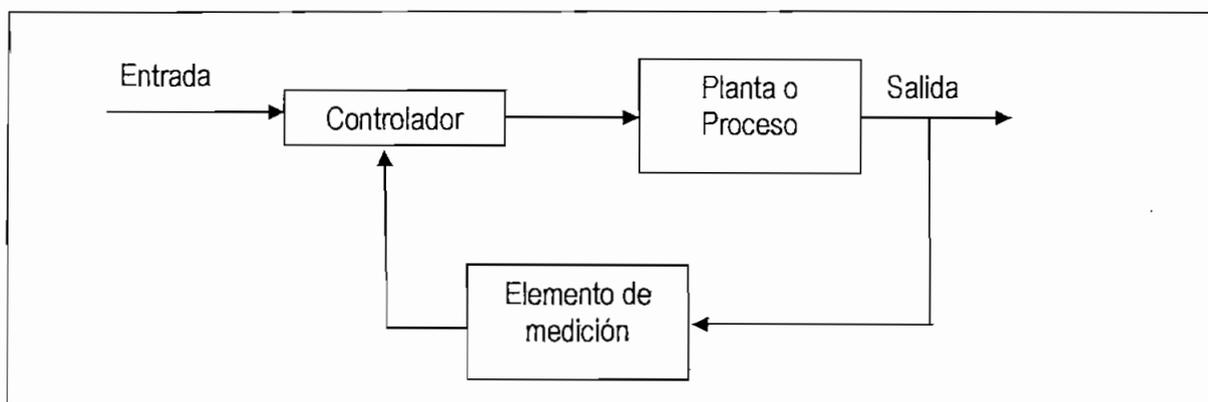
Existen dos tipos de sistemas de control: sistemas de control de lazo abierto o no realimentados y sistemas de control de lazo cerrado o con realimentación.

En un sistema de control de lazo abierto su ajuste depende de los criterios y cálculos humanos. No toma en cuenta variaciones en las condiciones del medio, ni tampoco perturbaciones o alteraciones externas. Un sistema de lazo abierto se muestra en la Figura 1-2.



**Figura 1-2: Sistema de control de lazo abierto**

En un sistema de control en lazo cerrado la variable de salida tiene efecto en la acción de control. Un sistema de lazo cerrado está compuesto de un controlador, la planta o proceso y de un elemento de medición.



**Figura 1-3: Sistema de control de lazo cerrado**

Como se ve en la Figura 1-3, se toma una parte de la señal de salida y se la lleva a un medidor, el cual realimenta la información del estado de la salida hacia el controlador. El controlador compara esta información con un valor de referencia y determina la acción requerida para llevar a la variable de salida al valor de referencia.

De esta manera, el control de lazo cerrado toma en cuenta las acciones de variables perturbadoras. Si la variable controlada solo se limitara a seguir a la variable de referencia, entonces cualquier perturbación se reflejaría totalmente en la variable deseada. Como la perturbación ocurre en cualquier momento, resulta más conveniente medir la variable controlada e intervenir continuamente en el lazo de mando en el caso de que su valor se aleje del valor prescrito por la variable de referencia.

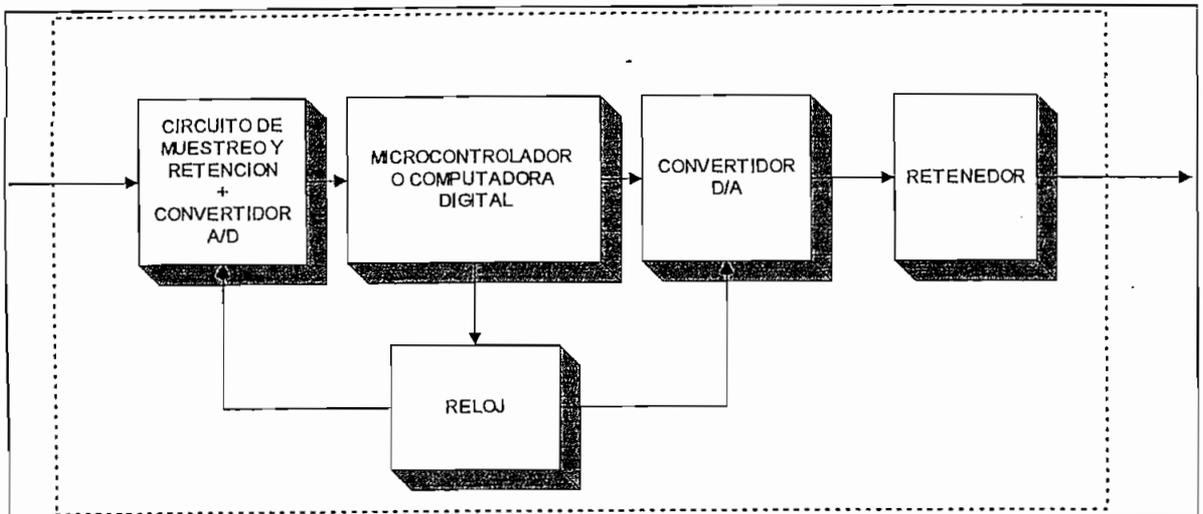
El elemento de medición normalmente es un transductor que convierte la señal de la variable física en otra señal de naturaleza diferente que sirve como señal de entrada al controlador.

El diseño de un sistema de control se facilita si se puede lograr un modelo matemático del proceso. Este modelo permite predecir el comportamiento que tendrá el sistema, tal como la estabilidad y la influencia de los fenómenos transitorios, etc. En la consecución del modelo matemático se tomará siempre en cuenta el compromiso entre simplicidad y exactitud.

## **1.1. EL CONTROLADOR DIGITAL**

Los controladores iniciales fueron hechos en base a sistemas mecánicos, neumáticos o hidráulicos. Con la aplicación de la electrónica, muchos elementos de control fueron implementados con esta tecnología usando básicamente amplificadores operacionales. En la medida en que la utilización del computador digital se ha extendido, también el control se ha digitalizado con lo que se ha llegado a tener circuitos de control digitales.

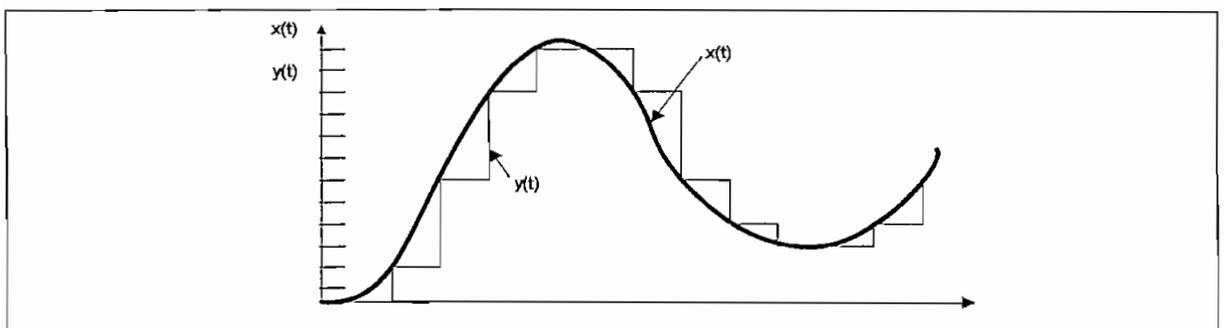
Un controlador digital está compuesto de un circuito de muestreo y de retención, un convertidor analógico – digital, una unidad de procesamiento central o computador digital, un reloj, un convertidor digital – analógico y de un retenedor como se muestra en la Figura 1-4.



**Figura 1-4: Diagrama de Bloques de un Controlador Digital**

Los microcontroladores trabajan con datos numéricos y principalmente con un sistema binario. Consecuentemente cualquier señal analógica deberá previamente ser transformada a un sistema binario, lo cual se logra con un circuito de muestreo y de retención y el convertidor analógico - digital. Esta señal digital se procesa en el microcontrolador y la salida resultante es otra señal muestreada que va al convertidor digital – analógico y alimenta a un circuito de retención para obtener una señal en tiempo continuo.

Al circuito de muestreo y de retención se define como un circuito que recibe una señal analógica como entrada y mantiene dicha señal en un valor constante durante un tiempo específico como se muestra en la Figura 1-5.



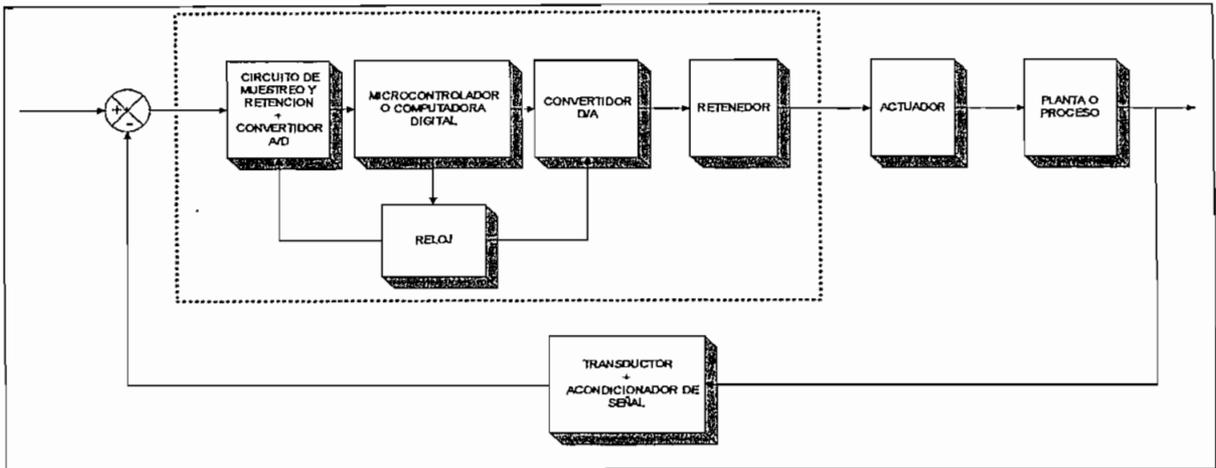
**Figura 1-5: Entrada Analógica  $x(t)$  y Salida Discreta  $y(t)$**

El convertidor analógico – digital es un codificador y además la interfaz entre un componente analógico y uno digital. Mientras que el convertidor digital – analógico es un decodificador debido a que convierte la señal digital (datos codificados

numéricamente) en una señal analógica y es la interfaz entre un componente digital y uno analógico.

El reloj sincroniza los eventos de adquisición y entrega de información entre los diferentes componentes.

En la Figura 1-6 se representa en diagrama de bloques a un sistema de control en el que se incluye el controlador digital.



**Figura 1-6: Diagrama de bloques de un sistema de control digital**

Cuando se trabaja con más de una variable generalmente se necesita de un multiplexor analógico; éste es un dispositivo que lleva a cabo la función de compartir en tiempo un convertidor A/D entre muchos canales analógicos. Es un interruptor múltiple, por cuanto conmuta en forma secuencial entre los diversos canales.

De manera similar se tendrá un demultiplexor a la salida sincronizando con las señales de muestreo de entrada, el mismo que separará los datos digitales de dicha señal compuesta hacia los diferentes canales conectados previamente a un convertidor D/A para producir la señal de salida analógica.

Hoy en día, las empresas fabricantes de microcontroladores han logrado compactar dentro de una sola arquitectura a componentes como conversores A/D y periféricos con funciones más específicas que facilitan su manejo lógico por medio de programación.

## **1.2. TIPOS DE CONTROLADORES DIGITALES**

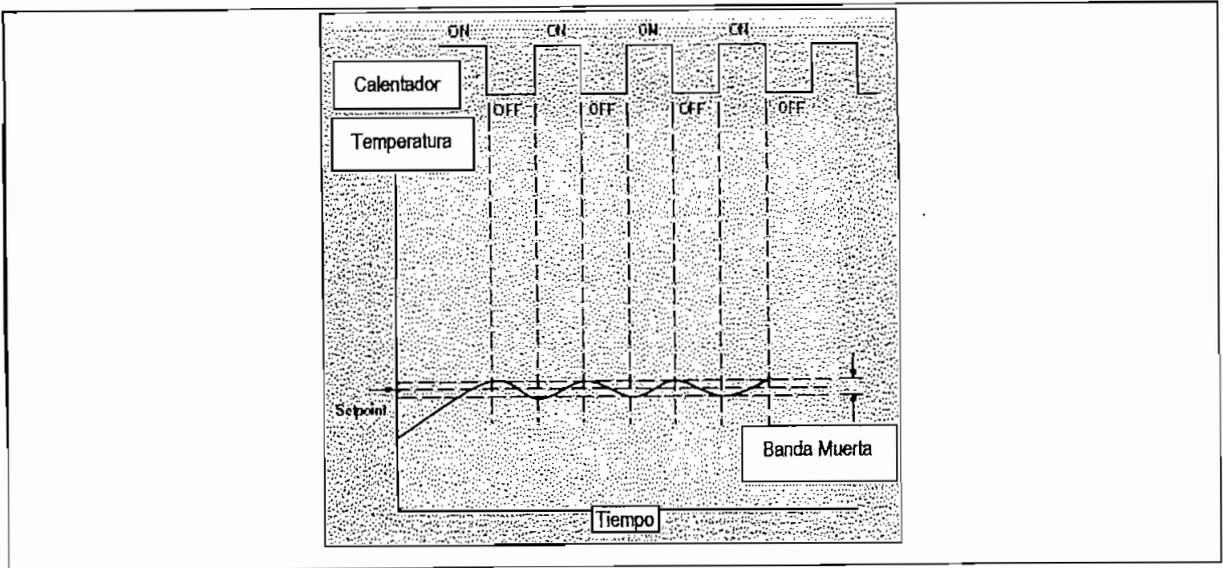
En sistemas de lazo cerrado, partiendo de la señal de error se forma la variable correctora precisa para conseguir un control del sistema, ajustando lo mejor posible los parámetros del controlador de acuerdo con la respuesta del sistema controlado.

Para el control de procesos se emplean algunos tipos de controladores. Se utilizan, por ejemplo, los controladores de tres niveles cuya salida solo puede adoptar tres valores discretos. La salida resultante en cada caso significa, marcha a la izquierda, parada y marcha a la derecha si se tratase de un motor, y se emplean para accionar válvulas, compuertas, etc. Los controladores todo o nada también denominados los «controladores a dos niveles», se utilizan, por ejemplo, para controlar la temperatura de los hornos. Los controladores caracterizados por una acción proporcional así como los controladores de tipo PID, adoptan una salida analógica de tensión o corriente comprendida en un margen de control que pasan de un valor a otro en forma continua o de intervalos, o si se trata de una proporción en el tiempo, en un intervalo varían la relación de encendido – apagado.

### **1.2.1. CONTROLADORES TODO O NADA**

Los controladores todo o nada, controladores a dos niveles o controladores ON – OFF, utilizan la técnica más sencilla de operación. Pasan de un estado de encendido (ON) a uno de apagado (OFF) y viceversa, sin ningún estado intermedio. Un controlador todo o nada conmutará la salida solamente cuando la variable de proceso cruce el punto de referencia y continuará haciéndolo en forma cíclica (Figura 1-7).

Muchas veces ocurren los ciclos rápidamente y esto provoca daños en los contadores o en las válvulas. Para evitar daños se incluye un diferencial (banda muerta) o histéresis a la operación del controlador. Este diferencial requiere que la variable controlada exceda al valor de referencia por una cierta cantidad antes de que la salida se apague o se encienda nuevamente.



**Figura 1-7: Acción del Control de Temperatura ON/OFF**

Mientras no sea indispensable un control muy preciso, la operación de tipo on – off funciona convenientemente. Este tipo de control se utiliza en sistemas en los cuales no se puede tener encendidos y apagados frecuentes, donde la masa del sistema es tan grande que los cambios de la variable controlada sean extremadamente pequeños. En alarmas es común la utilización del control a dos niveles.

### 1.2.2. CONTROLADORES DE ACCION PROPORCIONAL

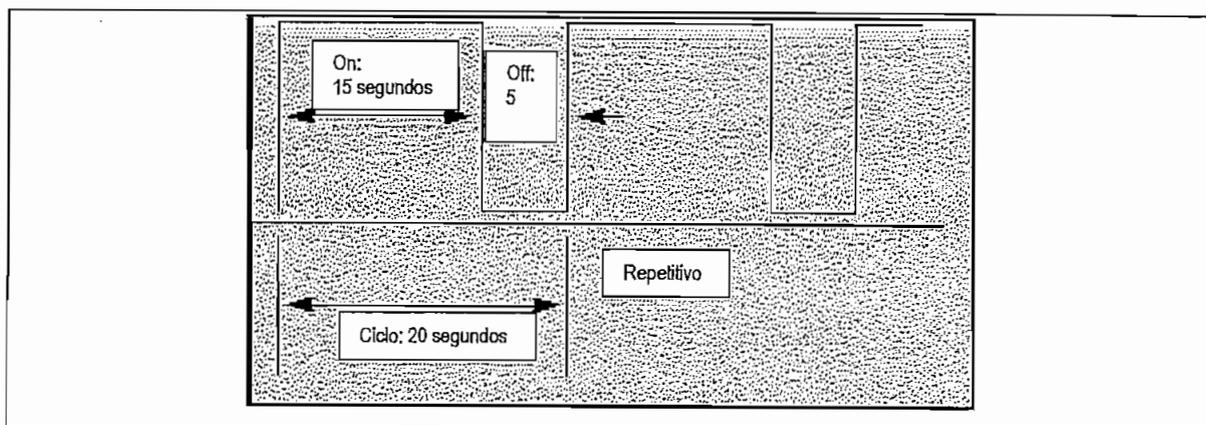
El control de acción proporcional está diseñado para eliminar los ciclos asociados con la operación de encendido - apagado. Un controlador proporcional disminuye, por ejemplo, el porcentaje de energía suministrada a un horno cuando la temperatura se aproxima a la de referencia, y baja lentamente la temperatura del horno manteniéndola luego estable. Esta acción proporcional puede ser suficiente para encendidos y apagados con intervalos de tiempo pequeños. La "proporción en tiempo" varía la relación del tiempo de encendido con respecto al tiempo de apagado.

La acción proporcional ocurre dentro de un ancho de banda proporcional alrededor del punto de referencia. Fuera de la banda proporcional, el controlador funciona como una unidad on - off, y dentro de ella, la salida se enciende o se apaga a razón de la diferencia medida que se tenga desde el punto de referencia.

En el valor de referencia o punto medio de la banda proporcional, la relación de la salida es 1:1, quiere decir, son iguales el tiempo de encendido y el tiempo de apagado. Cuanto más lejos la variable controlada esté de su referencia, más grande será la diferencia que exista entre los tiempos de encendido y apagado. Si la variable está bajo el punto de referencia, entonces la salida estará encendida más tiempo. De lo contrario, si la variable controlada sobrepasa el punto de referencia, la salida estará más tiempo en el estado de apagado.

La banda proporcional usualmente se expresa como un porcentaje de la escala completa y es una modificación en la entrada del control necesaria para producir un cambio de 100% en la salida del actuador. Es un término para definir la ganancia o sensibilidad de un control proporcional.

El control proporcional en tiempo suministra la potencia a un horno (si se trata de control de temperatura), pero en forma cíclica, de manera que el tiempo promedio de encendido y de apagado varía. En la mayoría de los casos, este ciclo de encendido – apagado que realiza el controlador y la banda proporcional se ajustan manualmente, permitiendo al controlador un mejor manejo de un proceso particular. En la Figura 1-8 se ilustra el control proporcional por ciclos.



**Figura 1-8: Tiempo Proporcional al 75% del Nivel de Salida**

Los controladores de acción proporcional son capaces de manejar salidas analógicas. Con este tipo de salidas, por lo general de corriente o voltaje, se varía el nivel actual de la variable controladora, en lugar de utilizar los tiempos de encendido y apagado.

Una de las ventajas del control proporcional es la simplicidad que tiene su operación. Puede requerir tal vez que el operador tenga que hacer pequeños ajustes que lleven a la referencia a su estado inicial o si las condiciones cambian significativamente.

### 1.2.3. CONTROLADOR PROPORCIONAL INTEGRAL DERIVATIVO

Las bandas proporcionales se utilizan para eliminar oscilaciones alrededor del punto de equilibrio. Sin embargo, el ancho de banda puede dar como resultado desajustes con cambios en la carga. Para compensar estos desajustes en el valor de la variable deseada, se incluye una acción integral. En cambio, una acción derivativa puede utilizarse en procesos lentos acelerando la recuperación al punto de trabajo después de generada una perturbación.

El controlador PID está provisto de un control proporcional con acción integral y derivativa. Es decir, el controlador combina la acción proporcional con los dos ajustes adicionales, los cuales ayudan a la unidad a compensar automáticamente los cambios en el sistema.

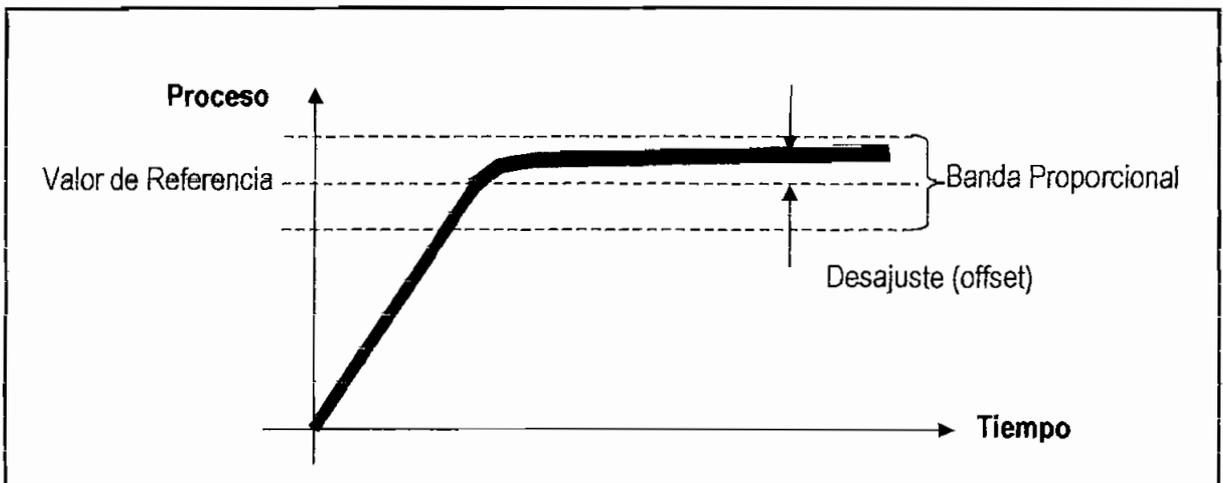


Figura 1-9: Proceso con un offset en la variable deseada

En la Figura 1-9 se puede observar un proceso con un desajuste en el valor deseado en estado estable. La diferencia entre el estado estable de la variable de proceso y el valor de referencia se conoce como "offset".

La acción proporcional se caracteriza porque la variable controladora que aparece como respuesta a la señal de error actuante no varía en el tiempo, y solamente viene expresada por un factor de proporcionalidad.

$$m(t) = K_p e(t), \quad \text{Ecuación (1-1)}$$

donde:  $e(t)$  es la variable de entrada (señal de error actuante),  $m(t)$  la variable de salida (variable controladora) y  $K_p$  es la ganancia proporcional.

La acción integral modifica a la variable controladora proporcionalmente a la señal de error actuante. Si la variable de entrada es igual a cero, no se modifica la variable controladora; ésta continúa presentando el valor que había alcanzado como consecuencia de la variable de entrada aplicada anteriormente, y éste puede ser cualquiera de los valores comprendidos en el margen de control. Si se tiene que el valor de la señal de error es pequeño, la variable controladora se modifica lentamente; por el contrario, si es grande, varía rápidamente.

La variable controladora puede tener un valor distinto de cero cuando la señal de error es cero. Únicamente con control proporcional esto es imposible debido a que una señal de control no nula requiere una señal de error no nula. Si bien la acción integral anula este efecto de corrimiento, puede llevar a una respuesta oscilatoria indeseable.

La acción integral se caracteriza por la integración a lo largo del tiempo de la señal de error actuante. La ecuación general en función del tiempo viene dada por:

$$m(t) = \frac{1}{T_i} \int_0^T e(t) dt \quad \text{Ecuación (1-2)}$$

El parámetro característico de este término es la constante de tiempo de acción integral (o tiempo de reajuste)  $T_i$ . De igual manera puede ser empleado el valor recíproco de la constante de tiempo de acción integral  $K_i$ .

A mayor constante de tiempo de acción integral se tiene un menor incremento de la variable controladora por unidad de tiempo para una señal de error dada.

Una acción de control proporcional e integral queda definida entonces por la siguiente ecuación:

$$m(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt, \quad \text{Ecuación (1-3)}$$

donde el tiempo integral regula la acción de control integral, y una modificación de  $K_p$  afecta tanto a la parte integral como al término proporcional de la acción de control.

Se denomina frecuencia de reposición (o frecuencia de reajuste) a la inversa del tiempo integral  $T_i$ , y es el número de veces por minuto que se duplica la parte proporcional de la acción de control.

Una acción de control proporcional y derivada queda definida por la ecuación:

$$m(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt}, \quad \text{Ecuación (1-4)}$$

donde  $T_d$  es el tiempo derivativo de la acción de control de velocidad o derivativo. En este tipo de control la salida es proporcional a la velocidad de la variación de la señal de entrada. El tiempo derivativo  $T_d$  es el intervalo de tiempo en que la acción de velocidad se adelanta al efecto de la acción proporcional.

La ventaja de la acción derivativa se orienta a que es una acción que se anticipa; sin embargo, amplifica las señales de ruido y puede producir efectos de saturación en el controlador.

No se puede tener una acción de control únicamente derivativa, debido a que este tipo de control es efectivo en períodos transitorios.

Cuando se tiene un controlador de tipo PID, se logran las ventajas de cada acción de control individuales. La ecuación para un controlador PID en función del tiempo está dada por

$$m(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt} + \frac{K_p}{T_i} \int_0^t e(t) dt \quad \text{Ecuación (1-5)}$$

Los términos proporcional, integral y derivativo deben ser ajustados individualmente para el uso en una prueba o error de un sistema en particular. Así es posible tener un control más estable y exacto entre los tres tipos de controladores, y es el mayormente utilizado en sistemas en los que se tiene una masa relativamente pequeña, que reaccionan rápidamente a cambios en la energía añadida al proceso. Además se recomienda utilizarlos en sistemas donde la carga cambia a menudo y cuando se espera que el controlador compense automáticamente debido a los cambios frecuentes de la referencia, la cantidad de energía aplicable o la masa a ser controlada.

### 1.3. ALGORITMOS DE CONTROL PID

En la práctica, los fabricantes de controladores no se adhieren a algoritmos de control PID estándares. Muchos fabricantes y vendedores utilizan algoritmos PID tan distintos e incluso algunos que sólo son válidos para sus propias líneas de productos.

De manera general existen 3 grandes clasificaciones de los algoritmos PID: serial, ideal y paralelo.

#### **Algoritmo ideal:**

$$SALIDA = K_c \left[ e(t) + \frac{1}{I} \int e(t) d(t) + D \frac{de(t)}{dt} \right] \quad \text{Ecuación (1-6)}$$

#### **Paralelo:**

$$SALIDA = K_p e(t) + \frac{1}{I} \int e(t) d(t) + D \frac{de(t)}{d(t)} \quad \text{Ecuación (1-7)}$$

**Serial:**

$$SALIDA = K_c \left[ e(t) + \frac{1}{I} \int e(t) dt \right] \left[ 1 + D \frac{de(t)}{dt} \right] \quad \text{Ecuación (1-8)}$$

donde  $K_c$ ,  $K_p$  son parámetros de ganancia,  $I$  corresponde a la acción integral y  $D$  a la acción derivativa. La estructura del algoritmo serial hace que actúe como un controlador electrónico donde un solo amplificador incluye los tres términos de control. En la Figura 1-10 se muestra un elemento de acción proporcional, integral y derivativa realizado con un amplificador inversor. Algunos fabricantes de controladores usan la forma serial en sus algoritmos digitales para mantener un ajuste similar a controladores electrónicos.

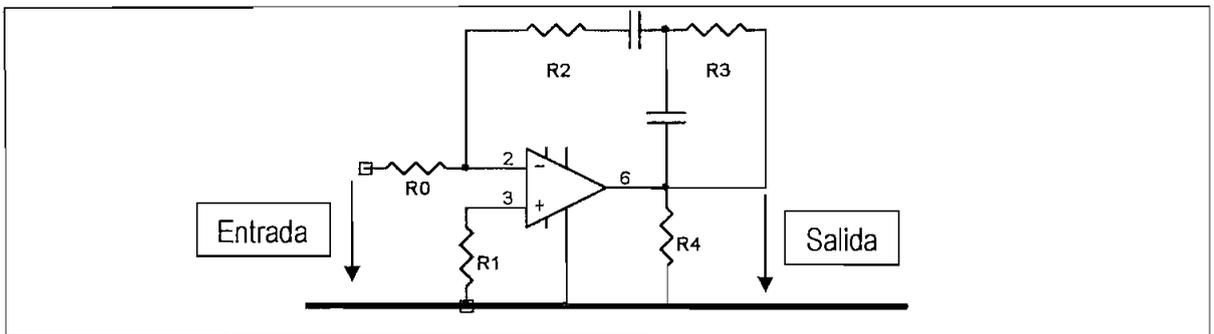


Figura 1-10: Controlador PID realizado con un amplificador electrónico

### 1.3.1. DIFERENCIAS ENTRE ALGORITMOS

Si se utiliza solamente una acción proporcional, la diferencia principal entre algoritmos seriales e ideales estaría en que unos utilizan a la banda proporcional en lugar de la ganancia proporcional. La banda proporcional es recíproca a la ganancia proporcional. En un controlador que utiliza parámetros de ganancia, al incrementar ésta, hace que el lazo sea más sensible y menos estable. El mismo efecto se tiene al decrecer la banda.

Una vez que se convierten los valores de integral a las mismas unidades, los controladores PI responden en su mayoría de igual forma para disturbios en la carga.

La variación más grande entre algoritmos está en la acción derivativa. Virtualmente no existen dos iguales. Esta es parte de la razón por qué muchas personas no usan dicha acción. Las diferencias existen a causa de la variedad de métodos de filtrado, si lo tienen, o donde el derivativo trabaja o no sobre los cambios del punto de referencia, y cómo interactúa o no el derivativo con la acción integral.

En un controlador serial, cuando se utiliza tanto acción integral como derivativa, los modos integral y derivativo interactúan.

Con controladores paralelos, la ganancia es multiplicada solamente por la señal de error. Las acciones integral y derivativa son "independientes" de la ganancia del controlador.

A simple vista se creería que el controlador paralelo es más fácil para trabajar debido a esta "independencia". Pero, los algoritmos paralelos requieren parámetros muy distintos de sintonización integral y derivativa que otros algoritmos.

#### **1.4. ESPECIFICACIONES DE UN CONTROLADOR PID**

La gama de estándares industriales para controladores PID no es amplia. Sin embargo, requieren de ciertas habilidades y características para que estos controladores tengan lazos robustos y óptimos de control.

Un controlador PID no depende de las unidades de medida. Es decir, es independiente de que esté en °C, °F, rpm, psi, etc. El controlador trabaja simplemente en una base de 0 – 100% o sobre una base de 0 – 1 para entradas, salidas y puntos de referencia. Las unidades de medida, si se utilizan, son únicamente para propósitos de visualización y son independientes del algoritmo PID.

Haciendo referencia a los controladores universales fabricados por Newport y Honeywell se ve que utilizan velocidades de muestreo de 3 a 4 veces por

segundo que si bien son bajas, resultan suficientes para procesos de temperatura. Sin embargo, en otro tipo de proceso no controlan adecuadamente.

Las unidades de acción proporcional normalmente están en porcentaje de la Banda Proporcional  $P$  o de la Ganancia Proporcional  $K_c$ , donde

$$K_c = 100/P \quad \text{Ecuación (1-9)}$$

La acción proporcional tiene que trabajar sobre la señal de error actuante o sobre la variable controlada dependiendo de la selección que se haga.

El rango para configurar la Banda Proporcional normalmente está entre 1 a 10,000 unidades. Por otro lado, si se usa la Ganancia Proporcional, el rango suele ser de 0.01 a 100 por recomendaciones de los fabricantes.

Las unidades de la acción integral generalmente están en minutos por repetición o segundos por repetición. La acción integral debe operar sobre la señal de error actuante. El tiempo integral es ajustable entre 0.002 a 1000 minutos.

Las unidades de acción derivativa están generalmente en minutos o segundos. La acción debería ser aplicada solo a la señal de error actuante. El tiempo derivativo en los controladores se ajusta en un rango de 0 a 500 minutos.

## 1.5. DISEÑO DEL ALGORITMO PID

Como se dijo anteriormente, en un sistema de lazo cerrado se ingresa la señal de la variable a ser controlada mediante un proceso de muestreo. Las características del lazo dictan la razón de muestreo y los cálculos deben completarse antes de que empiece el siguiente ciclo de muestreo. Estas exigencias en el lazo toman un lugar muy importante en el control digital de sistemas en lazo cerrado con señal de error en la realimentación.

Si el sistema de control tiene una resonancia u otro comportamiento con una constante de tiempo más corta que el tiempo de muestreo y de cálculo, simplemente se genera un caos. En otras palabras, la respuesta transitoria en un sistema de control en tiempo discreto depende del período de muestreo

### 1.5.1. ANALISIS EN EL PLANO "S"

El diagrama de bloques de una estructura analógica de control PID ideal se muestra en la Figura 1-11.

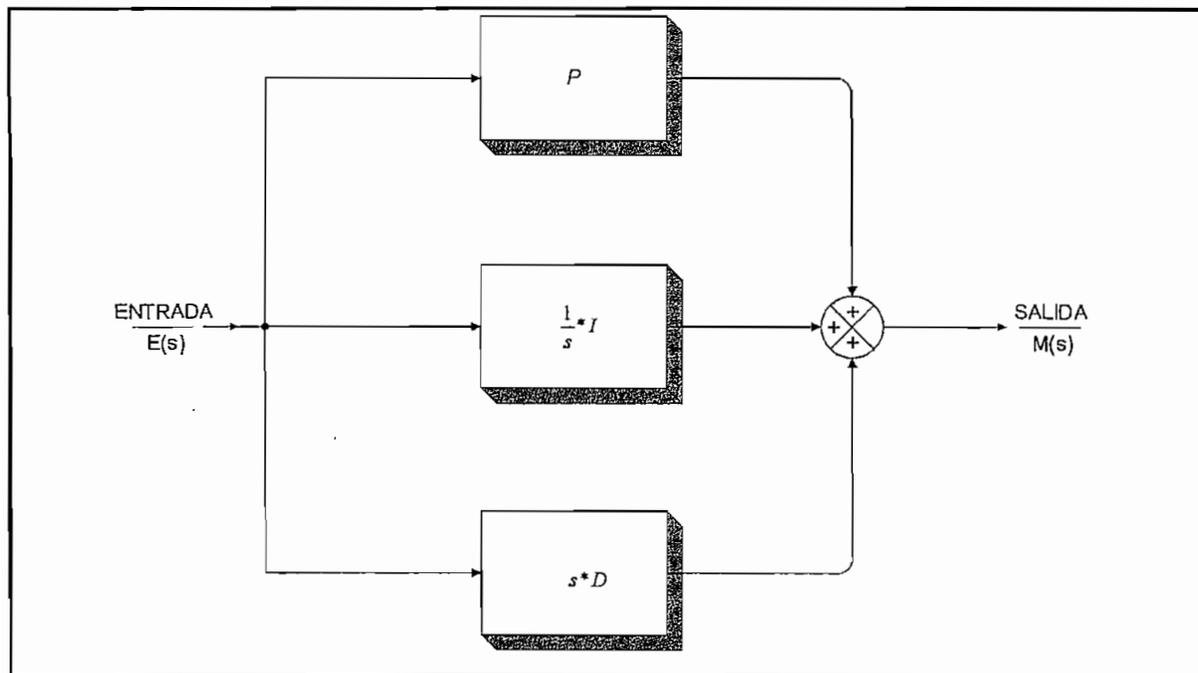


Figura 1-11: Diagrama de bloque de una estructura analógica de control PID

La función de transferencia (como función de s) es:

$$\frac{M(s)}{E(s)} = \frac{Ps + I + Ds^2}{s} \quad \text{Ecuación (1-10)}$$

donde:

- $M(s)$  es la salida de la sección PID
- $E(s)$  es la entrada de la sección PID
- $P$  es un multiplicador del error
- $I$  es el multiplicador de la integral del error
- $D$  es el multiplicador de la derivada del error
- $s$  es el operador de Laplace de frecuencia compleja.



### 1.5.3. LUGAR GEOMETRICO DE LAS RAICES PARA UNA ACCION INTEGRAL

En el dominio de la frecuencia, el término  $I$  también afecta la posición de los ceros como se puede observar en la Figura 1-13. Para  $I = 0$ , uno de los ceros está en  $s = -P/D$ , y el otro en  $s = 0$ , lo que significa que cancelará el integrador en  $s = 0$ . Conforme se aumenta el término  $I$  se tiende hacia un sistema subamortiguado que responderá rápidamente a un error de estado estacionario.

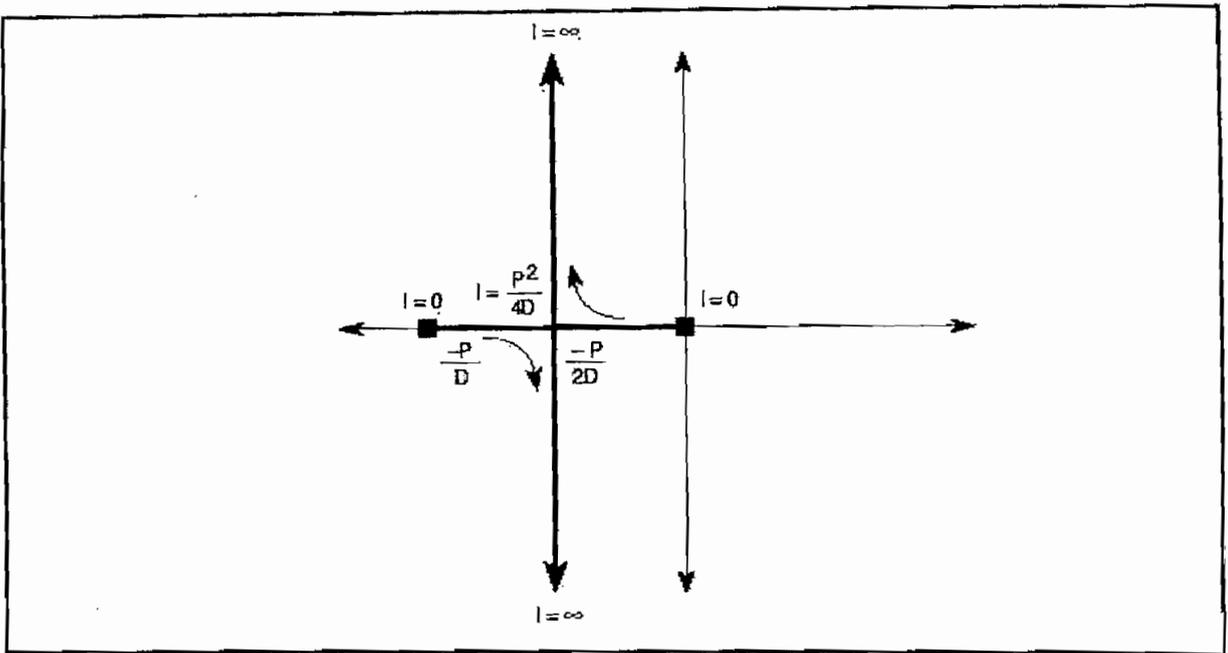


Figura 1-13: Lugar Geométrico de las raíces para una acción integral

Parecería que al añadir un integrador a un lazo éste llegaría a ser una panacea para los dolores de cabeza que produce el control. Sin embargo, mientras se añade un integrador que corrija el error de estado estacionario, se puede tener un impacto negativo sobre los sistemas dinámicos. Este efecto se lo puede apreciar de mejor manera en el dominio del tiempo.

En el caso de un servo motor, inicialmente éste está en "descanso", con un error de posición cero. Un torque se aplica hacia el eje del motor, cambiando su posición y manteniéndola en esta nueva posición. El sistema de control sensa un error en estado estable y trata de regresar el eje hacia la posición comandada. El voltaje de control continúa incrementándose como resultado del error integrado.

Mientras se incrementa el voltaje de control podría causar que el motor se sobrecaliente, esto es un efecto perjudicial aunque no es el único. Si el torque *aplicado* se retira repentinamente mientras la salida del integrador es grande, el eje del motor girará pasándose de la posición del eje deseada hasta que el voltaje de control se desvanezca. Eventualmente se logra una condición de error en estado estable, pero de una manera subamortiguada (potencialmente inaceptable).

Este fenómeno es similar a comprimir un resorte y soltarlo inmediatamente y se denomina en inglés como el efecto "Wind – up".

#### **1.5.4. LUGAR GEOMETRICO DE LAS RAICES PARA UNA ACCION DERIVATIVA**

Como se puede observar en la Figura 1-14, al incrementar el valor de  $D$  desde 0 hasta  $P^2/4I$  hace que los dos ceros se movilicen  $-2I/P$ . Al ocurrir esto, el cero de frecuencia más alta toma un valor que puede generar un adelanto de fase conveniente que se adhiere al retardo de fase introducido por los polos dondequiera que se encuentren en el sistema.

El diseño de la porción derivativa de un controlador PID es muchas veces crítico en el rendimiento de un sistema. Esta señal es diferenciada (sea directa o indirectamente) para generar una señal proporcional a la velocidad de la salida.

La ubicación del diferenciador en un lazo cerrado afecta su rendimiento. La entrada al diferenciador es la señal de error. Como la señal de referencia en la entrada es un componente de la señal de error, cualquier cambio brusco en dicha posición es diferenciada como si fuese una posición de realimentación, resultado de ello, se tiene un efecto igualmente brusco a la salida del algoritmo.

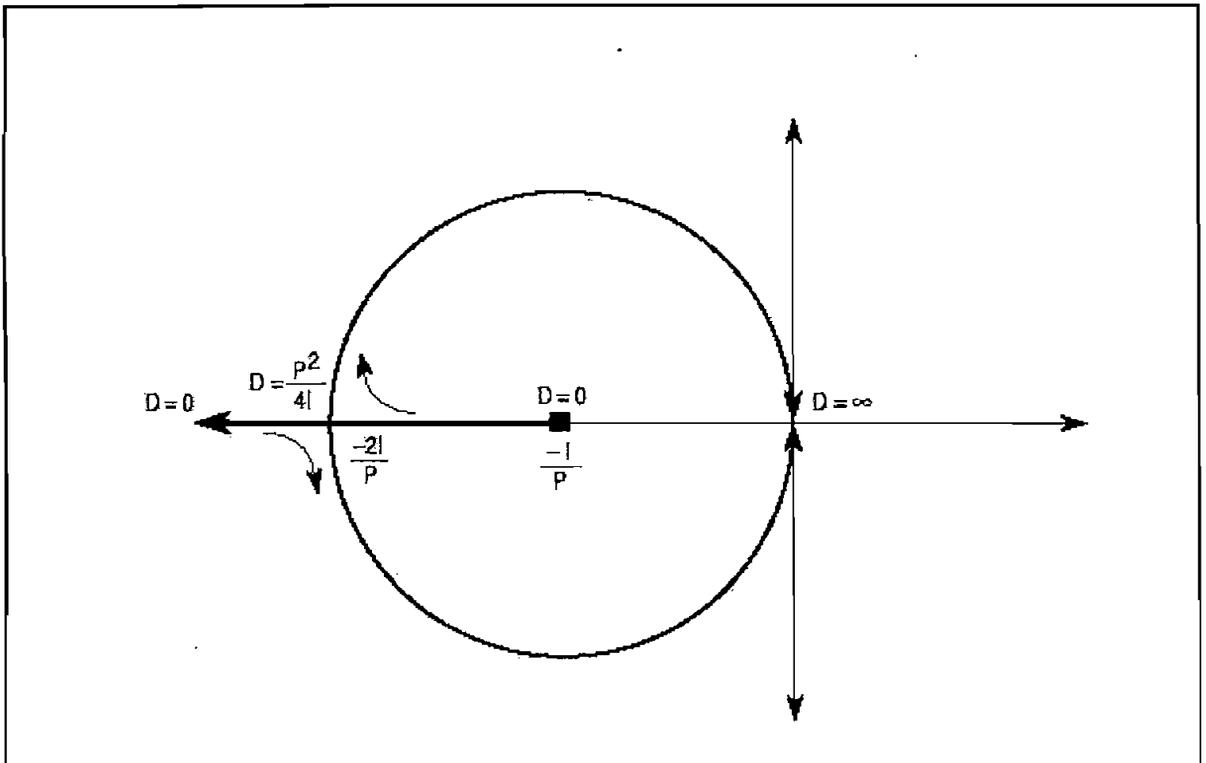


Figura 1-14: Lugar Geométrico de las raíces para una acción derivativa

### 1.5.5. ANALISIS EN EL PLANO Z

Desafortunadamente, debido a que se va a trabajar con controladores digitales en donde se usan sistemas muestreados, la transformada de Laplace o dominio  $s$  no puede ser directamente aplicado como en los cálculos para PID analógicos.

El espacio en tiempo discreto o frecuencia separada conocido también como dominio "z" se desarrolló justamente para los sistemas muestreados. Usando el dominio z, existen aproximaciones a las funciones comúnmente utilizadas representadas mediante la variable z en lugar de la variable s.

Dentro del procesamiento digital de señales, se define a un filtro digital como un algoritmo de cálculo que convierte una secuencia de números de entrada en una secuencia de salida, de modo que las características de la señal se cambien de manera predeterminada. Por esta razón, un controlador digital PID es una forma de filtro digital.

El diagrama de bloques de la estructura discreta del algoritmo PID se puede ver en la Figura 1-15.

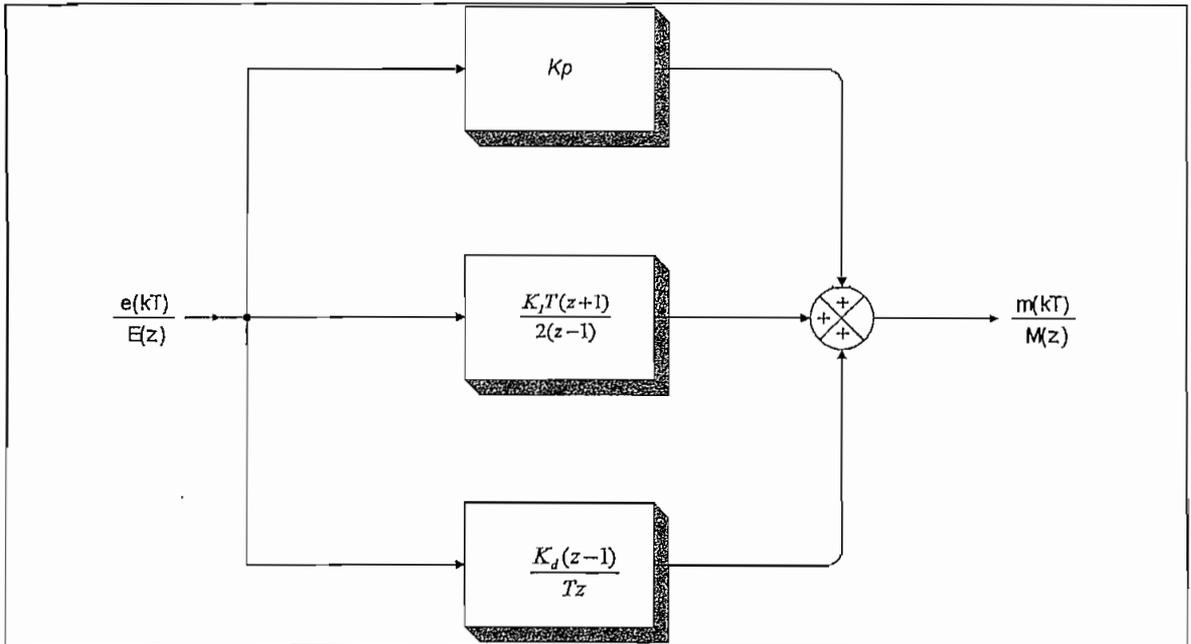


Figura 1-15: Diagrama de bloques de la estructura discreta del algoritmo PID

En el proceso existe un valor de referencia deseado ( $G_d$ ) y la variable a ser controlada  $G(t)$  con su valor actual en el tiempo. El error por tanto es:

$$e(t) = G_d - G(t) \quad \text{Ecuación (1-12)}$$

La variable de salida  $m(t)$  para un controlador PID es:

$$m(t) = K_p e(t) + K_I \int_0^t e(t) dt + K_D \left. \frac{de(t)}{dt} \right|_{t=T} \quad \text{Ecuación (1-13)}$$

donde  $K_p$ ,  $K_I$ ,  $K_D$  son constantes.

Reemplazando el término de  $e(t)$  en la integral tenemos:

$$m(t) = K_p e(t) + K_I \int_0^t (G_d - G(t)) dt + K_D \left. \frac{de(t)}{dt} \right|_{t=T} \quad \text{Ecuación (1-14)}$$

Al introducir el tiempo discreto, se cambia  $t$  por  $kT$ , donde  $k=1,2,\dots,n$  y  $T$  es el período de muestreo. Se define  $t_0 = (k-1)T$ . La integral es evaluada por tanto desde



$$\text{DIFERENCIADOR} = \frac{z-1}{Tz} \quad \text{Ecuación (1-17)}$$

El término de la derivada que actúa sobre la velocidad de cambio del error, puede generar ruido en un período, por tanto se utiliza un “promedio de n puntos” que tiene el comportamiento de un filtro pasabajas y que limita la velocidad de cuantificación del ruido, se promedia la información de velocidad sobre un número de muestras para obtener una resolución fina.

Sin embargo este filtro o promedio de n puntos produce un retardo proporcional al número de conteos. Un filtro práctico para los microcontroladores aproxima el término derivativo mediante el promedio de cuatro puntos tomados en forma central como indica la Figura 1-17.

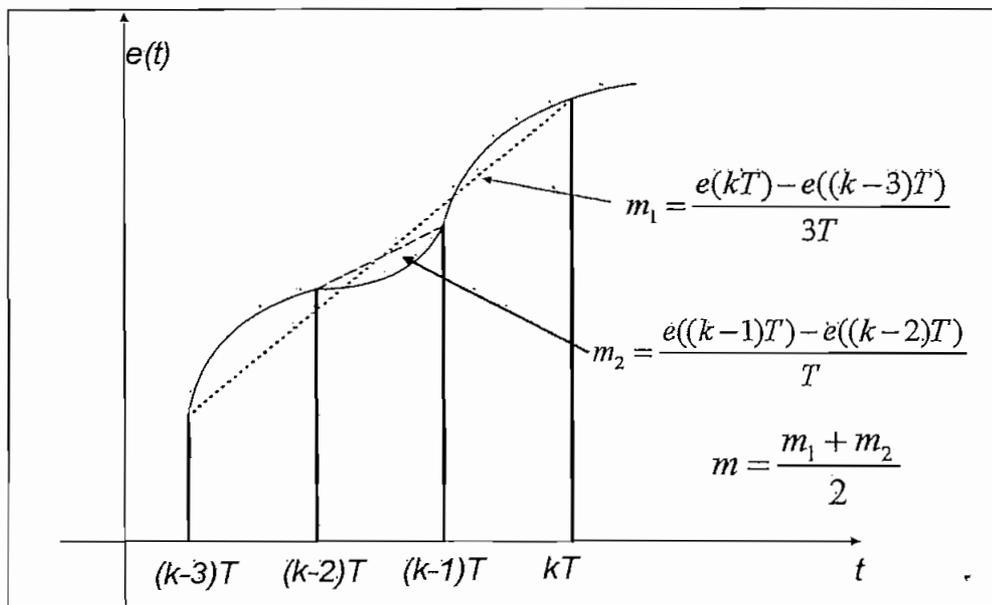
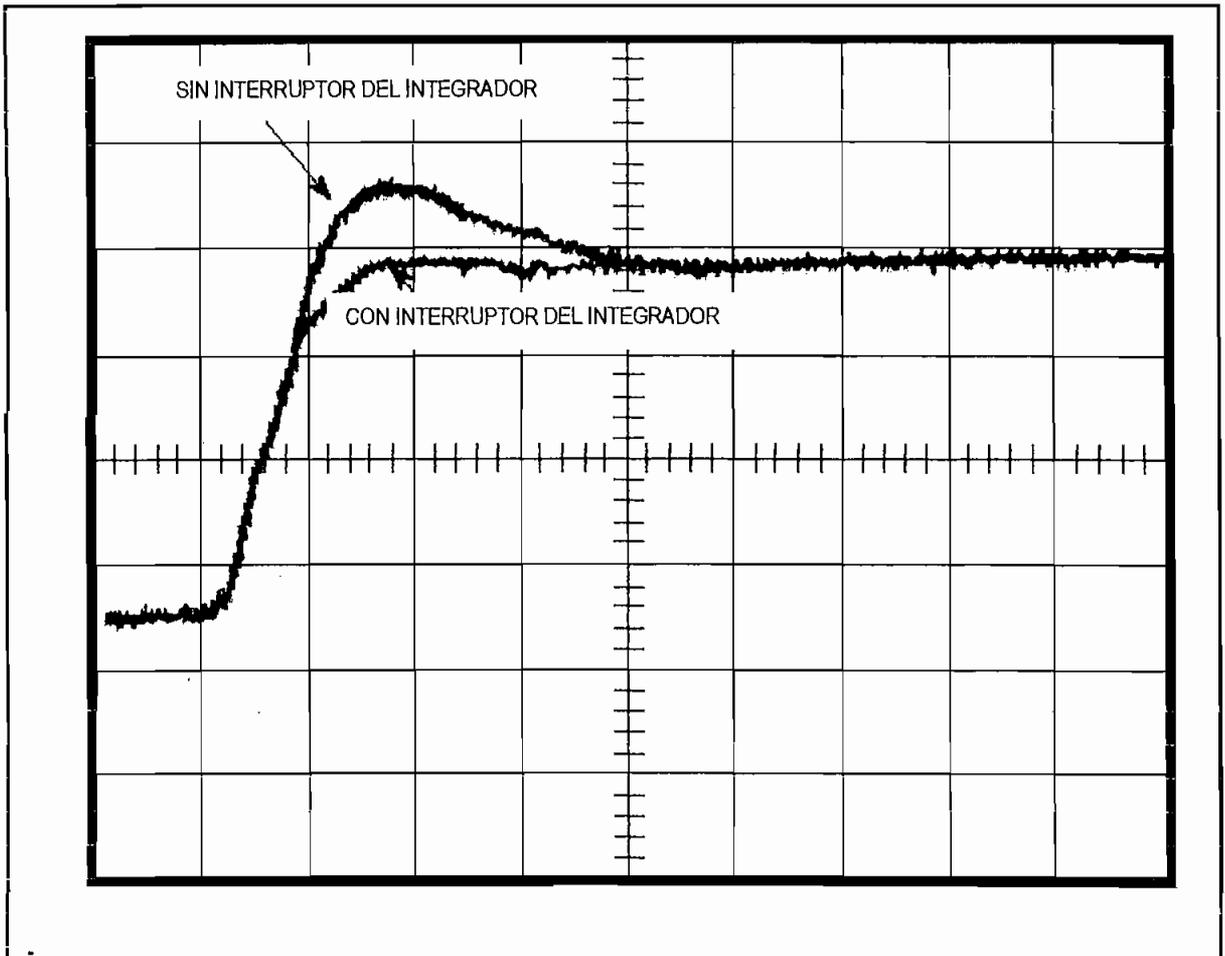


Figura 1-17: Diagrama que muestra el promedio de 4 puntos para la acción diferencial

$$D(kT) = m = \frac{1}{2} \left[ \frac{e(kT) - e((k-3)T)}{3T} + \frac{e((k-1)T) - e((k-2)T)}{T} \right] \quad \text{Ecuación (1-18)}$$

El integrador podría tener muchas características asociadas en el amortiguamiento del sistema. Una de ellas es la habilitación por medio de software de la acción integral únicamente cuando se requiera (Figura 1-18). Se asume que no es necesario el uso de la acción integral cuando la magnitud de la

velocidad se incrementa sobre un valor de umbral específico. Es decir, no se integra a la señal de error en todo el lapso del cambio de posición, evitando de esta forma sobrepicos cuando se detecta error.



**Figura 1-18: Habilitación del interruptor asociado a la acción integral**

La magnitud de salida del integrador está limitada para eliminar el efecto Wind-up asociado con los integradores PID. Este límite depende de la aplicación, y debería ser configurada para un valor mínimo que se requiere para generar una salida suficiente que sobrepase cualquier resistencia de carga anticipada.

En teoría, el proceso de muestreo está modelado como una serie de funciones impulso, lo que implica un tiempo de cálculo del filtro PID de cero. En realidad, la cantidad de tiempo que se requiera para ejecutar un algoritmo de filtro digital se debe contar para que éste se lo introduzca como retardo de fase en el sistema.

Por tanto, la forma como puede ser ejecutado el algoritmo directamente en el microprocesador es:

$$m(t) = K_p e(t) + K_I T \left( Gd - \frac{G(kT) + G[(k-1)T]}{2} \right) + \frac{K_D}{6T} [e(kT) - e(k-3) + 3(e(k-1) - e(k-2))] \quad \text{Ecuación (1-19)}$$

Estas expresiones no son las únicas representaciones de estas funciones, pero son las más ampliamente utilizadas en las aplicaciones de control.

## 1.6. EL CONTROLADOR UNIVERSAL

El término "Universal" hace referencia a que el controlador sea capaz de recibir una señal de entrada normalizada de corriente o de voltaje de un transductor, utilice un algoritmo de control apropiado (on-off, proporcional, PID), tenga salidas como relés electromagnéticos, SSR o señal de salida análoga, y salidas adicionales o requerimientos del sistema como por ejemplo display para la visualización de la variable y/o valor de referencia, alarmas, comunicación con computador, etc., que sean configurables por el usuario. Para efectos del diseño, motivo de este proyecto, se tomará en cuenta esta definición.

A veces se denominan controladores universales por tener entradas universales de termocupla, RTD, de corriente y/o de voltaje.

Otros controladores que han sido denominados universales, se caracterizan por tener una alimentación universal. Es decir trabajan en rangos amplios tanto de AC como de DC.

### 1.6.1. CONTROLADOR UNIVERSAL DISEÑADO

Los parámetros de diseño para el controlador se basaron en los controladores de compañías especializadas en la industria como Omega, Honeywell, Newport Electronics, etc.

A continuación se tiene una breve descripción de las características de dos controladores.

# Honeywell



UDC1000:  
48 X 48 mm



UDC1500:  
96 X 48 mm

<b>Descripción del Producto:</b>	UDC 1000 y 1500
<b>Tipos de Señales de Entrada:</b>	Termopares, termorresistencias, mV, mA
<b>Entradas Digitales:</b>	NO DISPONIBLE
<b>Salidas Analógicas:</b>	3 (0-20 mA, 4-20 mA, 0-5 V, 0-10 V)
<b>Control de Salidas Digitales:</b>	Hasta 2
<b>Alarmas de Salidas Digitales:</b>	Hasta 2
<b>Tipos de Salidas Digitales:</b>	Relé, driver SSR
<b>Alimentación:</b>	90 a 264 Vca, 24 a 48 Vca/cc
<b>Precisión :</b>	± 0,25% del alcance
<b>Velocidad de Muestreo:</b>	4 veces/s
<b>Lazos:</b>	1
<b>Algoritmos de Control:</b>	ON-OFF/PID
<b>Ajuste:</b>	Preajuste y Autoajuste
<b>Algoritmo de Salida:</b>	Tiempo, corriente
<b>Software de Configuración:</b>	SI
<b>Comunicación:</b>	RS485 ASCII o Modbus RTU

Figura 1-19: Controladores Honeywell

# NEWPORT



i16: 48 X 48 mm



i8: 96 X 48 mm

<b>Descripción del Producto:</b>	Serie i16 y Serie i8
<b>Tipos de Señales de Entrada:</b>	Entradas Universales: Termocuplas, RTD, Corriente, Voltaje
<b>Entradas Digitales:</b>	NO DISPONIBLE
<b>Salidas Analógicas:</b>	Voltaje y Corriente
<b>Control de Salidas Digitales:</b>	Hasta 2
<b>Alarmas de Salidas Digitales:</b>	Hasta 2
<b>Tipos de Salidas Digitales:</b>	Relé, SSR, Pulso DC
<b>Alimentación:</b>	90 a 240 Vca, 20-36 Vdc
<b>Precisión</b>	± 0,03% proceso de lectura
<b>Velocidad de Muestreo:</b>	3 veces/s
<b>Lazos:</b>	1
<b>Algoritmos de Control:</b>	ON-OFF/PID
<b>Ajuste:</b>	Preajuste y Autoajuste
<b>Display:</b>	Programable
<b>Software de Configuración:</b>	SI
<b>Comunicación:</b>	RS232, RS485 ASCII o Modbus RTU, protocolo TCP/IP

Figura 1 -20: Controladores NewPort

Con esta información, tomada como ejemplo, se desarrolló el controlador *universal* con las siguientes características:

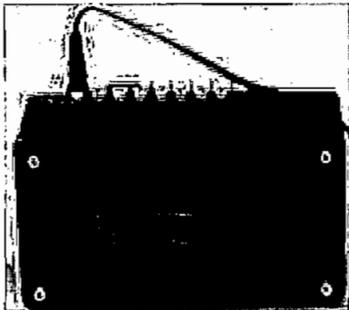
 <p>70 x 50 x 150 mm</p>	<b>Descripción del Producto:</b>	Controlador Universal
	<b>Número de Entradas</b>	1
	<b>Tipos de Señales de Entrada:</b>	Corriente, Voltaje
	<b>Salidas Analógicas:</b>	Voltaje
	<b>Control de Salidas Digitales:</b>	2
	<b>Alarma</b>	1
	<b>Tipos de Salidas Digitales:</b>	SSR
	<b>Alimentación:</b>	5 Vdc
	<b>Precisión</b>	± 0,2% proceso de lectura
	<b>Velocidad de Muestreo:</b>	37 veces/s aprox.
	<b>Algoritmos de Control:</b>	ON-OFF(con lazo de histéresis) /Proporcional /PID /Modo manual de Control
	<b>Acción de Control</b>	Inversa / Directa
	<b>Ajuste:</b>	Preajuste
	<b>Display:</b>	LCD alfanumérico de dos líneas con contraste variable
	<b>Software de Configuración:</b>	SI
<b>Comunicación:</b>	RS232 a distintas velocidades	

Figura 1 -21: Características del Controlador Universal Diseñado

Adicionalmente a las funciones de controlador, el equipo puede ser utilizado también como monitor, en donde se lee únicamente la variable de proceso.

- El controlador puede trabajar con mediciones de procesos en voltaje DC o corriente DC. El tipo de control se puede seleccionar entre on-off y PID.
- Entre sus características posee un display alfanumérico de cristal líquido LCD de 32 caracteres cuyo contraste se puede cambiar dentro de la configuración. Utiliza cuatro teclas para el ingreso a un menú de configuración del controlador.
- Posee dos salidas de SSR's o una salida analógica de voltaje con una salida SSR. Tiene comunicación serial RS232.
- La frecuencia de muestreo para el controlador diseñado es de aproximadamente 10 veces la frecuencia que utilizan los controladores de Newport y Honeywell. A 37,5 Hz el microcontrolador procesa toda la información para el control.
- Posee un filtro generado por software a la entrada del controlador. De esta manera se previene el ingreso de interferencias sobrepuestas a la señal.
- El controlador permite seleccionar entre modo manual y modo automático para el control de la salida y se fijan los límites máximos y mínimos de salida.
- La cubierta externa no está dimensionada bajo normas DIN. En su lugar utiliza una caja prefabricada para proyectos de RadioShack®. (70 X 50 X 150 mm).
- Tiene un sistema de Standby, el cual deshabilita las salidas en caso de emergencia. En este estado puede ser reconfigurado el controlador.

## **CAPITULO 2. DISEÑO DE HARDWARE**

## CAPITULO 2.

### DISEÑO DE HARDWARE

Una vez determinadas las necesidades y prioridades en la implementación del controlador, se trata de conseguir un microcontrolador que integre el mayor número de periféricos con el fin de simplificar en espacio y tiempo el diseño del controlador.

Entre la gama de microcontroladores, se eligió el MC68HC908GP32 de Motorola, debido a que presenta características técnicas y periféricos adecuados para el desarrollo del controlador. Entre las características se tiene 32Kbytes de memoria FLASH para programación, 8 canales de entrada para el conversor análogo-digital, 4 canales de temporizadores de 16 bits que permiten generar señales PWM, etiquetas especiales para teclado, módulos de comunicación serial y periférica serial, módulos de ahorro de energía, resistencias de pullups, etc.

Adicionalmente y una vez decidido con qué micro trabajar, se buscó una tarjeta adecuada para proyectos que incluya este micro y que posea ciertas particularidades como el puerto de comunicación serial en DB9, memoria externa para almacenar parámetros, conectores para LCD, teclado, etc. Se escogió la tarjeta MINIMAX/908-C de Bipom.

Aparte de la tarjeta se usa únicamente 4 pulsadores de teclado que permiten navegar por el menú a ser visualizado en un Display de Cristal Líquido, dos relés de estado sólido de 3A que servirán de salidas del controlador y operan en encendidos y apagados rápidos.

#### 2.1. TARJETA PRINCIPAL

La tarjeta principal MINIMAX/980-C tiene como base al microcontrolador MC68HC908GP32 de Motorola. Además posee una memoria EEPROM de 4Kbits. Conectores para teclado y LCD; un bus de expansión que permite tener una

interfaz con otros periféricos; un puerto serial RS232 y un PIC para la programación del microcontrolador y el contraste del display.

Las dimensiones que tiene la tarjeta MINIMAX/980-C son las siguientes: 5.97 x 6.10 centímetros (2.35 x 2.40 pulgadas). El rango de temperatura al cual puede operar está entre 0° a 70 °C. El rango de almacenamiento de datos entre -40 °C a + 85 °C.

El diagrama de bloques de la tarjeta MINIMAX/908-C se muestra en la Figura 2-1.

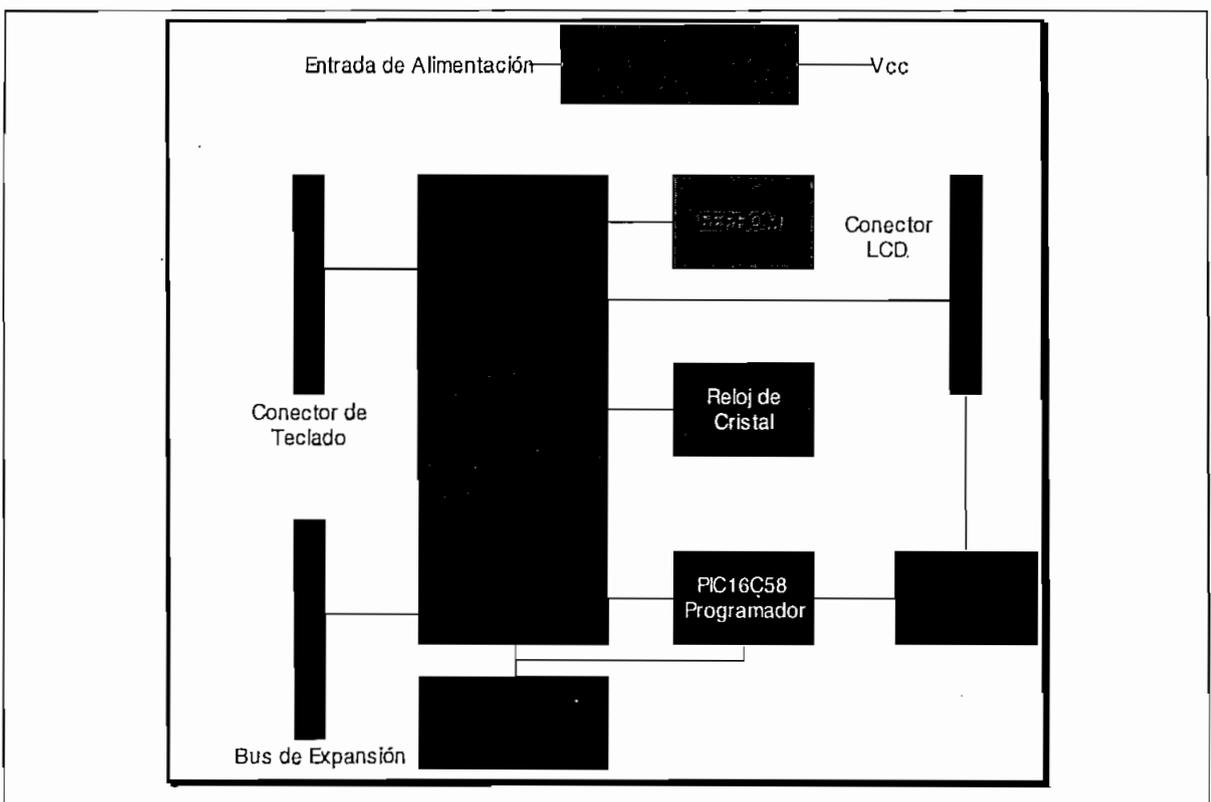
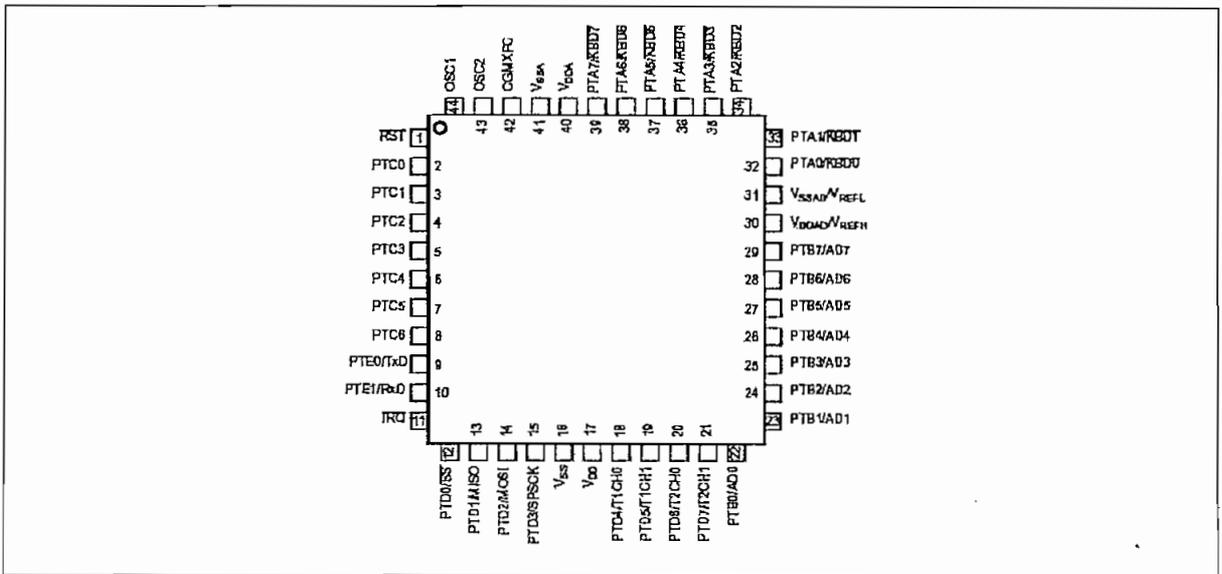


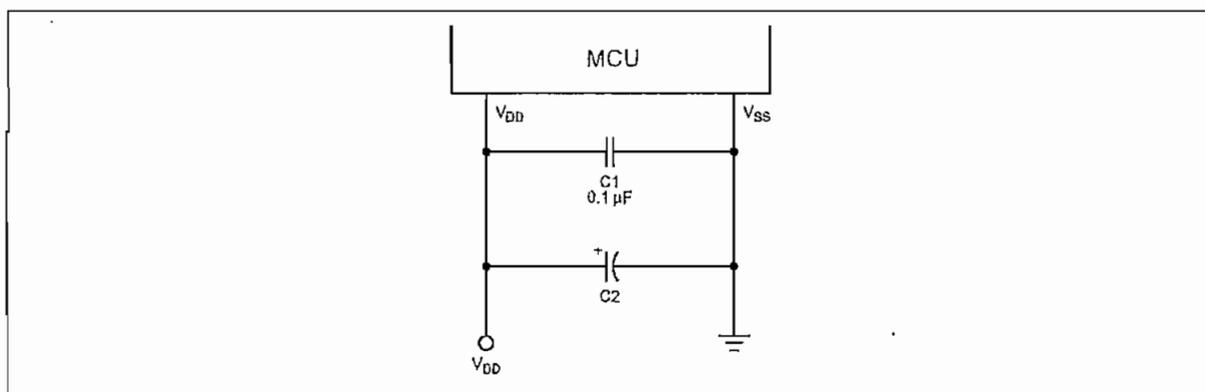
Figura 2-1: Diagrama de Bloques de la Tarjeta MINIMAX/908-C

### 2.1.1. EL MICROCONTROLADOR

Los puertos del microcontrolador y las líneas de alimentación están disponibles sobre un bus de expansión de 20 pines para las interfaces con periféricos u otros circuitos externos. El MC68HC908GP32 tiene cinco puertos: A (PTA), B (PTB), C (PTC), D (PTD) y E (PTE).

Los pines correspondientes al puerto PTC están dispuestos en el conector del LCD. Los pines del puerto PTA están dispuestos en el conector de teclado que además pueden utilizarse como pines de entrada-salida bi-direccionales para propósito general. Tanto el puerto PTB como el PTD se encuentran distribuidos en el conector de expansión. De igual manera los pines del puerto PTD pueden utilizarse como pines de entrada-salida o con características especiales tales como interfaz serial periférico (SPI) y entrada-salida de temporizadores. En cuanto al puerto PTB, éste puede funcionar sea para entrada-salida de propósito general o como entrada para el conversor Análogo-Digital. Los pines del puerto PTE sirven de interfaz para la comunicación serial. También está disponible en el conector de expansión el pin de interrupción externa (IRQ). En la Figura 2-2 se muestra la distribución de los pines del microcontrolador.





**Figura 2-3: Circuito de Bypass**

Los pines OSC1 (44) y OSC2 (43) corresponden a las conexiones del circuito oscilador o reloj. El circuito utiliza un reloj de 9,8304 MHz.

Para inicializar al micro se ingresa un 0 lógico en el pin RST. Esta tarjeta no permite tener acceso a dicho pin y únicamente está habilitado para propósitos de programación.

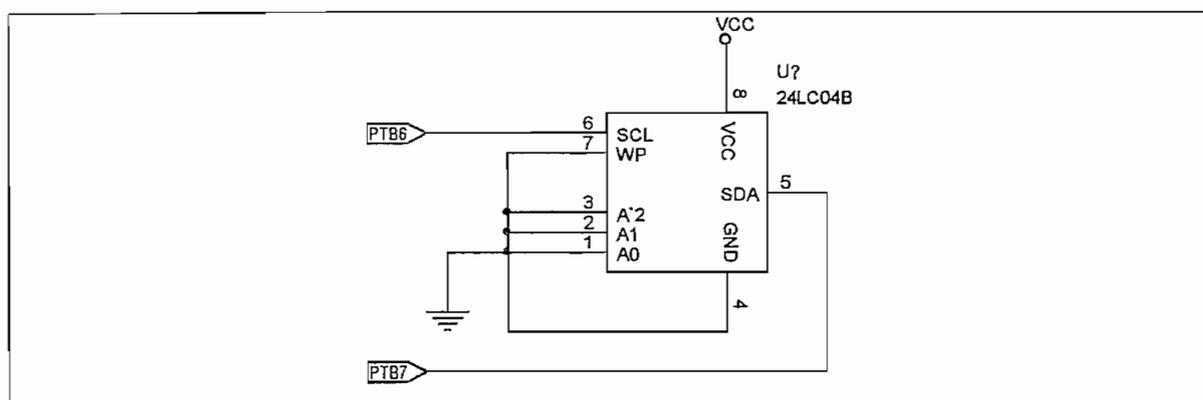
$V_{DDA}$  (40) y  $V_{SSA}$  (41) son los pines de alimentación para la porción analógica del módulo generador de reloj.

$V_{DDAD}$  (30) y  $V_{SSAD}$  (31) corresponden a la alimentación del convertor análogo-digital y son los voltajes de referencia para el convertor.

### 2.1.2. MEMORIA EEPROM

La tarjeta MINI-MAX/908-C incluye una Memoria Programable de Solo Lectura con Borrado Eléctrico (EEPROM) con capacidad de almacenamiento de 512 bytes. Con esta memoria se almacenan los parámetros de control, la identificación del usuario, registros, etc.

La comunicación entre el microcontrolador (MCU) y la EEPROM se la hace a través de los pines PTB6 y PTB7 que van conectados a los pines SCL y SDA de la memoria externa para tener una comunicación serial sincrónica de tipo I2C (Figura 2-4).



**Figura 2-4: Diagrama de Conexión del Micro con la Memoria EEPROM**

### 2.1.3. PROGRAMADOR PIC

El microcontrolador MC68HC908GP32 se programa de manera remota a través de comunicación RS232 usando para ello el PIC16C58 como programador. El pin PTA0 (32) está dedicado a la interfaz de programación del MC68HC908GP32 con el PIC16C58. Es decir, es un puente entre el puerto RS232 de una computadora y el microcontrolador.

Para la programación el computador cambia de estado al microcontrolador a un modo que se denomina Monitor. En esta operación la PC utiliza la línea RTS del puerto de comunicación como "0" lógico (+12V aparece en el pin RTS del puerto COM). El PIC16C58 chequea el estado de esta señal todo el tiempo. Se termina por tanto el modo Monitor al elevar el pin RTS a 1 lógico.

Las conexiones del PIC en la tarjeta MINIMAX/908- C se muestran en la Figura 2-5.

El PIC tiene una segunda aplicación. Genera una señal PWM constante que se encarga del control del contraste de un LCD y se puede variar mediante comunicación serial sincrónica I2C por los pines PTB6 y PTB7 del MCU. Esta señal se encuentra dispuesta en el conector para LCD de la tarjeta principal y se denomina PWM.

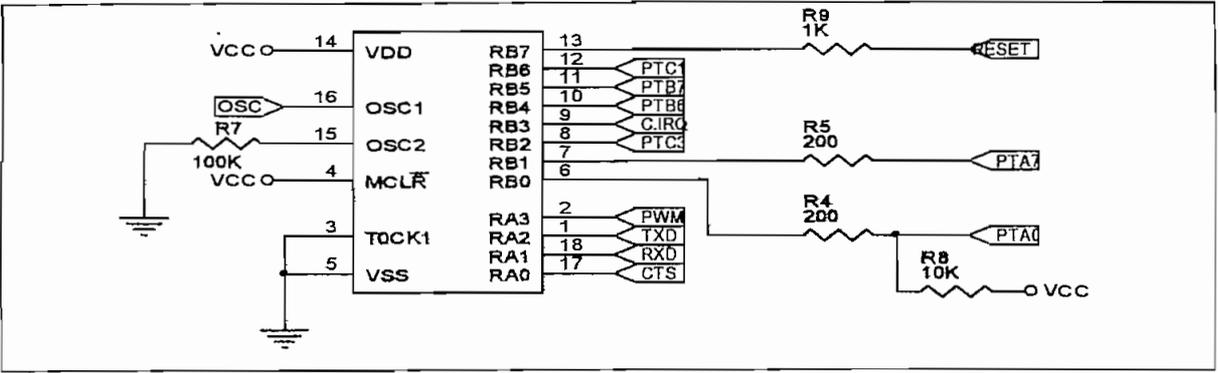


Figura 2-5: Diagrama de Conexiones del Programador PIC

### 2.1.4. CONECTORES

En la Figura 2-6 se puede observar la ubicación de los elementos en la tarjeta MINIMAX/908-C de Bipom.

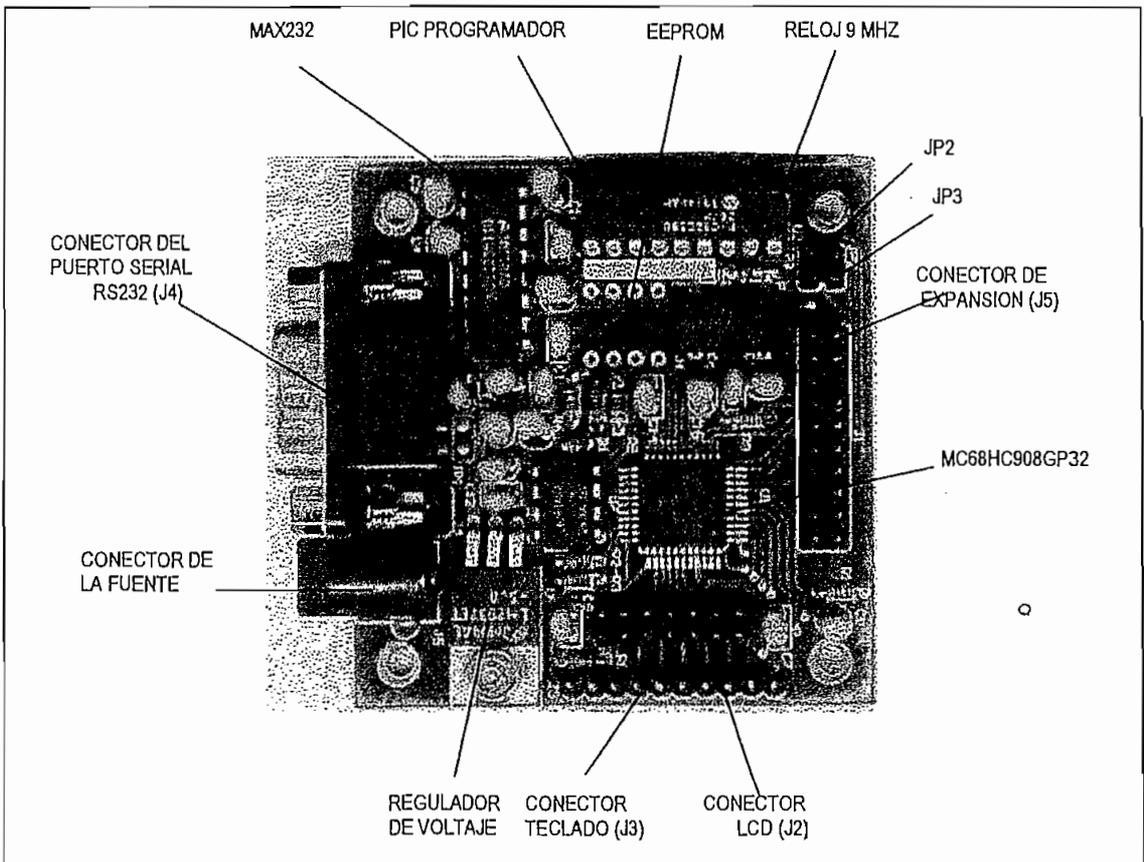


Figura 2-6: Ubicación de los Elementos en la tarjeta MINIMAX/908-C

La disposición de los pines que van al conector LCD (J2) se detalla en la Tabla 2-1.

**Tabla 2-1: Conector LCD (J2)**

SEÑAL	PIN	PIN	SEÑAL
TIERRA	1	2	Vcc
Vee	3	4	PTC0
PTC1	5	6	PTC2
TIERRA	7	8	TIERRA
TIERRA	9	10	TIERRA
PTC3	11	12	PTC4
PTC5	13	14	PTC6

En el conector de teclado (J3) se utilizan cuatro pines del puerto A (PTA0-PTA3) como teclas de manejo del controlador. La disposición de los pines correspondientes al conector de teclado se encuentra en la Tabla 2-2.

**Tabla 2-2: Conector de Teclado (J3)**

SEÑAL	PIN
PTA0	1
PTA1	2
PTA2	3
PTA3	4
PTA4	5
PTA5	6
PTA6	7
PTA7	8
TIERRA	9
Vcc	10

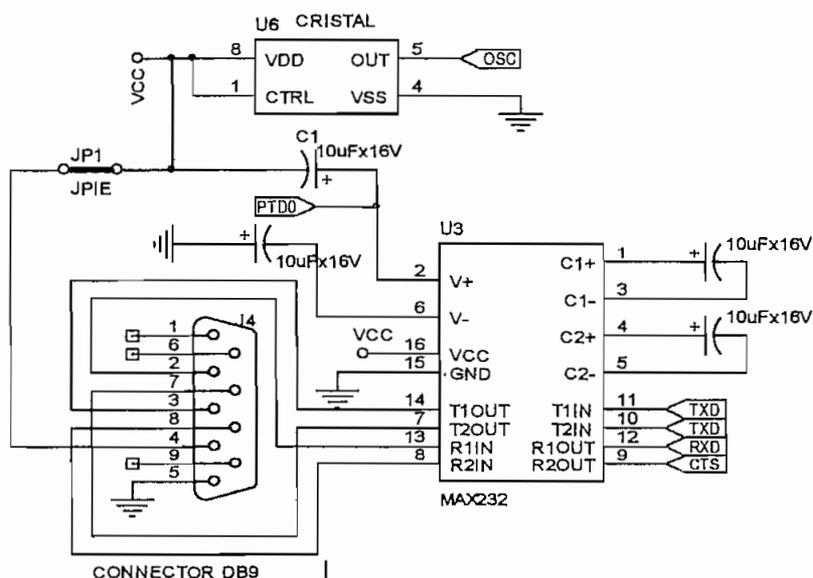
El MINI-MAX/908-C posee un puerto serial RS232 para comunicación asincrónica, el cual convierte a los pines RXD y TXD desde/a los niveles RS232. El puerto RS232 es accedido a través de un conector DB-9 macho (J4). Se pueden conectar computadores, módems, terminales Hand-held y otros elementos

seriales al puerto RS232. Las líneas de control MODEM CTS/RTS están provistas en el puerto RS232. El CTS se utiliza con servidores externos tales como un computador para poner al MINI-MAX/908-C en modo monitor.

En la Tabla 2-3 se describe la distribución de pines y en el diagrama de la Figura 2-7 las conexiones correspondientes a la comunicación serial RS232.

**Tabla 2-3: Distribución de los Pines para comunicación RS232 (J4)**

SEÑAL	PIN
NO ESTA CONECTADO	1
PIN DE RECEPCION (RXD)	2
PIN DE TRANSMISION (TXD)	3
Vcc	4
TIERRA	5
NO ESTA CONECTADO	6
RTS	7
CTS	8
NO ESTA CONECTADO	9



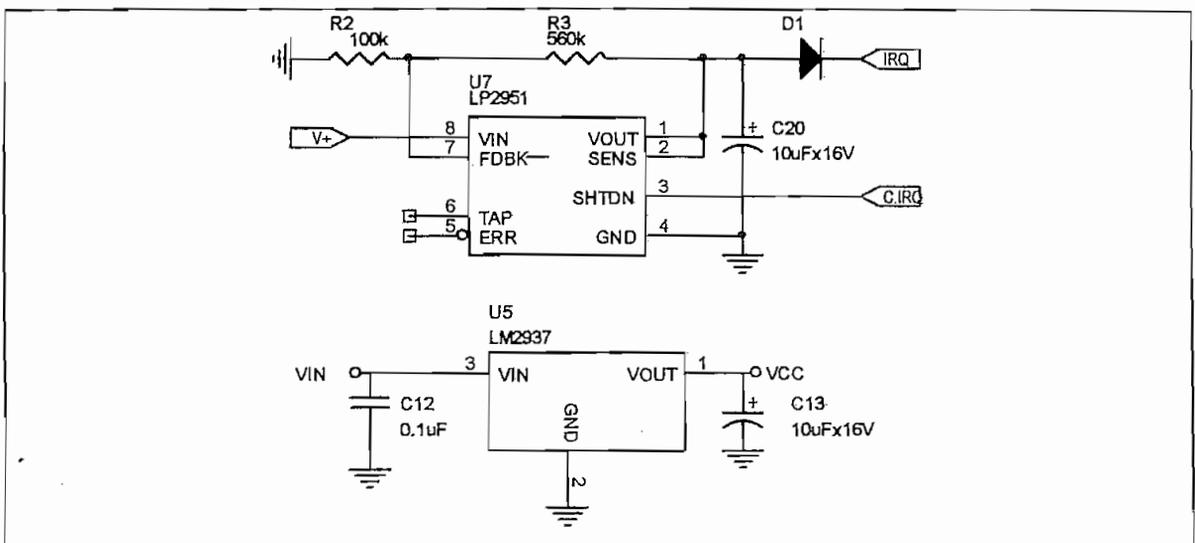
**Figura 2-7: Diagrama de Conexiones correspondiente a la Comunicación Serial RS232**

La distribución de los pines en el conector de expansión se indica en la Tabla 2-4.

**Tabla 2-4: Distribución de Pines en el conector de expansión (J5)**

SEÑAL	PIN	PIN	SEÑAL
PTD7	20	19	PTD6
PTD5 O IRQ	18	17	PTD4
PTD3	16	15	PTD2
PTD1	14	13	PTD0
PTB0	12	11	PTB1
PTB2	10	9	PTB3
PTB4	8	7	PTB5
PTB6	6	5	PTB7
Vcc	4	3	TIERRA
Vcc	2	1	TIERRA

La tarjeta incluye además un regulador de voltaje de 5 V (LM2937) en la estructura y un circuito que permite elevar el pin IRQ de interrupción externa hasta 8 V para el uso en la programación (LP2951).



**Figura 2-8: Regulador de Voltaje (LM2937) y circuito (LP2951)**

El diagrama esquemático completo de las conexiones que presenta la tarjeta MINIMAX/908-C se muestra en la Figura 2-9.

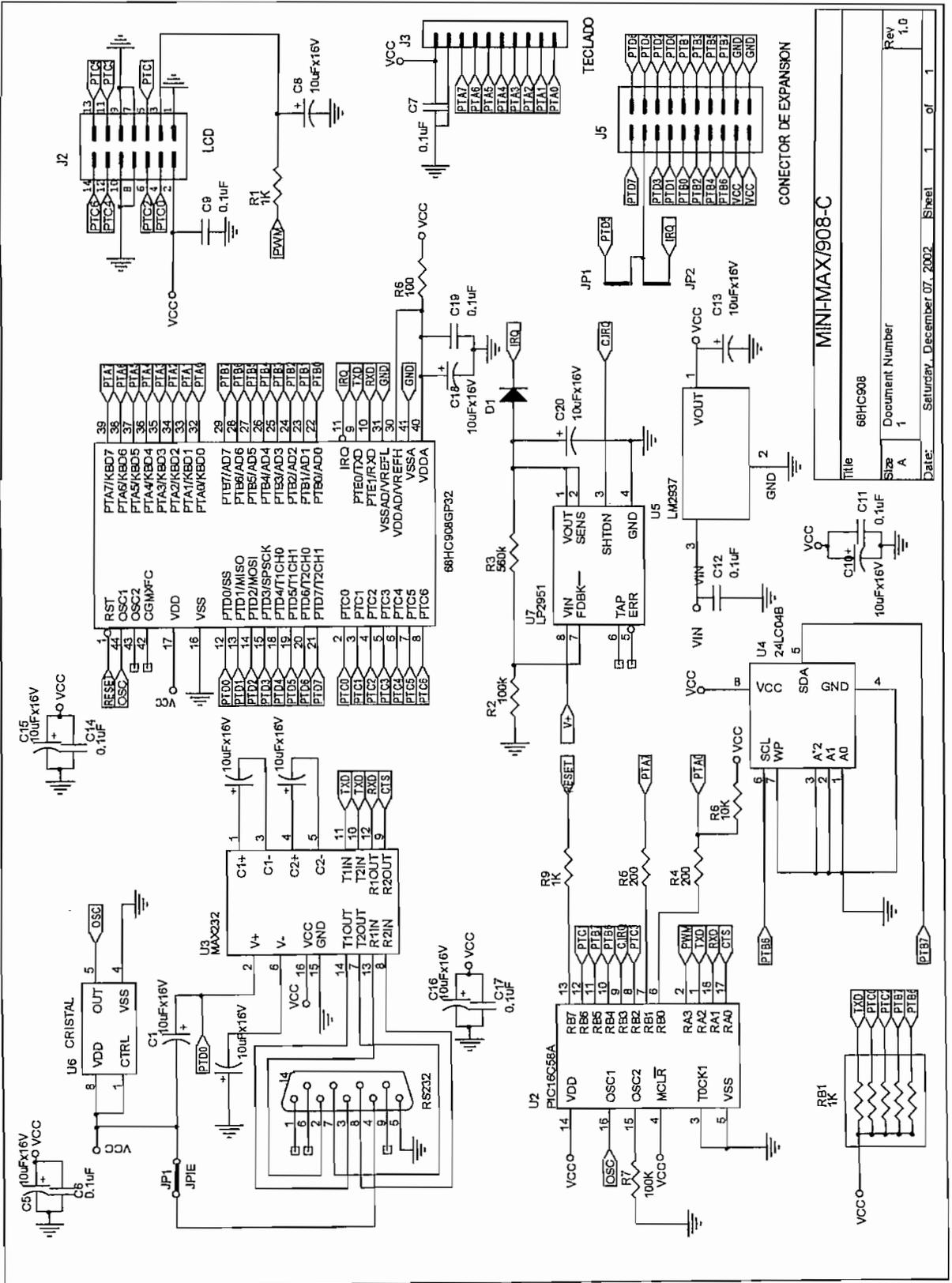


Figura 2-9: Esquemático de la Tarjeta MINIMAX/908-C

Title		68HC908	
Size	A	Document Number	1
Rev	1.0	Date:	Saturday, December 07, 2002
Sheet		1	of 1

Title		MINI-MAX/908-C	
Size	A	Document Number	1
Rev	1.0	Date:	Saturday, December 07, 2002
Sheet		1	of 1

## 2.2. PERIFERICOS Y COMPONENTES EXTERNOS

A la tarjeta MINIMAX/980-C se agregan los componentes de entrada y salida necesarios para completar la estructura del controlador. En la Figura 2-10 se indica los puertos del microcontrolador que manejan a cada elemento externo.

El puerto A se encarga de recibir cuatro entradas de teclado y del manejo de los dos relés de estado sólido. El puerto B recibe las señales de corriente o voltaje provenientes de un transductor. El puerto C está dedicado al manejo del display de cristal líquido. El puerto D genera una señal de tipo PWM para la salida analógica del control PID. Finalmente el puerto E se encarga de la comunicación serial RS232.

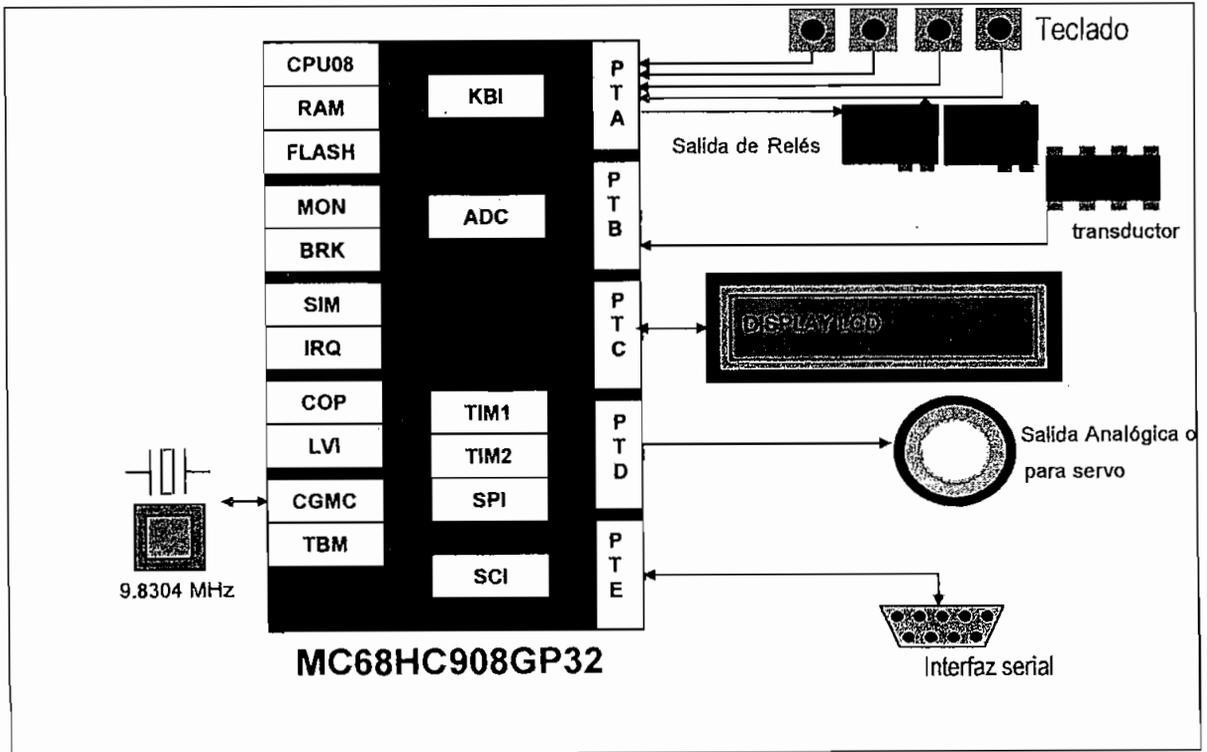


Figura 2-10: Componentes de Entrada y Salida del Controlador

### 2.2.1. DESCRIPCION

Para complementar los circuitos del controlador se utilizan dos tarjetas periféricas adicionales (Figura 2-11). En la primera están montados los conectores de LCD, el teclado y los dos relés de estado sólido. Se comunican a través de buses de

datos independientes. La segunda tarjeta tiene un conector especial para el bus de expansión de la tarjeta MINIMAX/908-C. En esta tarjeta se incluye el filtro RC a la señal PWM para obtener la salida analógica de voltaje y una configuración para transformar de corriente a voltaje en la entrada del convertor A/D para transductores de corriente.

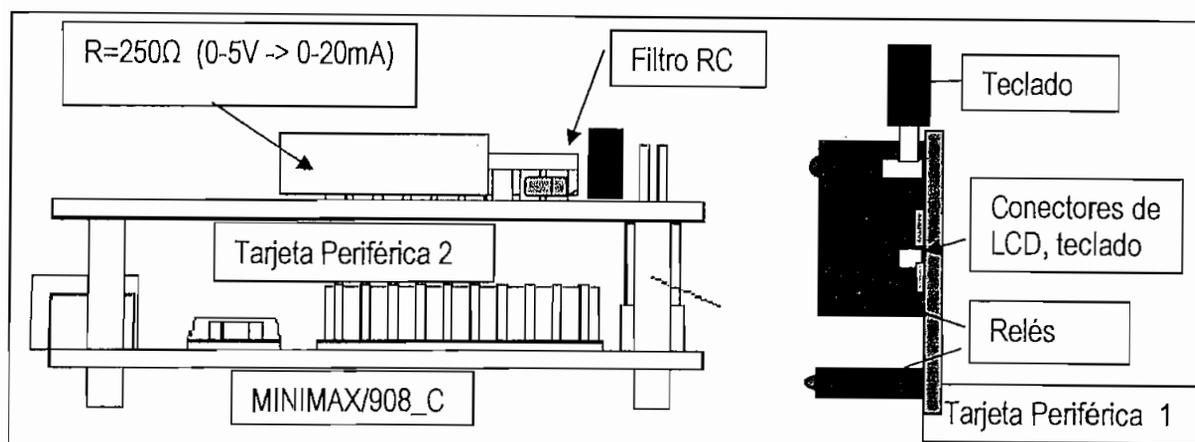


Figura 2-11: Tarjetas Periféricas

El teclado se conecta a la tarjeta MINIMAX/908-C mediante un bus de cuatro hilos y la designación se puede observar en la Tabla 2-5.

Tabla 2-5: Designación del Teclado

TECLA	ETIQUETA	COLOR	PIN
	TECLA MENU	AZUL	PTA3
	TECLA UP	MORADO	PTA2
	TECLA DOWN	GRIS	PTA1
	TECLA ENTER	BLANCO	PTA0

El módulo de Display LCD es de dos líneas y 16 caracteres alfanuméricos cada una, la pantalla es de color azul y tiene 7 cm x 2.5 cm de dimensiones.

Los relés de estado sólido utilizados en el controlador funcionan con un voltaje de control de 4 – 32 Vdc y su salida soporta 3 A @ 240 Vac.

La salida analógica es una señal PWM generada a 120 Hz. Esta señal es filtrada para obtener una señal dc. La resistencia mínima permitida en cada pin es de 250

ohmios para garantizar una corriente máxima de 20mA que soporta el microcontrolador. Para el diseño del filtro colocado en la salida analógica se utiliza una resistencia de 270 ohmios de acuerdo a la siguiente condición:

$$C \gg \frac{1}{2\pi f R} \quad \text{Ecuación (2-1)}$$

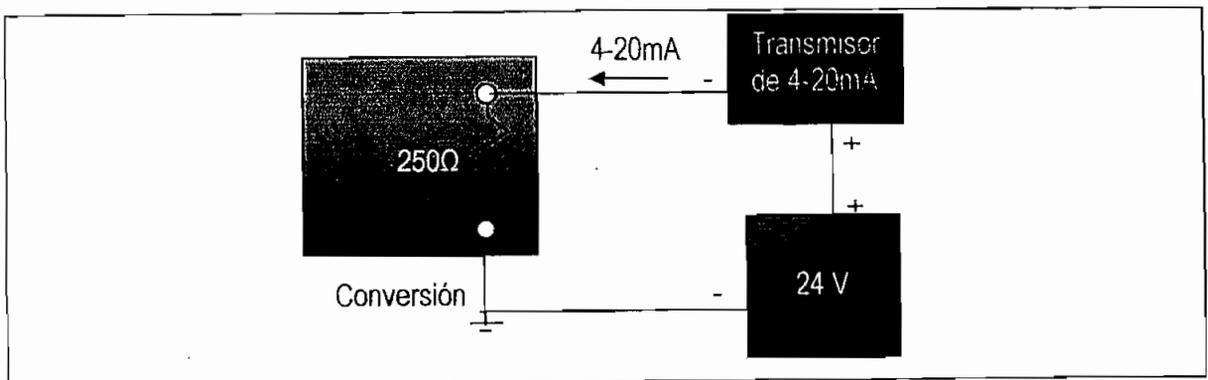
Como la frecuencia es de 120 Hz, el capacitor según la ecuación (2-1) tiene que ser mayor a los 50uF. Se utilizó un capacitor de 330uF.

La mayoría de los transmisores envían la señal analógica a los instrumentos industriales en forma de corriente en un rango de 4-20mA. El controlador transforma esta señal de corriente en una de voltaje con una resistencia de 250 ohmios. Aplicando la Ley de Ohm de  $V=IR$ , el controlador tiene la señal de 1 a 5V. (Tabla 2-6).

**Tabla 2-6: Conversión de Corriente a Voltaje**

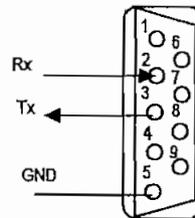
Señal en mA	X	multiplicada por la resistencia	=	convertido a voltaje
4mA	X	250	=	1V
20mA	X	250	=	5V

Se requiere de una fuente adicional de voltaje normalmente de 24 Vdc que alimente al transmisor como se muestran en la Figura 2-12.



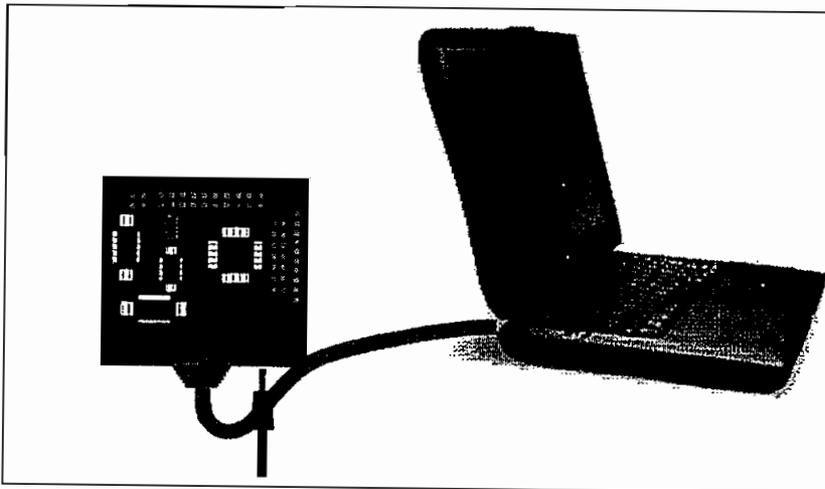
**Figura 2-12: Transmisor de 4-20mA**

La interfaz de comunicación serial entre el controlador y un computador es de tipo RS232 (punto a punto). El controlador opera con RS232 full-duplex por lo que usa tres hilos: recepción Rx, transmisión Tx y una tierra común GND. El largo del cable es limitado a 15 m. La mayoría de los computadores tiene un puerto RS232 para comunicación digital y usualmente utilizan un conector DB25 o DB9. En la Figura 2-13 se muestra el conector hembra DB9.



**Figura 2-13: Conector hembra DB9**

En la Figura 2-14 se puede observar la interfaz del controlador con un computador personal.



**Figura 2-14: Interfaz de comunicación entre el controlador con el computador**

**CAPITULO 3.**  
**DESARROLLO DEL SOFTWARE**

## CAPITULO 3.

### DESARROLLO DEL SOFTWARE.

El software que maneja el controlador fue diseñado en base a lenguaje de programación C. Se puede agregar que es un lenguaje de programación de alto nivel muy poderoso y flexible. Sin embargo, para producir un código de máquina eficiente, el programador debe ser cuidadoso en la construcción del programa y además poner atención en la implementación detallada del mismo.

Se ha dividido al programa en tres partes principales:

- Software de operación. Se encarga del funcionamiento interno del microcontrolador con los diferentes periféricos.
- Software de interfaz con el usuario. Se encarga de la visualización de la información en el display y del ingreso de la información a través del teclado.
- Software de interfaz con el computador. Permite la configuración y la visualización de los parámetros a través del computador.

Las herramientas de desarrollo permiten dividir la programación en distintos archivos y enlazarlos mediante encabezados. De esta manera, se obtiene un programa principal y todas las subrutinas e interrupciones separadas, de acuerdo a sus funciones, en sus respectivos archivos.

#### 3.1. CARACTERISTICAS DEL LENGUAJE C

Es un lenguaje de alto nivel que interactúa profundamente con el hardware del computador. Se puede considerar de fácil entendimiento, ampliamente estandarizado y muy productivo, permite reutilizar el código y se tiene una cierta abstracción con el hardware. Finalmente es considerado como una excelente herramienta de desarrollo para microcontroladores.

Se debe tener presente que un compilador no es más eficiente que un buen programador en ensamblador.

Es más fácil escribir un buen código en C, el cual puede ser convertido en un código eficiente de ensamblador, que escribir un buen código en ensamblador *manualmente*.

Desafortunadamente, no todas las arquitecturas de los microcontroladores permiten trabajar con lenguaje de programación C.

### 3.1.1. REQUERIMIENTOS DEL COMPILADOR



**Figura 3-1: Requerimientos del Compilador**

La escritura del programa en C o en C++ se define en los archivos

- header (.h)
- compiler(.c)
- compiler++(cpp)

El compilador genera un archivo de objeto (.o), donde se enlaza todos los archivos anteriores y es semejante al código en ensamblador (.asm).

Las características de los programas de desarrollo para microcontroladores se diferencian de las características que tienen los programas diseñados para operar en computadoras por tener:

- Limitada memoria RAM
- Limitada memoria ROM
- Programación orientada a Hardware
- Tiempos críticos (Subrutinas de interrupción, tareas, ...)

- Diferentes tipos de punteros (far, near, rom, uni, ...)
- Funciones especiales (@, interrupt, tiny, ...)

El código de arranque del compilador generalmente se lo escribe en lenguaje de ensamblador. Este código es el lazo para la ejecución del código de alto nivel.

La secuencia que se debe seguir es la siguiente:

- Deshabilitar las interrupciones
- Copiar los datos de inicialización (variables) desde la ROM a la RAM
- Poner ceros a todas las variables no inicializadas.
- Habilitar las interrupciones
- Ubicarse en la rutina principal.

CodeWarrior de Metroweks es una de las compañías asociadas a Motorola encargadas del desarrollo de software para programación de dichos microcontroladores. No es gratuito, y los costos varían sustancialmente dependiendo de las necesidades.

### **3.2. EL MC68HC08GP32**

Previo a la implementación de los algoritmos de control PID por medio del microcontrolador MC68HC08, es bueno hacer una comparación de las capacidades que tiene el control orientado mediante un DSP (Digital Signal Processing) con un microcontrolador.

Después de decidir el uso de un sistema digital, un diseñador debe evaluar los requerimientos de un sistema para determinar qué tipo de aparato es el más adecuado para su trabajo. La decisión a menudo recae en escoger entre un microcontrolador estándar o un DSP.

Todos los DSP están diseñados especialmente para la ejecución de cálculos de suma algebraica de productos a alta velocidad.

Los microcontroladores estándares están mejor situados para aplicaciones que requieren un control en tiempo real relativamente pequeño, y también para

ejecutar otras tareas, como iniciar un sistema de operación o una interfaz de usuario. Los DSP's, por otro lado, son utilizados generalmente cuando el algoritmo de control necesita de tiempo real "intenso", y cuando las tareas adicionales en el sistema son manejadas por un procesador principal. Por tanto existe una gran diferencia entre los microcontroladores de propósito general y los especializados, dedicados al manejo por DSP.

El MC68HC08 es un microcontrolador modular el cual es un puente entre los microcontroladores estándares y los DSP. El procesador del micro posee un gran set de instrucciones tan bueno como el que tiene un DSP, mientras los otros módulos del sistema generan una variedad de opciones de interfaz. El nivel alto de integración funcional en el MC68HC08 reduce la cantidad de hardware externo necesario para alcanzar una solución completa en un sistema. Posee ciertos registros dedicados de multiplicación y acumulación que lo hacen capaz de ejecutar operaciones de DSP, y características adicionales para control.

Esta diferencia es importante debido a que la mayoría de los DSP no tienen una capacidad de manipulación ni vectores múltiples de interrupción. Sin embargo el MC68HC08 no ejecuta las operaciones de acumulación y multiplicación tan rápido como lo hacen algunos DSP dedicados, que son ideales para aplicaciones tales como sistemas de control de movimiento.

Por las razones expuestas anteriormente, en el sentido de ser un micro intermedio entre los DSP's y los microcontroladores estándares, resulta atractiva su utilización en este trabajo.

Con el propósito de comprender cómo está distribuido el microcontrolador, la Figura 3-2 detalla la arquitectura interna y la Figura 3-3 el diagrama de bloques.

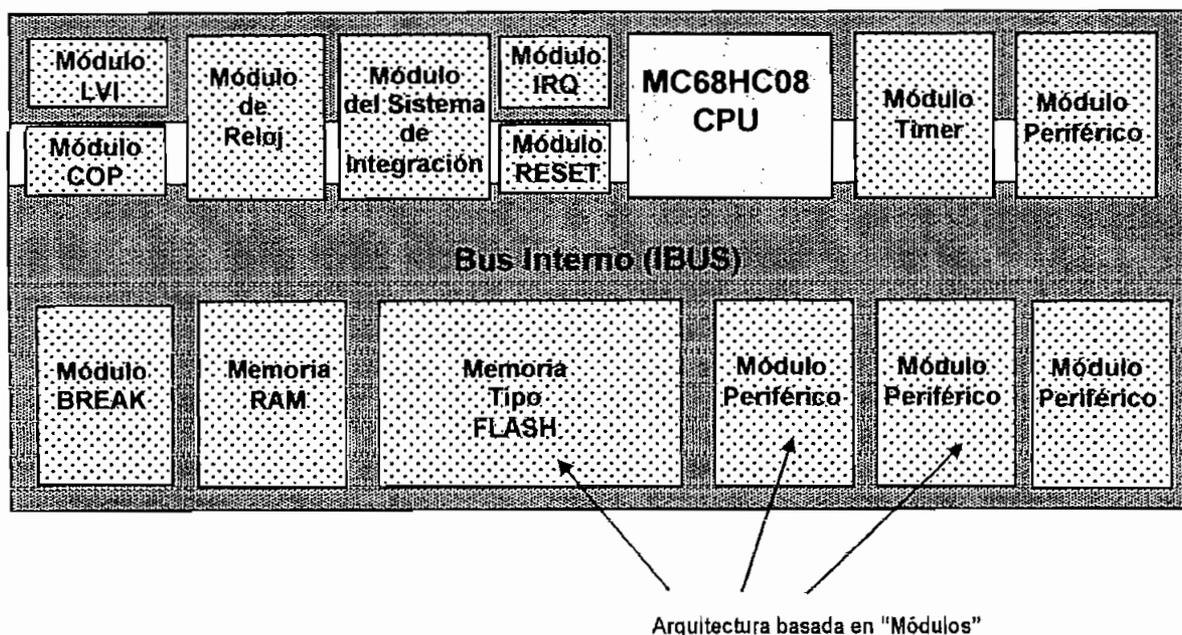


Figura 3-2: Arquitectura Interna del MC68HC08

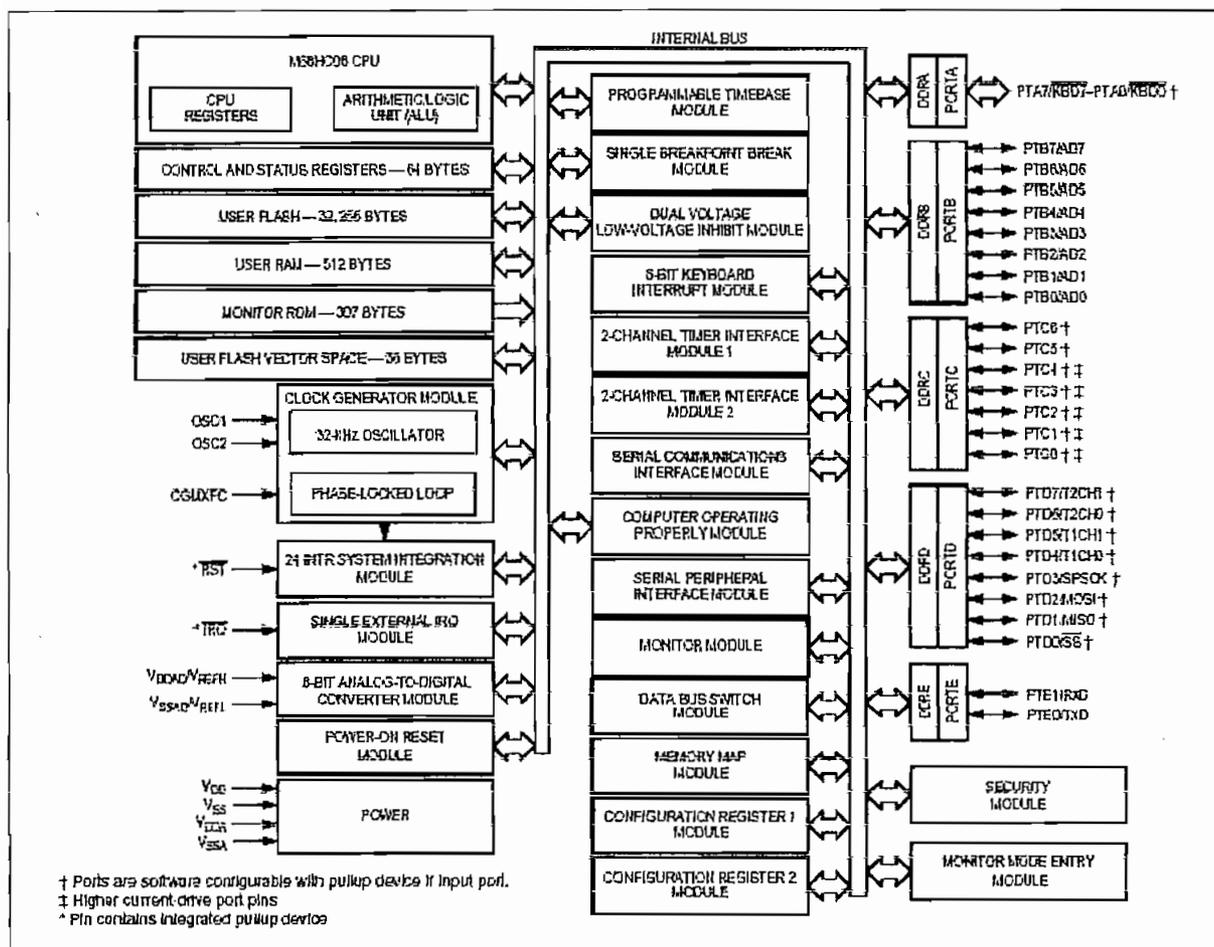


Figura 3-3: Diagrama de Bloques

El mapa de memoria del MC68HC08 que se muestra en la Figura 3-4, es de tipo "lineal", sin saltos de página y de acceso continuo. En la zona final del mapa de memoria se encuentran todos los vectores de interrupciones incluyendo el vector de reset y los registros de los distintos periféricos.

\$0000	I/O REGISTERS 64 BYTES
\$003F	
\$0040	
<b>\$0200</b>	<b>RAM 512 BYTES</b>
\$023F	REGISTROS DEL CONTROLADOR (RAM)
\$0240	
\$7FFF	
\$8000	
\$FDFF	MEMORIA FLASH 32,256 BYTES
\$FE00	SIM BREAK STATUS REGISTER ( SBSR )
\$FE01	SIM BREAK STATUS REGISTER ( SBSR )
\$FE02	
\$FE03	SIM Break Flag Control Register ( SBFCR )
\$FE04	INTERRUPT STATUS REGISTER 1 ( INT1 )
\$FE05	INTERRUPT STATUS REGISTER 2 ( INT2 )
\$FE06	INTERRUPT STATUS REGISTER 3 ( INT3 )
\$FE07	
\$FE08	FLASH Control Register ( FLCR )
\$FE09	Break Address Register High ( BRKH )
\$FE0A	Break Address Register LOW ( BRKL )
\$FE0B	Break Status and Control Register ( BRKSCR )
\$FE0B	LVI Status Register ( LVISR )
\$FE0D	
\$FE0F	
\$FE10	
\$FE1F	
\$FE20	
\$FF52	ROM Monitor 307 bytes
\$FF53	
\$FF7D	
\$FF7E	FLASH Block Protect Register ( FLBPR )
\$FF7F	
\$FFDB	
\$FFDC	FLASH Vectors 36 BYTES
\$FFFF	

Figura 3-4: Mapa de Memoria

### 3.3. REGISTROS DEL CONTROLADOR DISEÑADO

Los registros del controlador diseñado fueron creados bajo el mismo concepto de los registros predefinidos en el microcontrolador para el manejo de los periféricos. Están ubicados a partir de la dirección \$200h en la memoria RAM.

#### 3.3.1. REGISTROS DE SETPOINTS

Son registros de 16 bits que almacenan los valores de referencia para el control de la salida 1 y 2 respectivamente.

SETPOINTXL maneja los 8 bits menos significativos del registro SETPOINTX. SETPOINTXH los 8 bits más significativos del registro SETPOINTX.

En la Figura 3-5 se muestra el registro para el valor de referencia 1 (SETPOINT1).

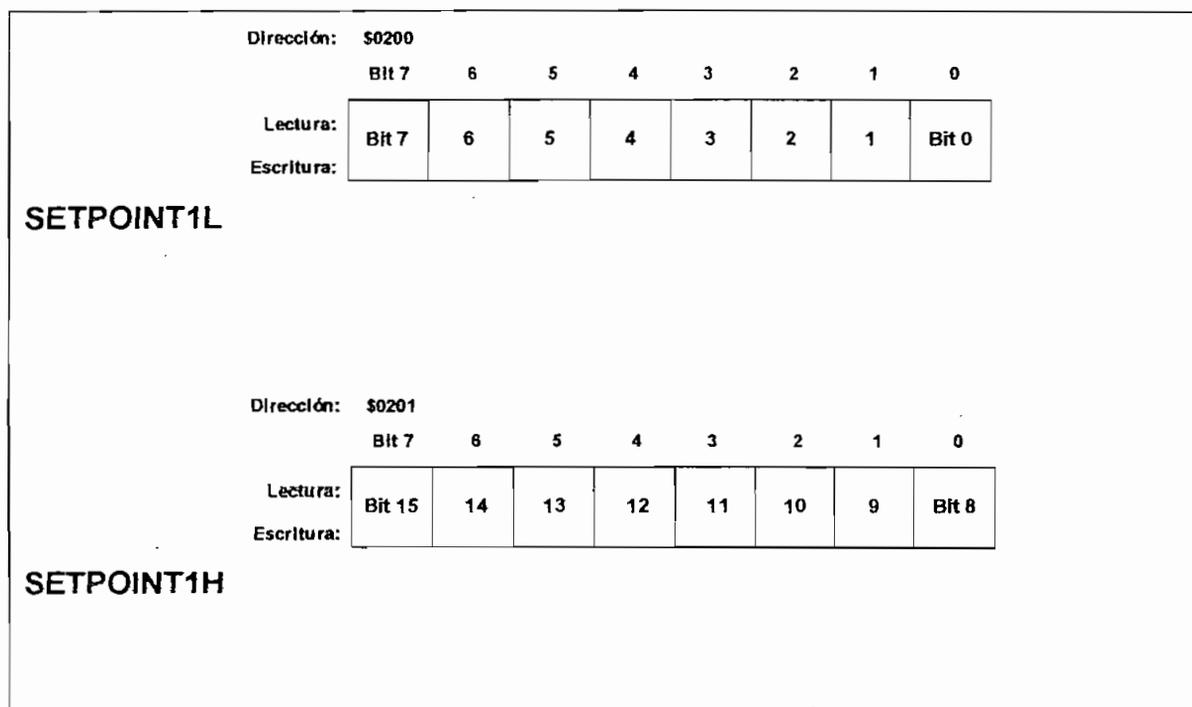


Figura 3-5: Registro del Valor de Referencia 1 (SETPOINT1)

En la Figura 3-6 se muestra el registro para el valor de referencia 2 (SETPOINT2).

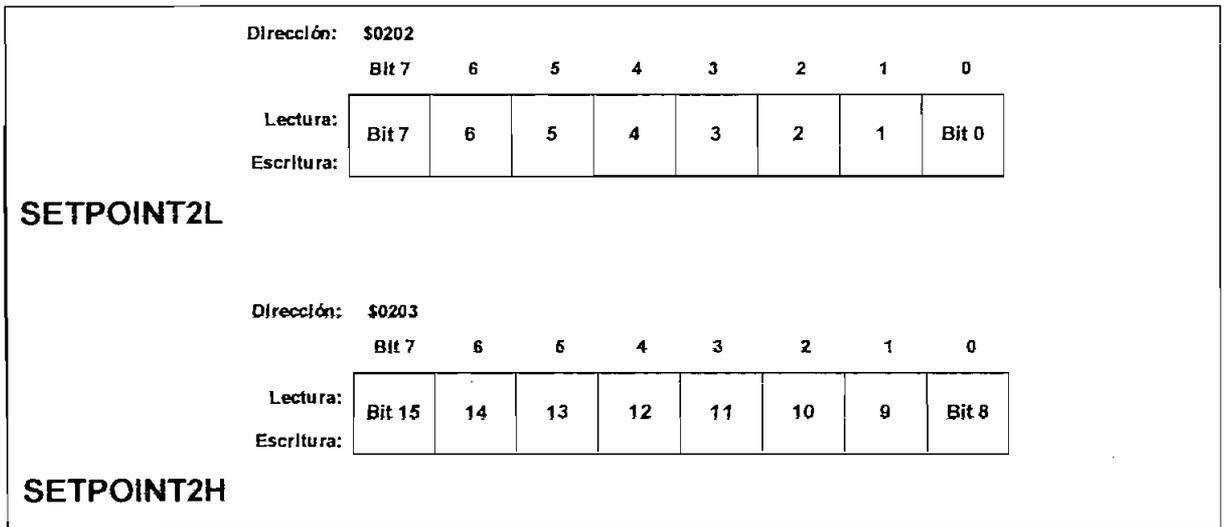


Figura 3-6: Registro del Valor de Referencia 2 (SETPOINT2)

### 3.3.2. REGISTRO DE IDENTIFICACION

Este registro almacena el código para seguridad del controlador. Es un registro de 16 bits. La clave de identificación tiene 4 valores, cada uno de los cuales ocupa 4 bits de los registros.

IDDRH es el registro más significativo y ocupan los dos primeros números. IDDRL es el registro menos significativo y ocupan los dos valores restantes como se muestra en la Figura 3-7.

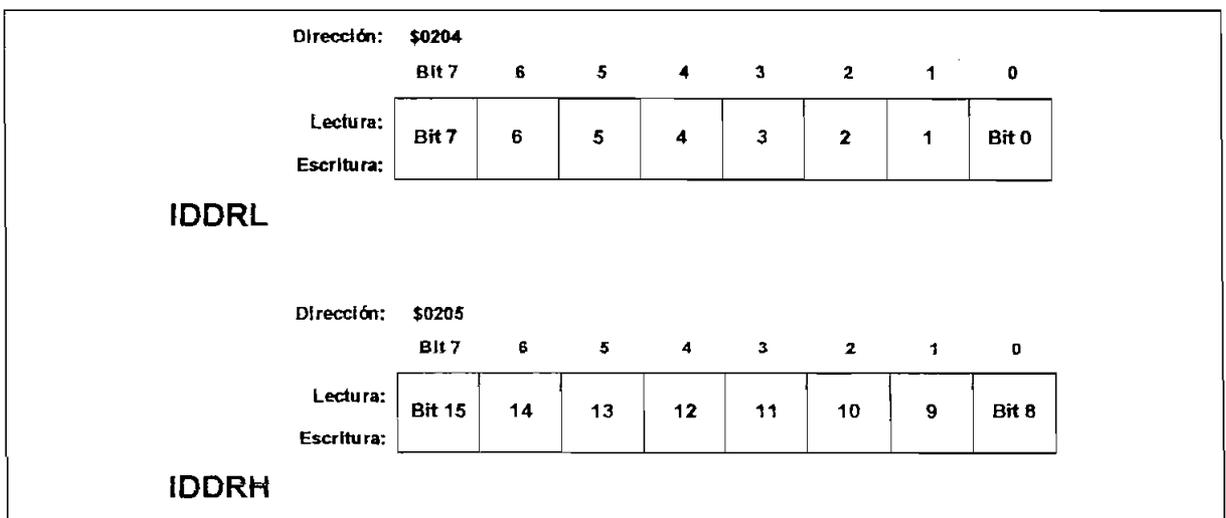


Figura 3-7: Registro de Identificación (IDDR)

### 3.3.3. REGISTRO DE ENTRADA

El registro de Entrada (INPUTR) permite seleccionar el tipo de entrada al controlador. La Figura 3-8 muestra los bits que componen este registro.

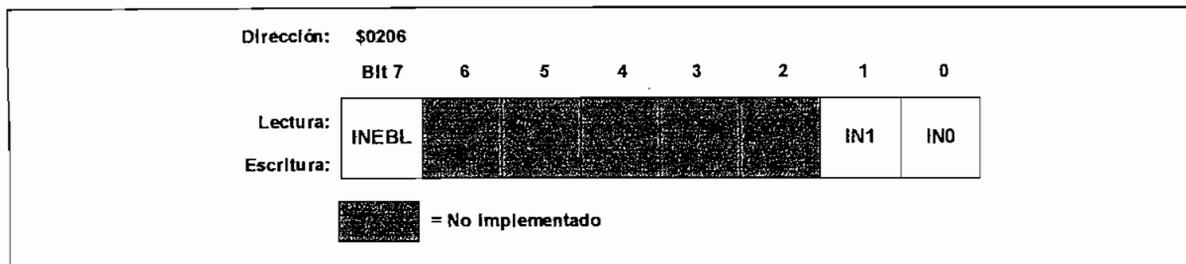


Figura 3-8: Registro de Entrada (INPUTR)

- INEBL –Este bit permite la habilitación de la entrada seleccionada.
- IN1 ^ IN0 – Bits de Selección de Entrada. En el controlador se tiene entradas de 0-5 V y de 0 – 20 mA.

### 3.3.4. REGISTRO DE CONFIGURACIÓN DE LECTURA

Con el registro de configuración de lectura (RDGCNG) se puede cambiar la posición del punto decimal en el valor numérico que aparece en el display. Este registro además permite cambiar la constante del filtro aplicado a la señal de entrada. En la Figura 3-9 se observa el registro (RDGCNG).

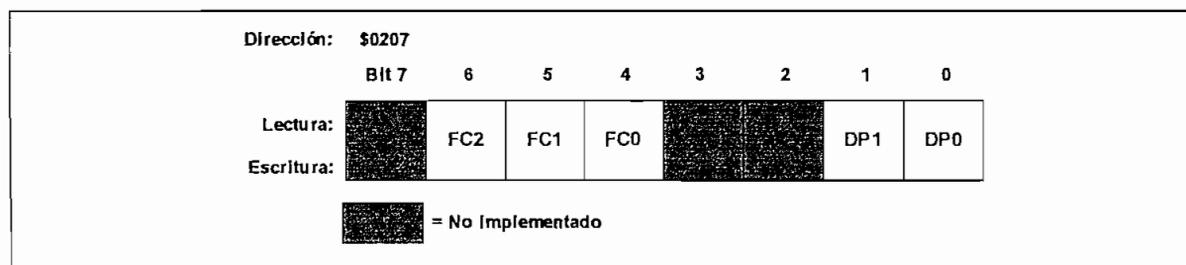


Figura 3-9: Registro de Configuración de Lectura (RDGCNG)

- FC2 – FC0 – Bits de Selección de la Constante de Filtro de Entrada. Mediante estos bits se selecciona la constante de filtro a la entrada del convertor analógico - digital. La constante de filtro seleccionada permite que el usuario especifique el número de lecturas almacenadas en el filtro

de entrada promediado. En la tabla 3-1 se determinan las constantes de filtro.

**Tabla 3-1: Constantes de Filtro de Entrada**

FC2 - FC0	FC
000	1
001	2
010	4
011	6
100	16
101	32
110	64
111	128

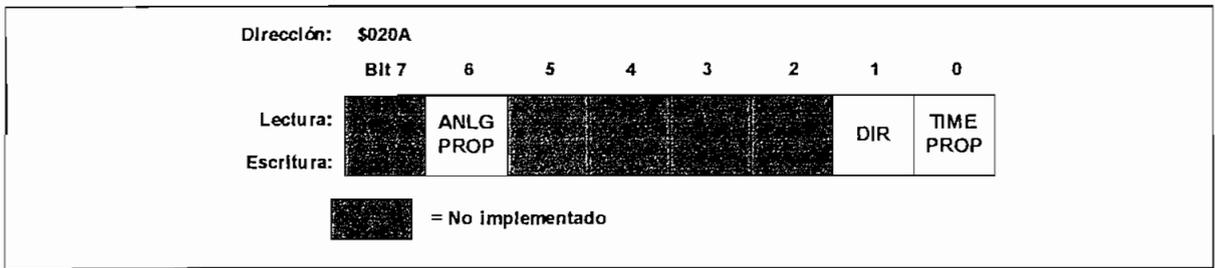
- DP1 – DP0 – Bits de Selección del Punto Decimal. Mediante estos bits de lectura/escritura se puede cambiar la visualización del punto decimal en el Display como se muestra en la Tabla 3-2.

**Tabla 3-2: Punto Decimal**

DP1 - DP0	Posición
00	FFFF.
01	FFF.F
10	FF.FF
11	F.FFF

### 3.3.5. REGISTROS DE SALIDAS

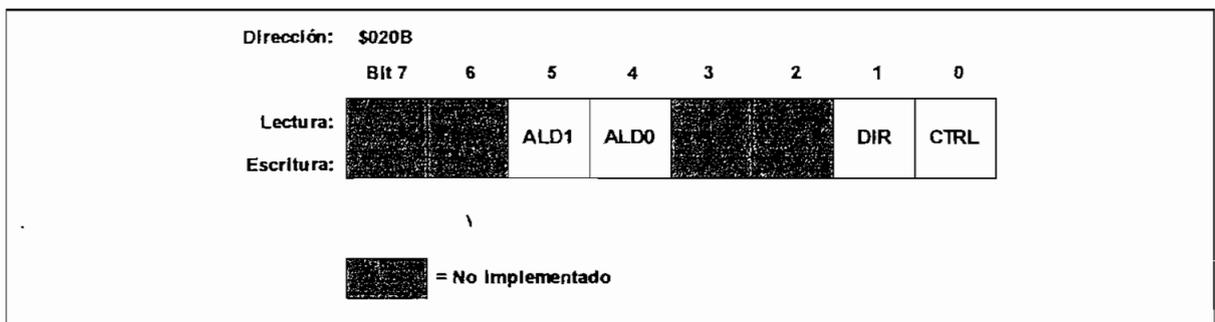
El registro de salida 1 determina si la salida es digital o analógica, la acción es directa o inversa y el tipo de control es on – off o PID como se muestra en la Figura 3-10.



**Figura 3-10: Registro de Salida 1 (OUT1CNG)**

- ANLGPROP - Bit de Habilitación de la Salida Analógica.
- DIR – Bit de selección del tipo de acción de control.
- TIMEPROP – Bit de selección del tipo de Control.

El registro de la salida 2 determina si la salida digital trabaja como alarma o control on-off, así como también cambia el modo de activación de la alarma (Figura 3-11)



**Figura 3-11: Registro de Salida 2 (OUT2CNG)**

- CTRL – Bit de selección del tipo de control.
- ALD1 – ALD0 – Bits de Selección del Tipo de Acción de la Alarma (Tabla 3-3).

**Tabla 3-3: Bits de Selección del Accionamiento de la Alarma**

ALD1 - ALD0	Se activa
00	Sobre
01	Bajo
10	Sobre-bajo
11	Banda

### 3.3.6. REGISTRO DE PARAMETROS DE COMUNICACIÓN

Este registro sirve para la configuración de los parámetros de comunicación serial con el computador (Figura 3-12)

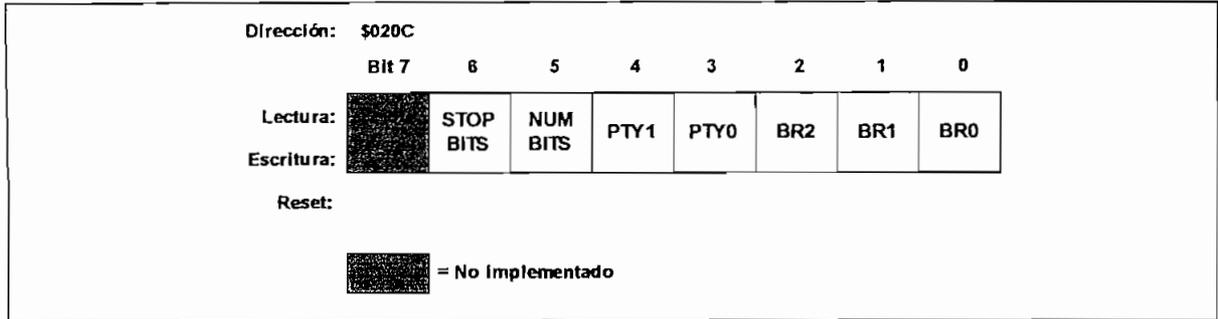


Figura 3-12: Registro de Parámetros De Comunicación (COMMPAR)

- STOPBITS – Bit de Parada. Este bit elige el número de bits de parada en la comunicación serial.
- NUMBITS – Número de Bits de Data. Este bit elige el número de bits de data en la comunicación serial entre 8 bits o 7 bits.
- PTY1 ^ PTY0 – Bits de Selección de Paridad. Mediante estos bits se selecciona el tipo de paridad en la comunicación serial como se muestra en la Tabla 3-4.

Tabla 3-4: Bits de Selección de Paridad

PTY1 - PTY0	Paridad
00	No paridad
[No Implementado]	[No Implementado]
10	Par
11	Impar

- BR2 – BR0 – Bits de Selección de la Velocidad de Comunicación (Baudios). Mediante estos bits se selecciona la velocidad de comunicación serial en Baudios como se define en la Tabla 3-5.

El valor predeterminado es de 9600.

Tabla 3-5: Bits de Selección de la Velocidad de Comunicación Serial

BR2 - BR0	BAUD
000	1
001	300
010	600
011	1200
100	2400
101	4800
110	9600
111	19200

### 3.3.7. REGISTRO DE CONFIGURACION DE SALIDAS

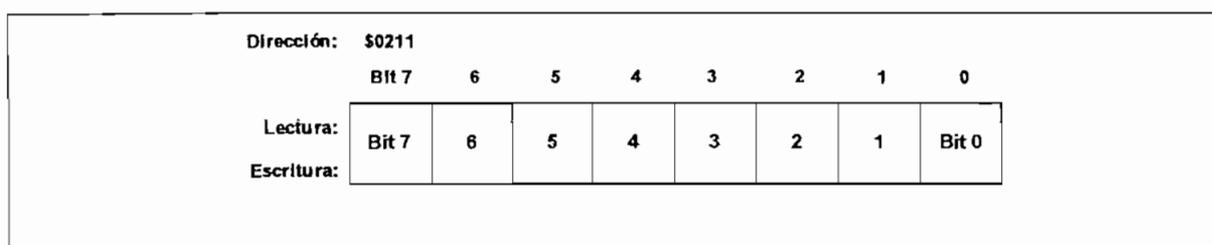


Figura 3-13: Registro de Valor Bajo (LOWDR)

El registro de Valor Bajo (LOWDR) especifica en porcentaje el valor mínimo (0000) para la salida de control. Es decir, si la salida es proporcional analógica, se especifica en porcentaje el valor mínimo de voltaje. Si por otro lado, la salida está dada por SSR se determina el valor mínimo de la razón on-off.

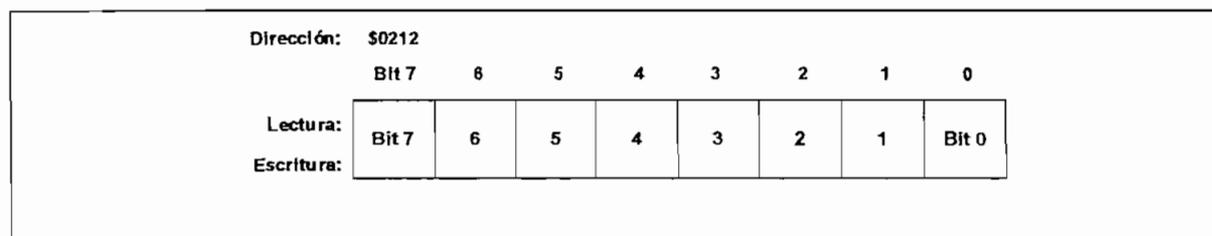
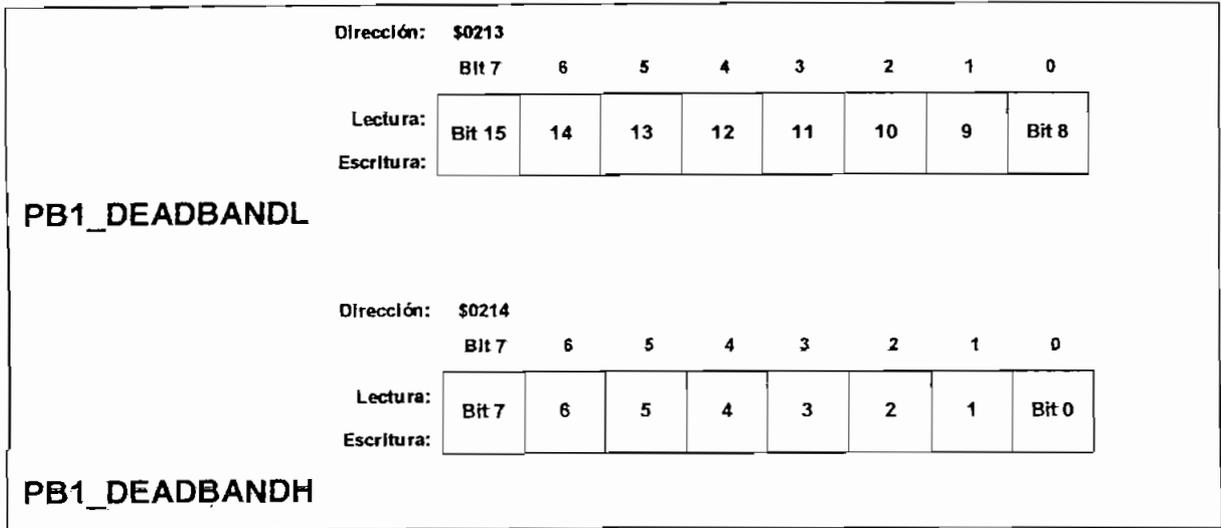


Figura 3-14: Registro de Valor Alto (HIDR)

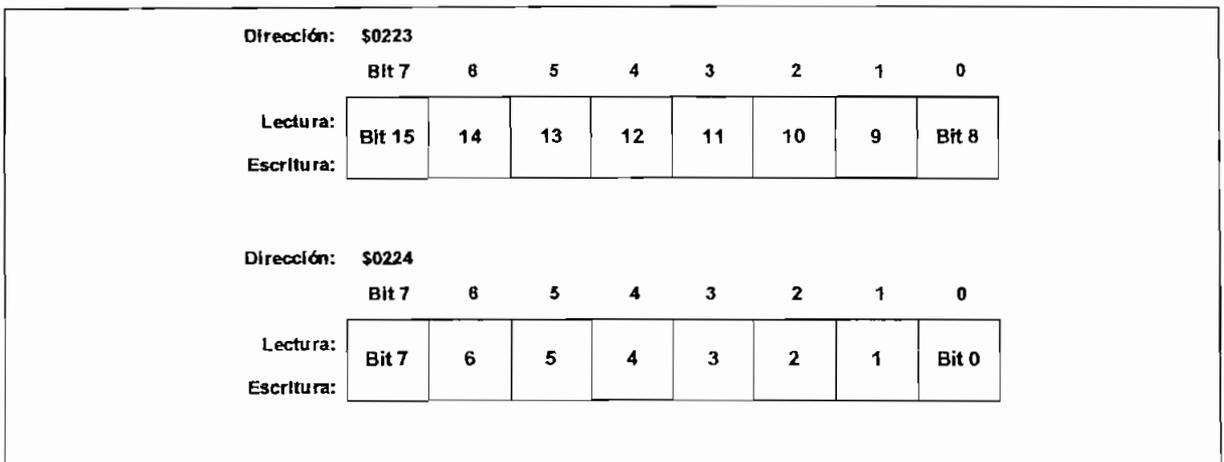
El registro de Valor Alto (HIDR) especifica en porcentaje el valor máximo (100%) para la salida de control. Si la salida es proporcional analógica, se especifica en

porcentaje el valor máximo de voltaje. Si la salida corresponde a SSR, determina la razón máxima de on- off.



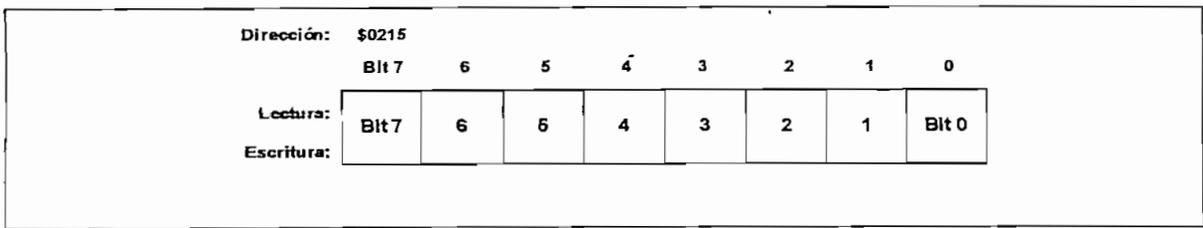
**Figura 3-15: Registro de Banda Proporcional 1 o Banda Muerta**

El registro de Banda Proporcional 1 (PB1\_DEADBAND) almacena el valor de la banda muerta o histéresis si el control de la salida 1 es de tipo on- off. Si por el contrario, el control es de tipo PID, este registro almacena el valor de la banda proporcional. Es un registro de 16 bits.



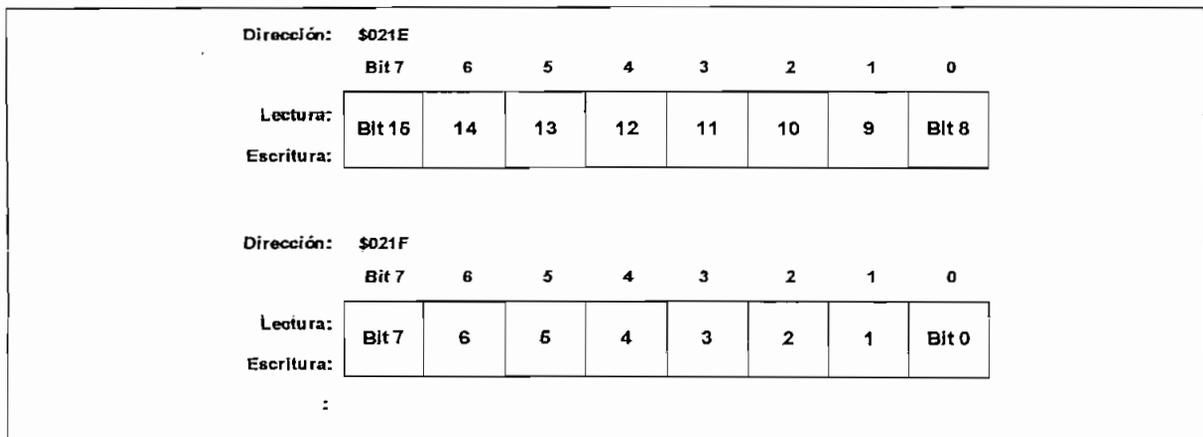
**Figura 3-16: Registro de Banda Proporcional 2 o Banda Muerta**

El registro de Banda Muerta 2 almacena el valor de la banda muerta o histéresis si el control de la salida 2 es de tipo on- off, También representa el valor de banda para cuando trabaja la salida 2 como alarma. Es un registro de 16 bits.



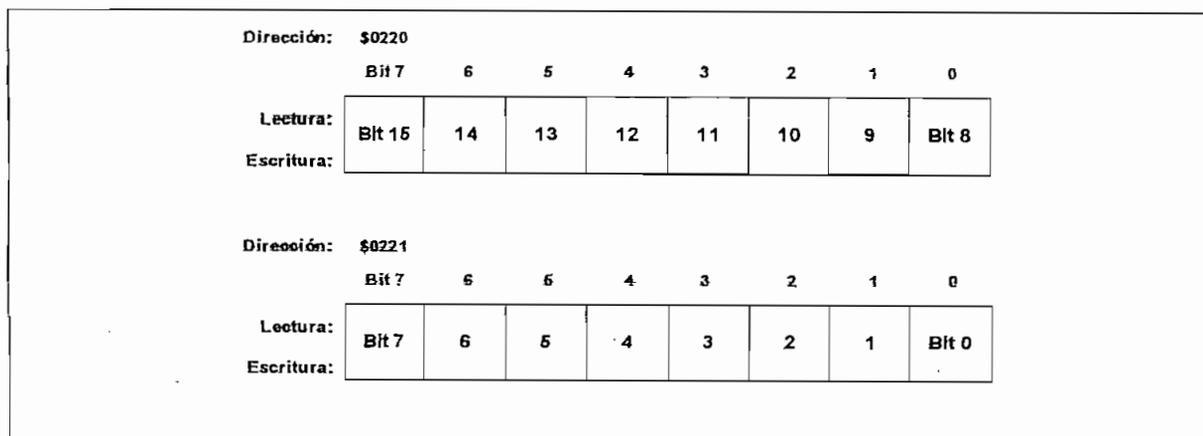
**Figura 3-17: Registro de Ciclo de Tiempo (CYCLEDR)**

El registro de ciclo (CYCLEDR) almacena el valor del ciclo de tiempo en segundos que determina el tiempo total de encendido/apagado en cada ciclo proporcional. Por ejemplo, un tiempo de ciclo de 15 segundos significa que cada 15 segundos la salida es encendida como parte o el total del ciclo.



**Figura 3-18: Registro para el Término Integral (RESETDR)**

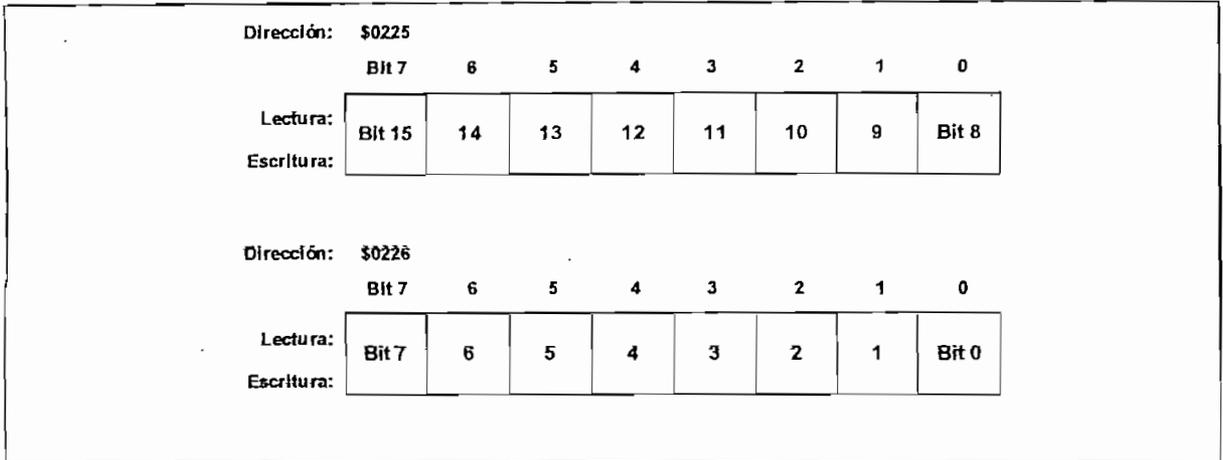
El registro (RESETDR) almacena el valor del término integral o reajuste para el control PID y las unidades están en segundos por repetición.



**Figura 3-19: Registro Término Derivativo (RATEDR)**

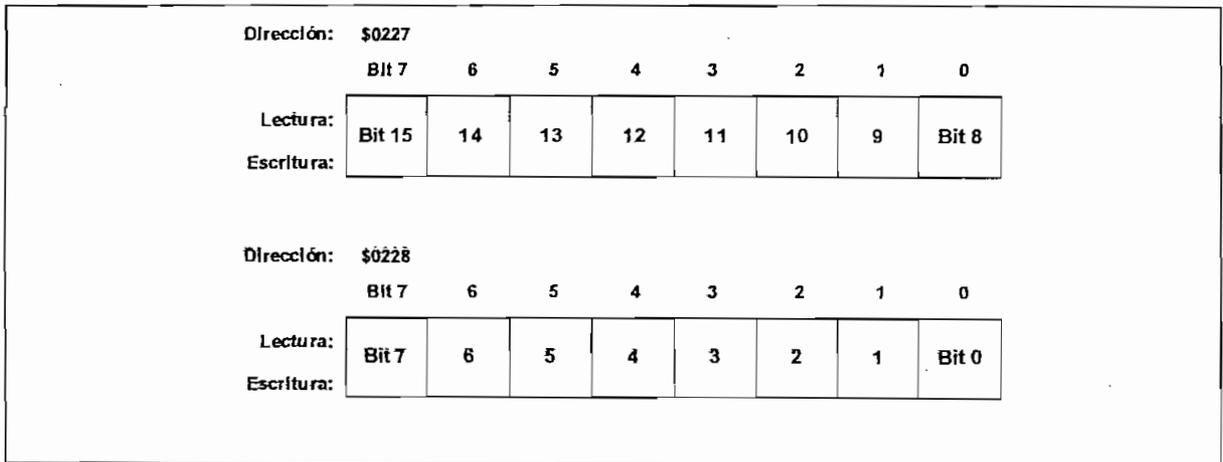
En el registro (RATEDR) se almacena el valor del término derivativo para el control PID y las unidades están en segundos.

En las Figuras 3-20 y 3-21 se muestran los registros que almacenan los valores alto y bajo para la alarma de la salida 2.



**Figura 3-20: Registro del Valor Bajo para Alarma (LOWDR2)**

El registro (LOWDR2) en la salida 2 configurada como alarma, representa el valor bajo al cual se activa la misma. Es un registro de 16 bits.



**Figura 3-21: Registro del Valor Alto para Alarma (HIDR2)**

Si la salida 2 está configurada como alarma, el registro (HIDR2) representa el valor alto al cual se activa la alarma. Es un registro de 16 bits.

El registro (SELFDR) mantiene la salida en un valor determinado para el control manual (Figura 3-22).

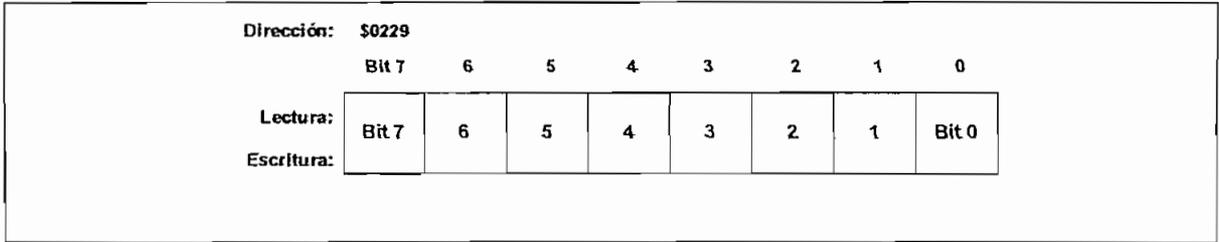


Figura 3-22: Registro del Valor del Modo Manual (SELFDR)

### 3.3.8. REGISTROS DE ESCALA Y OFFSET DE LECTURA

Con estos registros, la señal de entrada de voltaje o de corriente puede ser convertida o escalada a valores apropiados para la lectura en el display. Esta característica permite tener unidades virtuales de ingeniería perfectas para uso en temperatura, presión, flujo u otro tipo de procesos.

El usuario ingresa dos valores de entrada con sus correspondientes valores de lectura. La entrada 1 y 2 son el producto del voltaje o corriente de entrada con un valor de conversión. Este valor es un coeficiente de conversión entre los valores de entrada y el rango total del display (10.000 conteos)

El registro (VALORIN1) almacena el primer valor de entrada para la conversión (Figura 3-23). Tiene predeterminado el valor (0000).

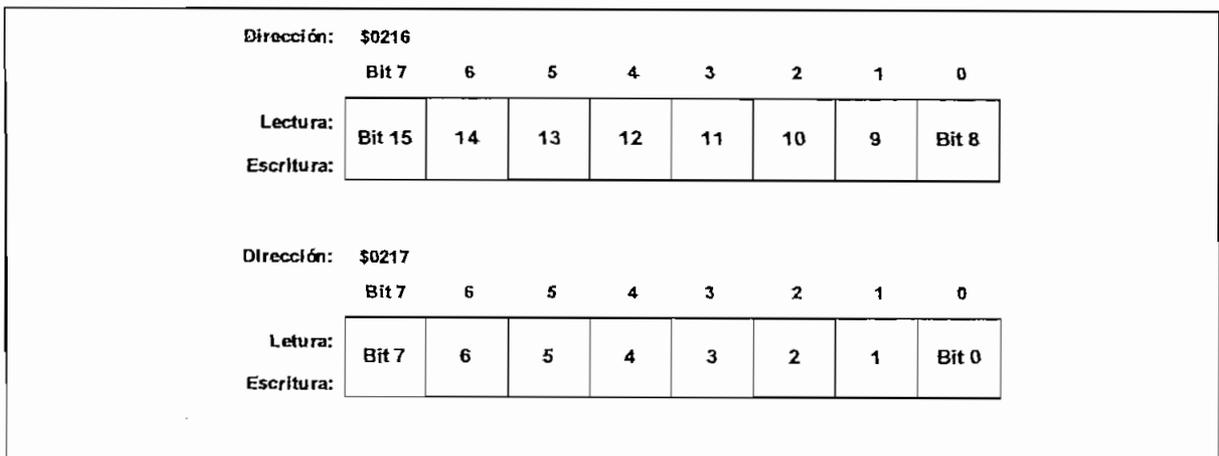


Figura 3-23: Registro De Valor De Entrada 1 (VALORIN1)

El registro (VALORRD1) almacena el valor de lectura para el valor de entrada 1 (Figura 3-24). El valor predeterminado es (0000).

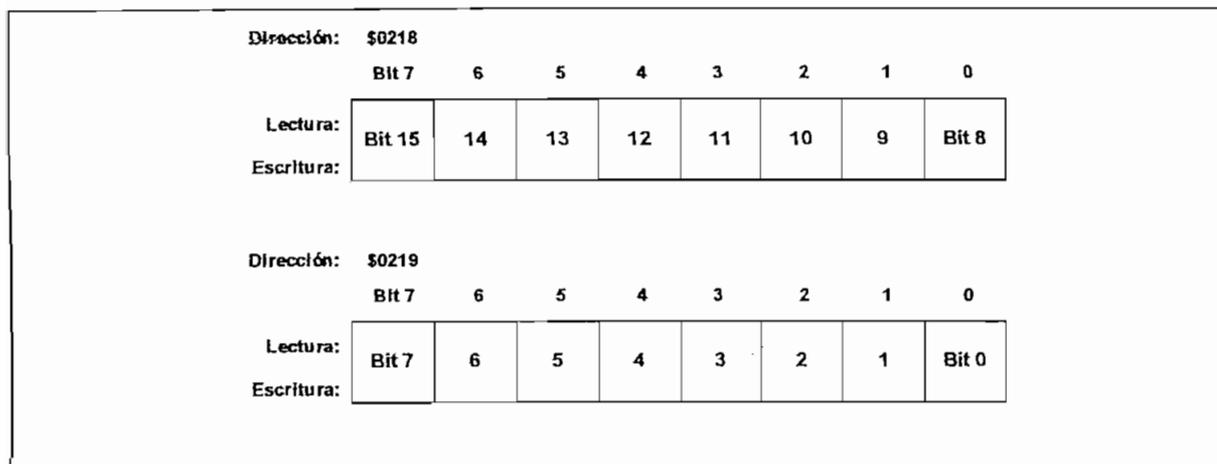


Figura 3-24: Registro De Valor De Lectura 1 (VALORRD1)

El registro (VALORIN2) almacena el segundo valor de entrada para la conversión (Figura 3-25). Tiene predeterminado el valor (10000).

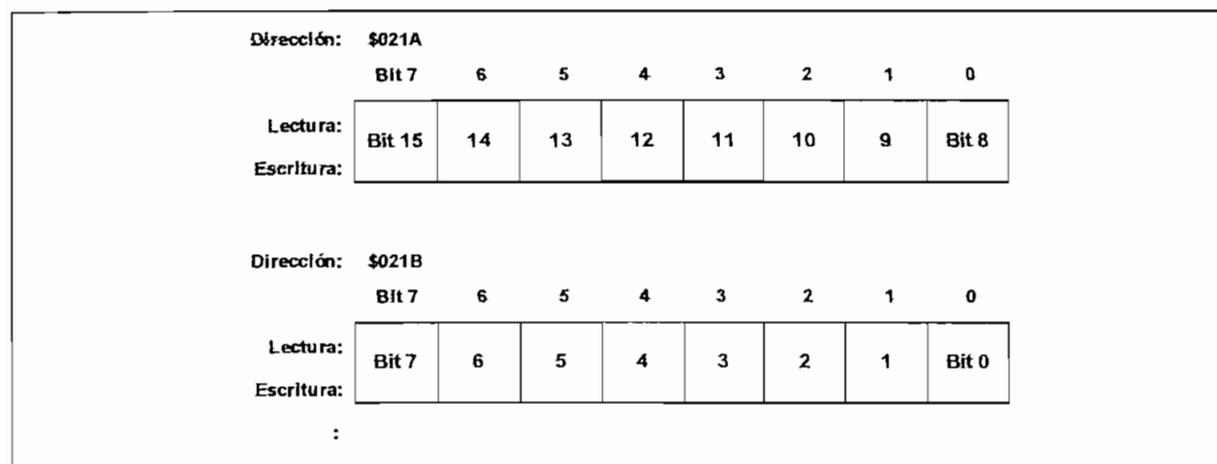


Figura 3-25: Registro De Valor De Entrada 2 (VALORIN2)

El registro (VALORRD2) almacena el valor de lectura para el valor de entrada 2 (Figura 3-26). El valor predeterminado es (10000).

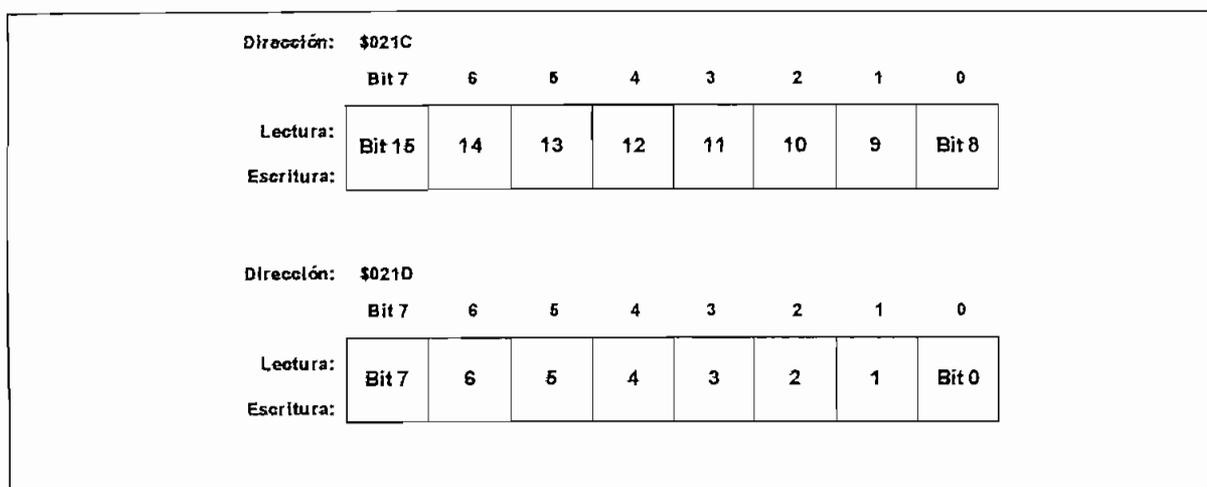


Figura 3-26: Registro De Valor De Lectura 2 (VALORRD2)

### 3.3.9. REGISTRO MISCELLANEOUS

El registro Miscellaneous habilita la seguridad del equipo y contiene el selector entre modo manual y modo automático para el control de la salida. Este registro se muestra en la Figura 3-27.

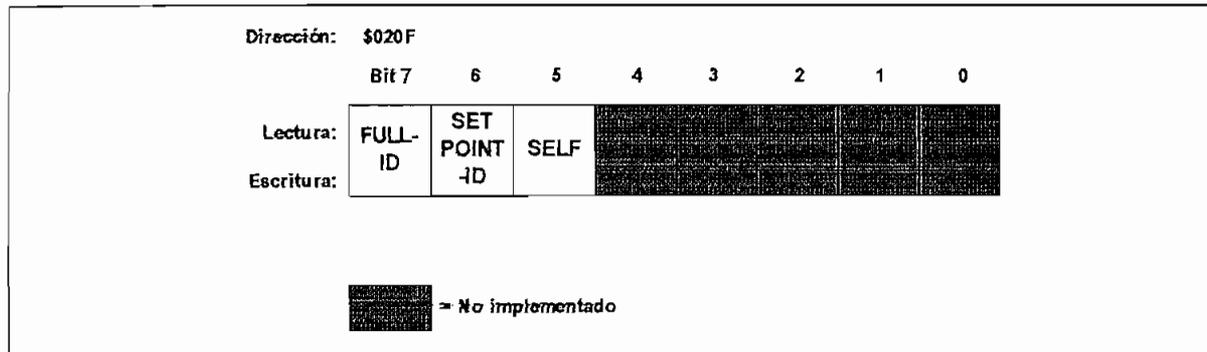
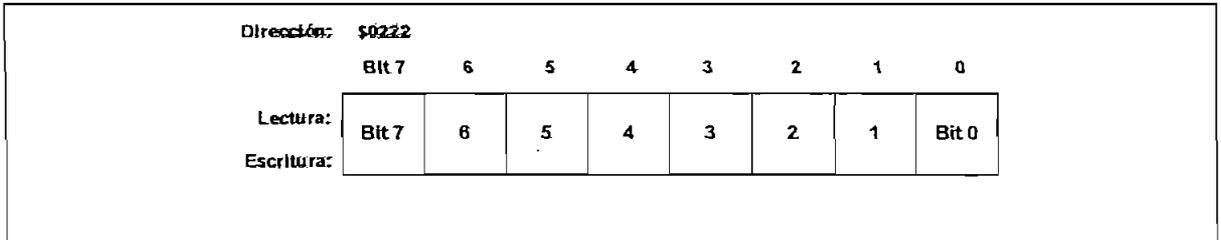


Figura 3-27: Registro Miscellaneous (MISCELLANEOUS)

- FULLID – Bit de Habilitación Completa de Seguridad. Este bit habilita la seguridad completa del controlador una vez configurado.
- SETPOINTID – Bit de Habilitación de Seguridad – Setpoint. Permite modificar los valores de referencia, pero no ingresa al menú de configuración sin la identificación correcta.
- SELF – Bit de Habilitación Modo Manual. Este bit habilita la opción de controlar la salida en forma manual

### 3.3.10. REGISTRO CONTRASTE DISPLAY (LCDDR)

En este registro se maneja el contraste del display LCD. Mediante una señal PWM se aumenta o disminuye el voltaje de entrada correspondiente al contraste del LCD.



**Figura 3-28: Registro De Contraste Del Display (LCDDR)**

El valor predeterminado es (0000) y viene a ser el valor máximo de voltaje para el contraste del display.

### 3.4. SOFTWARE DE OPERACIÓN

El software de operación del controlador sigue un flujo de programación para la utilización de los periféricos con las características necesarias para el diseño del controlador.

#### 3.4.1. PROGRAMA PRINCIPAL

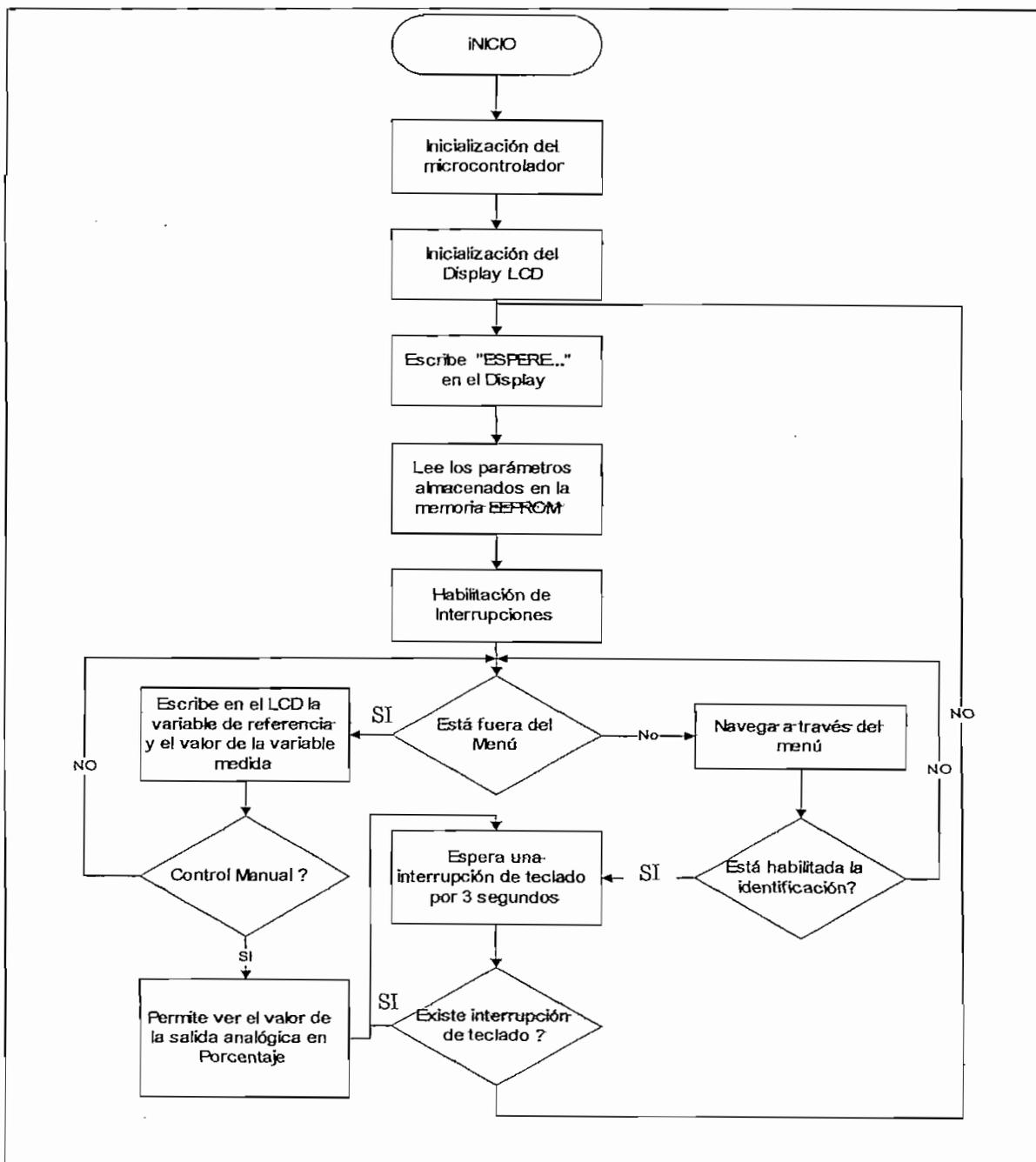


Figura 3-29: Diagrama de Flujo del Programa Principal

El programa principal (Figura 3-29) se encarga de inicializar los módulos y periféricos del microcontrolador. Inicializa al display para su visualización. Toma todos los parámetros almacenados en la memoria EEPROM y los ubica en localidades de memoria del microcontrolador. Finalizado todo el proceso de inicio habilita las interrupciones y permanece en un lazo que escribe en el display el estado de la variable medida y la de referencia. Tiene un tiempo de espera de 3 segundos para operaciones especiales como el ingreso de la clave, que reinicializa el controlador en caso de no completar la operación.

### 3.4.2. SUBROUTINA DEL ALGORITMO DE CONTROL

En la Figura 3-30 se encuentra el esquema en bloques del algoritmo de control PID utilizado para el controlador diseñado.

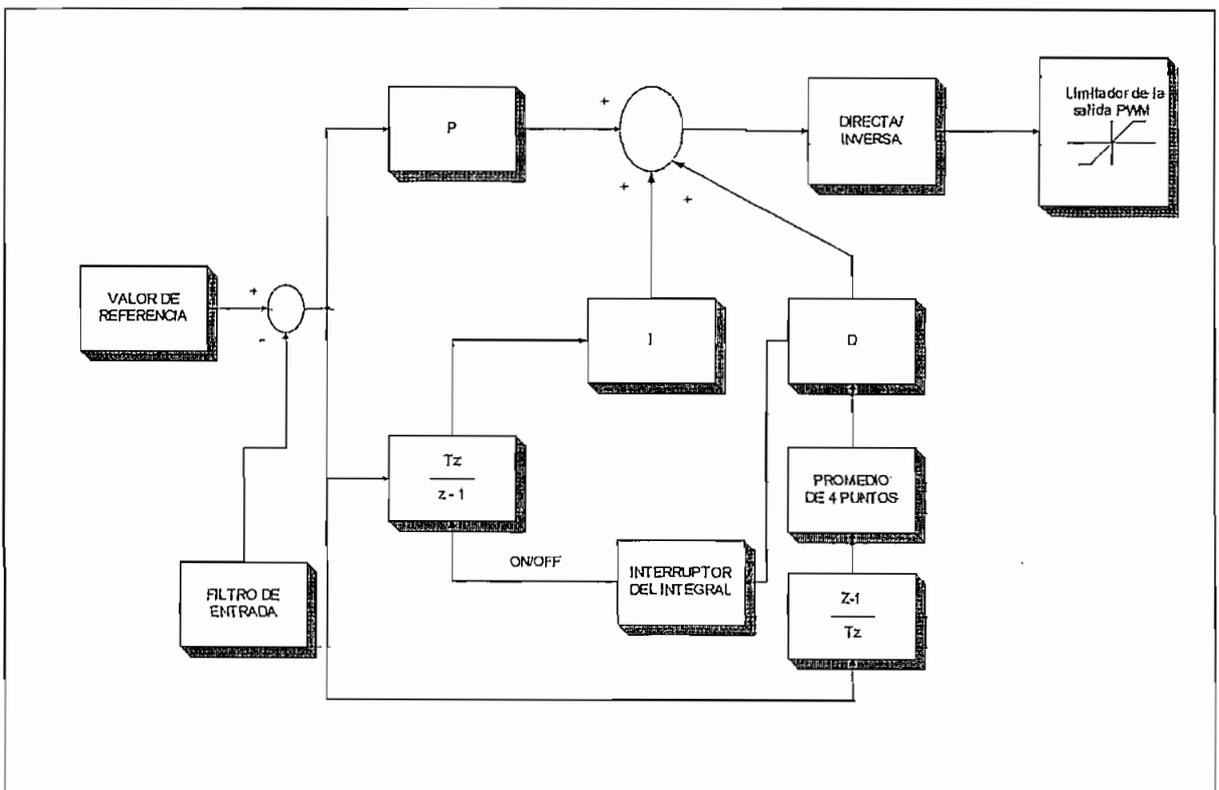


Figura 3-30: Algoritmo de Control PID

Se ingresa por teclado el valor de referencia. La entrada es filtrada antes de ingresar al lazo cerrado de control. Este filtro se configura en el menú. El valor promediado de la variable depende de la constante que se utilice en el filtro.

Se resta el valor de referencia del valor medido y se obtiene el valor de error (ERRX). El error ingresa a los tres términos de control: proporcional, integral y derivativo como se describe en la ecuación siguiente:

$$NEWDTY = KP * (ERRX) + KI * PERDT * [SETPOINT - (ADRCX + ADRCXM1) / 2] + [KD / (6 * PERDT)] * [(ERRX - ERRM3X) + 3 * (ERRM1X - ERRM2X)]$$

*Ecuación (3-1)*

NEWDTY es la salida del controlador, PERDT es el período de muestreo, SETPOINT es el valor de referencia, ADRCX es la entrada actual del convertor, ADRCXM1 es la entrada anterior del convertor, ERRMxX son los valores de error de los cuatro períodos anteriores.

Se determina si el valor de entrada actual con respecto al valor de entrada anterior es más grande que el valor de umbral especificado (5 unidades). En caso de ser mayor el valor del término integral perteneciente al período de muestreo anterior se anula. De esta manera se genera el interruptor de integral, el cual ayuda a mejorar el comportamiento del controlador. El término integral que se ingresa en el menú es recíproco al valor Ki del algoritmo, por tal razón, se divide el período de muestreo para este valor. En caso equivalente a cero, se anula la acción integral.

Para la acción derivativa se utiliza en el algoritmo el promedio de 4 puntos. Se almacenan 3 valores del error en períodos de muestreo anteriores para su cálculo.

Una vez sumado el valor integral y el derivativo al valor de error, se divide a todo por el valor de la Banda Proporcional. Luego se determina si la acción que se

requiere es directa o inversa en la salida del controlador. Verificado si el valor de salida actual está dentro de los límites requeridos, actualiza los datos y envía la señal al módulo PWM.

En el caso que se trate de control por tiempo proporcional, se realiza el encendido y apagado de la salida digital correspondiente a los relés dependiendo del ciclo seleccionado en configuración.

Para realizar este procedimiento de manera periódica existe un módulo del microcontrolador que genera una interrupción de forma cíclica conocido como el módulo de base de tiempo (TBM).

En este mismo módulo se generan los algoritmos correspondientes al control de tipo on – off y la alarma. El control on – off toma el valor de error  $ERRX$  y compara con el valor absoluto de la mitad de la banda muerta para encender o apagar la salida digital. La alarma tiene tres acciones posibles de encendido: sobre, bajo y en la banda.

$$ERRX \geq | banda\_muerta / 2| \quad \text{Ecuación (3-2)}$$

### 3.4.3. SUBROUTINA DE BASE DE TIEMPO PARA EL CONTROL

La función que se ejecuta periódicamente tiene la siguiente sintaxis:

*interrupt void TBM\_ISR(void)*

Dependiendo del tipo de control deseado se ejecuta el algoritmo correspondiente de control. Utiliza el módulo de base de tiempo (TBM) del microcontrolador a una frecuencia de 300 Hz. Sin embargo, los algoritmos de control, especialmente el PID, realizan sus cálculos en un período mayor al seleccionado. Por tal razón, se añade un contador que compense este retardo de aproximadamente 8 veces el período seleccionado y se obtiene, por tanto, un período de muestreo de 37.5 Hz como máximo.

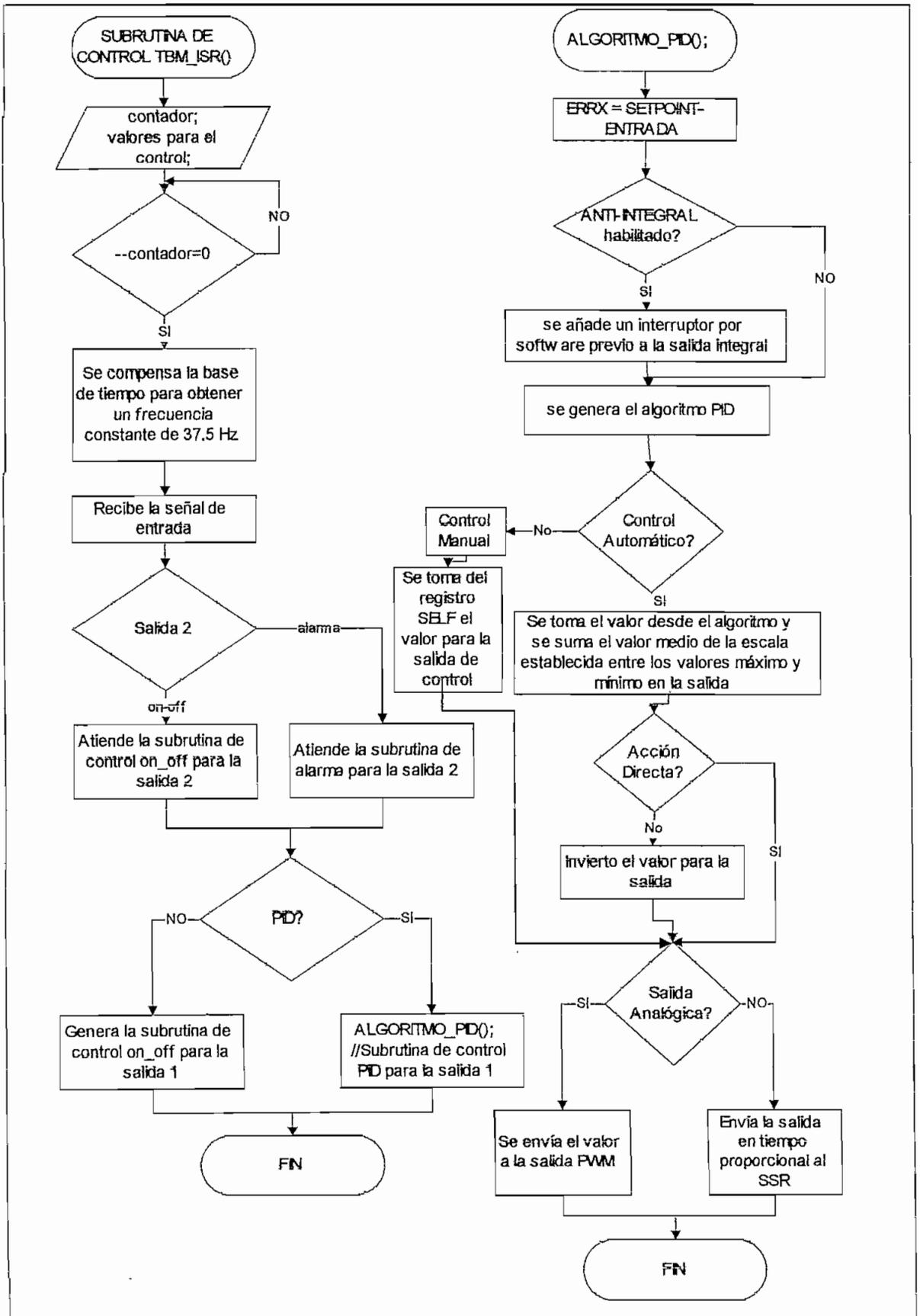


Figura 3-31: Diagrama de Flujo de la Subrutina de Control

### 3.4.4. SUBRUTINAS DE USO COMUN

#### 3.4.4.1. Subrutina de Retardo

La subrutina Delay(), suspende la ejecución del programa por un intervalo de tiempo en milisegundos. Recibe el valor en milisegundos requerido para el retardo. Su sintaxis es:

*void Delay (unsigned int ms)*

La frecuencia del cristal es de 9,8304 MHz y como la frecuencia del bus interno es la cuarta parte, entonces para generar 1 ms se tiene la siguiente ecuación:

$$\frac{9830400}{1000 * 4} = 2458 = (3 * \text{contador\_A} + 2 + 1 + 1 + 3) * \text{contador\_X} + 2 \quad \text{Ecuación (3-3)}$$

donde: contador\_A = 100, contador\_X = 8 y las constantes son retardos generados por cada instrucción.

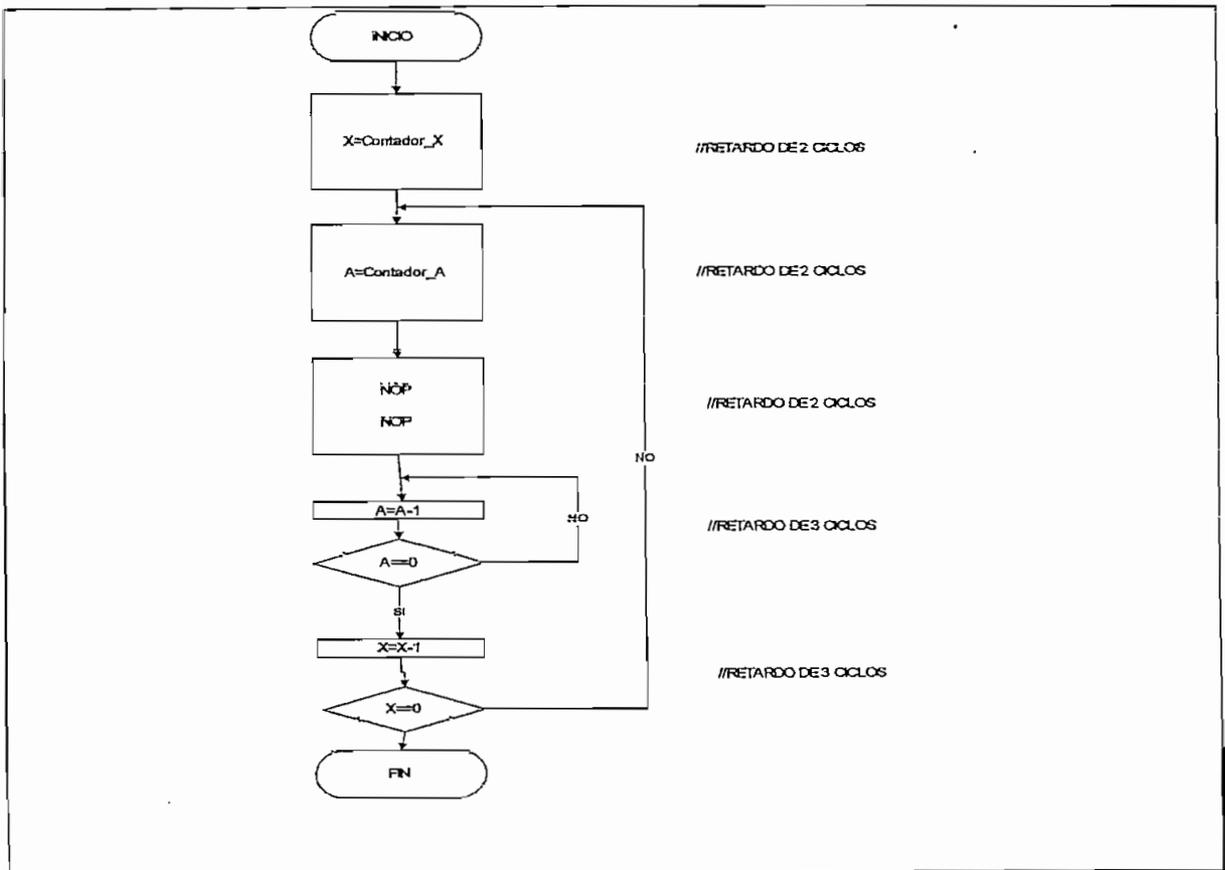


Figura 3-32: Diagrama de Flujo de la Función Delay

### 3.4.4.2. Subrutina de Conversión a Caracteres ASCII

La función `sprintf()` escribe la salida formateada a una cadena. Evalúa los argumentos, los convierte de acuerdo al formato y escribe el resultado en una cadena, terminada con un carácter cero. Su sintaxis es:

```
int sprintf (char *s, const char *format, ...)
```

Esta subrutina sirve para escribir en el display un mensaje de texto añadido valores numéricos con signo o para enviar información por medio del puerto serial a un computador.

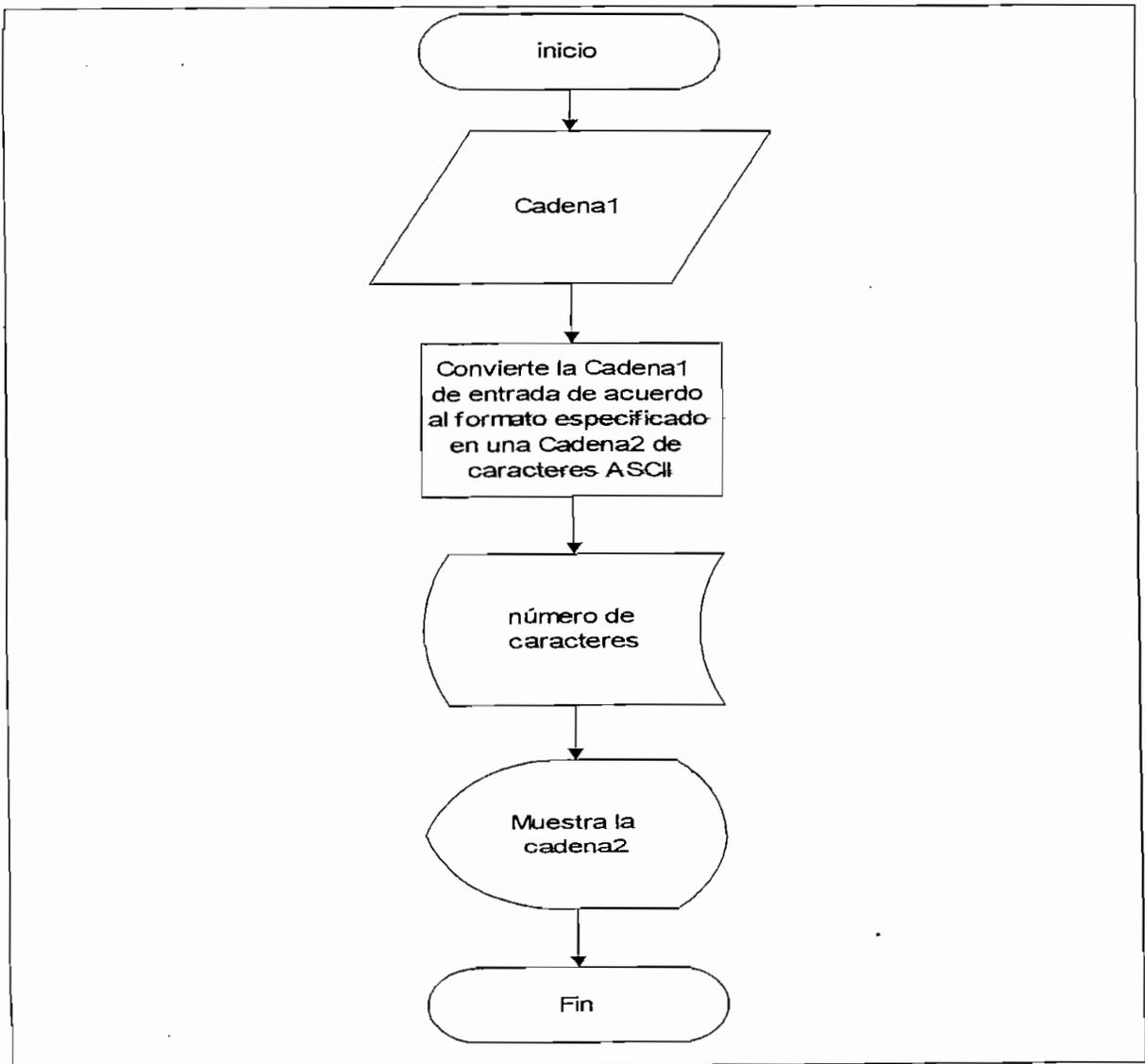


Figura 3-33: Diagrama de Flujo de la Función `sprintf`

### 3.4.5. SUBROUTINAS DE COMUNICACIÓN SERIAL I2C

La comunicación serial I2C es un protocolo de transmisión de datos síncrono que se utiliza en este caso para la comunicación entre el microcontrolador y la memoria EEPROM o el PIC mediante dos pines de un bus bidireccional. Un dispositivo que envía los datos sobre el bus se define como transmisor y el que recibe la información es el receptor. El bus tiene que ser controlado por un elemento principal o maestro, en este caso el microcontrolador, el cual genera una señal de reloj (SCL), controla el acceso al bus y genera las condiciones de inicio y de parada, mientras que la memoria EEPROM o el PIC serían los elementos esclavos. Tanto el elemento controlador como los elementos esclavos pueden operar como transmisor o receptor, pero el microcontrolador determina en qué modo debe ser actuado.

El protocolo de bus debe cumplir lo siguiente:

- La transferencia de datos puede iniciarse solamente cuando el bus está libre.
- Durante la transferencia de datos, la línea de datos (SDA) debe permanecer estable mientras la línea de reloj (SCL) está en ALTO. Los cambios en la línea de datos mientras está la línea de reloj en ALTO puede interpretarse como condiciones de inicio o de parada.

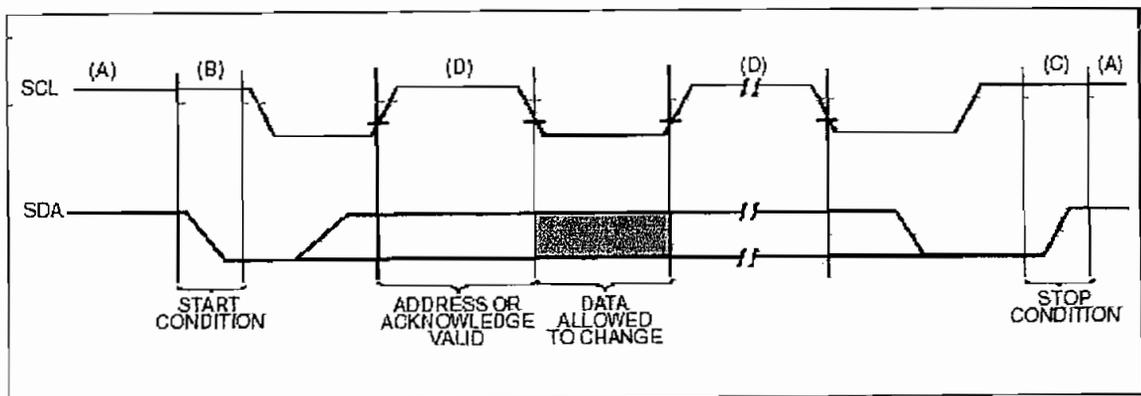


Figura 3-34: Secuencia de la Transferencia de Datos y las Condiciones del Bus de Transferencia

El bus está libre (A) cuando tanto la línea de datos como la del reloj permanecen en ALTO (1 lógico).

Un flanco negativo en la línea del SDA mientras el reloj está en ALTO determina la condición de Inicio (B). Todos los comandos deben ser precedidos por una condición de inicio.

Un flanco positivo en la línea del SDA mientras el reloj está en ALTO determina una condición de parada (C). Todas las operaciones deben ser finalizadas con esta condición.

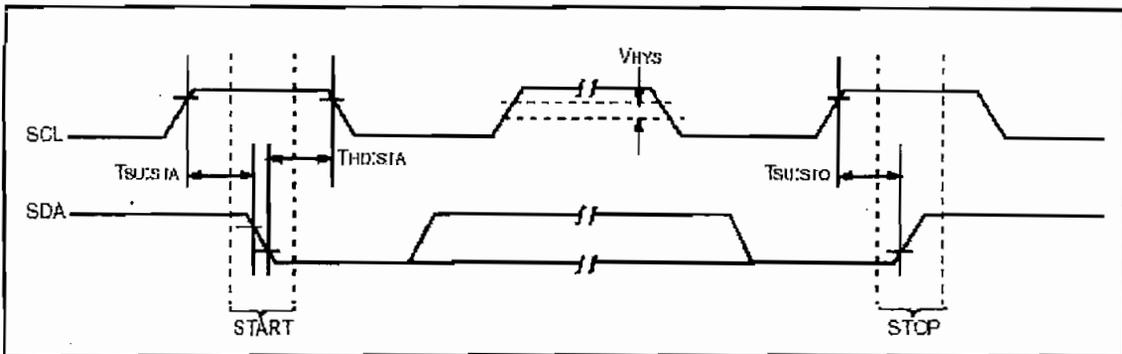


Figura 3-35: Tiempos en el Bus para las Condiciones de Inicio y de Parada

Después de una condición de inicio, el estado de la línea de datos representa datos válidos. Esta línea permanece en ALTO durante un período de reloj

Los datos en la línea deben ser ingresados durante el período BAJO de la señal de reloj. Existe un pulso de reloj por cada bit de datos.

Cada transferencia de datos se inicia con una condición START y termina con una condición STOP. El número de bytes de datos transferidos entre las condiciones de inicio y de parada está determinado por el microcontrolador y teóricamente es ilimitado, pero solamente los últimos 16 bytes serán almacenados cuando se realice una operación de escritura.

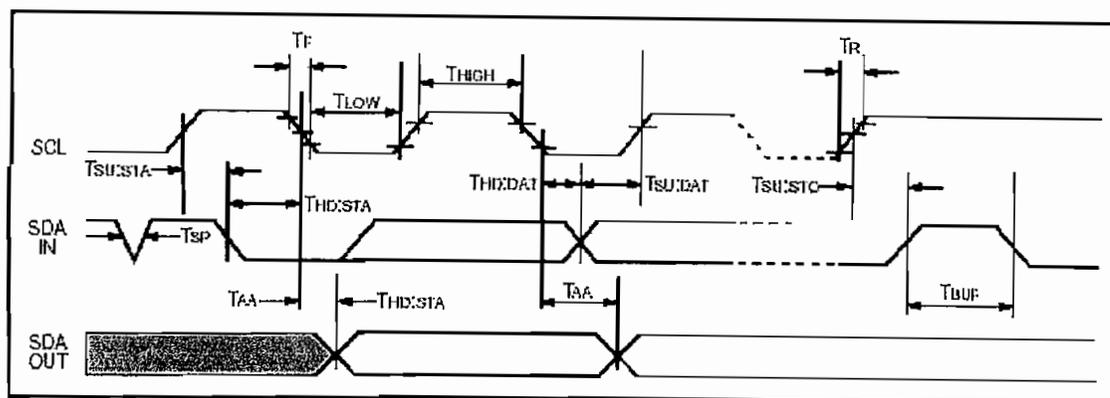


Figura 3-36: Tiempos del Bus para la Transferencia de Datos

Tabla 3- 6: Tiempos requeridos para la Comunicación Serial I2C

Parámetro	Símbolo	Min	Max	Unidades
Frecuencia del Reloj	$F_{CLK}$	-	400	KHz
Tiempo en Alto Reloj	$T_{HIGH}$	600	-	Ns
Tiempo en Bajo Reloj	$T_{LOW}$	1300	-	Ns
Tiempo de elevación SDA y SCL	$T_R$	-	300	Ns
Tiempo de caída SDA y SCL	$T_F$	-	300	Ns
Tiempo de establecimiento para la condición START	$T_{HD:STA}$	600	-	Ns
Tiempo de la condición $\overline{START}$	$T_{SU:STA}$	600	-	Ns
Tiempo de establecimiento para el ingreso de datos	$T_{HD:DAT}$	0	-	Ns
Tiempo de ingreso de los datos	$T_{SU:DAT}$	100	-	Ns
Tiempo de la condición STOP	$T_{SU:STO}$	600	-	Ns
Tiempo del bus libre: Este tiempo se requiere antes de generar una nueva transmisión.	$T_{BUF}$	1300	-	Ns

### 3.4.5.1. Subrutinas de Inicio y Parada

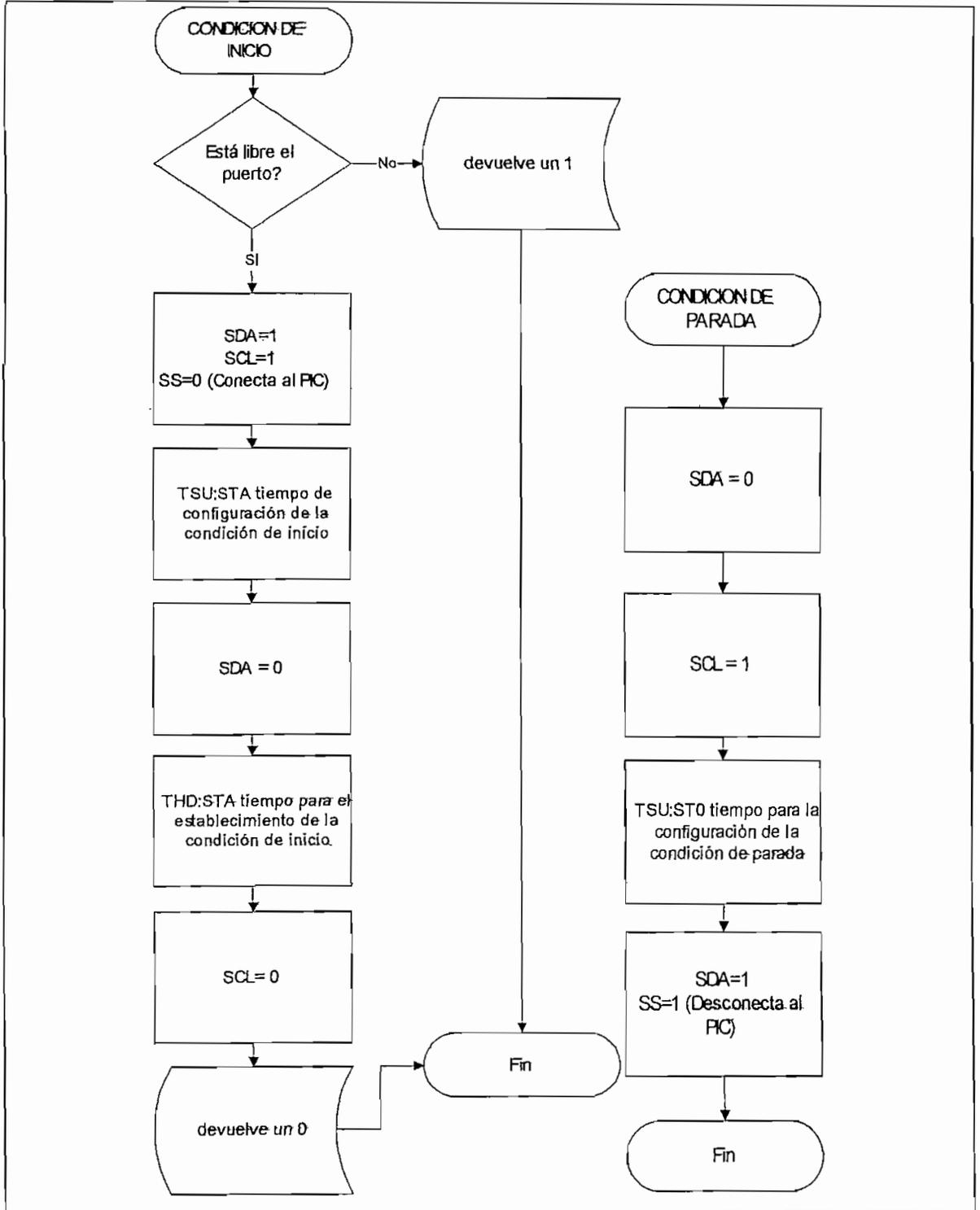
La función destinada para la generación de la condición de inicio (START) descrita anteriormente tiene la sintaxis:

*int Start (void)*

Stop () es una función destinada para la generación de la condición de parada (STOP) descrita anteriormente cuya sintaxis es:

*void Stop (void)*

En la Figura 3-37 se muestra el diagrama de flujo de las subrutinas de inicio y de parada.



**Figura 3 – 37: Diagrama de Flujo de las Condiciones de Inicio y Parada para la Comunicación Serial I2C.**

Cada elemento receptor, cuando se lo direcciona, está obligado a generar una señal de acuse después de la recepción de cada byte. El microcontrolador debe generar un pulso de reloj adicional el cual es asociado con esta señal.

### 3.4.5.2. Direccionamiento

Primero se envía un byte de control seguido de la condición de inicio desde el microcontrolador. El byte de control consiste en un código de 4 bits. Para la memoria EEPROM se configura el número binario 1010 tanto en las operaciones de escritura como de lectura y para el PIC el número 1011 para ambas operaciones.

Los siguientes tres bits del byte de control determinan el bloque de selección (B1, B0). Estos se utilizan, en el caso de la memoria, para seleccionar uno de los dos bloques de memoria (256 bytes cada uno). Estos bits serán los más significativos en el direccionamiento.

El último bit del byte de control define la operación de escritura o lectura. Si es un 1 lógico, la operación es de lectura, de lo contrario, la operación es de escritura, esto se resume en la Tabla 3-7.

**Tabla 3-7: Direccionamiento de un elemento para comunicación I2C**

OPERACIÓN	ELEMENTO	CODIGO DE CONTROL	SELECCIÓN DE BLOQUE	E/L <sup>-</sup>
LECTURA	EEPROM	1010	Dirección del Bloque	1
	PIC	1011		1
ESCRITURA	EEPROM	1010	Dirección del Bloque	0
	PIC	1011		0

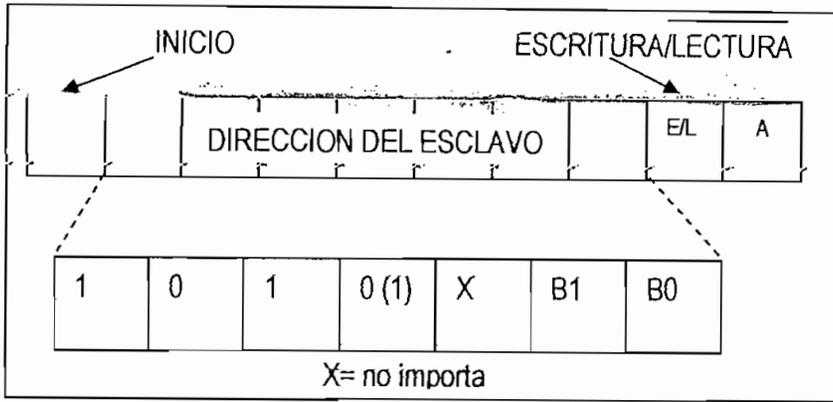


Figura.3-38: Ubicación del Byte de Control.

### 3.4.5.3. Subrutinas Para La Operación de Escritura

Luego de la condición de inicio desde el microcontrolador, el código del elemento (4 bits), la selección de bloques (3 bits) y el bit de lectura/escritura, el mismo que es un 0 lógico, se ubican en el transmisor del microcontrolador. Esto indica al receptor que un byte con la dirección seguirá después de generar la señal de acuse durante el noveno ciclo de reloj. Así, el siguiente byte transmitido por el microcontrolador es la dirección de la localidad y será escrita en el puntero de la memoria. Después de recibir la otra señal de acuse desde la memoria, el microcontrolador transmitirá el byte de datos para dicha localidad de memoria. La EEPROM manda nuevamente la señal de acuse y el microcontrolador genera la condición de parada. Todo lo anterior comprende el ciclo de escritura. Durante este tiempo, la memoria no generará ninguna señal de acuse.

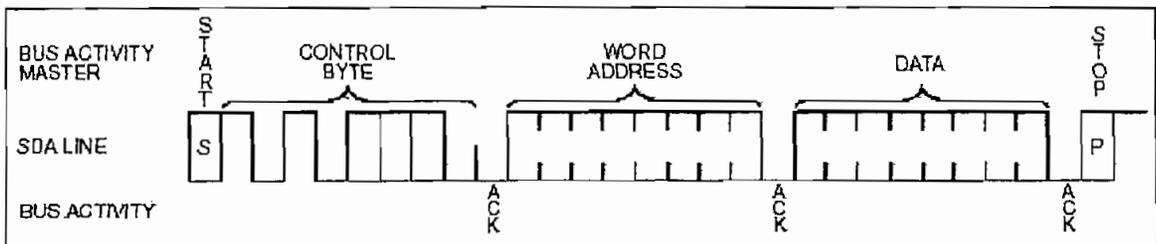


Figura 3-39: Operación de Escritura

La función destinada para transmitir al PIC o a la memoria EEPROM un byte tiene la siguiente sintaxis:

*int Transmit ( uchar data1 )*

La función para la comunicación con el PIC que envía un caracter para generar la señal PWM que controla el display tiene la siguiente sintaxis:

*void WritePort ( uchar data )*

La función para comunicación con la EEPROM que envía un byte en una dirección específica de la memoria tiene la siguiente sintaxis:

*void WriteByte( uchar deviceNumber, uint address, uchar data )*

Por ejemplo: *WriteByte(0,50,30)* ingresa el valor 30 en la localidad 50 del bloque 0 de memoria.

*WriteMsg()* es una función que incluye a la anterior para el envío de un conjunto de caracteres o un mensaje en particular. Su sintaxis es la siguiente:

*void WriteMsg(uint address, uchar\* mensaje)*

En la Figura 3-40 se observa el diagrama de flujo de la operación de escritura al PIC y en la Figura 3-41 se muestra el diagrama de flujo de la operación de escritura a la memoria EEPROM.

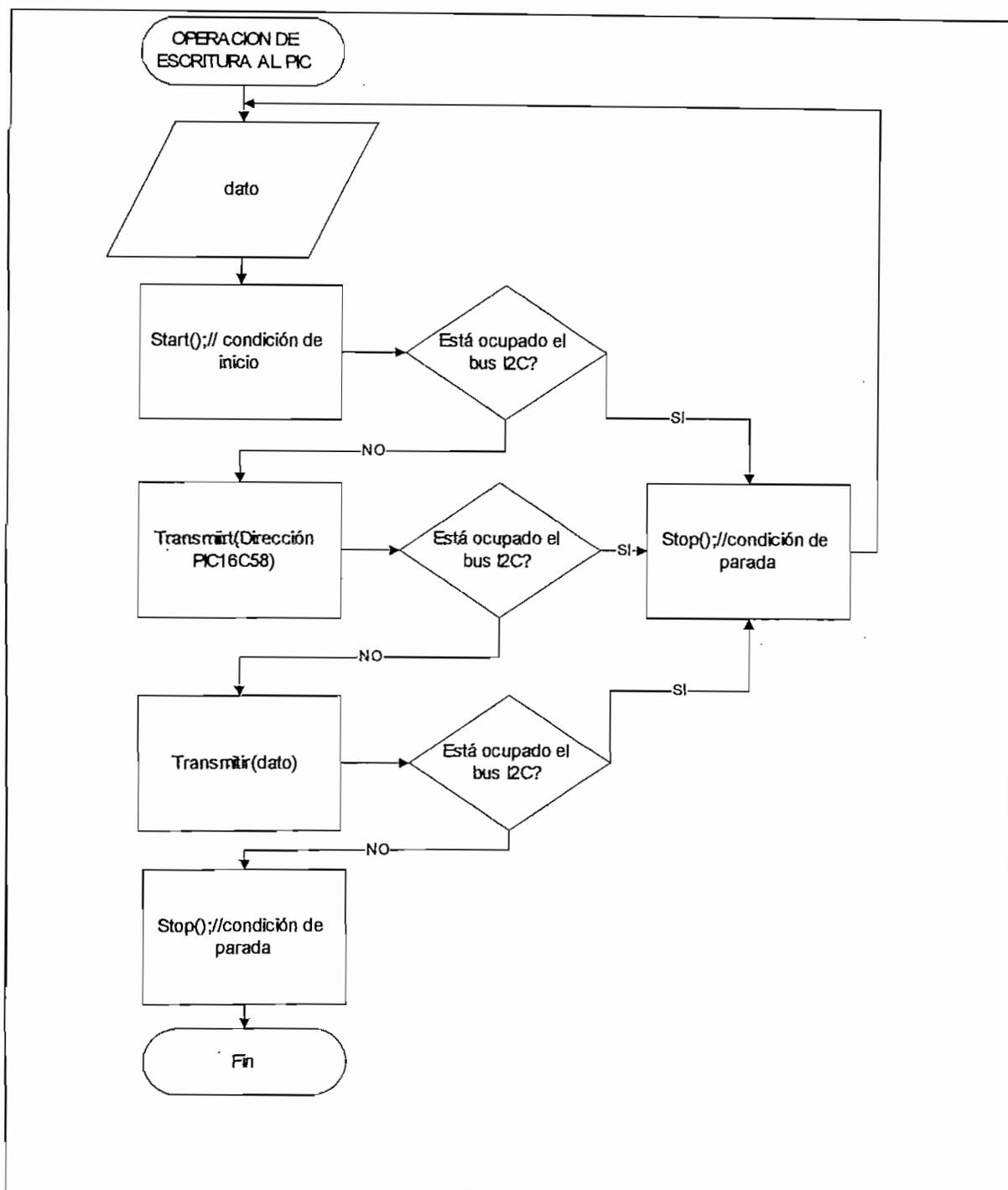


Figura 3-40: Diagrama de Flujo de la Operación de Escritura al PIC.

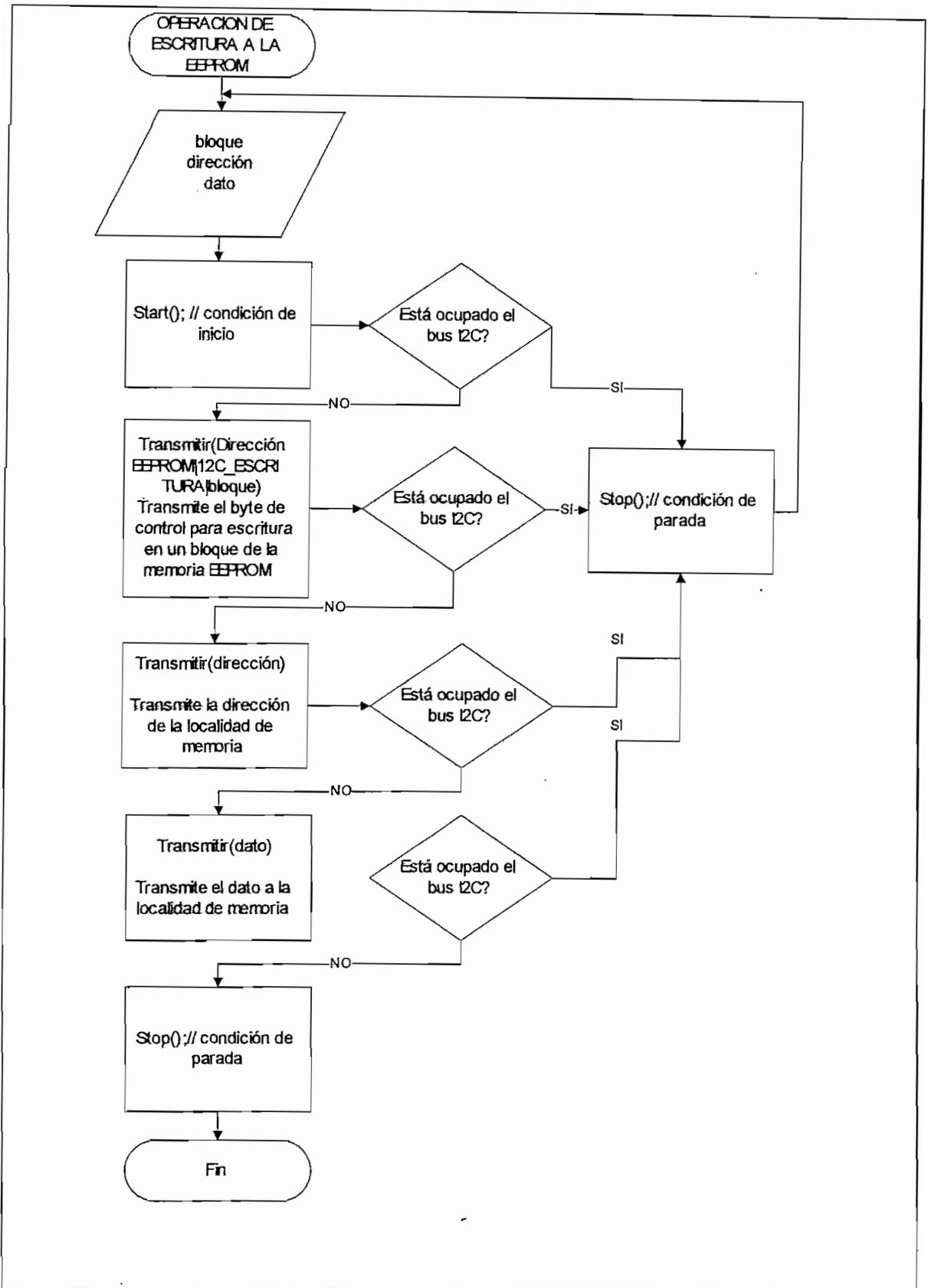


Figura 3-41: Diagrama de Flujo de la Operación de Escritura a la memoria EEPROM.

### 3.4.5.4. Subrutina para la Recepción de la Señal de Acuse

Si el elemento no genera una señal de acuse durante el ciclo de escritura, este indicativo puede usarse para determinar si el ciclo de escritura se culminó. Una vez que la condición de parada para un comando de escritura generada por el microcontrolador se ejecuta, el elemento inicializa internamente un ciclo de escritura. El microcontrolador puede enviar una señal de inicio seguida por el byte de control en modo de escritura. Si el elemento está todavía ocupado con el ciclo de escritura, no entregará la señal ACK. Si el ciclo se terminó entregará esta señal ACK y el microcontrolador puede proceder al siguiente comando de escritura o lectura.

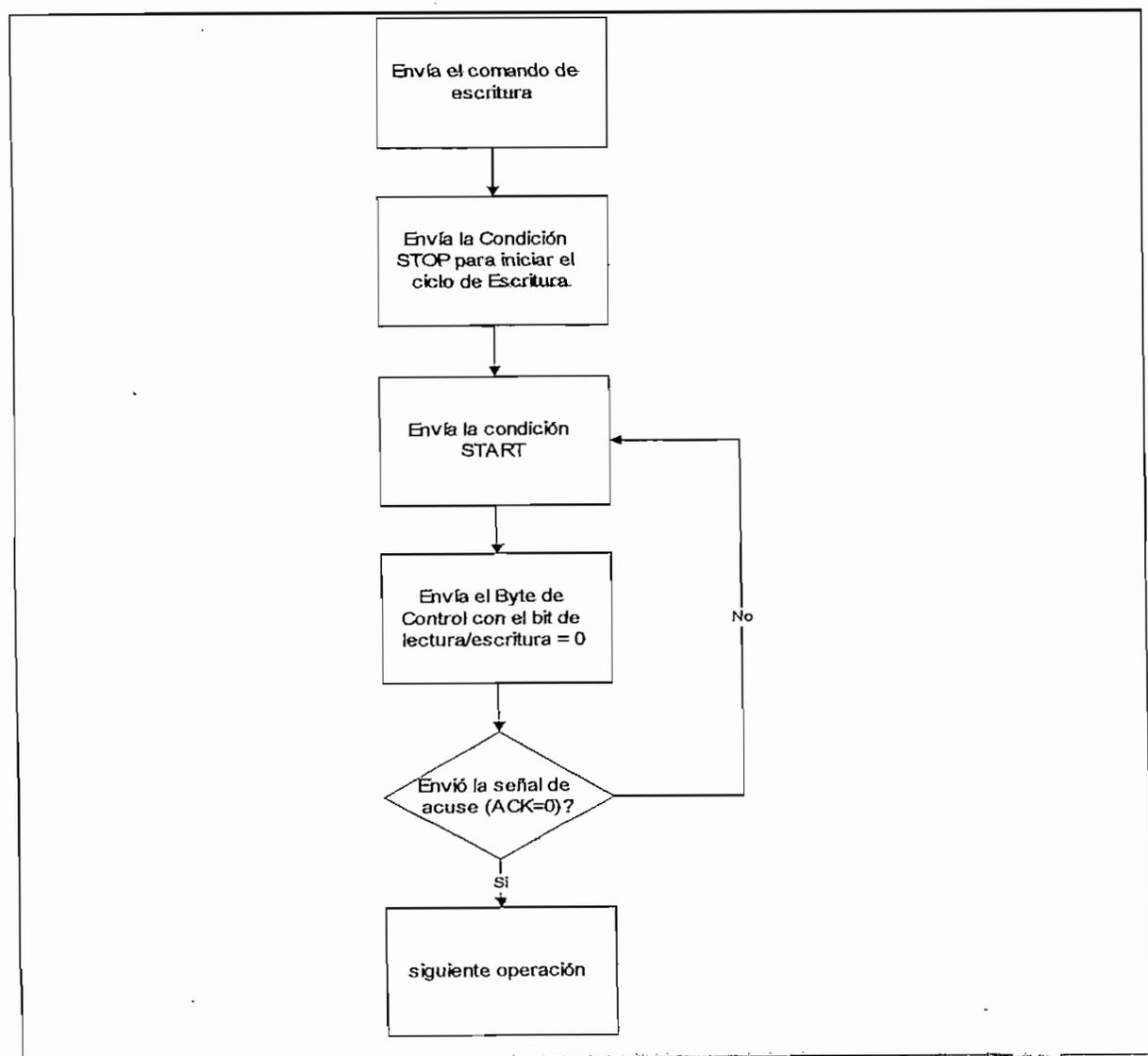


Figura 3-42: Diagrama de Flujo de la Recepción de la señal de acuse

### 3.4.5.5. Subrutinas para la Operación de Lectura

La operación de lectura se inicia de la misma manera que la operación de escritura con la excepción de que el bit de lectura/escritura para el direccionamiento del esclavo se coloca en 1 lógico.

Para ejecutar este tipo de operación, primero se direcciona. Una vez enviada la dirección a la EEPROM como parte de una operación de escritura, el microcontrolador genera una condición de inicio seguida por su acuse. Así se termina la operación de escritura, pero no antes de que el puntero de dirección interno se configure. Luego, el microcontrolador genera nuevamente el byte de control pero con el bit de lectura/escritura en 1 lógico. La memoria generará la señal de acuse y el byte de datos será transmitido. El microcontrolador no genera un acuse pero genera una condición de parada y de esta manera la memoria EEPROM descontinúa su transmisión.

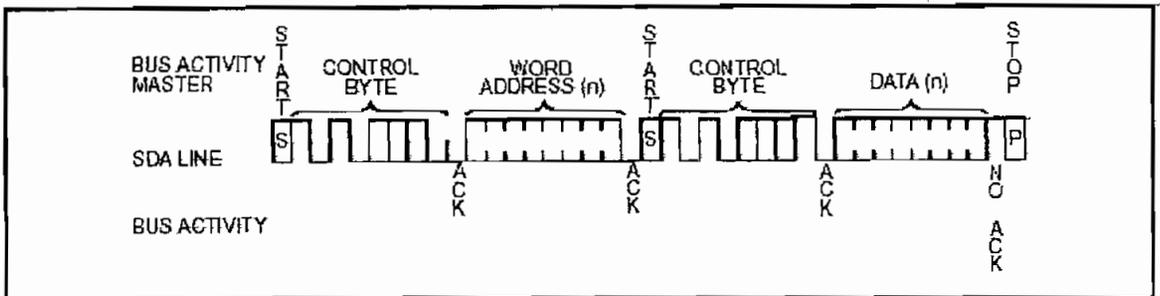


Figura 3-43: Operación de Lectura

`Nak ()` es una función creada con el objeto de generar un acuse negativo en la línea SDA. Su sintaxis es la siguiente:

```
void Nak (void)
```

`Receive()` es una función destinada para recibir un byte sea del PIC o de la memoria EEPROM. Su sintaxis es la siguiente:

```
uchar Receive (void)
```

ReadByte() es una función que genera las condiciones de lectura adecuadas para la recepción de información desde la memoria EEPROM. Su sintaxis es:

```
void ReadByte (uchar deviceNumber, uint address,volatile uchar* BitReceive);
```

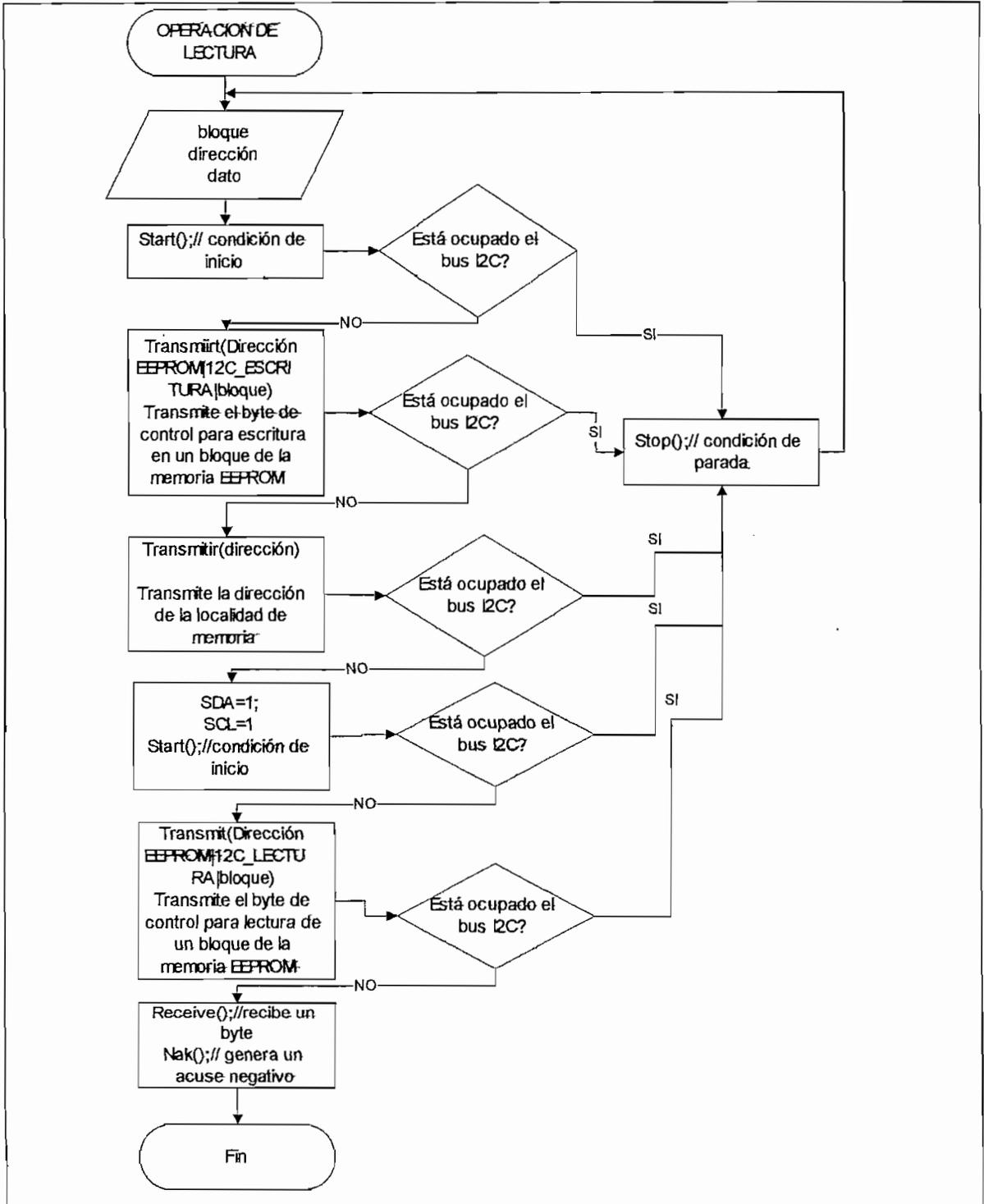


Figura 3-44: Diagrama de Flujo de la Operación de Lectura

### 3.4.6. SUBROUTINAS PARA EL MANEJO DEL DISPLAY LCD

Los displays de cristal líquido son dispositivos que se pueden programar para presentar caracteres alfanuméricos y otros caracteres especiales. Se operan a través de comandos enviados en forma de datos de 4 u 8 bits. Se siguen los siguientes pasos:

1. Esperar a que el display se configure internamente.
2. Definir el ancho de los datos que se van a enviar desde el microcontrolador.
3. Definir si el display permanece estático o se desplaza.
4. Encender el display.
5. Enviar los caracteres.

El puerto C se define como el puerto de comunicación con el módulo de display de cristal líquido LCD. Los bits del puerto C están configurados de la siguiente manera:

PTC0 = Señal de Control (CTRL)

- CTRL = 0 Accede a los registros de instrucción
- CTRL = 1 Accede a los registros de datos

PTC1 = Señal de Lectura /Escritura (READ)

- READ = 0 Modo de Escritura
- READ = 1 Modo de Lectura

PTC2 = Señal de Habilitación de las operaciones de lectura o escritura (STROBE)

PTC3 – PTC6 = líneas para la transferencia bidireccional de los datos en modo de 4 bits entre el microcontrolador y el módulo LCD (DATA).

### 3.4.6.1. Subrutinas de Escritura en los Registros de Control y de Datos

WriteCtrl () se encarga de escribir el valor en el registro de control. Su sintaxis es:

```
void WriteCtrl (unsigned int valor)
```

WriteData() se encarga de escribir el valor en el registro de datos. Su sintaxis es:

```
void WriteData (unsigned int valor)
```

WriteCommon() es una función común para las dos subrutinas anteriores.

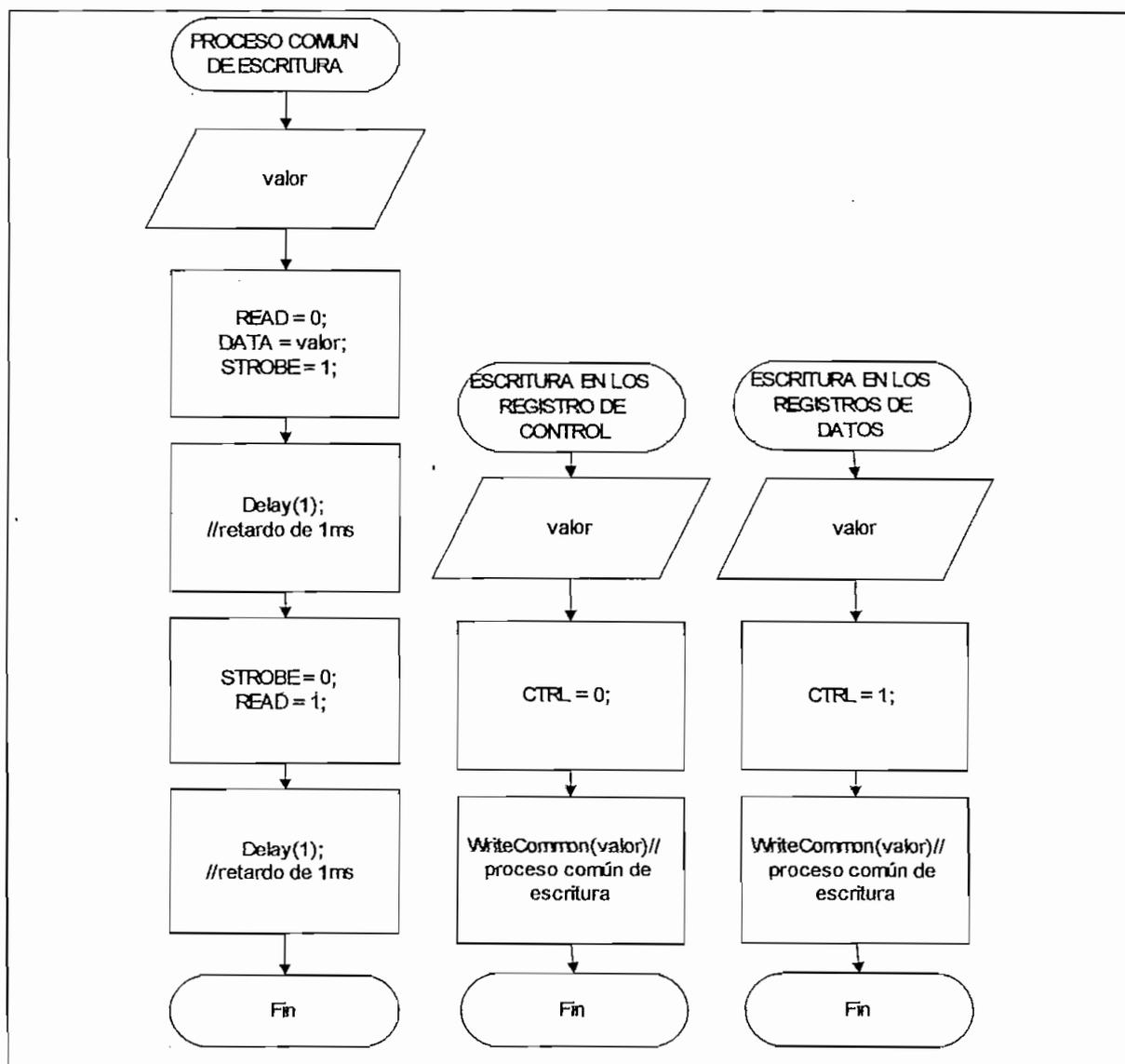


Figura 3-45: Diagramas de Flujo de las Operaciones de Escritura en los Registros de Control y Datos del Módulo LCD y Subrutina de uso común

### 3.4.6.2. Subrutinas para la Ubicación del Cursor

CursorHome () es una función que se encarga de regresar el cursor a la posición inicial. Su sintaxis es: *void CursorHome (void)*.

CursorSet () es una función que se encarga de ubicar el cursor en una posición deseada. Su sintaxis es la siguiente: *void CursorSet (unsigned char pos )*.

SetLineTwo () es una función que ubica al cursor en la segunda línea del display. Su sintaxis es: *void SetLineTwo (void)*.

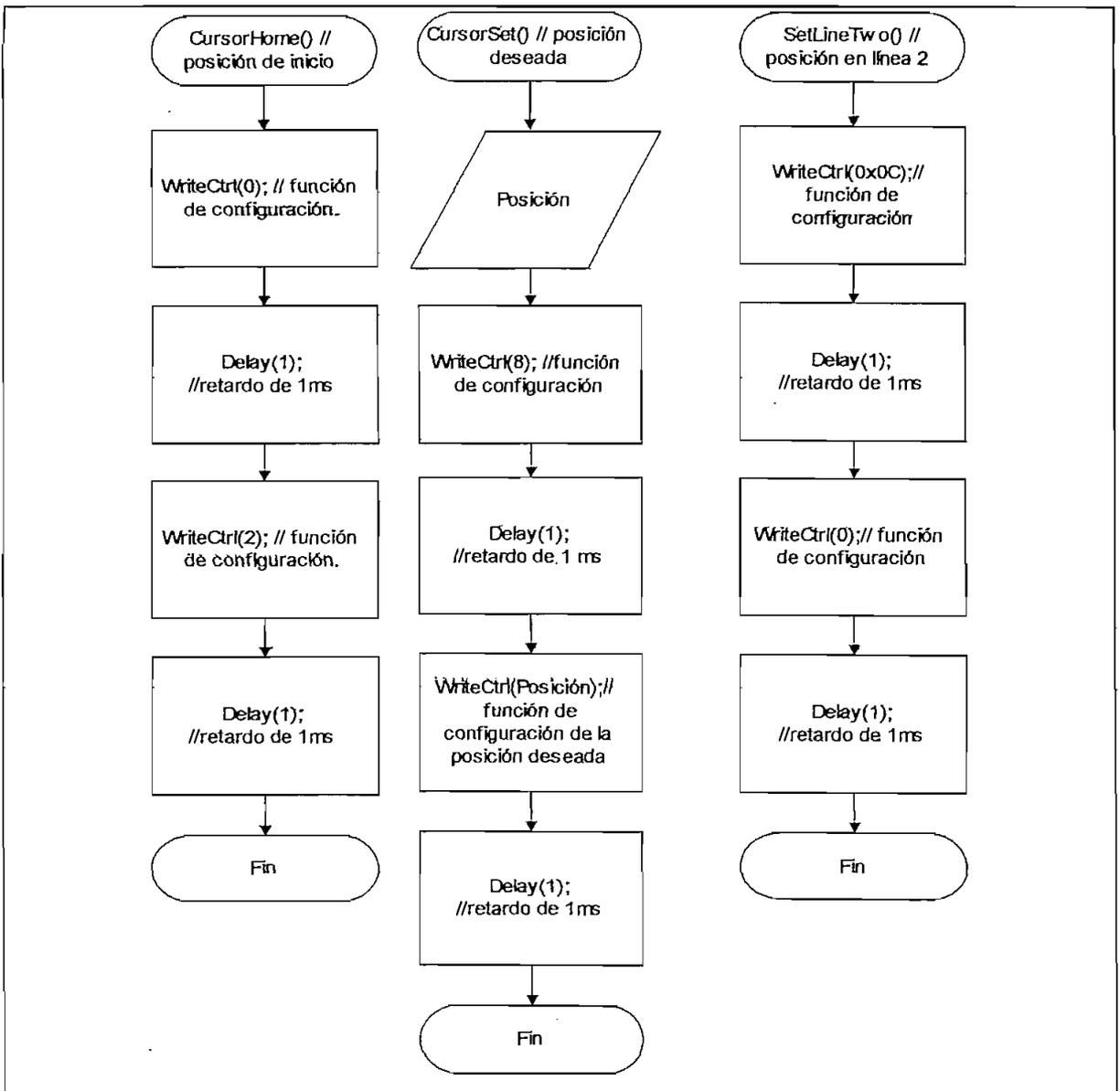


Figura 3-46: Diagrama de Flujo para las Operaciones de Ubicación del Cursor

### 3.4.6.3. Subrutinas de Borrado y Escritura en el LCD

ClearLCD () es una subrutina creada para borrar el contenido del LCD. La sintaxis es la siguiente:

```
void ClearLCD (void)
```

La subrutina WriteLCD () se encarga de enviar un mensaje a la pantalla del LCD. La sintaxis es la siguiente:

```
void WriteLCD (void)
```

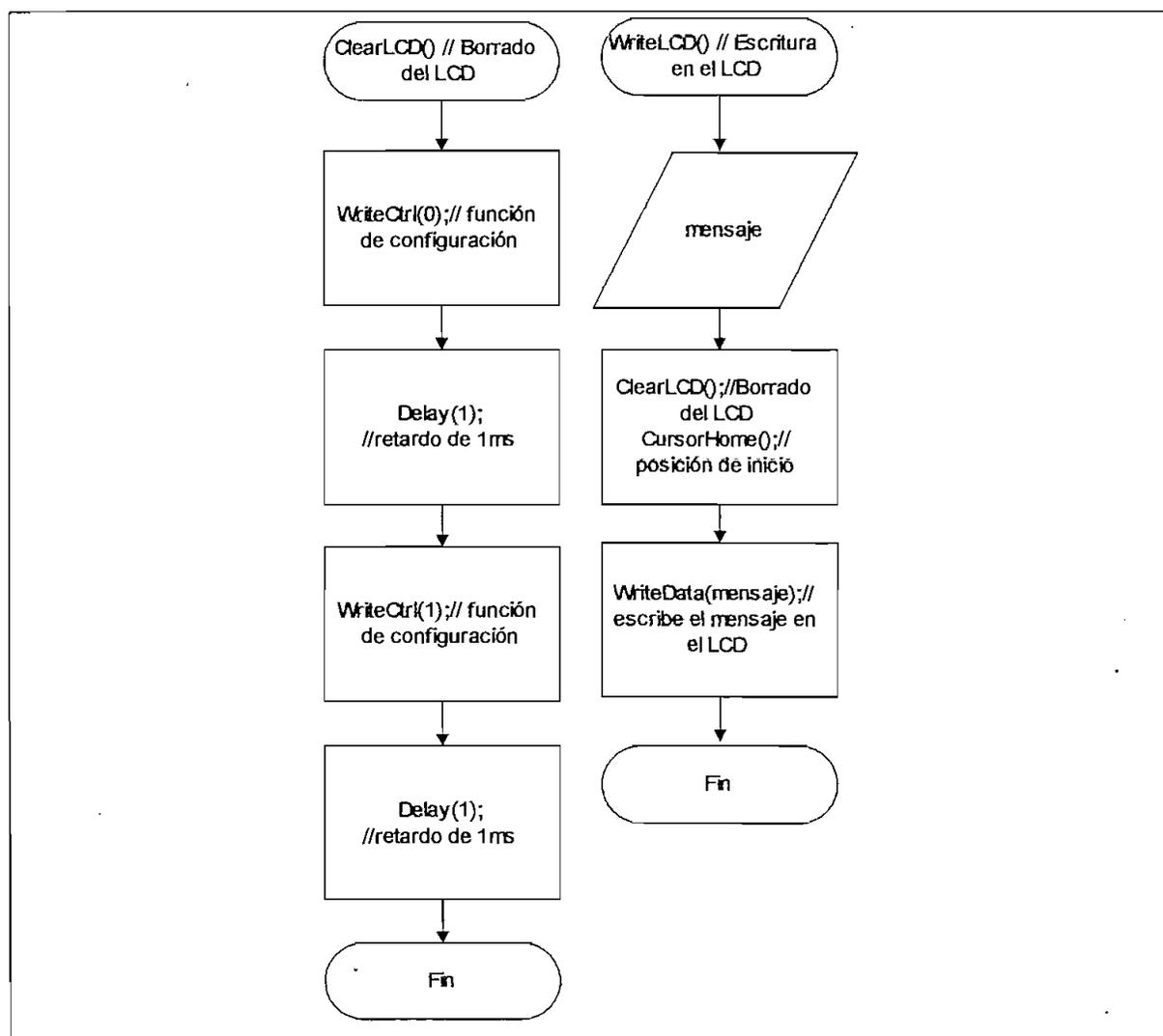


Figura 3-47: Diagramas de Flujo de las Operaciones de Borrado y Escritura en el LCD

### 3.4.6.4. Subrutina de Inicialización del LCD

La subrutina que se encarga de la inicialización del LCD a 4 bits se denomina `InitLCD` y tiene la siguiente sintaxis:

```
void InitLCD (void)
```

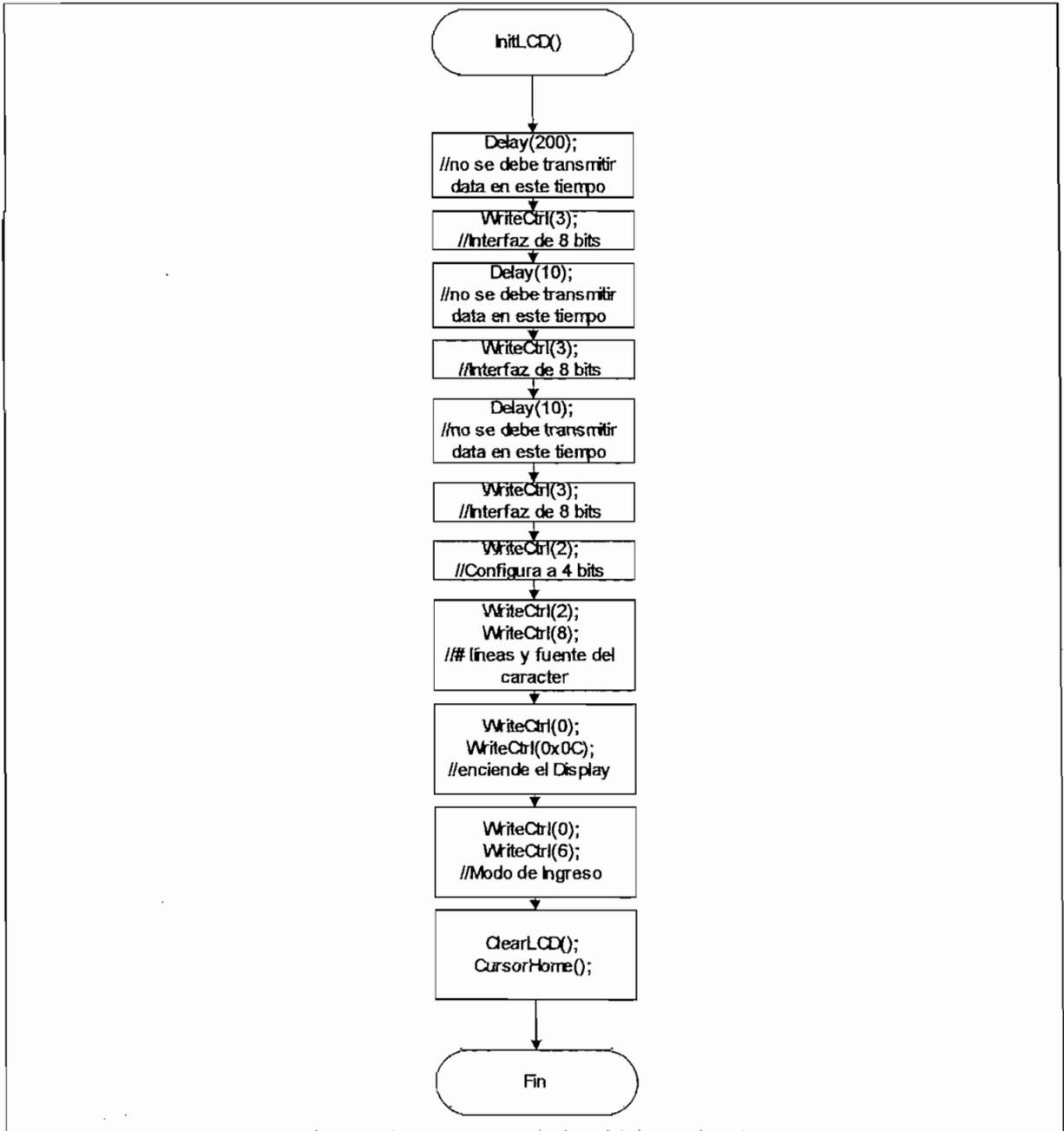


Figura 3-48: Subrutina de Inicialización del módulo LCD.

### 3.4.7. SUBRUTINAS PARA LA COMUNICACIÓN SERIAL RS232

El microcontrolador posee un módulo dedicado para la interfaz de comunicación serial, el cual permite en este caso la comunicación con el computador. Entre las características importantes están:

- Operación asincrónica full-duplex de alta velocidad
- Formato estándar NRZ
- 32 velocidades (baudios) programables
- Programable el tamaño del carácter de 8 o 9 bits
- Separadas las habilitaciones de Transmisión y Recepción
- Doble Buffer de recepción y transmisión

Aprovechando estas características y bajo una combinación entre los registros generados para el controlador diseñado y los del microcontrolador se manipula los parámetros para que al final se pueda configurar al controlador de acuerdo a las necesidades requeridas de comunicación. Por tal razón, a continuación se hace una breve explicación de los registros e interrupciones que componen el módulo serial.

Al micro le llegan interrupciones generadas en la transmisión, en la recepción, por efectos de ruido, de paridad, etc. Los registros que maneja se muestran en la Tabla 3-8.

**Tabla 3-8: Registros del Módulo de Comunicación Serial**

SCC1: SCI control register 1
SCC2: SCI control register 2
SCC2: SCI control register 3
SCSI: SCI status register 1
SCS2: SCI status register 2
SCDR: SCI data register
SCBR: SCI baud rate register

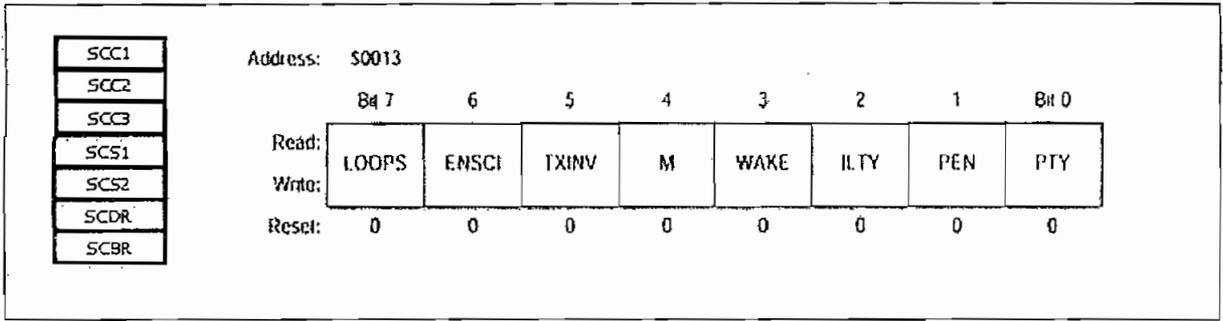


Figura 3-49: Registro SCC1 del Módulo de Comunicación Serial

Del registro SCC1 que se muestra en la Figura 3-49 se determina el formato de datos que se quiere dar para la comunicación.

Para la programación de la velocidad de comunicación (en baudios) se utiliza la siguiente fórmula:

$$\text{velocidad en baudios} = \frac{\text{fuente de reloj del SCI}}{64 \times \text{PD} \times \text{BD}} \quad \text{Ecuación (3-4)}$$

donde la fuente de reloj es la frecuencia del bus interno, PD y BD son dos parámetros de los registros del microcontrolador.

Por ejemplo, para generar una velocidad a 9600 se procede de la manera siguiente:

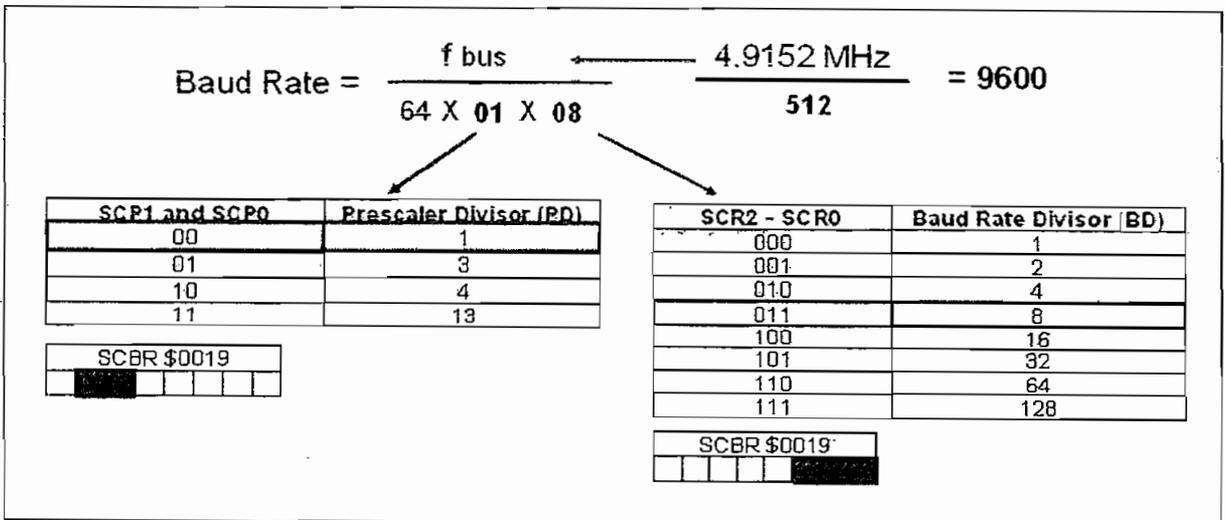


Figura 3-50: Selección de la velocidad de comunicación serial

### 3.4.7.1. Subrutinas de Transmisión y Recepción de Datos

En las Figuras 3-51 y 3-52 se indica el procedimiento que requiere el microcontrolador para la transmisión de datos. La bandera SCTE en 1 lógico indica que el puerto está libre y se puede enviar información. Una vez transmitida la información a través del pin 0 del puerto E (TxD) se levanta la bandera TC que indica que se completó la transmisión.

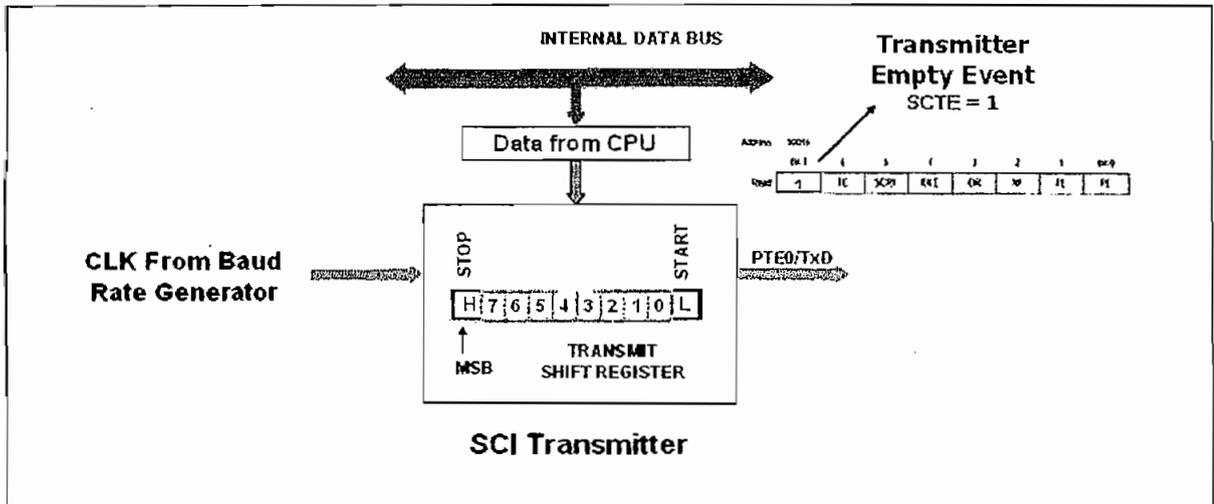


Figura 3-51: Inicio del Proceso para la transmisión de datos en el puerto serial

TransmitComm() es una subrutina creada con el objetivo de enviar la información desde el microcontrolador a un periférico externo y su sintaxis es la siguiente:

```
void TransmitComm (uchar* sMessage)
```

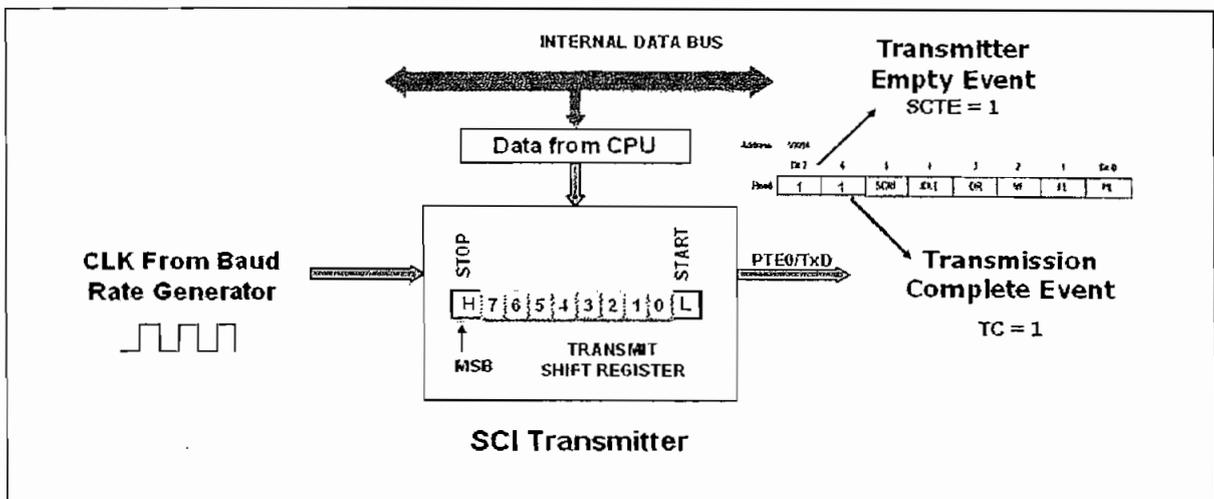


Figura 3-52: Fin del Proceso de Transmisión Serial

SCI\_Rx\_ISR () es una subrutina de interrupción en la recepción de datos que toma acciones de configuración en el controlador basado en un protocolo determinado en la subrutina Command\_Prefix (). La sintaxis es la siguiente:

```
interrupt void SCI_Rx_ISR (void)
```

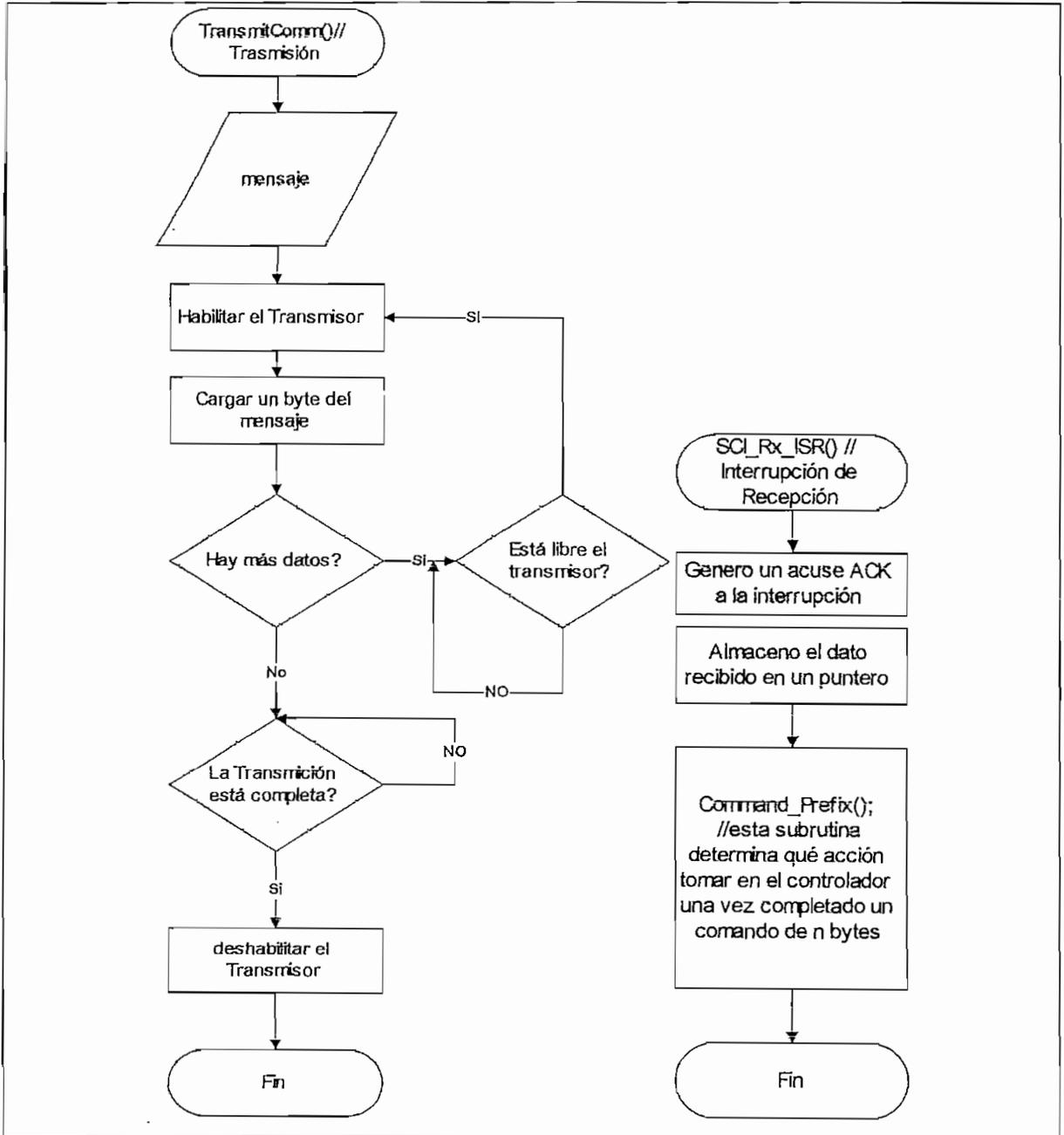


Figura 3-53: Diagramas de Flujo de la Transmisión Serial e Interrupción de Recepción

### 3.4.8. SUBROUTINAS PARA LA ADQUISICION DE LA VARIABLE MEDIDA

El ingreso de la variable a ser medida por el controlador se la realiza a través del módulo del conversor análogo a digital (ADC) que comparte el puerto B. Entre sus características principales se tiene:

- Posee ocho canales con una entrada multiplexada
- Resolución de 8 bits

Mediante la manipulación de los registros del microcontrolador descritos en la Tabla 3-9, se configura el controlador diseñado para que varíe de acuerdo a los requerimientos de la variable de medición.

Tabla 3-9: Registros del Módulo ADC del microcontrolador

ADC Status and Control Register (ADSCR)
ADC Data Register (ADR)
ADC Clock Register (ADCLK)

Dentro del registro ADSCR que se muestra en la Figura 3-54, se determina el canal de entrada al conversor.

Address: \$0003C									
		Bit 7	6	5	4	3	2	1	Bit 0
Read:		COCO/ IDMAS	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
Write:									
Reset:		0	0	0	1	1	1	1	1

Figura 3-54: Registro ADSCR del conversor análogo – digital del microcontrolador

La subrutina de adquisición de la variable medida selecciona primero el canal y el filtro de entrada, entrega el valor promediado al algoritmo de control convertido en un valor de punto flotante y al display para que sea visualizado. Su sintaxis es:

```
void ADC_adquirir (uchar inCanal)
```

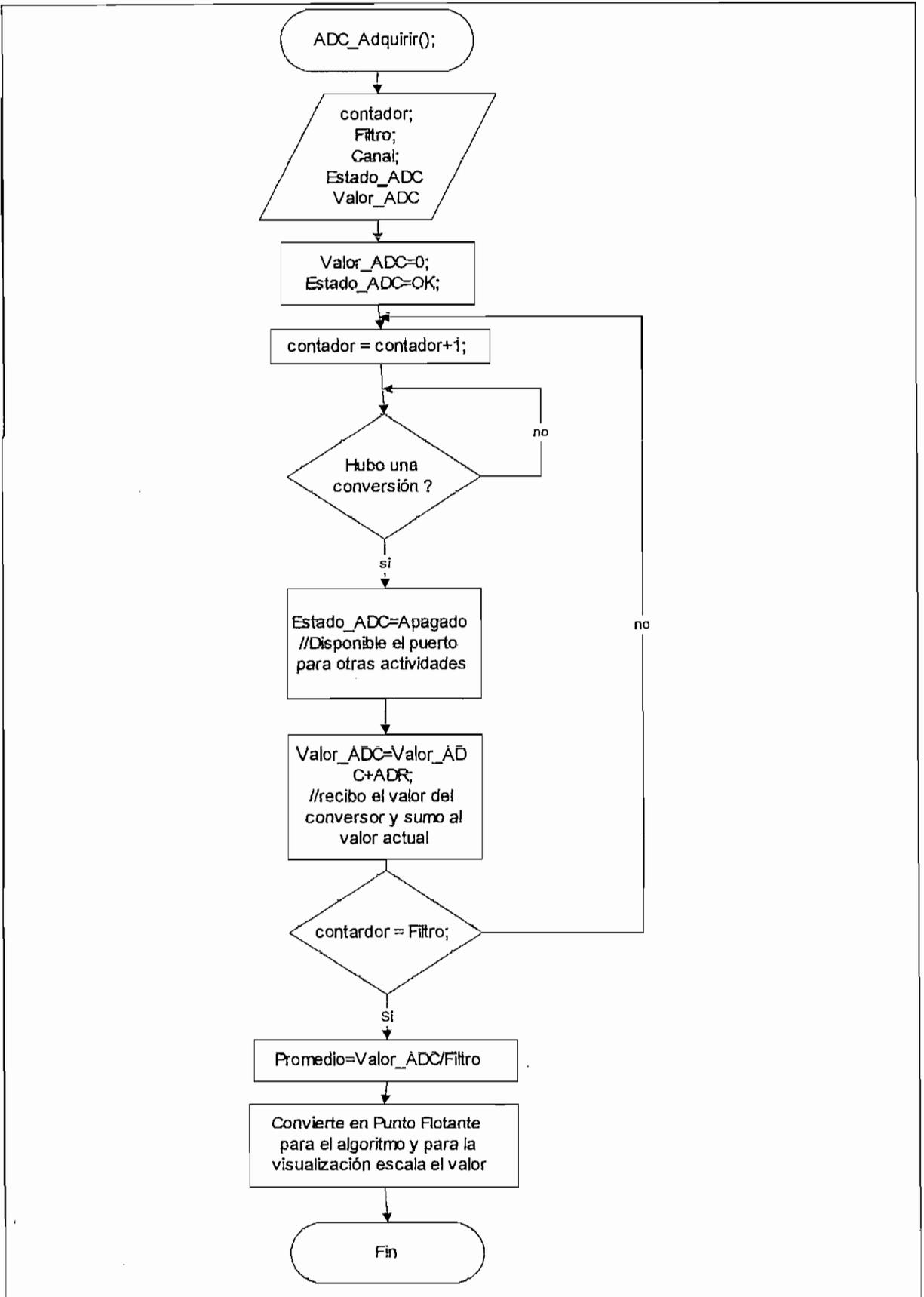


Figura 3-55: Diagrama de Flujo de la Subrutina de Adquisición de la Variable Medida

### 3.4.9. SUBRUTINAS PARA LA INTERFAZ CON EL TECLADO

Una de las características importantes en el desarrollo del controlador universal es el manejo de un menú para su correcta configuración. Es aquí la parte más extensa en la programación debido al número de lazos que se debe manejar correctamente a través de un teclado reducido. Esta estructura de programación, aparentemente compleja, cada vez es más utilizada en equipos que poseen un menú de configuración como la telefonía celular en la que los usuarios pueden acceder sin mayor dificultad.

El módulo de interfaz con el teclado del microcontrolador solamente contiene dos registros: Registro de Estado - Control y Registro para la Habilitación de las Interrupciones de Teclado.

Entre las características más importantes que tiene el módulo de interfaz con el teclado están:

- 8 entradas de teclado por el puerto A
- Atiende interrupciones de teclado independientes
- Tiene búferes de histéresis para eliminar rebotes

La subrutina que responde a un requerimiento de interrupción de teclado y que determina qué tecla ha sido presionada, tiene la siguiente sintaxis en la programación:

*interrupt void Polling (void)*

La subrutina que toma una acción dependiendo de la posición en el menú tiene la siguiente sintaxis:

*void key\_interface (keyE inkey)*

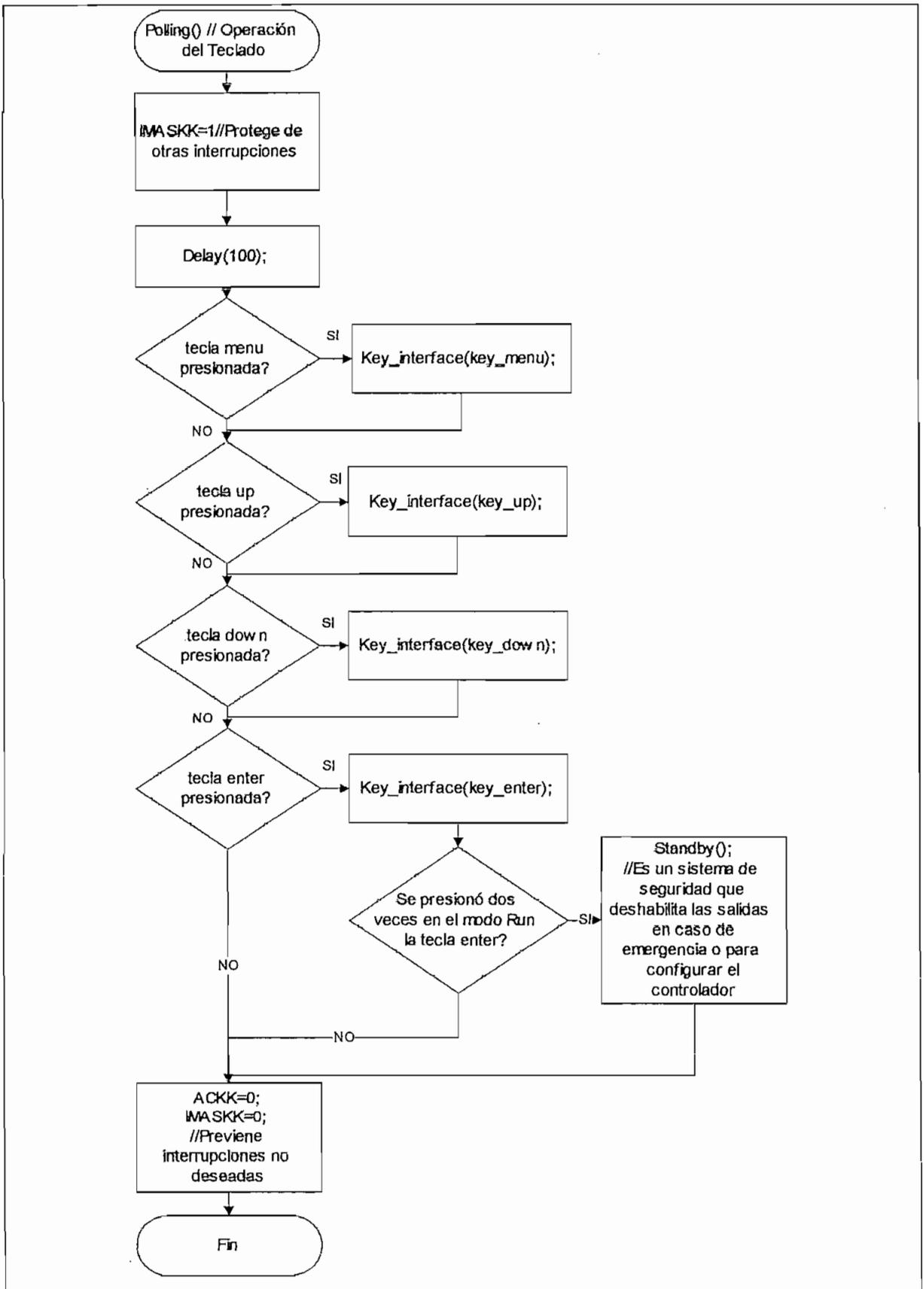


Figura 3-56: Diagrama de Flujo de la Operación de Teclado

### 3.5. SOFTWARE DE INTERFAZ CON EL USUARIO

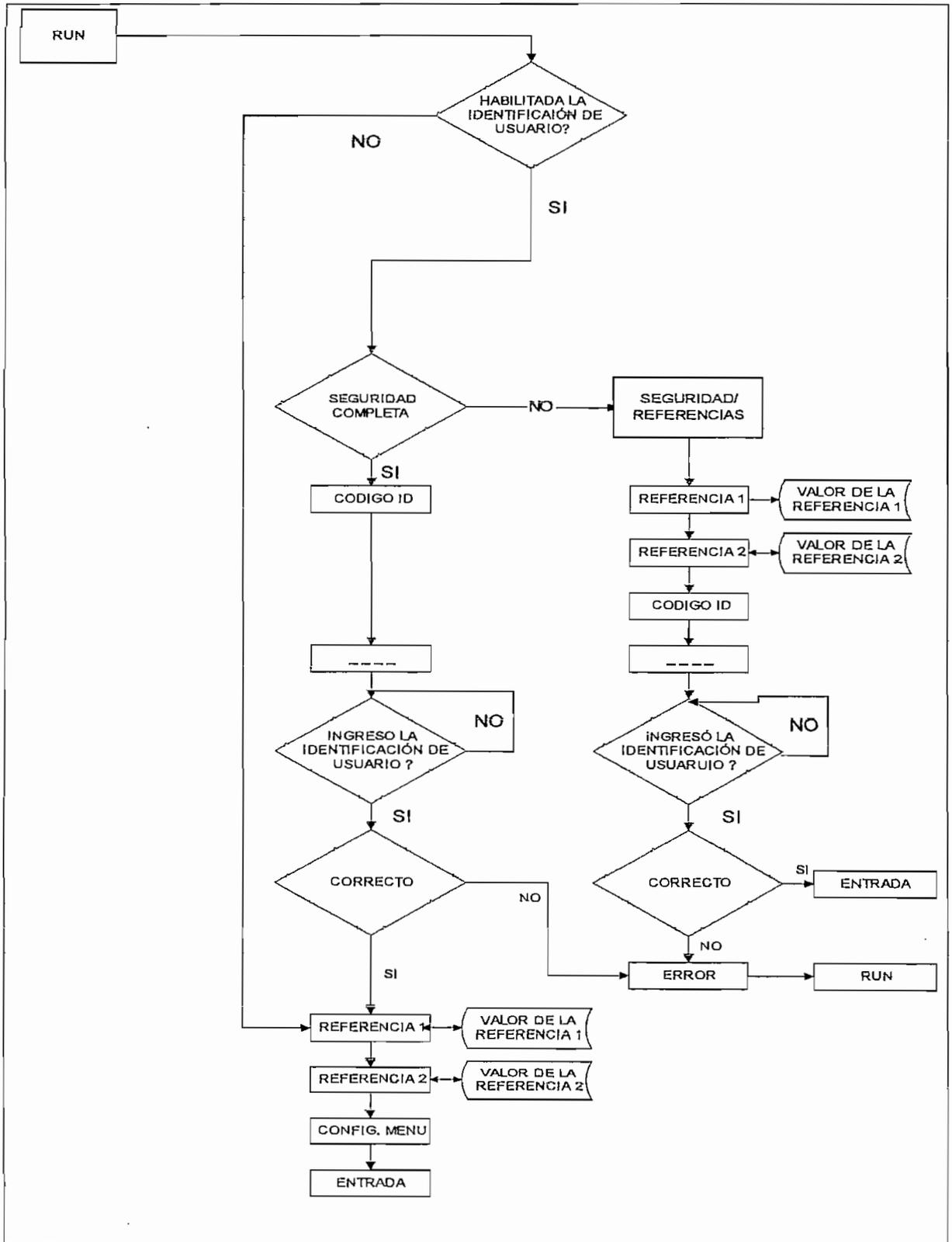


Figura 3-57: Diagrama de Flujo para ingreso al Menú de Configuración

El software de interfaz con el usuario permite navegar en el menú del controlador por medio del teclado y el display. En la Figura 3-57 se puede observar el diagrama de flujo para el ingreso al menú. Si no está habilitada ninguna identificación de usuario, se ingresa hacia los valores de referencia y al menú de configuración, caso contrario, requiere del ingreso de los dígitos.

Cuando está operando el controlador, el display indica los valores de referencia y el valor actual de la variable a ser medida como se indica en la Figura 3-58.

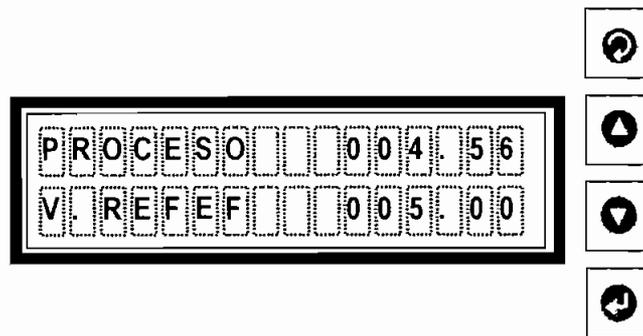


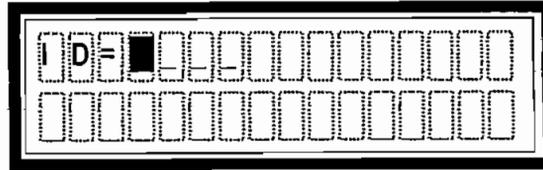
Figura 3-58: Visualización en modo de operación

Para ingresar al menú, se define al teclado como indica la Tabla 3-10.

Tabla 3- 10: Descripción de las teclas del controlador diseñado

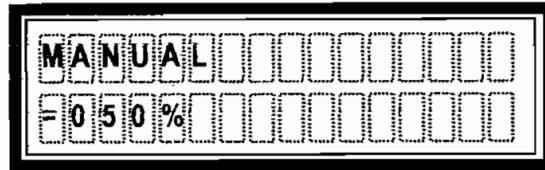
TECLA	DESCRIPCION
	Cambia el display al modo de configuración y navega a través de los ítems del menú.
	Usado en el modo de programación y control manual
	Usado en el modo de programación y control manual
	Accede a los submenús en el modo de configuración y almacena los valores seleccionados.

La pantalla que pide la clave para ingresar parpadea en el primer dígito y espera por un lapso de 3 segundos (Figura 3-59). En caso de no ingresar correctamente los cuatro dígitos, sale un mensaje de **ERROR** y regresa al estado de operación.



**Figura 3-59: Ingreso de los Cuatro Dígitos para la Identificación del Usuario**

Al presionar la tecla  mientras esté presionada la tecla  se habilita o deshabilita la opción del control manual para la salida 1. El display muestra la pantalla que indica la Figura 3-60 al presionar las teclas  o .



**Figura 3-60: Visualización del Control Manual**

En la Tabla 3-11 se indican todos los mensajes que aparecen en el display de acuerdo al ítem del menú seleccionado.

**Tabla 3-11: Detalle de los Ítems del Menú de Configuración**

ITEM DEL MENU	DISPLAY	VALOR PREDEFINIDO	DESCRIPCION
SP1	REFERENCIA 1	0000.0	Valor de referencia para la salida 1
SP2	REFERENCIA 2	0000.0	Valor de referencia para la salida 2
CNFG	CONFIGURACION		Configuración del controlador
INPT	A ENTRADA		Entrada al convertidor
PROC	ENTRADA DE PROCESO:	0-5 V	Determina la entrada al controlador
RDG	B LECTURA		Menú de configuración de lectura
DEC	B1 DECIMAL	FFF.F	Indica la posición del punto decimal
FLTR	B2 CNST.FILTRO	4	La constante del filtro a la

			entrada
IN.RD	B3 ENT-LECTURA		Para la visualización
IN1	ENTRADA1	0000.0	
RD1	LECTURA1	0000.0	
IN2	ENTRADA2	1000.0	
RD2	LECTURA2	1000.0	
OUT1	C SALIDA1		Configuración de la salida 1
%LOW	MIN%bajo	000%	Mínimo valor de la salida
%HI	MAX%alto	100%	Máximo valor de la salida
ANLG	SALIDA ANALOGICA	Habilitada	Salida de voltaje
CTRL	CONTROL	PID	Tipo de control
ON-OFF	ON-OFF		
ACCION	ACCION	DIRECTA	Tipo de acción directa o inversa
DEAD	BD MUERTA	0020.0	Ancho de la banda muerta o histéresis.
ANTI	ANTI - INTEGRAL	DESHABILITADA	
PROP	BD PROPORCIONAL	0020.0	Banda Proporcional
REAJUSTE	REAJUSTE [S]	0000.0	Acción Integral
RATE	DERIVATIVO [S]	0000.0	Acción Derivativa
CICLO	CICLO DE TIEMPO	007.	El ciclo de encendido y apagado para la salida proporcional en el tiempo
OUT2	D SALIDA2		Configuración de la salida 2
ALARMA	ALARMA		Alarma en la salida 2
ACTIVAR	ACTIVAR	SOBRE	Tipo de accionamiento de la alarma
ALR.L	VALOR BAJO	0900.0	Valor bajo para la alarma
ALR.H	VALOR ALTO	0010.0	Valor alto para la alarma
ID	E ID		Identificación
DEFT	PREDETERMINADO	0000	Valor por defecto
CH.ID	CAMBIAR ID		Cambia el valor del ID
COMP.	COMPLETA/ID	Deshabilitada	Seguridad completa del controlador
SP.ID	REFERENCIA/ID	Deshabilitada	Permite cambiar los valores de referencia pero no ingresa al menú de configuración
COMM.	f COMM		Comunicación serial
C.PAR	PARAMETROS DE		Configuración de los

	COMUNICACION		parámetros para la comunicación
BAUD	BAUD RATE	9600	Velocidad de comunicación
PRTY	PARIDAD	IMPAR	No, par o impar
D.BIT	BITS DE DATA	8 bits	7 u 8 bits
STOP	BITS DE PARADA	1 bit	
LCD	G CONTROL DEL LCD		
CONTR.	CONTRASTE		Determina el contraste del display

En las Figuras 3-61 y 3-62 se muestran los modos de ingreso a los menús de configuración del controlador mediante el teclado.

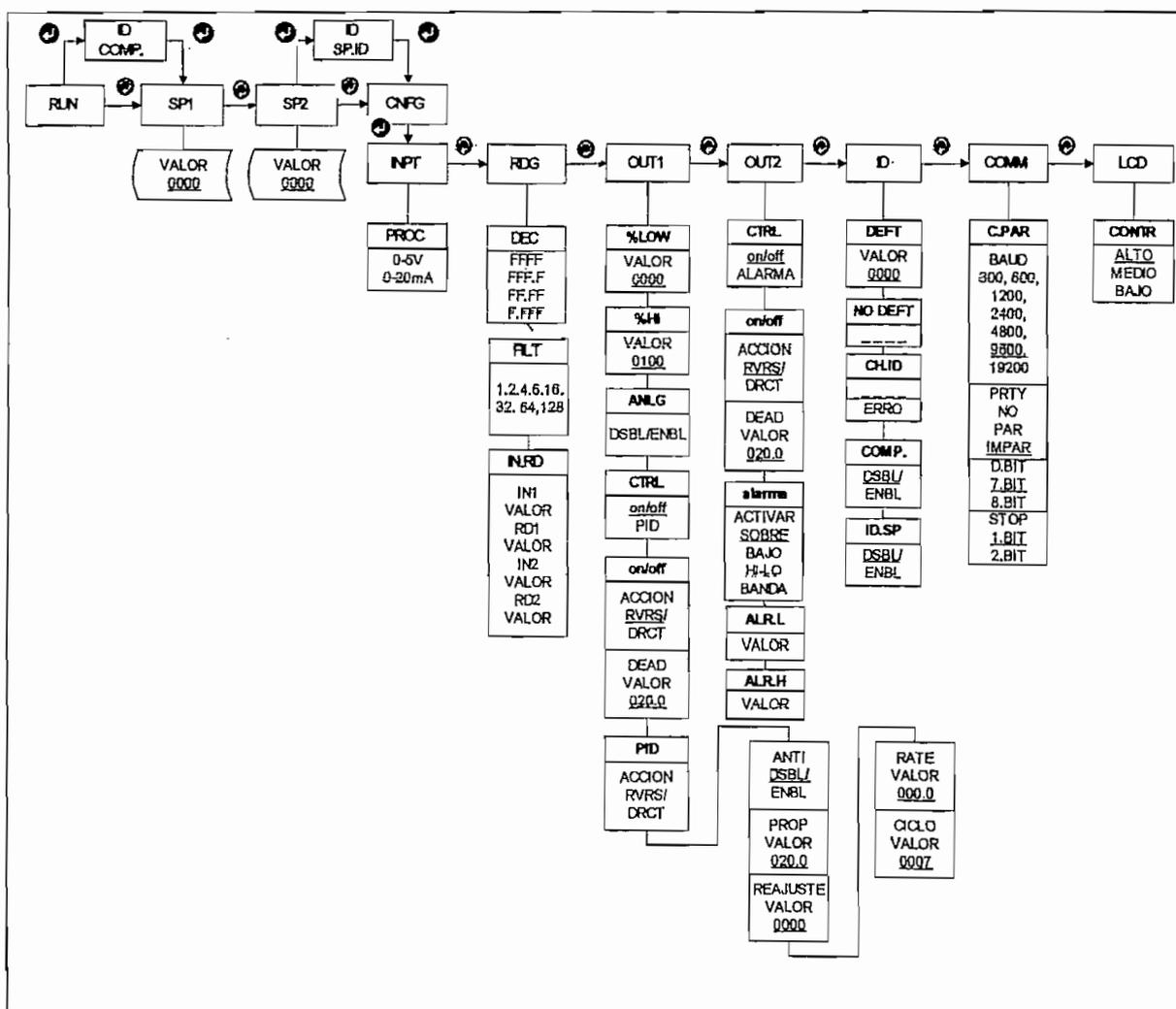


Figura 3-61: Diagrama de Bloques del Menú de Configuración

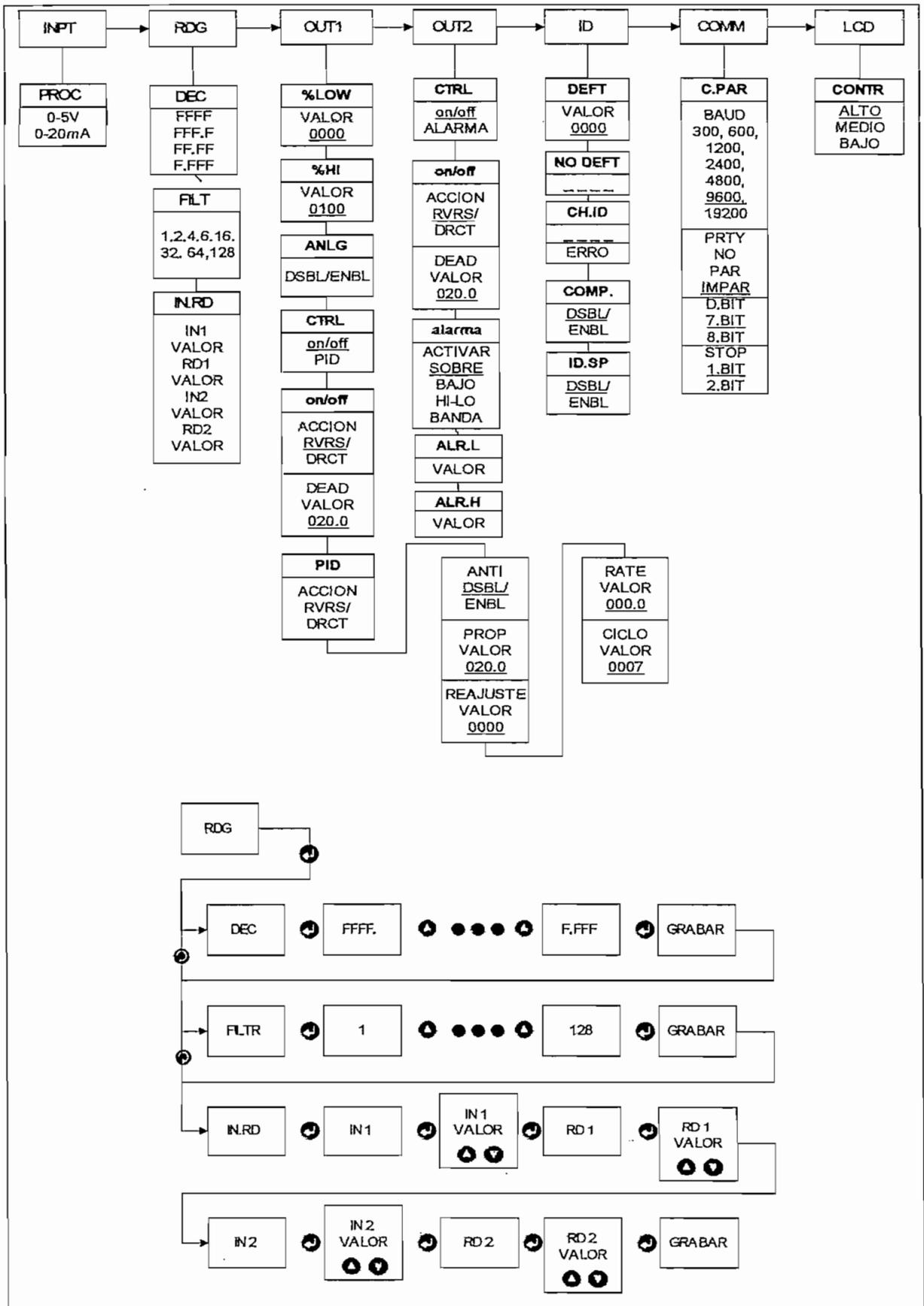


Figura 3-62: Menú de Configuración y Submenú de la Configuración de Lectura.

### 3.6. SOFTWARE DE INTERFAZ CON EL COMPUTADOR

El software de interfaz con el computador contiene el protocolo de comunicación de datos para las reglas y la estructura de los mensajes utilizados en el controlador para la transmisión. Una transacción normal consistirá en el envío de un requerimiento desde el computador seguido de la respuesta del controlador.

Se debe configurar primeramente los parámetros de comunicación en el controlador mediante el teclado del panel frontal.

Existen diferentes tipos de comandos asociados en la comunicación como se indica en la Tabla 3 -12, la misma que indica los prefijos de comando.

**Tabla 3-12: Letras Prefijos de Comando**

PREFIJO DEL COMANDO	SIGNIFICADO
P o p(Put)	Escribe un valor HEX en la RAM del controlador
W o w(Write)	Escribe un valor HEX en la EEPROM
R o r(Read)	Lee un valor HEX desde la RAM
D o d	Deshabilita
E o e	Habilita
Z o z	Reset

El formato de los comandos para el controlador es:

**Tabla 3-13: Formatos de los comandos**

"P" y "W"	"R", "D", "E" y "Z"
ccc<data><cr>	ccc<cr>

Donde:

"ccc" representa el caracter ASCII del prefijo del comando seguido por dos caracteres ASCII que son los sufijos que identifican los datos, las características del controlador o los ítems del menú para los que el comando está diseccionado.

“data” es la cadena de caracteres que contiene la información de la variable que el computador está enviando al controlador

“<cr>” representa el caracter ASCII de la tecla retorno en un computador.

En la Tabla 3-14 se indica las letras de los prefijos y sufijos para los comandos del controlador diseñado.

**Tabla 3-14: Letras de los Prefijos y Sufijos para los comandos**

COMANDO	INDICE	FUNCION	BYTES	# DE CARACTERES
RWP	01	SP1	3	6
RWP	02	SP2	3	6
RWP	03	IN1	2	4
RWP	04	RD1	2	4
RWP	05	IN2	2	4
RWP	06	RD2	2	4
RWP	07	INPT	1	2
RWP	08	RDG	1	2
RWP	0A	ALARMA	1	2
RWP	0C	OUT1	1	2
RWP	0D	OUT2	1	2
RWP	10	C.PAR	1	2
ED	03	standby	0	0
Z	02	RESET	0	0

Después de cada modificación que se haga a la configuración a través de los comandos con el prefijo W, se debe enviar un comando de RESET para que los cambios se efectúen.

Para resetear el controlador, se envía el comando “Z02” (Tabla 3-14).

Si se quiere leer el valor de Referencia 1, el comando que se debe escribir en el computador es “R01”.

Cuando se quiere cambiar el valor de Referencia 1, por ejemplo a 10.00 se envía el comando "W012003E8" como se explica en la Tabla 3-15.

A los 3 bytes (2 caracteres hexadecimales en cada byte) del valor de referencia se define como SETPOINT.23-0. Donde 23-0 son las 24 (3\*8 bits) posiciones binarias de los bits.

**Tabla 3-15: Descripción del comando de escritura del valor de referencia 1**

SETPOINT 23=	SETPOINT 21-20=	SETPOINT 19-0=
0 = Signo positivo	00 – Punto decimal 1 (FFFF.)	Dato del valor de Referencia
1 = Signo negativo	01 – Punto decimal 2 (FFF.F)	
	10 – Punto decimal 3 (FF.FF)	
	11 – Punto decimal 4 (F.FFF)	

Para 10.00: Signo positivo = 0, Punto decimal 3 = 10 Bin, Dato 1000 = 3E8 Hex = 001111101000 Bin.

El comando = 0010 0000 0000 0011 1110 1000 Bin = 2003E8 Hex

2 0 0 3 E 8 Hex

Se envía por tanto W012003E8.

Los espacios no están permitidos en la cadena de datos.

Para cambiar el valor de referencia a -10.00 se envía el comando "W01A003E8". Signo negativo = 1, Punto decimal 3 = 10 Bin, Dato 1000 = 3E8 Hex = 001111101000 Bin.

El comando = 1010 000 0000 0011 1110 1000 Bin = A003E8 Hex

A 0 0 3 E 8 Hex

Para modificar la configuración del controlador se hace referencia primeramente a los registros que fueron creados descritos al inicio de este capítulo. Así por ejemplo, el registro de configuración de lectura tiene el formato que se describe en la Tabla 3-16.

**Tabla 3-16: Formato para la Configuración de Lectura**

BITS DEL REGISTRO								FUNCIÓN
7	6	5	4	3	2	1	0	
						0	0	Punto decimal 1 (FFFF.)
						0	1	Punto decimal 2 (FFF.F)
						1	0	Punto decimal 3 (FF.FF)
						1	1	Punto decimal 4 (F.FFF)
	0	0	0					Constante de Filtro 1
	0	0	1					Constante de Filtro 2
	0	1	0					Constante de Filtro 4
	0	1	1					Constante de Filtro 8
	1	0	0					Constante de Filtro 16
	1	0	1					Constante de Filtro 32
	1	1	0					Constante de Filtro 64
	1	1	1					Constante de Filtro 128

Para modificar la configuración de lectura (índice 08) a un punto decimal 2 y con una constante de Filtro 16 se tiene en el comando de datos: 01000001 Bin = 41 Hex. Se envía por tanto el comando "W0841".

Con este protocolo se puede utilizar cualquier terminal que se comuniquen con el controlador a través de comunicación serial y enviar de esta manera los caracteres necesarios. También se pueden crear aplicaciones o integrar a un sistema de control.

### 3.6.1. SOFTWARE EN VISUALBASIC

Para facilitar la programación a través de un computador se ha diseñado un software en base a Visual Basic que se comunica con el controlador y que envía los caracteres pero a través de una interfaz gráfica amigable con el usuario. La Figura 3-63 presenta la pantalla de inicio del Software desarrollado para la Comunicación Serial con el controlador.



**Figura 3-63: Pantalla de Inicio del Software para Comunicación Serial**

El programa tiene varias ventanas o formularios que se activan en el momento de ser requeridos. La ventana principal se aprecia en la Figura 3-64. Esta ventana presenta una carpeta con varias páginas que permiten ingresar los parámetros que se requiera modificar en el controlador. Si los valores que se ingresan son superiores o inferiores a los permitidos por el controlador o si se altera el punto decimal, aparecerán mensajes de alerta para evitar datos erróneos. Si no se quiere que aparezcan este tipo de mensajes nuevamente, el usuario puede desactivarlos después de que hayan aparecido por primera vez. O si se desea que los mensajes de alerta nuevamente se presenten, se presiona el botón de Preferencias y llama a una ventana como se aprecia en al Figura 3-65.

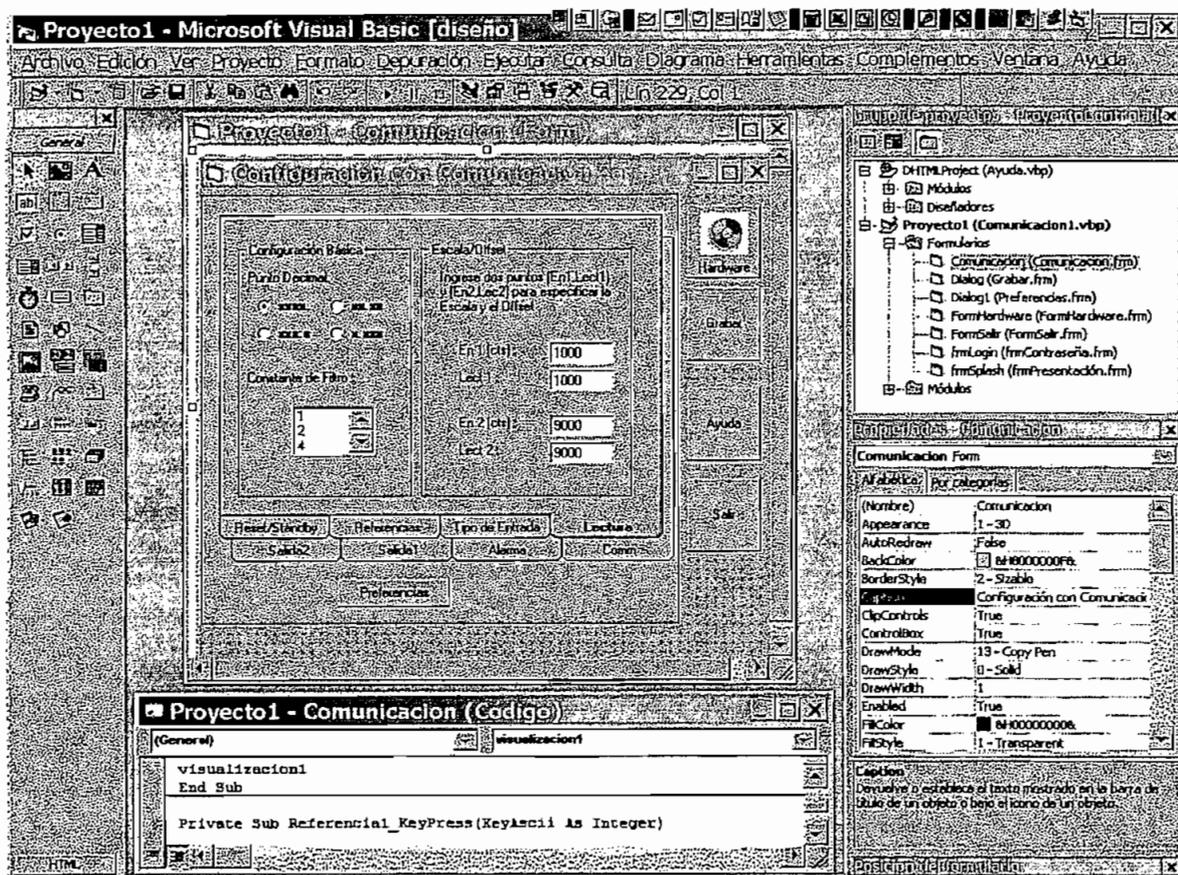


Figura 3-64: Software de Comunicación Serial

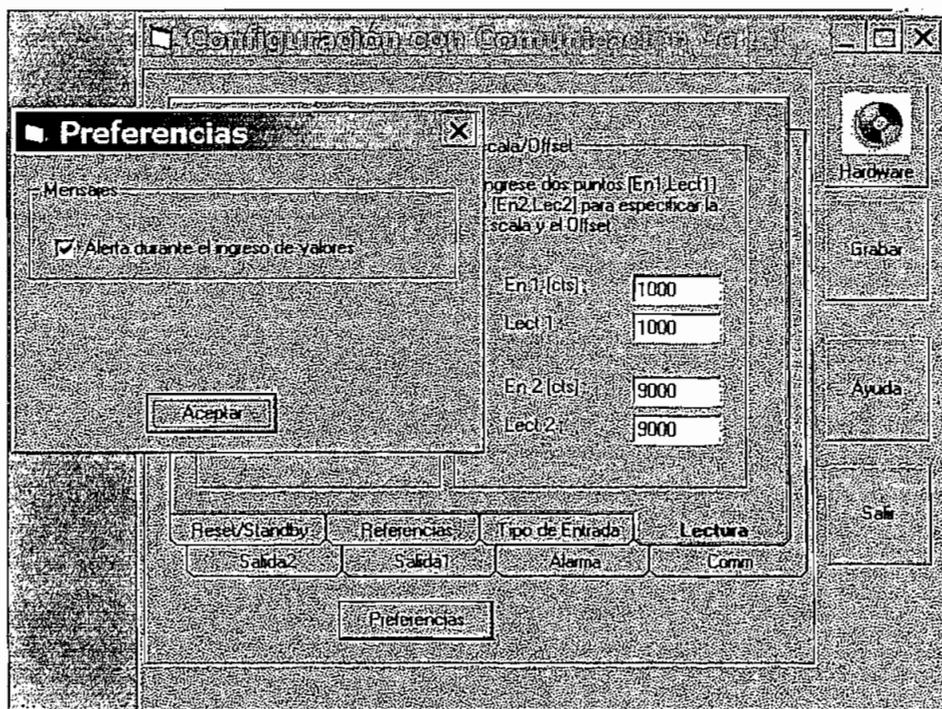


Figura 3-65: Ventana de activación de mensajes de alerta

En la Figura 3-66 se encuentran las diferentes ventanas que presenta el software de VisualBasic para la configuración del controlador.

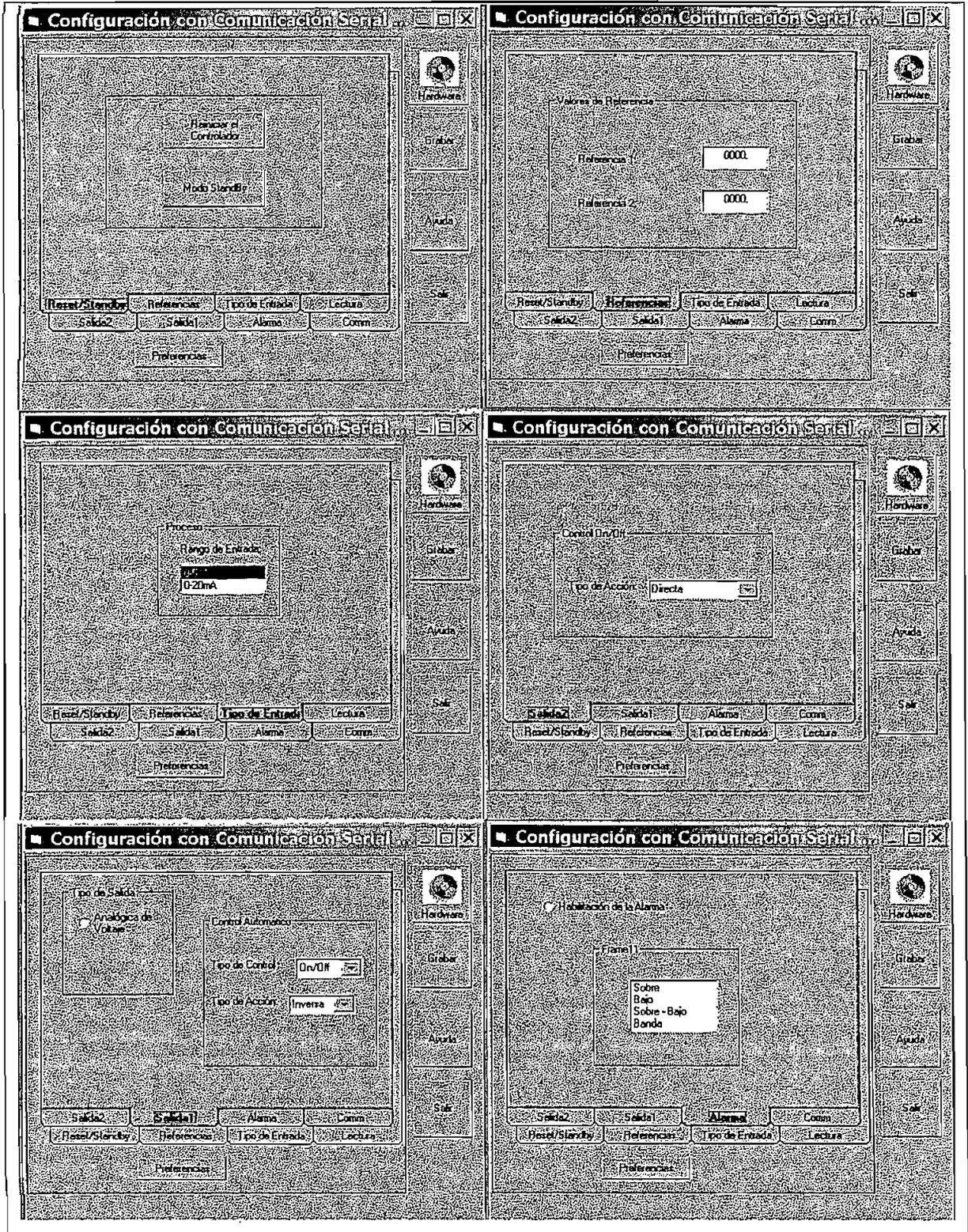


Figura 3-66: Ventanas de Configuración en VisualBasic

Este programa automáticamente cambia el punto decimal y el signo al modificar los valores de referencia utilizando el comando Format\$ como se indica a continuación:

```
Referencia1 = Format(Referencia1.Text, "0000.")
If Referencia1.Text >= 0 Then
ValorSetpoint1 = "W0100" & Format$("0000" + Hex(Referencia1.Text), "@@@@!")
Else
ValorSetpoint1 = "W0180" & Format$("0000" + Hex(-Referencia1.Text), "@@@@!")
End If
```

La primera línea cambia el punto decimal a la posición deseada. La condición if determina si el valor es positivo o negativo. Si este valor es positivo, agrega el prefijo W0100 al valor de referencia, de lo contrario agrega el valor W0180.

En la Figura 3-67 se muestra la ventana de Hardware. En esta ventana se puede simplificar la configuración al indicar las características que se quiere tener en Hardware.

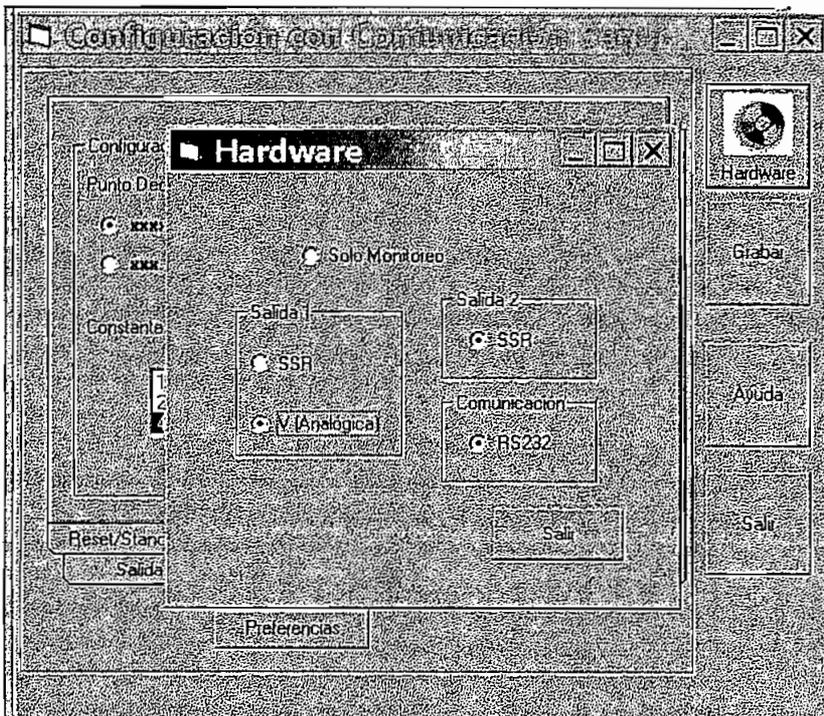


Figura 3-67: Ventana de Hardware

Una vez ingresados los datos que se requiera en el controlador, se configuran los parámetros de comunicación. El software detecta automáticamente cual puerto del computador está disponible para la comunicación con el controlador. Si no existe ningún puerto disponible aparecerá un mensaje resaltado en color rojo indicando el estado del puerto seleccionado, caso contrario aparecerá un mensaje resaltado en color verde indicando el puerto activo como se puede ver en la Figura 3-68. Se presiona el botón de Grabar una vez terminada la configuración.



**Figura 3-68: Ventana de los Parámetros de Comunicación del Computador**

Toda la programación utilizada en este proyecto tanto en el microcontrolador como en el computador, facilita al usuario tener una ágil, dinámica y sencilla configuración del equipo para acoplarlo a un sistema de control.

## **CAPITULO 4. PRUEBAS Y RESULTADOS**

## CAPITULO 4.

### PRUEBAS Y RESULTADOS

Con la finalidad de comprobar el correcto funcionamiento del controlador diseñado se han realizado pruebas de las características que presenta éste.

- Pruebas a la entrada pertenecientes a la variable medida.
- Pruebas a la salida del controlador e integración en sistemas de lazo cerrado.
- Pruebas de comunicación con el computador.

#### 4.1. CONSIDERACIONES ESPECIALES EN LA PROGRAMACIÓN

El algoritmo utiliza variables en punto flotante y hace multiplicaciones y divisiones en cada etapa del control PID. Estos cálculos se deben realizar dentro de un período constante que está determinado por una interrupción con una base de tiempo de 3,333ms que es arbitraria. En realidad el rendimiento del microprocesador y los requerimientos del sistema controlado determinan el período. El caso más crítico se tiene al incluir los tres términos: proporcional, integral y derivativo. El tiempo mínimo que se obtiene para realizar todos los cálculos es de aproximadamente 30ms (37.5Hz). Y en los casos menos exigentes se realizan compensaciones que mantienen el período en este valor.

El formato para las variables de punto flotante comprende 4 bytes o 32 bits:

SEEEEEEE EMMMMMMM MMMMMMMM MMMMMMMM

S representa el bit de signo, E representa un exponente de 8 bits restado 127 y M representa la Mantisa fraccional de 23 bits con un valor de 1 implícito. En un rango de 0 a 255 correspondiente a un puerto de 8 bits de entrada y salida, se escalan los 8 bits más significativos de la variable de punto flotante en el rango de 1.0 (3F800000 Hex) a 1.998046875 (3FFF8000 Hex). Con la siguiente expresión se evalúa un número en punto flotante.

$$F=[(-1)^S][2^{(E-127)}][1.M]$$

*Ecuación (4-1)*

Las operaciones matemáticas mediante el lenguaje C pueden ser escritas directamente como una ecuación. Sin embargo, por tratarse de puntos flotantes, es preferible realizar por tramos las operaciones, de otro modo tiende a desbordarse el programa o a entregar datos erróneos.

Como las variables pertenecientes al algoritmo están siendo utilizadas la mayor parte del tiempo, se las coloca en la sección más baja de la memoria RAM conocida como página cero, la misma que permite un direccionamiento directo y por tanto su acceso es más rápido. Además deben ser variables de tipo global para que estén disponibles fuera del lazo de control, es decir, pueden interactuar con el resto de subrutinas.

La memoria RAM es limitada, por tal razón las designaciones de variables en punto flotantes también son limitadas. Para el cálculo del algoritmo, se reduce a tres variables de punto flotante que ejecuten todas las operaciones de cálculo del algoritmo de control.

## **4.2. PRUEBAS DE LECTURA DE LA VARIABLE MEDIDA**

El rango de visualización que maneja el controlador para la variable de proceso es de 0 a 10.000 unidades. Sin embargo, la resolución del conversor de entrada es de 8 bits, lo que permite tener en realidad 255 incrementos. Es decir se tiene aproximadamente 39 unidades en el rango por incremento (Tabla 4-1).

**Tabla 4-1: Conversión de Unidades**

<b>Voltaje de Entrada [V]</b>	<b>Valor Digital</b>	<b>Visualización [unidades]</b>
0	0	0
2.5	127	5000
5	255	10000

La mayoría de los transmisores de corriente trabajan en un rango de 4 a 20 mA, por esto se debe asegurar que el punto decimal esté configurado en la posición correspondiente, ya que el rango de 4 a 20 mA debe corresponder al rango de 0 a 100 unidades.

Para mostrar en el display de 0 a 100, el punto decimal se configura en la posición 1 (FFFF.).

Para mostrar en el display de 0 a 100.0, el punto decimal se configura en la posición 2 (FFF.F) y los cálculos de la Escala y el Offset se realizan para un rango de 0 a 1000.

El valor de entrada bajo = mínimo valor de entrada \* número de conversión =  $4(\text{mA}) \times 500 = 2000$ .

El valor de entrada alto = máximo valor de entrada \* número de conversión =  $20(\text{mA}) \times 500 = 10.000$ .

donde: el número de conversión es un coeficiente entre los valores de entrada y el rango real de visualización.

El rango completo del display = 10.000, el número de conversión =  $10.000/20=500$ .

La Tabla 4.2 muestra los números de conversión apropiados.

**Tabla 4-2: Número de Conversión**

<b>Rango de Entrada</b>	<b>Número de Conversión</b>
0-5V	$10.000 / (1.000 \times 5) = 2 \text{ cts/mV}$
0-20mA	$1.000 / (20 \times 1) = 500 \text{ cts/mA}$

Escalando:

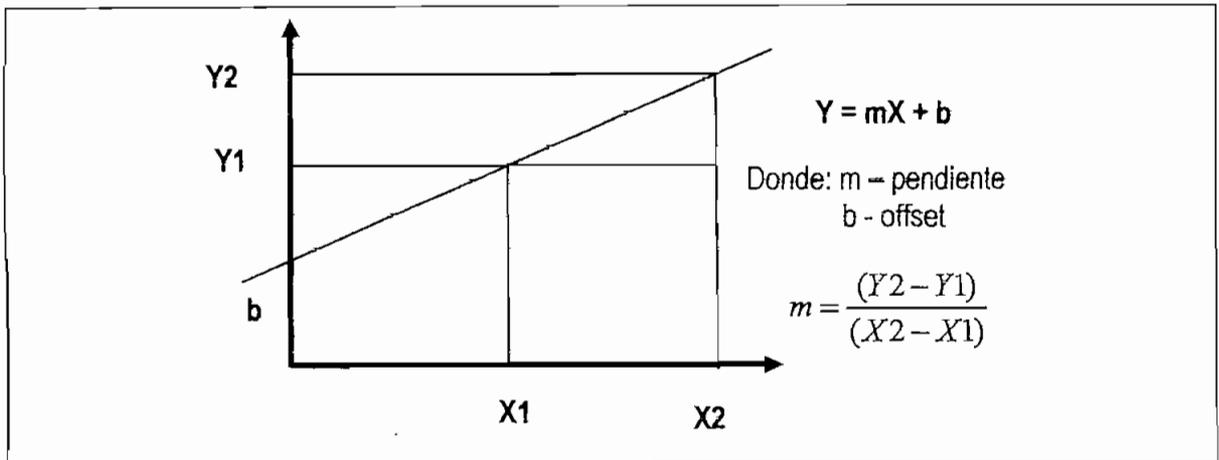


Figura 4-1: Escala y Offset

Para reubicar los 4 – 20 mA en el rango del display desde 0 a 100 se utiliza una pendiente.

$$\text{Pendiente(Escala)} = \frac{RD2 - RD1}{IN2 - IN1} \quad \text{Ecuación (4-2)}$$

donde: RD2 – Lectura Alta del Display(100), RD1 – Lectura Baja del Display(0), IN2 – Entrada Alta (20 x 500), IN1 – Entrada Baja (4 x 500). Estos valores son los que ingresa el usuario en el controlador. Se obtiene un factor de escala:

$$\text{Escala} = (100 - 0) / (10.000 - 2000) = 0.0125 \quad \text{Ecuación (4-3)}$$

### 4.3. PRUEBAS A LA ENTRADA DEL CONTROLADOR

Una vez comprobadas las conversiones de lectura, se continúa con las pruebas pertenecientes a la entrada del controlador. En la entrada de voltaje se utiliza un elemento pasivo (un potenciómetro) al que se le hace variar entre 0 a 5V como elemento transductor. Ayudados por las características de conversión que tiene el controlador para la visualización, se escala para que el rango varíe entre 0,000 y 5,000 unidades.

Los pines  $V_{REFH}$  y  $V_{REFL}$  del conversor analógico digital del microcontrolador determinan la pendiente de conversión. En la tarjeta MINIMAX/908-C existe una pequeña caída de voltaje en el pin  $V_{REFH}$  y no alcanza los 5V de alimentación. Por tal razón, la pendiente es ligeramente más pronunciada.

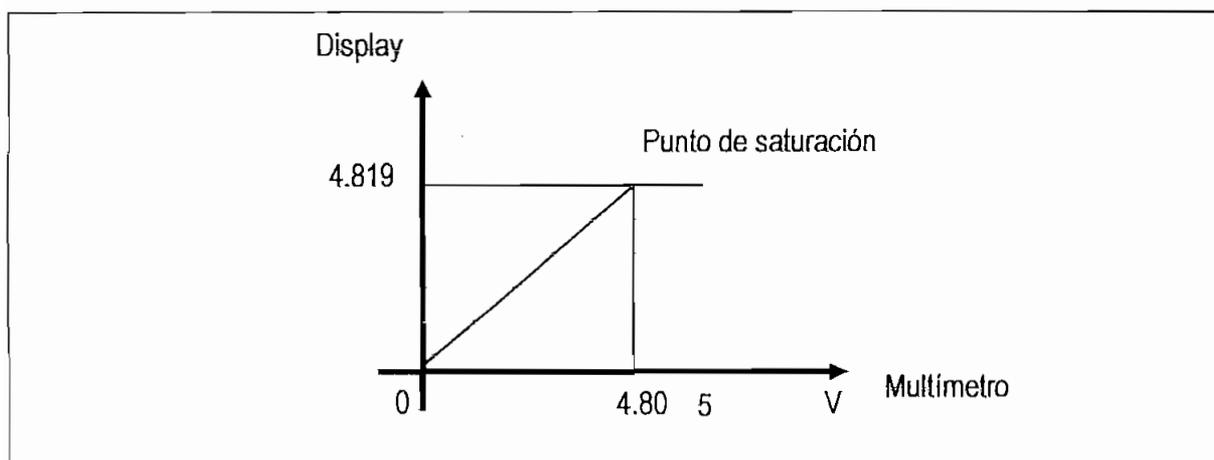
Con el fin de compensar la curva de conversión a través de software, se multiplica la entrada perteneciente al conversor por una constante o coeficiente de compensación. Se determinó un coeficiente de compensación igual a 0,964.

En la Tabla 4-3 se compara la medición que entrega el multímetro digital con respecto al valor del display.

**Tabla 4-3: Comparación del voltaje de entrada al controlador con un multímetro**

	MULTIMETRO DIGITAL	CONTROLADOR
1	0.00	0.000
2	0.13	0.132
3	0.26	0.264
4	0.39	0.396
5	1.06	1.058
6	1.54	1.531
7	2.16	2.154
8	2.43	2.438
9	3.30	3.307
10	4.69	4.706
11	4.80	4.819
12	4.97	4.819

Con el coeficiente de compensación se mantiene la linealidad en todo el rango. Sin embargo, esta compensación satura el valor de entrada a los 4.8 V como puede observarse en la Figura 4-2.



**Figura 4-2: Punto de Saturación para la Entrada al Controlador**

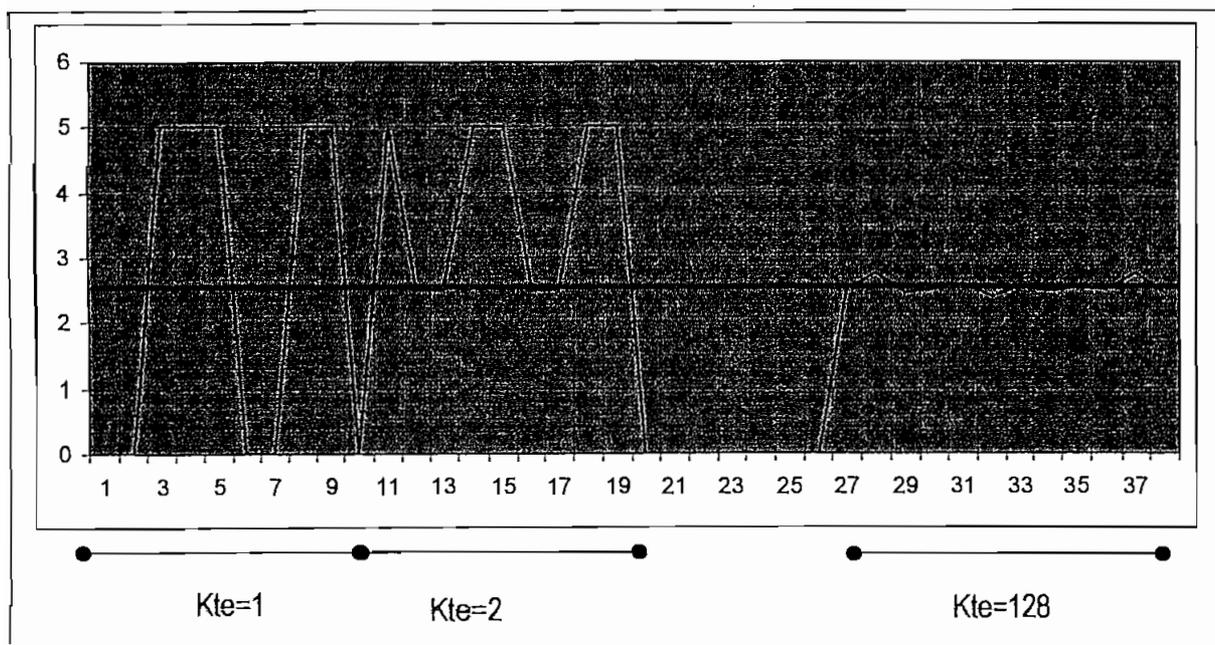
A través de software se implementa un filtro digital a la entrada de la variable medida con la finalidad de evitar la medición de valores erróneos provocados por interferencias.

Este filtro es configurable y toma el número de valores de entrada dependiendo de la constante de filtro utilizada. Las constantes de filtro válidas son: 1, 2, 4, 8, 16, 32, 64 y 128.

Un método de verificar el comportamiento del filtro, es simular una señal ruidosa que ingrese al filtro digital y visualizar el resultado en el display.

Se tomó una señal modulada de ancho de pulso (PWM) proveniente del mismo controlador a una frecuencia de 120 Hz con una relación de trabajo del 50% como entrada de la variable medida y se varió el valor de la constante de filtro.

Cuando se utiliza un filtro de constante 1, recibe valores de 0 o 5V dependiendo en qué estado esté la señal PWM. Al incrementar la constante a 2, se tienen valores de 0, 2.5 y 5V. En un filtro de constante 4 se pueden tener valores de 0, 1.25, 2.5, 3.75 y 5V repitiéndose con más frecuencias los valores intermedios. Cuando toma 128 valores de la señal PWM generada, la mayoría de los valores promediados bordean el valor de 2.5V alejándose de los extremos y se tiene una señal con un rizado aceptable como se aprecia en la Figura 4-3.



**Figura 4-3: Efecto de la Constante de Filtro en la Entrada del Controlador**

Para comprobar la entrada de corriente, se utilizó un transmisor de temperatura de 4 a 20 mA en un rango de 0 a 255 °C.

RD2 = 250; RD1 = 0; IN2=10.000; IN1=2.000

Escala =  $(250 - 0) / (10.000 - 2000) = 0,03125$

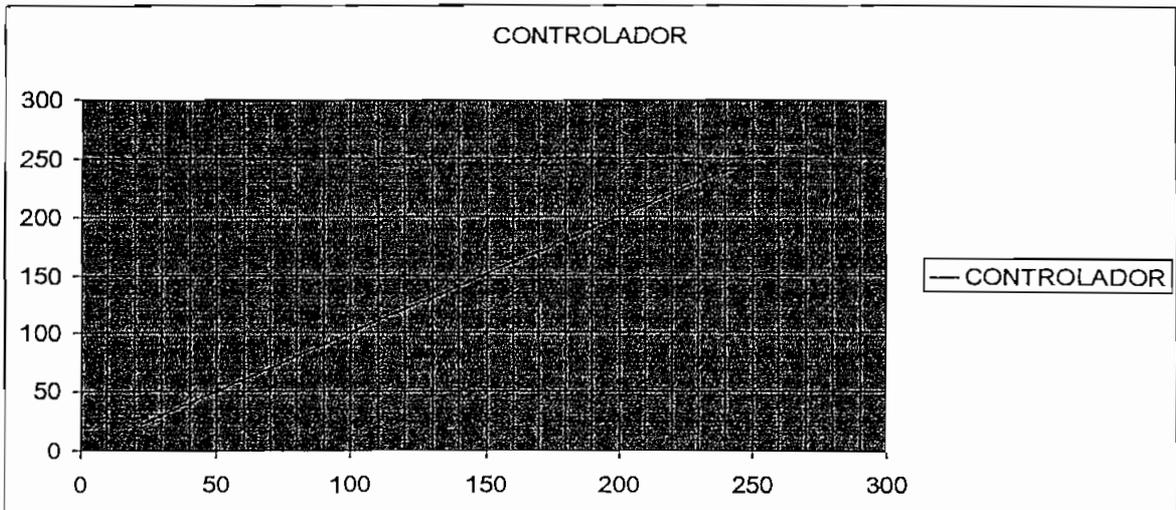
En la Tabla 4-4 se detalla la prueba realizada con el transmisor de temperatura.

	TRANSMISOR DE TEMPERATURA	CONTROLADOR
1	22.50	22.5
2	30.75	30.8
3	40.25	40.2
4	50.75	49.7
5	60.1	60.3
6	70.00	69.6
7	78.75	79.2
8	91.25	91.0
9	102.75	102.8

10	119.25	119.4
11	131.25	131.2
12	144.25	144.2
13	151.25	151.3
14	167.75	167.8
15	176.25	176.1
16	182.50	182.5
17	191.25	191.4
18	208.50	208.0
19	240.00	238.7
20	245.00	238.7

**Tabla 4-4: Prueba de la Entrada de Corriente al Controlador**

Como se puede observar en la Figura 4-4, la curva presenta un comportamiento lineal en el rango de temperatura medido y solamente al final de la curva se tiene una pequeña saturación antes de llegar a su límite superior, por las razones que se expusieron al inicio de esta sección.



**Figura 4-4: Medida del Controlador con respecto a la medida del Transmisor**

## 4.4. PRUEBAS DE LA SALIDA DEL CONTROLADOR

### 4.4.1. SALIDA ANALOGICA

Una vez generada la señal PWM por el microcontrolador, se filtra mediante un circuito RC para obtener la salida analógica de voltaje. Este filtro fue diseñado para la frecuencia de 120 Hz que es la generada en la señal PWM.

Existe una caída de tensión por efecto de la resistencia que presenta el filtro y de la impedancia a la que se acopla la salida del controlador. Por tanto, se utiliza una resistencia relativamente pequeña. Sin embargo, no es recomendado tener valores menores a los 250 ohmios, debido a que el microcontrolador soporta como máximo 20 mA por pin y el capacitor descargado en un instante resulta ser un cortocircuito. Se utilizó una resistencia de 270 ohmios y un capacitor de 330uF. Con capacitores mayores se mejora el rizado pero se incrementa el retardo que produce el tiempo de carga y descarga del mismo y por tanto se tiene una respuesta más lenta a los cambios en un sistema de control.

En la prueba se configura la salida en modo manual y se compara el valor deseado en porcentaje que presenta el display con el valor medido por el multímetro como muestra en la Tabla 4-5

**Tabla 4-5: Prueba de la Salida de Voltaje del Controlador**

Salida [%]	Multímetro [V]
10	0.49
20	0.99
30	1.49
40	1.98
50	2.48
60	2.98
70	3.48
80	3.97
90	4.47
100	4.97

Como se puede observar en el gráfico de la Figura 4-5, se tiene una linealidad en todo el rango de la salida y se debe principalmente a que los incrementos de la señal PWM que gobierna a la salida analógica son pequeños ( $2^{16}$  incrementos de 0 a 5 V) por tratarse de un registro de 16 bits.

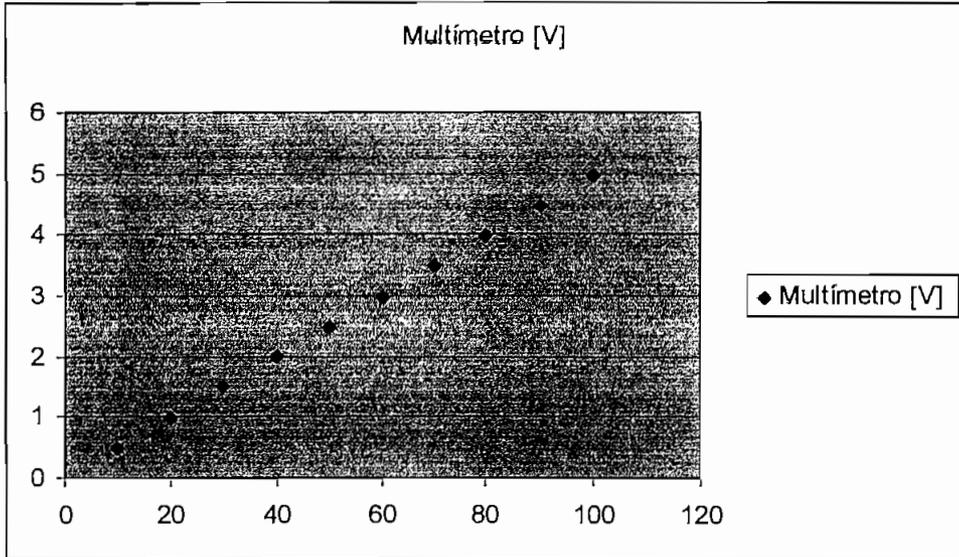


Figura 4-5: Curva de la Salida de Voltaje del Controlador

#### 4.4.2. SALIDA DIGITAL PARA EL RELE DE ESTADO SÓLIDO

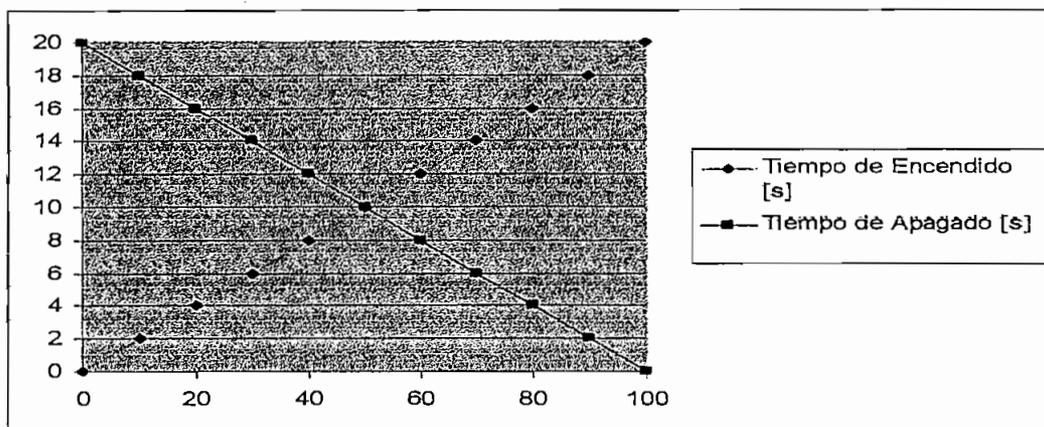
Para tener la salida digital correspondiente al relé de estado sólido, se deshabilita primero en el menú de configuración la salida analógica. De igual manera se configura el período de encendido – apagado que se desea obtener en esta salida digital. Este período de encendido – apagado para la salida digital es un múltiplo del período de muestreo que realiza el controlador (37.5 Hz).

**MODO DE CONFIGURACION:** En el ítem **c SALIDA1** del menú principal se presiona **↻** hasta que aparezca en el display **SALIDA ANALOGICA**. Se presiona la tecla **⏪** y luego **⏩** hasta que aparezca en el display el mensaje **deshabilitada** y se presiona nuevamente la tecla **⏪**. Para cambiar el periodo o ciclo de tiempo proporcional de la salida digital, se avanza con la tecla **⏩** dentro del submenú de la salida 1 hasta que aparezca en el display el mensaje de texto **CICLO DE TIEMPO**. Mediante las teclas **⏩** y **⏪** se incrementa o disminuye

el ciclo del tiempo proporcional comprendido entre 1 a 199 segundos. Para un período de 20 segundos se tiene los siguientes resultados (Tabla 4-6):

**Tabla 4-6: Control en la Salida Digital para el SSR con un período de 20 segundos**

Salida [%]	Tiempo de Encendido [s]	Tiempo de Apagado [s]
0	0	20
10	2	18
20	4	16
30	6	14
40	8	12
50	10	10
60	12	8
70	14	6
80	16	4
90	18	2
100	20	0



**Figura 4-6: Tiempos de Encendido y Apagado de la Salida Digital del Controlador**

Como se puede observar en la Figura 4-6, al aumentar el tiempo de encendido, disminuye el tiempo de apagado y viceversa. Se tiene una linealidad en todo el rango de la salida para los tiempos de encendido y apagado debido a que son factores de la señal generada PWM de la salida analógica.

## 4.5. PRUEBAS DE CONTROL EN SISTEMAS DE LAZO CERRADO

### 4.5.1. PRUEBAS EN MODULOS DIDACTICOS DEL LABORATORIO

Los módulos didácticos son proyectos diseñados como modelos de control para procesos. Se ha utilizado dos procesos: control de flujo y control de presión.

#### 4.5.1.1. Módulo Didáctico De Control De Flujo

El módulo didáctico de control de flujo tiene un transductor cuya salida es de 0 a 5V que es entrada del controlador; y, como variable controladora utiliza un variador de velocidad que requiere de voltaje para su manipulación. Se utiliza únicamente un control proporcional, debido a que no necesita ajustes por medio de una acción integral y no se puede utilizar el término derivativo en este tipo de procesos de acuerdo a la Tabla 4-7.

**Tabla 4-7: Parámetros Iniciales para Lazos de Control Comunes para Algunos Controladores**

TIPO DE LAZO	PB %	INTEGRAL Min/rep	INTEGRAL rep/min	DERIVATIVO min
Flujo	50 a 500	.005 a .05	20 a 200	No
Presión de Líquido	50 a 500	.005 a .05	20 a 200	No
Presión de gas	1 a 50	1 a 50	.02 a 10	.02 a 1
Nivel	1 a 50	1 a 100	.01 a 1	.01 a .05
Temperatura	2 a 100	2 a 50	.02 a 5	1 a 20

Fuente: [www.expertune.com](http://www.expertune.com), 2002

Estos parámetros se asumen como adecuados para diseños de lazos de control con algoritmos ideales o seriales pero no son aplicables para todos los controladores.

La Banda Proporcional aplicada en el controlador para el módulo didáctico de control de flujo se ajustó al 90% de la escala completa, cumpliendo con los rangos que se indican en la Tabla 4-7. Para valores menores en la banda proporcional se

vuelve inestable el control y perjudicial para el funcionamiento del variador de velocidad.

Este control tiene una respuesta rápida a variaciones en la entrada, y por tal razón la velocidad de muestreo del controlador debe ser lo suficientemente rápida para poder controlar al sistema.

Se realizó las pruebas a distintas velocidades de muestreo y se obtuvo los siguientes resultados (Tabla 4-8).

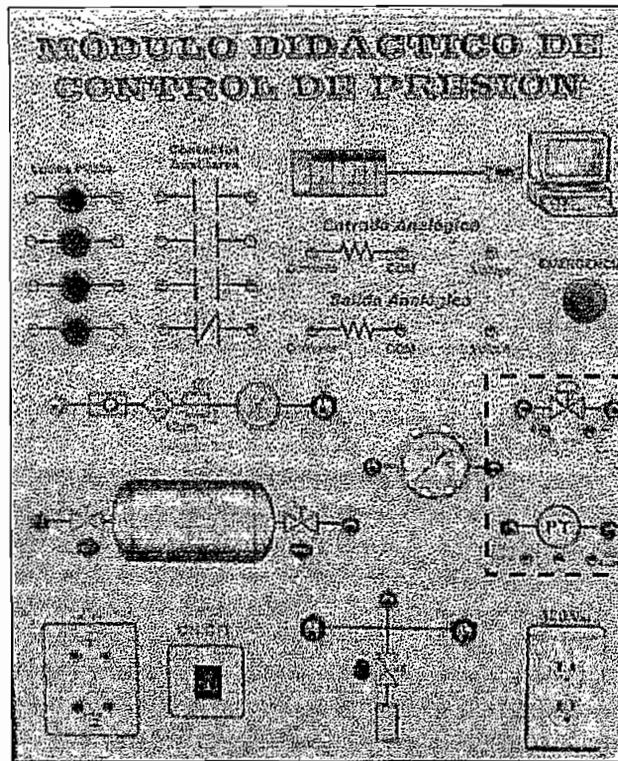
**Tabla 4-8: Velocidad de Muestreo para el Control en el Módulo de Flujo**

Frecuencia de Muestreo (Hz)	Control
3	No controla
4	No controla
10	bueno
37,5	muy bueno
> 50	óptimo

A partir de los 10 Hz se tiene un control en el módulo didáctico de flujo y se mejora conforme aumenta la velocidad de muestreo. El control se limita a una velocidad de muestreo de 37,5 Hz debido a que los cálculos para el algoritmo PID no pueden realizarse a una velocidad mayor. Con el propósito de verificar si necesita el control velocidades de muestreo mayores se aplicó frecuencias superiores a los 50 Hz.

#### **4.5.1.2. Módulo Didáctico De Control De Presión**

En la Figura 4-7, se puede observar el panel frontal del módulo didáctico de Control de Presión. En el recuadro entre líneas punteadas se encuentran tanto la servo-válvula para el control de la planta como el transductor de presión.



**Figura 4-7: Panel Frontal del Módulo Didáctico de Control de Presión**

Las variaciones de la presión en el módulo didáctico son lentas, por lo tanto el margen en el cual se pueden hacer cambios del punto de referencia es pequeño. El sistema se ayuda de una electro-válvula de escape para bajar la presión. El control que se puede utilizar en la electro-válvula es de tipo on-off.

Con un control únicamente proporcional en la servo-válvula, se trabaja con bandas proporcionales pequeñas del 3 al 4%. Si se incrementa el valor de la banda, el error de posición por efecto de esta acción se incrementa. Si por el contrario, se utilizan valores inferiores al 3% en la banda, el error disminuye, pero es más sensible y tiende a ser inestable.

En la Figura 4-8 se puede apreciar la respuesta del sistema a un cambio en el valor de referencia con un control únicamente proporcional (BP=4%).

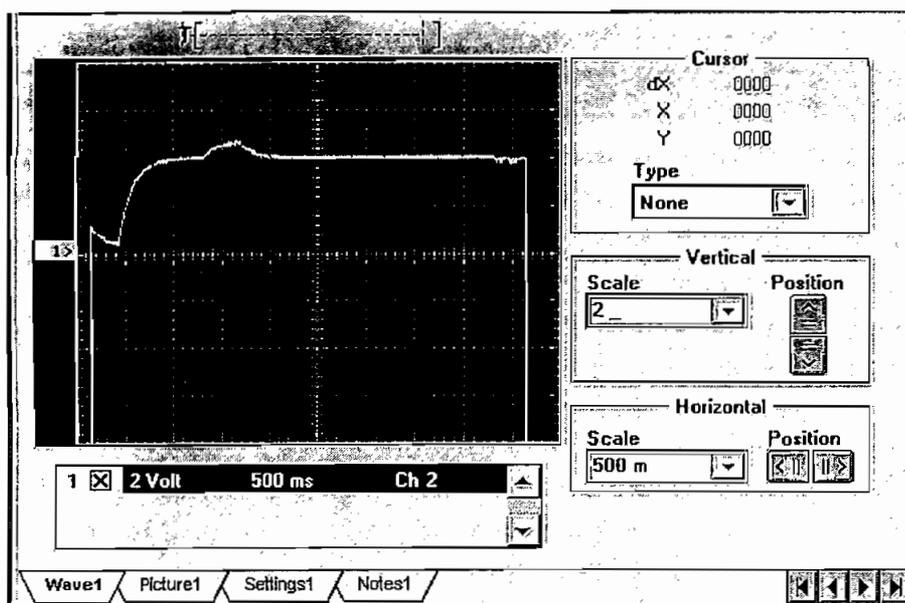


Figura 4-8: Control Proporcional con BP = 4%

La Figura 4-9 representa la respuesta que tiene el sistema a un cambio en el valor de referencia con control proporcional pero con una BP=3%.

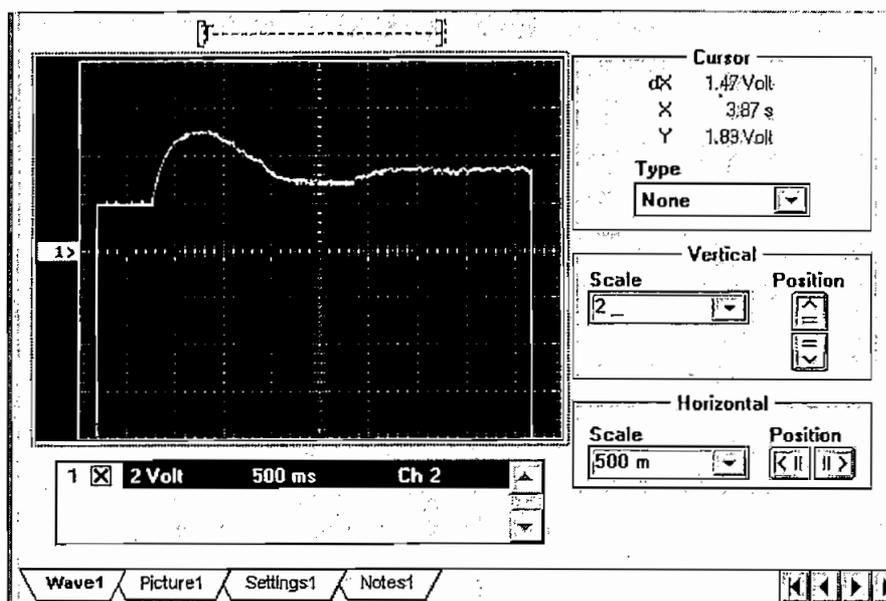


Figura 4-9: Control Proporcional con BP = 3%

Utilizando la acción derivativa como se indica en la Figura 4-10, se gana velocidad en la respuesta del sistema pero se vuelve más sensible a interferencias, es por esto que la acción derivativa no se usa en este tipo de procesos.

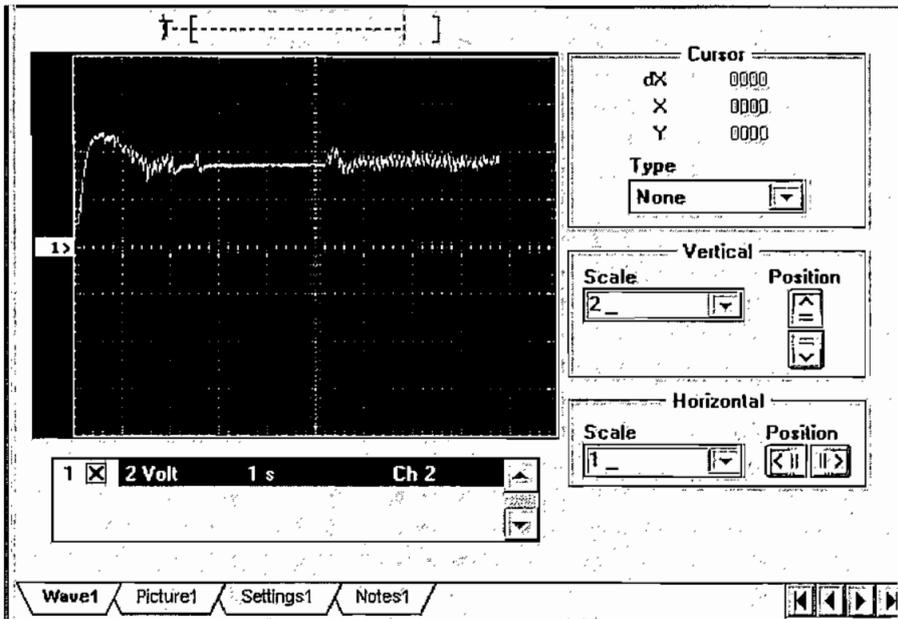
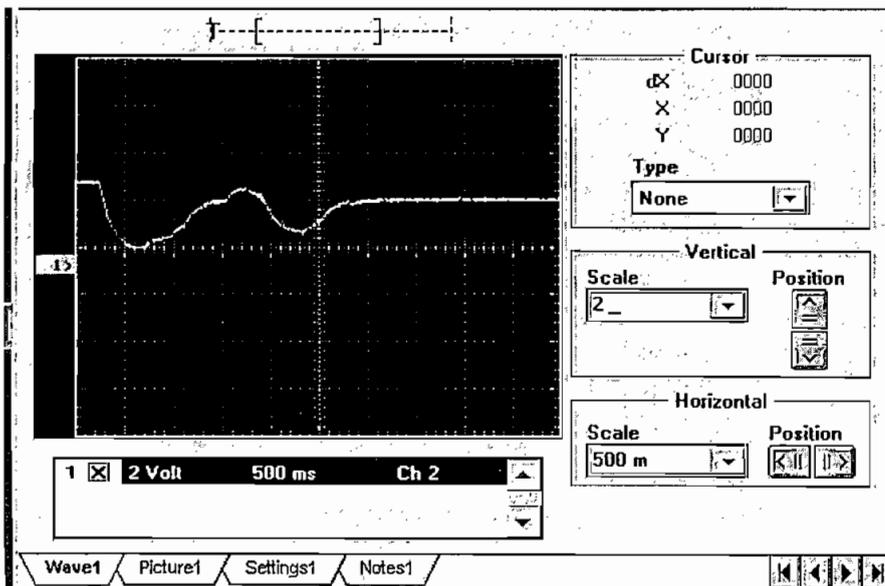


Figura 4-10: Control Proporcional - Derivativo

Una acción integral reduce los offsets ocasionados por la acción proporcional. En la Figura 4-11 se utiliza una acción proporcional e integral y se varía de un valor de referencia a uno de menor valor.



VP	SP	Error
55.6	55.8	0.2
54.4	54	0.4
59.1	59.4	0.3

Figura 4-11: Respuesta del Sistema con control PI a un cambio en el valor de referencia

Utilizando un control PID en el módulo didáctico de presión se obtiene la siguiente respuesta del sistema (Figura 4-12).

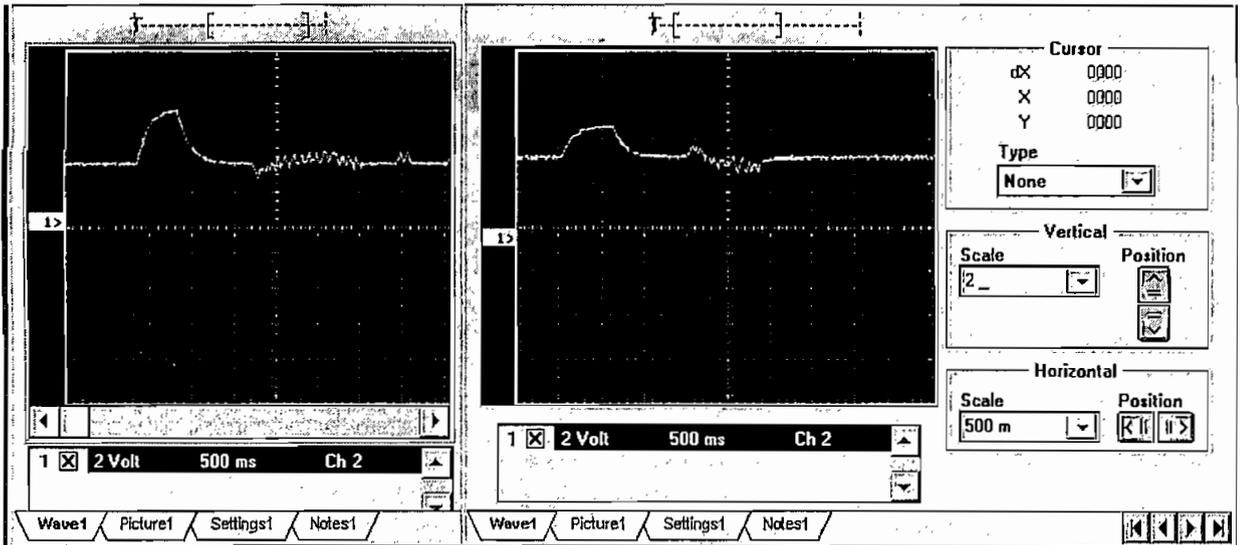


Figura 4-12: Control PID en el Módulo Didáctico de Control de Presión

La sintonización del control PID para el sistema se puede mejorar variando los 3 parámetros.

Si se quiere tener un mayor rango de trabajo, se puede incluir la segunda salida de relé del controlador en el sistema para el manejo de la electro-válvula con control on-off e ir variando los dos puntos de referencia simultáneamente.

#### 4.5.2. SIMULACION DE PLANTAS A TRAVES DEL COMPUTADOR

Se ha desarrollado un programa en LabView que permite simular procesos que puedan ser controlados por el controlador diseñado mediante una tarjeta de adquisición de datos. Los algoritmos se implementaron mediante sistemas discretos. Para demostrar el funcionamiento del controlador se ha utilizado una planta que tenga la función de transferencia

$$G_p = \frac{1}{s(s+1)} \quad \text{Ecuación (4-4)}$$

y el período de muestreo de 1 segundo. Entonces la función de transferencia en el plano Z es:

$$G(z) = \frac{0.3679z^{-1} + 0.2642z^{-2}}{1 - 1.3679z^{-1} + 0.3679z^{-2}} \quad \text{Ecuación (4-5)}$$

El programa permite ingresar los valores de la función de transferencia de la planta en el dominio z. La Figura 4-13 muestra el diagrama de diseño de la planta en LabView.

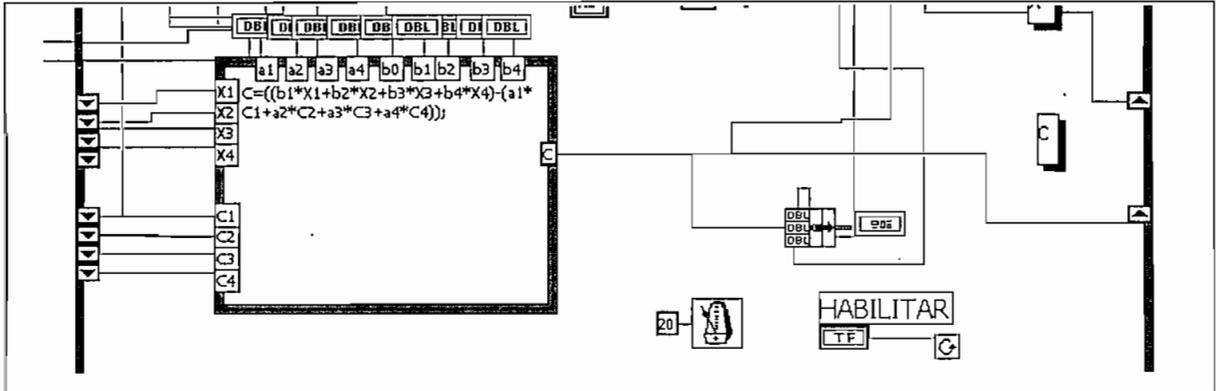


Figura 4-13: Diagrama de diseño de la planta en LabView

En la Figura 4-14 se aprecia las respuestas en el tiempo de la variable controlada y el error en un control proporcional de la planta simulada. Para este control se requiere de una banda mayor al 100% de la escala completa, es decir un  $K_p >= 1$ .

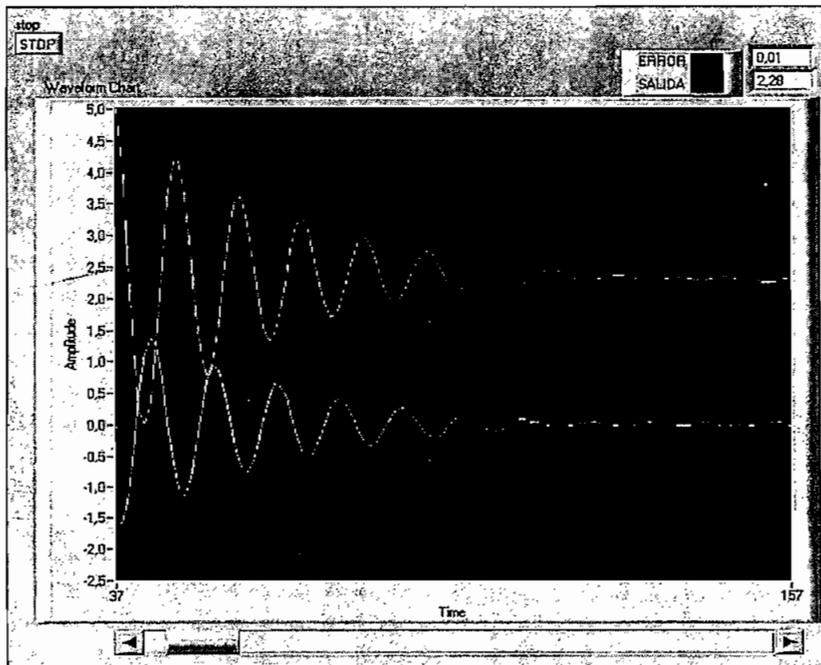


Figura 4-14: Respuestas en el Tiempo del Sistema con Control Proporcional

A continuación, en la Figura 4-15, se puede observar la respuesta del sistema al incluir en el control la acción derivativa. La constante de la acción derivativa está alrededor de los 0.4s

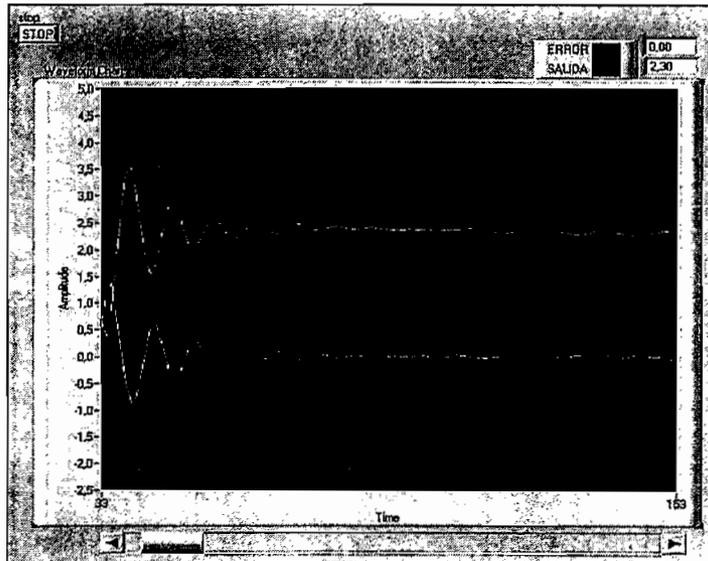


Figura 4-15: Respuestas en el Tiempo del Sistema con Control PD

Finalmente se incluyen los tres términos del control PID. Las Figuras 4-16 y 4-17 muestran las respuestas en el tiempo que tiene el sistema manteniendo el valor proporcional y derivativo constante y variando la acción integral de 0.1 segundos por repetición a 1 segundo por repetición.

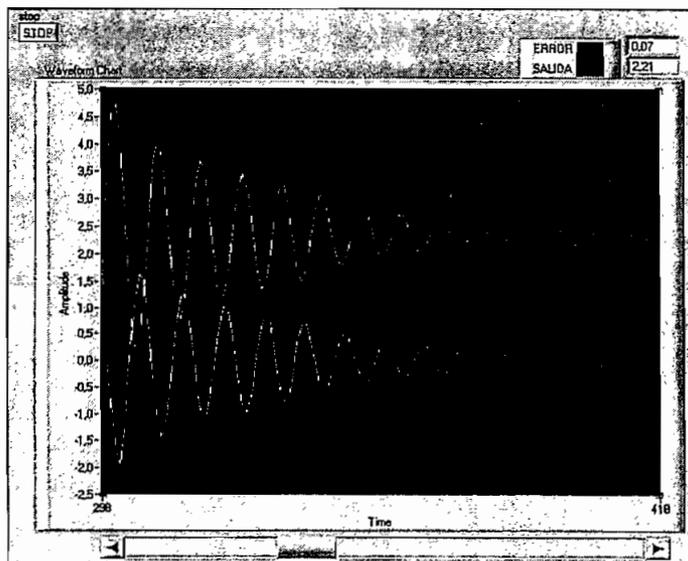


Figura 4-16: Respuestas en el Tiempo del Sistema con Control PID (BP=160%,I=0.1,D=0.4)

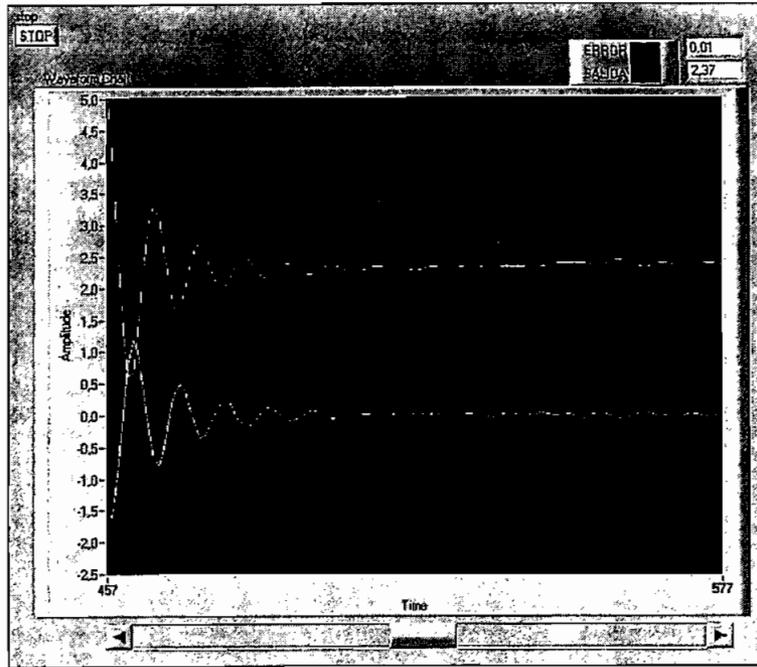


Figura 4-17: Respuestas en el Tiempo del Sistema con Control PID (BP=160%,I=1,D=0.4)

#### 4.6. PRUEBAS DE COMUNICACIÓN SERIAL CON EL COMPUTADOR

Para las pruebas de comunicación serial con el computador se utilizó un terminal que permite visualizar tanto el ingreso como la recepción de los caracteres que corresponden al protocolo generado. En el terminal se define la velocidad y el formato de envío de los caracteres.

En la Figura 4-18 se puede observar la configuración que debe tener el terminal para una velocidad de 19200 baudios, no paridad, 8 bits de datos y 1 bit de parada

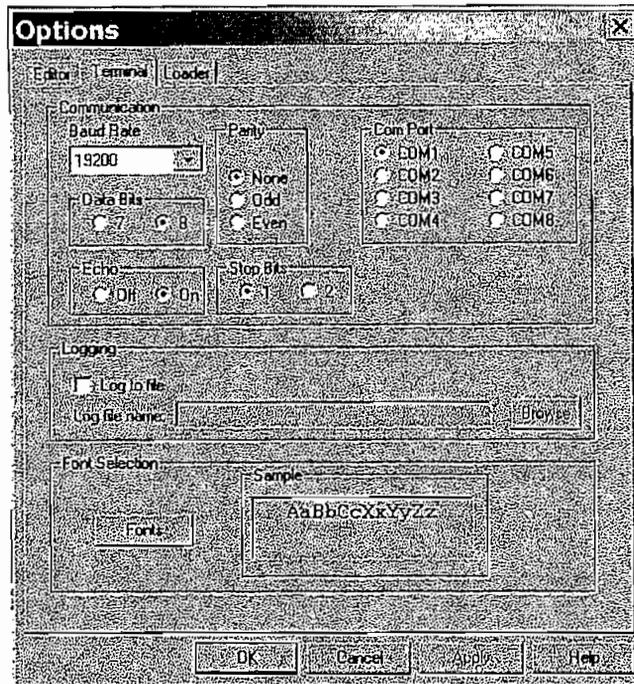


Figura 4-18: Parámetros de Comunicación del Terminal

En la Figura 4-19 se muestra la ventana que presenta el terminal para la comunicación del computador con el controlador.

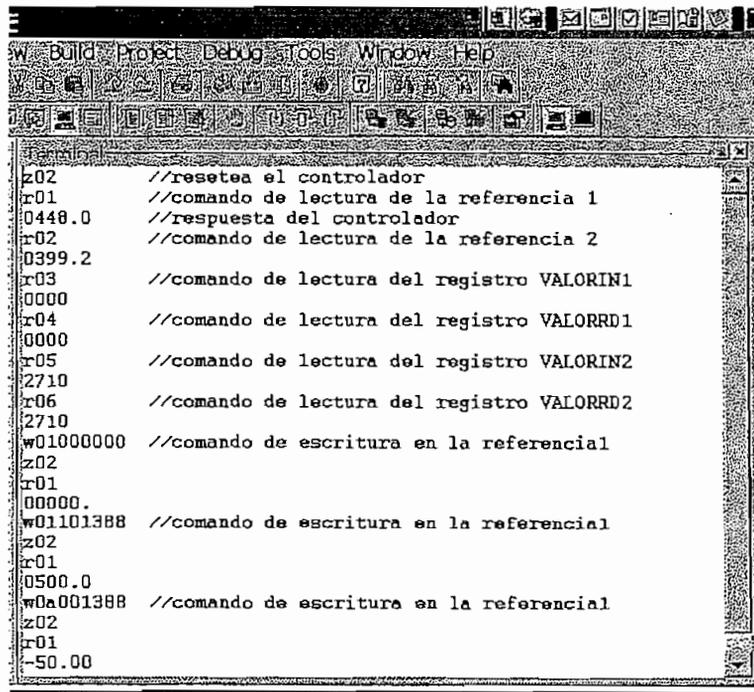


Figura 4-19: Comunicación Serial entre una PC y el controlador por medio de un Terminal



## **CAPITULO 5. CONCLUSIONES Y RECOMENDACIONES**

## CAPITULO 5.

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1. CONCLUSIONES

- Con el uso del microcontrolador MC68HC08 de Motorola se ha logrado cumplir con el propósito del proyecto de titulación que era la implementación de un controlador universal configurable mediante teclado con una interfaz amigable con el usuario y comunicación serial con un computador personal.
- Es importante recalcar que el controlador no fue diseñado para ninguna aplicación en particular o planta específica. En la práctica, mediante la manipulación de los tres parámetros de control: proporcional, integral y derivativo, se han logrado los ajustes necesarios para el control de las plantas con las que se ha deseado trabajar y no ha requerido de acomodados empíricos para un proceso singular. Es decir, el algoritmo utilizado cumple con la teoría de control.
- Los controladores comerciales incluyen normalmente en su arquitectura relés electromecánicos hasta 2 A, relés de estado sólido hasta 0,5 A o transistores para el manejo de relés de estado sólido mayores. El controlador diseñado incluye relés de estado sólido de 3 A que podrían combinar las características de los controladores comerciales, no requieren de disipadores y poseen led's con lo que se puede comprobar su funcionamiento.
- La utilización de la tarjeta MINIMAX/908-C permitió reducir en tiempo, en espacio y en costo para el desarrollo del hardware: Además los elementos adicionales han sido fácilmente adquiridos en el mercado local.
- Las pequeñas dimensiones que posee el aparato se aproximan a las dimensiones usuales de las versiones comerciales de características

semejantes, sin que signifique el máximo provecho en espacio, dado que se ha utilizado un receptáculo de dimensiones prediseñadas.

- Las características del microcontrolador, como son: ser un procesador de 8 bits, tener una limitada memoria RAM, etc. no están orientadas a operaciones en punto flotantes. Sin embargo, la programación en lenguaje C acompañada del manejo adecuado de las variables y de la velocidad de procesamiento del microcontrolador han permitido que el controlador trabaje con punto flotante.
- A diferencia de los controladores comerciales que muestrean la señal de entrada entre 3 a 5 veces por segundo, este controlador está diseñado para un muestreo 10 veces mayor, aparentemente innecesario para procesos de temperatura, pero no así para procesos rápidos, como por ejemplo, aquellos gobernados por un variador de velocidad y que gracias a esta característica ha sido posible controlarlos.
- La frecuencia de muestreo utilizada y las rutinas de punto flotantes permiten al microcontrolador únicamente trabajar con uno de los cuatro canales destinados para salidas PWM. Es por esta razón que se tiene una sola salida analógica de control.
- El controlador permite definir el rango de trabajo de la salida, con esta característica se evita daños en los servomotores o instrumentos de control que estén acoplados a la salida; se puede determinar también con esta característica el punto de operación en estado estable que se requiera para un proceso en particular.
- Al integrar una memoria EEPROM en la arquitectura del controlador, se logra el almacenamiento de los parámetros de configuración sin que requiera ninguna clase de pila que mantenga por un cierto tiempo esta información.
- Con la comunicación serial se puede lograr un mejor desempeño del controlador diseñado para procesos que siguen curvas en distintos puntos

de trabajo. Además se pueden generar en el computador archivos que almacenen toda la información correspondiente al comportamiento de una planta o proceso y de esta manera tomar acciones en base a los resultados que presente.

## 5.2. RECOMENDACIONES

- El uso de la tarjeta MINIMAX/908-C, si bien facilitó la implementación del hardware, exigió un software complejo y denso con un buen número de subrutinas y de instrucciones. Desafortunadamente, no se pudo utilizar todas las herramientas de desarrollo disponibles en el programa CodeWarrior por razones económicas, cuyas ventajas están encaminadas hacia una programación a nivel gráfico y no literal. Para una producción en escala se recomendaría la compra de las licencias.
- En el caso que se requiera tener características de microcontrolador, y no se desee utilizar DSP's pero que impliquen operaciones en sus algoritmos a mayor velocidad, se recomienda la utilización de microcontroladores, preferiblemente Motorola, pero a 16 bits.
- Se puede reemplazar el display alfanumérico por display matriciales que permitan una mejor navegación a través del menú en la configuración facilitando al usuario la programación, así como también se podría realizar histogramas de las variables medidas.
- Si se desea aprovechar las entradas adicionales que tiene el microcontrolador para el conversor analógico – digital, se puede trabajar directamente con entradas de termocupla, RTD's, etc.
- La salida analógica de voltaje tiene la posibilidad de convertirse en salida de corriente a través de un conversor de 1-5 V a 4-20mA.. También se puede ampliar las características de la salida analógica para que sea utilizada como retransmisión de la variable de proceso.

- Si se reemplazan los SSR actualmente utilizados en el controlador por Relés comunes, la salida de control no debe ser seleccionada para ciclos de tiempo menores a 7 segundos debido a que el tiempo de vida útil de los relés se acortaría.
- La comunicación serial que tiene el controlador con un computador no está destinada a un software específico, lo que permite integrarlo a un sistema mayor para su control. Además, si se quiere tener una red de controladores, se debe utilizar la comunicación RS485, en donde cada controlador es designado con una dirección y se utiliza el protocolo FIELDBUS para la comunicación entre éstos. Existe también una tendencia hacia el control por medio del Internet, donde al controlador se le asignaría una dirección IP.
- Otra característica que se le puede dar al controlador es la generación de rampas que lleven al proceso al valor de referencia en un tiempo determinado o también determinar un tiempo para el funcionamiento en un valor seteado.

## **REFERENCIAS BIBLIOGRAFICAS**

### **MANUALES**

MOTOROLA SEMICONDUCTORS, MC68HC908GP32 Technical Data, 5ta Edición, DigitalDNA & Motorola, Denver Colorado USA, Julio 2001.

BIPOM Electronics, MINI-MAX7908-C Single Borrada Computer Technical Manual, Revision 1.11, BIPOM Electronics, Houston Texas USA, Marzo 17 del 2001.

METROWERKS, Motorola HC08 Compiler, Revision 1.1, Metroweks, Austin Texas USA, Julio 2002.

NEWPORT Electronics, Temperature & Process Controller Manual, Santa Ana CA USA, Julio 2002.

NEWPORT Electronics, Monitor/Controller Communication Manual, Santa Ana CA USA, Julio 2002.

### **DIRECCIONES ELECTRONICAS**

[www.motorola.com/semiconductors](http://www.motorola.com/semiconductors)

[www.bipom.com](http://www.bipom.com)

[www.metrowerks.com](http://www.metrowerks.com)

[www.newportUS.com/i](http://www.newportUS.com/i)

[www.expertune.com](http://www.expertune.com)

## **ANEXOS**

## Appendix A. MC68HC08GP32

### A.1 Contents

A.2	Introduction .....	398
A.3	MCU Block Diagram .....	398
A.4	Memory Map .....	400
A.5	Mask Option Registers .....	401
A.6	Reserved Registers .....	402
A.7	Monitor ROM .....	402
A.8	Electrical Specifications .....	403
A.8.1	Functional Operating Range .....	403
A.8.2	5.0-V DC Electrical Characteristics .....	403
A.8.3	3.0-V DC Electrical Characteristics .....	404
A.8.4	Memory Characteristics .....	405
A.9	ROM MC Order Numbers .....	406

## A.2 Introduction

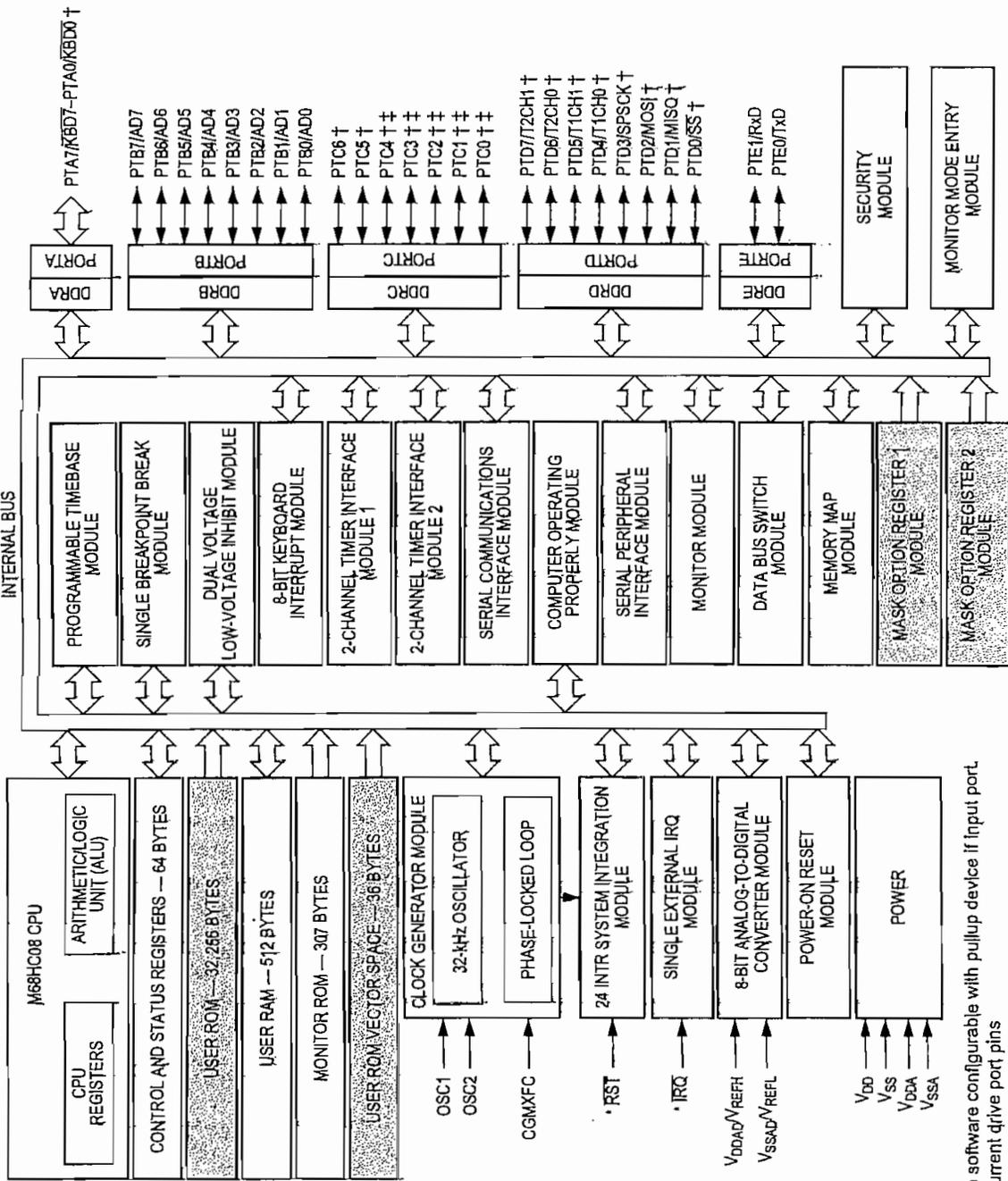
This section introduces the MC68HC08GP32, the ROM part equivalent to the MC68HC908GP32. The entire data book apply to this ROM device, with exceptions outlined in this appendix.

**Table A-1. Summary of MC68HC08GP32 and MC68HC908GP32 differences**

	<b>MC68HC08GP32</b>	<b>MC68HC908GP32</b>
<b>Memory (\$8000–\$FDFF)</b>	32,256 bytes ROM	32,256 bytes FLASH
<b>User vectors (\$FFDC–\$FFFF)</b>	36 bytes ROM	36 bytes FLASH
<b>Registers at \$001E and \$001F</b>	Mask option registers; defined by mask; read only. \$001E — MOR2 \$001F — MOR1	Configuration registers; write once after reset. \$001E — CONFIG2 \$001F — CONFIG1
<b>Registers at \$FE08 and \$FF7E</b>	Not used; locations are reserved	FLASH related registers. \$FE08 — FLCR \$FF7E — FLBPR
<b>Bit 2 at \$FE01</b>	Not used; bit is reserved	MODRST: monitor mode entry by blank reset vector bit.
<b>Monitor ROM (\$FE20–\$FF52)</b>	Used for testing purposes only.	Used for testing and FLASH programming/erasing.
<b>Available Packages</b>	42-pin SDIP 44-pin QFP	40-pin PDIP 42-pin SDIP 44-pin QFP

## A.3 MCU Block Diagram

**Figure A-1** shows the block diagram of the MC68HC08GP32.



† Ports are software configurable with pullup device if input port.  
 ‡ Higher current drive port pins  
 • Pin contains integrated pullup device  
 [Shaded blocks indicate differences to MC68HC908GP32]

Figure A-1. MC68HC908GP32 Block Diagram

## A.4 Memory Map

The MC68HC08GP32 has 32,256 bytes of user ROM from \$8000 to \$FDFF, and 36 bytes of user ROM vectors from \$FE00 to \$FE0B. On the MC68HC908GP32, these memory locations are FLASH memory.

Figure A-2 shows the memory map of the MC68HC08GP32.

\$0000	I/O Registers 64 Bytes
↓	
\$003F	RAM 512 Bytes
↓	
\$023F	
\$0240	Unimplemented 32,192 Bytes
↓	
\$7FFF	ROM 32,256 Bytes
↓	
\$8000	SIM Break Status Register (SBSR)
↓	
\$FDFF	SIM Reset Status Register (SRSR)
\$FE00	
\$FE01	Reserved (SUBAR)
\$FE02	
\$FE03	SIM Break Flag Control Register (SBFCR)
\$FE09	
\$FE0A	Interrupt Status Register 1 (INT1)
\$FE0B	
\$FE0C	Interrupt Status Register 2 (INT2)
\$FE0D	
\$FE0E	Interrupt Status Register 3 (INT3)
\$FE0F	
\$FE10	Reserved
\$FE11	
\$FE12	Reserved
\$FE13	
\$FE14	Break Address Register High (BRKH)
\$FE15	
\$FE16	Break Address Register Low (BRKL)
\$FE17	
\$FE18	Break Status and Control Register (BRKSCR)
\$FE19	

Figure A-2. MC68HC08GP32 Memory Map

\$FE0C	LVI Status Register (LVISR)
\$FE0D	Unimplemented 3 Bytes
↓	
\$FE0F	Unimplemented 16 Bytes Reserved for Compatibility with Monitor Code for A-Family Parts
\$FE10	
↓	
\$FE1F	Monitor ROM 307 Bytes
\$FE20	
↓	Unimplemented 48 Bytes
\$FF52	
\$FF53	Reserved
↓	
\$FF7D	Unimplemented 93 Bytes
\$FF7E	
\$FF7F	ROM Vectors 36 Bytes
↓	
\$FFDB	ROM Vectors 36 Bytes
\$FFDC	
↓	
\$FFFF	

Note: \$FFF6–\$FFFD reserved for 8 security bytes

Figure A-2. MC68HC08GP32 Memory Map (Continued)

## A.5 Mask Option Registers

The two mask option registers at \$001E and \$001F (see **Figure A-3** and **Figure A-4**) are read-only registers. They are defined by mask options (hard-wired connections) specified at the same time as the ROM code submission.

On the MC68HC908GP32, these two registers are called configuration registers (CONFIG2 and CONFIG1).

Address: \$001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	OSC-STOPENB	SCIBD-SRC
Write:								
Reset:	Mask defined							

**Figure A-3. Mask Option Register 2 (MOR2)**

Address: \$001F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	LVI5OR3	SSREC	STOP	COPD
Write:								
Reset:	Mask defined							

**Figure A-4. Mask Option Register 1 (MOR1)**

The bit functions for these two registers are the same as the configuration registers in MC68HC908GP32 (see **Section 8. Configuration Register (CONFIG)**).

## A.6 Reserved Registers

The two registers at \$FE08 and \$FF7E are reserved locations on the MC68HC08GP32.

On the MC68HC908GP32, these two locations are the FLASH control register and the FLASH block protect register respectively.

## A.7 Monitor ROM

The monitor program (monitor ROM, \$FE20–\$FF52) on the MC68HC08GP32 is for device testing only.

The monitor mode entry by blank reset vector bit, MODRST bit (bit 2 at \$FE01), is not used in the ROM device — the reset vector will always contain data in the MC68HC08GP32.

## A.8 Electrical Specifications

Electrical specifications for the MC68HC908GP32 apply to the MC68HC08GP32, except for the parameters indicated below.

### A.8.1 Functional Operating Range

Characteristic	Symbol	Value			Unit
		C	V	M	
Operating temperature range	$T_A$	-40 to +85	-40 to +105	-40 to +125	°C
Operating voltage range	$V_{DD}$	$3V \pm 10\%$	$3V \pm 10\%$	—	V
		$5V \pm 10\%$	$5V \pm 10\%$	$5V \pm 10\%$	

### A.8.2 5.0-V DC Electrical Characteristics

Characteristic <sup>(1)</sup>	Symbol	Min	Typ <sup>(2)</sup>	Max	Unit
$V_{DD}$ supply current					
Run <sup>(3)</sup>		—	15	20	mA
Wait <sup>(4)</sup>		—	4	8	mA
Stop <sup>(5)</sup>					
25 °C	$I_{DD}$	—	2	—	μA
25 °C with TBM enabled <sup>(6)</sup>		—	20	—	μA
25 °C with LVI and TBM enabled <sup>(6)</sup>		—	300	—	μA
-40 °C to 125 °C		—	—	35	μA
-40 °C to 85 °C with TBM enabled <sup>(6)</sup>		—	50	—	μA
-40 °C to 85 °C with LVI and TBM enabled <sup>(6)</sup>		—	500	—	μA
Low-voltage inhibit, trip falling voltage	$V_{TRIPF}$	3.90	4.25	4.50	V
Low-voltage inhibit, trip rising voltage	$V_{TRIPR}$	4.00	4.35	4.60	V
Low-voltage inhibit reset/recover hysteresis ( $V_{TRIPF} + V_{HYS} = V_{TRIPR}$ )	$V_{HYS}$	—	100	—	mV

Notes:

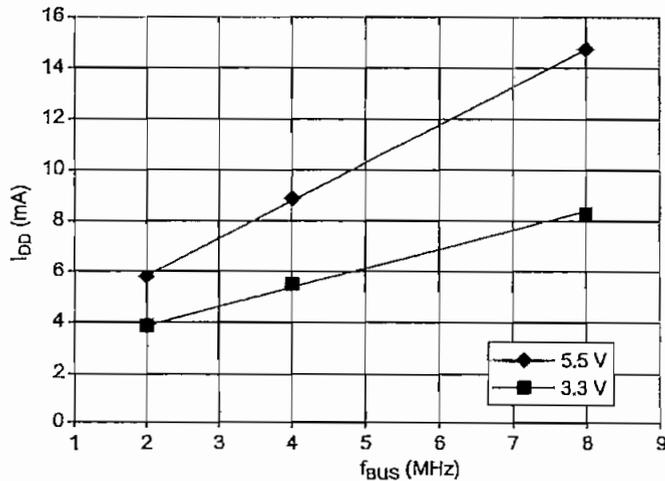
- $V_{DD} = 5.0 \text{ Vdc} \pm 10\%$ ,  $V_{SS} = 0 \text{ Vdc}$ ,  $T_A = T_L$  to  $T_H$ , unless otherwise noted.
- Typical values reflect average measurements at midpoint of voltage range, 25 °C only.
- Run (operating)  $I_{DD}$  measured using external square wave clock source ( $f_{OSC} = 32.8 \text{ MHz}$ ). All inputs 0.2V from rail. No dc loads. Less than 100 pF on all outputs.  $C_L = 20 \text{ pF}$  on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects run  $I_{DD}$ . Measured with all modules enabled.
- Wait  $I_{DD}$  measured using external square wave clock source ( $f_{OSC} = 32.8 \text{ MHz}$ ). All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs.  $C_L = 20 \text{ pF}$  on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects wait  $I_{DD}$ . Measured with PLL and LVI enabled.
- Stop  $I_{DD}$  is measured with  $OSC1 = V_{SS}$ .
- Stop  $I_{DD}$  with TBM enabled is measured using an external square wave clock source ( $f_{OSC} = 32.8 \text{ MHz}$ ). All inputs 0.2V from rail. No dc loads. Less than 100 pF on all outputs. All inputs configured as inputs.

### A.8.3 3.0-V DC Electrical Characteristics

Characteristic <sup>(1)</sup>	Symbol	Min	Typ <sup>(2)</sup>	Max	Unit
$V_{DD}$ supply current					
Run <sup>(3)</sup>		—	5.2	8	mA
Wait <sup>(4)</sup>		—	1.65	4	mA
Stop <sup>(5)</sup>					
25 °C		—	1	—	μA
25 °C with TBM enabled <sup>(6)</sup>	$I_{DD}$	—	12	—	μA
25 °C with TBM enabled <sup>(7)</sup>		—	25	—	μA
25 °C with LVI and TBM enabled <sup>(6)</sup>		—	200	—	μA
-40 °C to 85 °C		—	—	5	μA
-40 °C to 85 °C with TBM enabled <sup>(7)</sup>		—	—	50	μA
-40 °C to 85 °C with LVI and TBM enabled <sup>(6)</sup>		—	300	—	μA
Low-voltage inhibit, trip falling voltage	$V_{TRIPF}$	2.45	2.60	2.70	V
Low-voltage inhibit, trip rising voltage	$V_{TRIPR}$	2.50	2.66	2.80	V
Low-voltage inhibit reset/recover hysteresis ( $V_{TRIPF} + V_{HYS} = V_{TRIPR}$ )	$V_{HYS}$	—	60	—	mV

**Notes:**

- $V_{DD} = 3.0 \text{ Vdc} \pm 10\%$ ,  $V_{SS} = 0 \text{ Vdc}$ ,  $T_A = T_L$  to  $T_H$ , unless otherwise noted
- Typical values reflect average measurements at midpoint of voltage range, 25 °C only.
- Run (operating)  $I_{DD}$  measured using external square wave clock source ( $f_{OSC} = 16.4 \text{ MHz}$ ). All inputs 0.2V from rail. No dc loads. Less than 100 pF on all outputs.  $C_L = 20 \text{ pF}$  on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects run  $I_{DD}$ . Measured with all modules enabled.
- Wait  $I_{DD}$  measured using external square wave clock source ( $f_{OSC} = 16.4 \text{ MHz}$ ). All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs.  $C_L = 20 \text{ pF}$  on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects wait  $I_{DD}$ . Measured with PLL and LVI enabled.
- Stop  $I_{DD}$  is measured with  $OSC1 = V_{SS}$ .
- Stop  $I_{DD}$  with TBM enabled is measured using an external square wave clock source ( $f_{OSC} = 16.4 \text{ MHz}$ ). All inputs 0.2V from rail. No dc loads. Less than 100 pF on all outputs. All inputs configured as inputs.
- Measured with TBM enabled using 32kHz crystal.



**Figure 24-1. Typical Operating  $I_{DD}$**

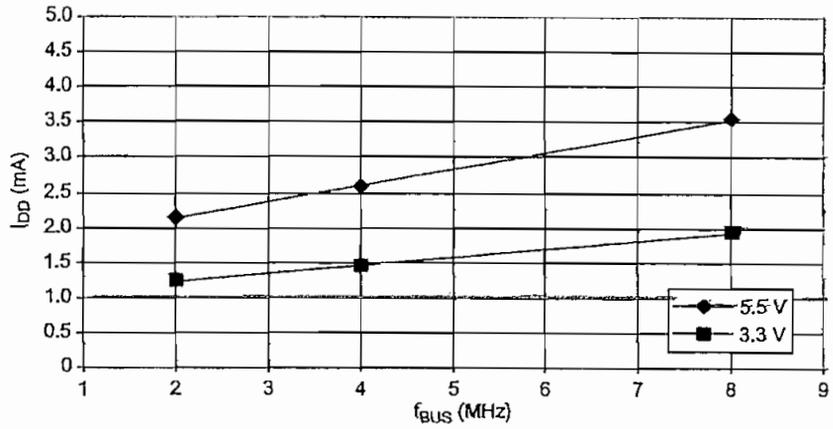


Figure 24-2. Typical Wait Mode  $I_{DD}$

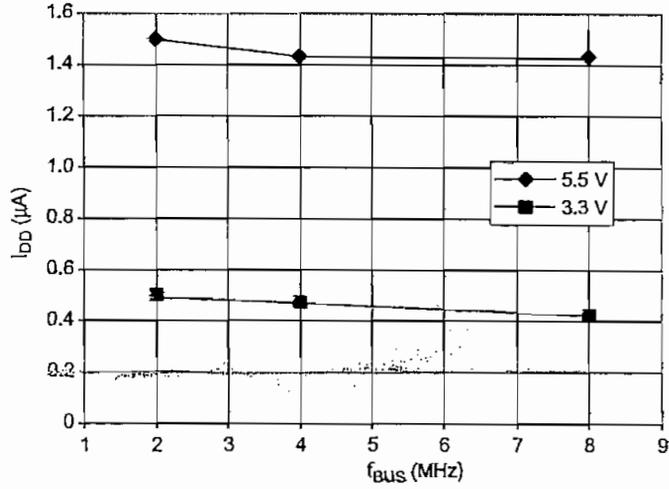


Figure 24-3. Typical Stop Mode  $I_{DD}$

#### A.8.4 Memory Characteristics

Characteristic	Symbol	Min	Max	Unit
RAM data retention voltage	$V_{RDR}$	1.3	—	V

Notes:

Since MC68HC08GP32 is a ROM device, FLASH memory electrical characteristics do not apply.

## A.9 ROM MC Order Numbers

These part numbers are generic numbers only. To place an order, ROM code must be submitted to the ROM Processing Center (RPC).

**Table 24-1. MC Order Numbers**

MC order number	Operating temperature range	Package
MC68HC08GP32CB	-40 °C to +85 °C	42-pin SDIP
MC68HC08GP32VB	-40 °C to +105 °C	
MC68HC08GP32MB <sup>(1)</sup>	-40 °C to +125 °C	
MC68HC08GP32CFB	-40 °C to +85 °C	44-pin QFP
MC68HC08GP32VFB	-40 °C to +105 °C	
MC68HC08GP32MFB <sup>(1)</sup>	-40 °C to +125 °C	

Notes:

1. Temperature grade "M" is available for 5V operating voltage only.



## Reguladores PID

- ▣ Control por relé
- ▣ Controlador de realimentación
- ▣ Acción proporcional, integral y derivativa
- ▣ Reguladores P, PI, PD, PID



## Control por Relé

- ▣ Esquema básico de control en lazo cerrado
- ▣ Descripción
  - $Si \ y \leq y_{ref} \Rightarrow Act = ON$
  - $Si \ y > y_{ref} \Rightarrow Act = OFF$
- ▣ Corresponde con un lazo de realimentación con ganancia infinita de control
- ▣ Evolución del control (Ej.: Secaderos industriales)
  - ✓ Temporización
  - ✓ Control por Relé
  - ✓ Regulación continua
  - ✓ Estructuras complejas de regulación (¿Actual?)



## Controlador de realimentación

### Tareas realizadas

- ✓ Recibir la señal de medida de la variable controlada
- ✓ Comparación con el valor de referencia
- ✓ Actuación sobre la variable manipulada

### Utilidades básicas

- ✓ Selector manual/automático
  - ✓ Sirve para llevar al proceso al punto de operación
- ✓ Selector del tipo de actuación
  - ✓ Directa
    - Salida aumenta  $\Rightarrow$  Control aumenta
  - ✓ Inversa
    - Salida aumenta  $\Rightarrow$  Control disminuye
- ✓ Capacidad de programación (implementar algoritmo de control)

Tema V-II.1 Técnicas de diseño

Control e Instrumentación de P. Q.

Dpto Informática y Automática, USAL 3



## Acción proporcional I

### Definición $m(t) = \bar{m} + K_c e(t)$

✓ Error  $e(t) = y_r(t) - y_m(t)$

✓ Señal de *bías*  $\bar{m} = [m(t) - K_c e(t)]_{t=0}$

- ✓ Al poner el controlador en automático ( $t=0$ ), la señal de control no debe variar bruscamente

### Parámetros

✓ Ganancia  $K_c$

✓ Banda proporcional  $BP = X_p = \frac{100}{K_c}$

- ✓ Cambio en el error para producir un cambio del 100% en la señal de control
- ✓ BP baja supone que un error pequeño provoca una acción de control grande

Tema V-II.1 Técnicas de diseño

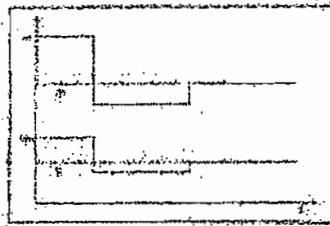
Control e Instrumentación de P. Q.

Dpto Informática y Automática, USAL 4



## Acción proporcional II

- ✓ Actuación directa
  - ✓ Salida aumenta y control aumenta  $\Rightarrow K_c < 0$
- ✓ Actuación inversa
  - ✓ Salida aumenta y control disminuye  $\Rightarrow K_c > 0$



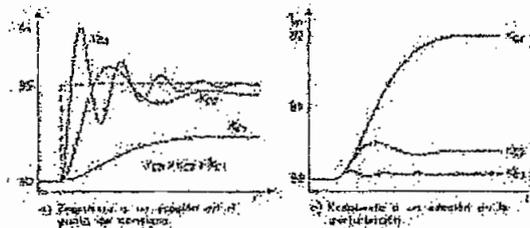
Acción proporcional  
(actuación directa)



## Acción proporcional III

### Efecto

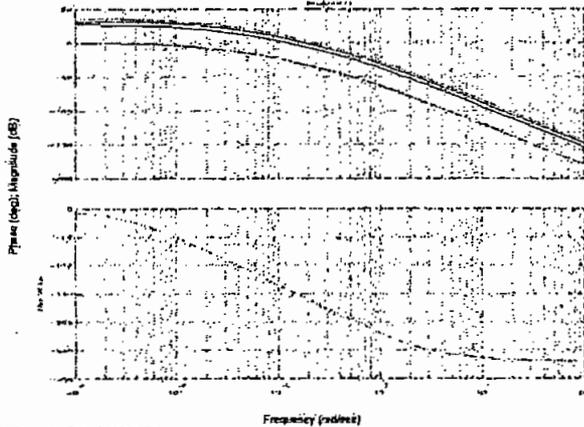
- ✓ Aumento de la ganancia
  - ✓ disminuye el error estacionario y aumentan las oscilaciones





## Acción proporcional IV

### Respuesta en frecuencia



$$G(s) = K_c \frac{1}{(1+s)(1+10s)(1+100s)}$$

$$K_c = 1,204060$$

Tema V-II.1 Técnicas  
de diseño

Control e Instrumentación de P. Q.

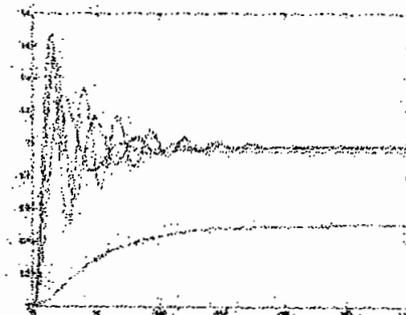
Dpto Informática y Automática. USAL 7



## Acción proporcional (y V)

### Respuesta en frecuencia

- ✓ La ganancia proporcional
  - ✓ Modifica la curva de ganancias desplazándola verticalmente
  - ✓ No altera la curva de fases
- ✓ Aumento de la ganancia
  - ✓ Disminuye el margen de fase
    - Aumenta la rapidez de la respuesta



Tema V-II.1 Técnicas  
de diseño

Control e Instrumentación de P. Q.

Dpto Informática y Automática. USAL 8



## Acción integral I

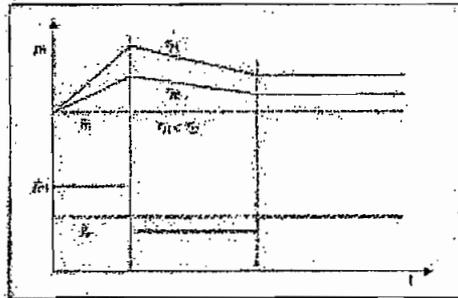
Definición  $m(t) = \bar{m} + \frac{1}{\tau_i} \int e(t) dt$

Parámetro  
✓ Tiempo integral  $\tau_i$

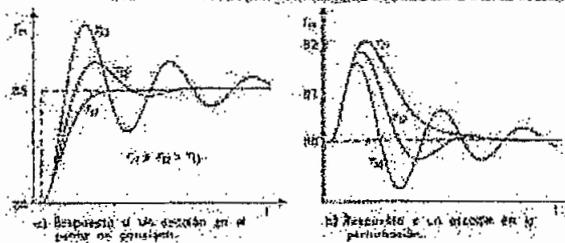
Propiedades  
✓ Cambia la salida del controlador en la dirección correcta

$$\frac{dm(t)}{dt} = \frac{1}{\tau_i} e(t)$$

✓ Elimina error estacionario



## Acción integral II



✓ Va acompañado de acción proporcional (PI)

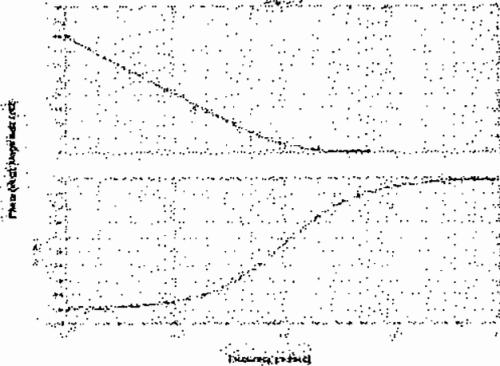
✓ Evitar respuestas lentas

$$m(t) = \bar{m} + K_c \left( e(t) + \frac{1}{\tau_i} \int e(t) dt \right) \quad G_c(s) = K_c \left( \frac{1 + \tau_i s}{\tau_i s} \right)$$



## Acción integral III

### Respuesta en frecuencia



- ✓ Al disminuir el tiempo Integral
  - ✓ Aumenta el efecto integral
  - ✓ Se desplazan a la derecha las curvas de ganancia y fase
  - ✓ Se aporta retardo de fase a frecuencias altas
  - ✓ Disminuye el margen de ganancia y fase
    - La respuesta se hace más oscilatoria



## Acción derivativa I

### Definición

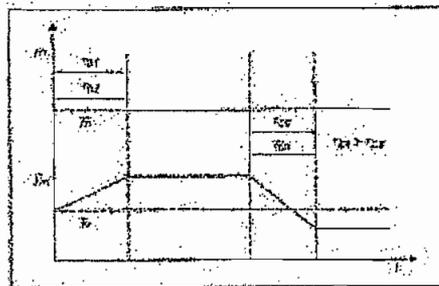
$$m(t) = \bar{m} + \tau_d \frac{de}{dt}(t)$$

### Parámetro

- ✓ Tiempo derivativo  $\tau_d$

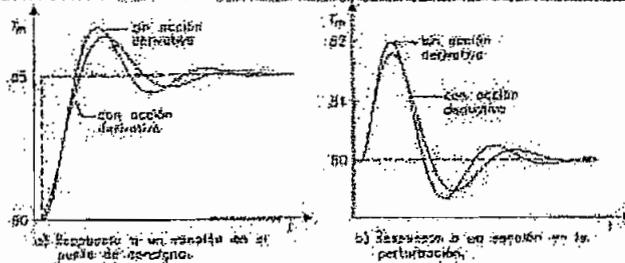
### Propiedades

- ✓ Carácter anticipativo
  - ✓ Aporta más a la señal de control cuanto mayor sea la velocidad de cambio del error
  - ✓ Aporta una fuerte acción correctora con errores pequeños





## Acción derivativa (y II)



- ✓ Estabiliza la respuesta
  - ✓ Amortigua las oscilaciones y permite elevar la ganancia
- ✓ No debe emplearse en presencia de medidas con ruido
- ✓ Normalmente se añade a actuaciones PI

$$m(t) = \bar{m} + K_c \left( e(t) + \frac{1}{\tau_i} \int e(t) dt + \tau_d \frac{de(t)}{dt} \right) \quad G_c(s) = K_c \left( 1 + \frac{1}{\tau_i s} + \tau_d s \right)$$

Tema V-II.1 Técnicas de diseño

Control e Instrumentación de P. Q.

Dpto Informática y Automática, USAL 13



## Reguladores P,PI,PID I

- ✎ Regulador proporcional
  - ✓ Cuando sea tolerable un error estacionario
  - ✓ Proceso con término integral (Nivel en un tanque sin salida)
- ✎ Regulador PI
  - ✓ Error en estacionario no es aceptable
  - ✓ Procesos rápidos (Control de caudal)
  - ✓ Sistemas con ruido
- ✎ Regulador PID
  - ✓ Error en estacionario no es aceptable
  - ✓ Permiten elevar la ganancia
  - ✓ Sistemas lentos sin ruido
    - ✓ Sistemas térmicos, múltiples capacidades, lazos de composición

Tema V-II.1 Técnicas de diseño

Control e Instrumentación de P. Q.

Dpto Informática y Automática, USAL 14



## Reguladores P,PI,PID II

### Reguladores PID

- ✓ No interactivo
  - ✓ Acciones integral y derivativa operan en paralelo independientemente
- ✓ Interactivo (PID comerciales)
  - ✓ Implementación en serie
- ✓ Relación entre ambas

$$G_c(s) = K_c \left( 1 + \frac{1}{\tau_i s} + \tau_d s \right)$$

$$G_c(s) = K_c' \left( \frac{1 + \tau_i' s}{\tau_i' s} \right) (\tau_d' s + 1)$$

$$K_c = K_c' \frac{\tau_i' + \tau_d'}{\tau_i'}$$

$$\tau_i = \tau_i' + \tau_d'$$

$$\tau_d \ll \tau_i'$$

$$\tau_d = \frac{\tau_i' \tau_d'}{\tau_i' + \tau_d'}$$

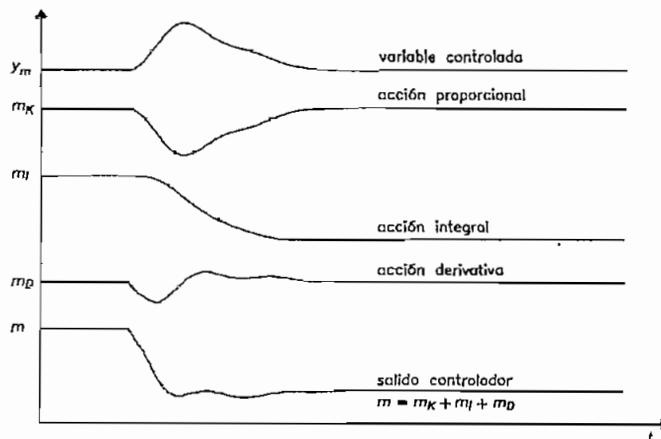
Tema V-II.1 Técnicas de diseño

Control e Instrumentación de P. Q.

15  
Dpto Informática y Automática, USAL



## Reguladores P,PI,PID III



Tema V-II.1 Técnicas de diseño

Control e Instrumentación de P. Q.

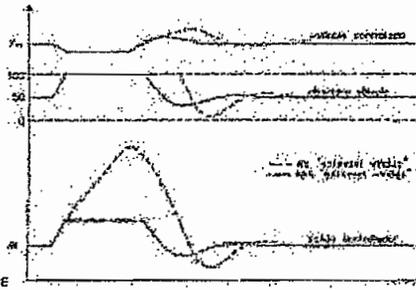
16  
Dpto Informática y Automática, USAL



## Reguladores P,PI,PID (y IV)

### Modificaciones sobre el PID

- ✓ Acción derivativa sobre la medida
  - ✓ Evita pico derivativo en cambios de referencia
- ✓ Evitar la saturación del término Integral (antireset-windup)
  - ✓ Suspender la acción integral cuando  $m(t)$  llega a un límite



Tema V-II.1 Técnica de diseño

Control e Instrumentación de P. Q.

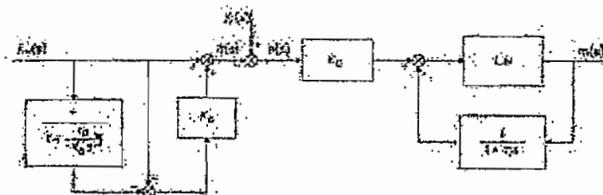
17

Dpto Informática y Automática, USAL



## Ejercicios

- Obtener el diagrama de bode de un PID no interactivo
- Demostrar que el siguiente diagrama de bloques realiza la operación de un PID con las siguientes características:
  - ✓ Interactivo
  - ✓ Derivada en la medida
  - ✓ antireset-windup



Tema V-II.1 Técnica de diseño

Control e Instrumentación de P. Q.

18

Dpto Informática y Automática, USAL